HistoCrypt 2021

Proceedings of the 4th International Conference on Historical Cryptology

20-22 June 2022
Amsterdam

Proceedings of the 4th International Conference on Historical Cryptology

HistoCrypt 2021

Editor
Carola Dahlke

June 20-22, 2022
Institute of Informatics
University of Amsterdam
The Netherlands

Published by:

NEALT Proceedings series 48
Linköping University Electronic Press, Sweden
Linköping Electronic Conference Proceedings, No. 183
ISSN: 1650-3686
eISSN: 1650-3740
ISBN: 978-91-7929-026-9
URL: https://doi.org/10.3384/ecp183
Cover photo: Ecolex I, CryptoMuseum Eindhoven

SPONSORS

UNIVERSITY OF AMSTERDAM
Informatics Institute

NETHERLANDS INTELLIGENCE
NISA
STUDIES ASSOCIATION

UPPSALA
UNIVERSITET

Preface

We are very pleased to present the proceedings of the 4th International Conference on Historical Cryptology, HISTOCRYPT 2021. Due to the COVID-19 pandemic, the respective conference has been postponed to June 20-22, 2022 in Amsterdam, the Netherlands. This means that all accepted papers and posters from 2021 will be presented together with the next year's contributions in Amsterdam in 2022.

HISTOCRYPT addresses all aspects of historical cryptography and cryptanalysis including work in closely related disciplines, such as history, history of ideas, computer science, artificial intelligence, computational linguistics, linguistics, or image processing with relevance to historical cipher-texts and codes. The conference's subjects include, but are not limited to the use of cryptography in military, diplomacy, business, and other areas, the analysis of historical ciphers with the help of modern computerized methods, unsolved historical cryptograms, the Enigma and other encryption machines, the history of modern (computer-based) cryptography, special linguistic aspects of cryptology, the influence of cryptography on the course of history as well as teaching and promoting cryptology in schools, universities, and the public.

The program committee welcomed submissions in two distinct tracks: *regular papers* on substantial, original, and unpublished research, including evaluation results, where appropriate, and *short papers* on smaller, focused contributions, work in progress, negative results, surveys, tutorials, or opinion pieces. The conference received 24 submissions from all over Europe including Austria, the Republic of Cyprus, the Czech Republic, Germany, Hungary, Italy, the Netherlands, Poland, Slovakia, Spain, Sweden and the U.K. as well as from Australia, Canada, Israel and the United States. Following the previous events, the primary goal of the program committee was to deliver a high quality program with a wide variety of topics by performing a double-blind review process. At least, three experts in the corresponding field evaluated each submission, and gave their recommendations to accept or decline. The reviews were synchronized and if ambiguous, were thoroughly discussed among the reviewers and the senior members of the PC, who made the final selection based on the recommendations and discussions. Finally, we rejected six papers and accepted 75% of the submissions, of which twelve papers were submitted as long and four were submitted as short papers. All accepted submissions are collected in this volume in alphabetical order after the last name of the first author.

Since the conference has been postponed, we do of course hope to be able to carry out a large part of our previously planned program in 2022. Nevertheless, we would like to take this opportunity to thank the invited keynote speakers who kindly accepted our invitation to speak at the conference this year: *Tanja Lange*, professor of Mathematics and expert in modern cryptography and leader of a European post-quantum project, *Maarten Oberman*, cryptologist specialized on Cold War cryptology and the Dutch cipher machine *ECOLEX*, *David Oranchak, Sam Blake, Jarl van Eycke* who solved *the Z-340 Zodiac Killer's Cipher*, *Paul Reuvers*, engineer, Hagelin expert, crypto collector and curator of *Crypto Museum* and *Gerhard F. Strasser*, professor emeritus of German and Comparative Literature, The Pennsylvania University.

Organizing a conference and a peer-review process always relies on the goodwill of many colleagues who take their valuable time to contribute to an interesting and fruitful program. First of all, I would like to thank Karl de Leeuw for the great collaboration, and my special thanks go to Beáta Megyesi and Karl de Leeuw for your help in publishing these proceedings. Furthermore, I want to thank all senior members of the program committee, Bernhard Esslinger, Benedek Láng, George Lasry, Karl de Leeuw, Beáta Megyesi, and Dermot Turing for your both active and mental support, for our spontaneous meetings and wise decisions on difficult issues. I am glad that I can rely on you for another HistoCrypt period. As well, I want to thank the 28 members of our extended Program Committee for your time and effort to give constructive and collegial feedback to help in the review process and selection of papers. In addition, I would like to thank all the authors for making these proceedings again interesting, diverse and impressive. Furthermore, many thanks go to the Local Committee under Karl de Leeuw's leadership for the organisation in Amsterdam so far, even though it was unfortunately not feasible this year, to Arno Wacker and Christoph Ruhl for helping out with the conference website, and to the Steering Committee.

As the physical conference has been postponed to June 20-22, 2022 in Amsterdam, Netherlands, we are planning a half-day online event this year on 20 September 2021, which will feature an exciting small program with a keynote, an online workshop, time for administrative information about HISTOCRYPT and, most importantly, the opportunity to share and network together. The complete program, and the respective registration information, is available on the HISTOCRYPT homepage www.histocrypt.org. Hopefully, we will finally meet again for real in 2022. Until then, I wish you all the best, good health, and enjoyment of this year's HISTOCRYPT publications.

Carola Dahlke
Program Chair of HISTOCRYPT 2021

Program Committee

- Carola Dahlke (program chair), Deutsches Museum, Germany
- Bernhard Esslinger, University of Siegen, Germany
- Benedek Láng, Budapest University of Technology and Economics, Hungary
- George Lasry, The CrypTool Team, Germany
- Karl de Leeuw, University of Amsterdam, Netherlands
- Beáta Megyesi, Uppsala University, Sweden
- Dermot Turing, Kellogg College, Oxford, UK

Local Organizing Committee

- Karl de Leeuw (general chair)
- Jan Bergstra
- Matthijs Koot
- Paul Reuvers
- Jaap van Tuyll

Steering Committee

- Joachim von zur Gathen, Emeritus, Bonn-Aachen International Center for Information Technology, Germany
- Marek Grajek, Poland
- Klaus Schmeh, Private researcher, Germany
- Arno Wacker, Bundeswehr University Munich, Germany

Extended Program Committee: Reviewers

- Eugen Antal, Slovak University of Technology in Bratislava, Slovakia
- Paolo Bonavoglia, Mathesis Venezia, Italy
- Nicolas Courtois, University College London, U.K.
- Camille Desenclos, Centre d'Histoire des Sociétés, des Sciences et des Conflits, Université de Picardie Jules Verne, France

- Ekaterina Domnina, Moscow State Lomonosov University, Russia
- John Dooley, Knox College, U.S.A.
- Joseph Fitsanakis, Coastal Carolina University, U.S.A.
- Otokar Grošek, Slovak University of Technology in Bratislava, Slovakia
- Emrah Safa Gurkan, Istanbul 29 Mayis University, Turkey
- Julio Hernandez-Castro, School of Computing, University of Kent, U.K.
- Kevin Knight, DiDi Labs, USA
- Grzegorz Kondrak, University of Alberta, Canada
- Nils Kopal, University of Siegen, Germany
- Jakub Mírka, The State Regional Archives in Pilsen, Czech Republic
- Valerie Nachef, UCY Cergy Paris Université, France
- Diego Navarro, Carlos III University of Madrid, Spain
- Ingo Niebel, Historian and Journalist, Germany
- Marie-Louise Rodén, Kristianstad University, Sweden
- Klaus Schmeh, Cryptovision, Germany
- Betsy Rohaly Smoot, Independent scholar, U.S.A.
- Gerhard F. Strasser, The Pennsylvania State University, U.S.A.
- Jörg Ulbert, Université Bretagne Sud, France
- Serge Vaudenay, Ecole Polytechnique Fédérale de Lausanne, Switzerland
- Arno Wacker, Bundeswehr University of Munich, Germany
- Michelle Waldispühl, Göteborg University, Institute for språk och litteraturer, Sweden

Extended Program Committee: Subreviewers

- Colin Choi, University of Alberta, Canada
- Bradley Hauer, University of Alberta, Canada
- Abram Hindle, University of Alberta, Canada

Contents

HCPortal Modules for Teaching and Promoting Cryptology

Eugen Antal
Slovak University of
Technology in Bratislava
Slovakia
eugen.antal@stuba.sk

Pavol Zajac
Slovak University of
Technology in Bratislava
Slovakia
pavol.zajac@stuba.sk

Abstract

HCPortal is an online portal focusing on historical cryptology. The structure of the portal is logically divided into modules. In this paper, we are presenting three new modules focusing on teaching and promoting cryptology. The first module is called Education. It contains a demonstration of selected classical ciphers and their respective cryptanalytic techniques. The second module focuses on nomenclators. The third module represents a virtual museum of historical ciphers.

1 Introduction

The Portal of Historical Ciphers (HCPortal) is an online portal consisting of several web pages and tools (logically divided into modules). Each module represents a specific topic related to historical cryptology.

In the first years of development, a series of modules were released. These modules are: *Home page* (entry point of the portal with navigation and information centre), *ManuLab* and *ManuLab online* (software product for statistical analysis, with a public API and example web page), *Tools and web pages* (links to external projects) and *Glossary* (glossary for historical cryptology, including codes and nomenclator terminology). The portal also features a special *Database of cryptograms*, containing a collection of cryptograms. A detailed overview of these modules is available in Antal and Zajac (2020).

In 2020, new modules were designed and developed (some of them are still in progress) focusing on teaching and promoting cryptology. The first module is called *Education*. It contains a demonstration of some classical ciphers and their respective cryptanalytic techniques. Each technique is accompanied by a visualization. Currently, its main use is as a support tool for a Classical Ciphers course at the Slovak University of Technology in Bratislava.

The second module focuses on *nomenclators*. This module is not yet released to the public. It contains a special online tool designed to create, use, and share nomenclator keys. The second part of this module is a special client-server application. The server contains an online database containing digitized nomenclator keys with a public API. The client part is a desktop application (with access to the server part) that supports transcription of nomenclator keys from images.

The third module represents a *virtual museum* of historical ciphers. It is built on a virtual reality framework supported by modern web browsers. The goal is to promote public interest in ciphers using modern technologies. This module is also a work in progress, and not yet released to the public. The estimated release date of the nomenclator and virtual museum modules is at the end of the year 2021.

2 The Education Module

There are several websites on the internet, dedicated to historical cryptography. Some of them, including dCode (2020), Cryptii (2020), CTO (2020), provide implementations of different classical ciphers. These and similar other sources pro-vide students with an opportunity to interact with the cipher algorithm and learn basic encryption and decryption steps. However, there is only a limited number of publicly available tools that contain fully described algorithms used in cryptanalysis. Moreover, a lot of the sources for cryptanalytic techniques lack interactive visualization that is suitable for educational purposes.

The primary goal for the inclusion of the *Education* module to HCPortal was to support an online education in our course Classical Ciphers taught at the Slovak University of Technology in Bratislava. However, we have designed the tools in such a way, that they can be used by other students and the general public.

The *Education* module is a collection of interactive tools with graphical visualization of the data designed for a better understanding of attacks on selected classical ciphers. At the moment, the *Education* module contains:[1]

- Brute-force attack on *Caesar Cipher*
- Hill-Climbing attack on *Simple Substitution Cipher*
- Friedman test and brute-force attack on *Vigenère Cipher*

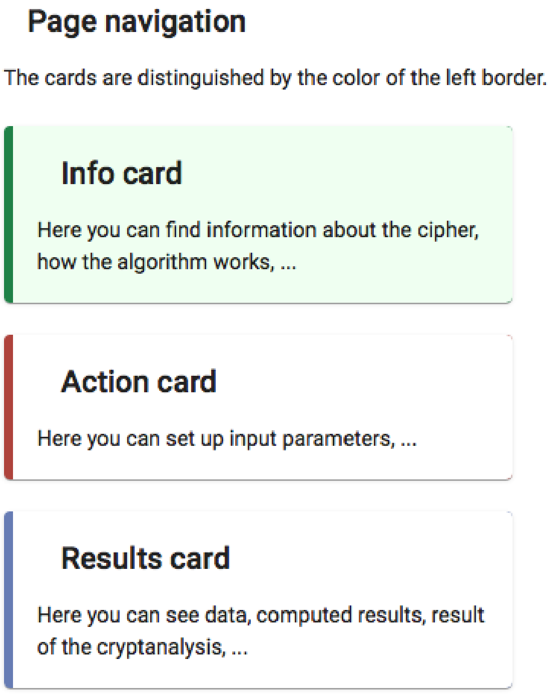
Currently available attacks were implemented in Angular. The implemented attacks are accessible through the navigation menu option or the main screen (Figure 3).

Each demonstrated attack is divided into logical steps. For visualization, we attach special *cards* to each of these steps. There are three card types, which are distinguished by the color of the left border (Figure 1). The cards are used to describe details of the current step of the attack, to display the computed results, or to serve as user input.
The *Education* module is available at the following web address: `https://www.edu.hcportal.eu`.

2.1 Brute-force Attack on Caesar Cipher

The first implemented attack is a brute-force attack on Caesar Cipher. The Caesar cipher is a simple type of substitution cipher suitable as a basic introduction to the topic of encryption algorithms even for young children. It replaces every plain text letter with a different one. Each letter is shifted by *n* letters in the used alphabet (26 letters for the English alphabet). The shift is defined by the key (originally, Caesar used a shift by 3 letters). The keyspace is very small, there are only 25 possible shifts (if we exclude the identity).

Despite the fact that Caesar Cipher can be cracked easily by hand, a good example can be created to demonstrate a brute-force attack. We have also decided that the Caesar cipher example is a good way to introduce a more general topic of frequency analysis, which is used to break the general monoalphabetic substitution cipher.


Page navigation

The cards are distinguished by the color of the left border.

Info card

Here you can find information about the cipher, how the algorithm works, ...

Action card

Here you can set up input parameters, ...

Results card

Here you can see data, computed results, result of the cryptanalysis, ...

Figure 1: Education module - cards description

The demonstration in our tool consists of the following steps. In the first step, a short description of the Caesar cipher is presented in an *info card*. The next card is used by the student to set up the input plaintext for the example. Currently, we only allow using English text as an input. After the input plaintext and the key is provided, the ciphertext is computed by the engine and passed to the next step.

Frequency analysis is performed on the encrypted text. From the obtained result we are guessing the used language by the index of coincidence (Figure 5). The measured index of coincidence is compared with the reference value of 6 languages and with the minimal value of the index (representing a random text). This step does not influence the attack.

The attack starts with a short description in an *info card*. Computed frequencies of letters in the encrypted message are presented. In the next step, an exhaustive search algorithm through all possible combinations of the key is performed. Unlike other similar tools, our goal is to demonstrate the automation of such attacks. Instead of evaluating possible plaintexts by the user, each candidate plaintext is evaluated automatically with a

[1] Additional specialized attacks on transposition ciphers will be also included soon.

specific scoring function: the Manhattan distance (Minkowski's L1 distance) of letter frequencies of the decrypted text from reference values. The overall result is displayed in a table for all possible keys. The table is ordered by the obtained score, and also contains the distances of measured frequencies from the reference values for each letter.

In the last step, a special frequency chart is displayed, where the letter frequencies can be compared with the reference values for every possible Caesar shift (Figure 4).

2.2 Hill-Climbing Attack on Simple Substitution Cipher

Simple substitution replaces letters of the plaintext with letters of the ciphertext based on predefined rules. Each letter from the plaintext alphabet maps exactly to one letter from the ciphertext alphabet. In comparison with the Caesar cipher[2] described in section 2.1, the keyspace (26! for English letters) of the generalized simple substitution can not be searched by brute-force.

This demonstration is similar to that presented in section 2.1. Instead of searching the whole keyspace, we show the students how to use an optimization algorithm (in this case the Hill-Climbing algorithm) to find the key. The goal was to provide insights on the details of the Hill-Climbing algorithm, and on what happens during the keyspace search. The main principle of the attack, however, stays the same: we explore a part of the keyspace in an intelligent way, score the used key based on a characteristic of plaintext candidates, and present the student with the scores and visualization of the steps. As a scoring function, we use bigram statistics instead of just letter frequencies. The input plaintext must be again an English text.[3]

After setting up the input, a short description of the attack follows. The attack has only two parameters: the number of iterations and the number of restarts (Figure 6). The best found key candidate is presented in a result card.

There are three additional and important result cards. In the first one, the evolution of the score and the match rate is presented. There are key candidates in a table, picked in different computation cycles (iterations). Each row contains the key, its score, the text decrypted using that key, and the match rate of the decrypted text. This gives some information about how the key/text candidate was changed during the search process. The second card visually compares the change of the score and match rate during the search process (Figure 7, first card).

Hill-Climbing is designed to accept only better candidates (improvements). It is also visible on the first card on Figure 7 - the score (marked as Sum) is only decreasing[4] during the whole search process. Using the third result card the relative change of the match rate is presented. It's important to show, that the match rate does not strictly follow the change of the score. We can accept a new candidate that produces a better score but worse match rate than the previous candidate (Figure 7, second card).

2.3 Friedman Test and Brute-force Attack on Vigenère Cipher

The Vigenère cipher is a polyalphabetic substitution. The encryption consists of series of Caesar ciphers, depending on the letters of a keyword. The keyword is repeated periodically. The first step of the analysis is the Friedman test, which helps to determine the length of the key. The next step is to try all possible combinations of the key. Each key is evaluated with a specific scoring function to find the correct one. The input plaintext must be an English text.

After setting up the input text, the encrypted text is displayed. Letters shifted by the same Caesar shift are highlighted with the same color. This visualization helps to better understand the encryption process. See Figure 2 as an example (text is encrypted with a key of length 4).

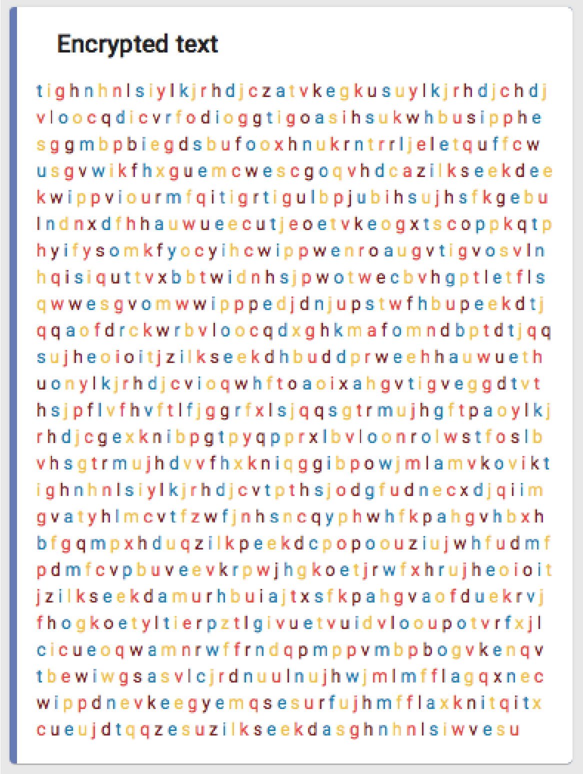
The Friedman test is a well-known method for determining the length of the key in the Vigenère cipher and is based on the notion of the index of coincidence (IC) . If the text is written in English, then we expect the value of IC to be close to 0.065. If the text was generated randomly, then the expected value of IC would be 0.038.

An English text encrypted by the Vigenère cipher with a key of length r, can be divided into r cosets where each coset contains letters shifted by the same Caesar shift. Therefore, the value of IC of each coset is expected to be close to 0.065.

[2] Which is a subset of a generalized simple (also called monoalphabetic) substitution cipher.

[3] The reference English bigram values are calculated from the Open American National Corpus, as probabilities.

[4] Minimisation problem - the scoring function calculates the distance of two vectors. The goal is to find a key that produces minimal distance.



Encrypted text

tighnhnlsiylkjrhdjczatvkegkusuylkjrhdjchdj
vloocqdicvrfodioggtigoasihsukwhbusipphe
sggmbpbiegdsbufooxhnukrntrrljeletquffcw
usgvwikfhxguemcwescgoqvhdcazilkseekdee
kwippviourmfqitigrtigulbpjubihsujhsfkgebu
lndnxdfhhauwueecutjeoetvkeogxtscoppkqtp
hyifysomkfyocyihcwippwenroaugvtigvosvln
hqisiquttvxbbtwidnhsjpwotwecbvhgptletfls
qwwesgvomwwipppedjdnjupstwfhbupeekdtj
qqaofdrckwrbvloocqdxghkmafomndbptdtjqq
sujheoioitjzilkseekdhbuddprweehhauwueth
uonylkjrhdjcvioqwhftoaoixahgvtigveggdtvt
hsjpflvfhvftlfjggrfxlsjqqsgtrmujhgftpaoylkj
rhdjcgexknibpgtpyqpprxlbvloonrolwstfoslb
vhsgtrmujhdvvfhxkniqggibpowjmlamvkovikt
ighnhnlsiylkjrhdjcvtpthsjodgfudnecxdjqiim
gvatyhlmcvtfzwfjnhsncqyphwhfkpahgvhbxh
bfgqmpxhduqzilkpeekdcpopoouziujwhfudmf
pdmfcvpbuveevkrpwjhgkoetjrwfxhrujheoioit
jzilkseekdamurhbuiajtxsfkpahgvaofduekrvj
fhogkoetyltierpztlgivuetvuidvlooupotvrfxjl
cicueoqwamnrwffrndqpmppvmbpbogvkenqv
tbewiwgsasvlcjrdnuulnujhwjmlmfflagqxnec
wippdnevkeegyemqsesurfujhmfflaxknitqitx
cueujdtqqzesuzilkseekdasghnhnlsiwvesu

Figure 2: Education module - text encrypted with the Vigenère cipher

The most probable value for *r* is displayed including the cosets highlighted with different colors as in Figure 2. For each coset the value of IC is presented as their average (see Figure 8). The user can manually change the key length and watch the changes in the IC value of new cosets.

In the second phase, all possible keys of the length determined by the Friedman test are generated[5] (brute-force). The ciphertext is decrypted with each key and its score is evaluated. The correct key should produce a text with letter frequencies that are closest to reference values (calculated from a large English corpus).

We calculate the frequency of letters by counting the occurrence of every letter in the text. Converting the absolute frequencies into relative ones helps to compare the letter frequencies independently of the text length. For each key, the computer calculates the statistical distance of letter frequencies of the decrypted text from reference values.

The top 10 results based on the score are presented in a table using a result card. The most probable key is the row with the lowest score value (column named sum). Each row contains the key, its score, and the decrypted text.

3 The Nomenclator Module

The Nomenclator module is a currently hosting two projects related to nomenclators. The first project, called *CipherCreator*, is a tool that allows the user to create custom nomenclator keys, while the second (unnamed) project focuses on management of historical nomenclator keys.

The *CipherCreator* is an Angular application for creating custom nomenclator keys. Keys created with this application have both textual (JSON format) and graphical (PDF format) representation. The actual *CipherCreator* web site allows the user to encrypt and decrypt custom text messages with the nomenclator keys.

The nomenclators created with *CipherCreator* can have the following cipher parts:

- simple/homophonic substitution,
- bigrams and trigrams,
- code words,
- nulls.

User can select which parts should be included in the nomenclator, and in which order (priority). To simplify the nomenclator creation, the application provides some predetermined bigrams, trigrams, and codewords. The graphical version of the nomenclator is created using a specific handwriting font (there are five preconfigured handwriting fonts available). By default, only numbers are used as ciphertext symbols. However, letters and (Unicode) symbols can be also set to make our system compatible with other systems and databases. The application also provides the ability to use custom symbols based on the images uploaded by the user while setting up the nomenclator key. Finally, there is a tool that allows the user to manually draw new symbols directly on the web page (Figure 9). The final nomenclator image can than be displayed on the background based on several paper types (Figure 10).

The second project is a special client-server application that can be used to digitize and store (historical) nomenclator keys. Compared to the DECODE database (Megyesi, 2020), (Megyesi et al.,

[5] Due to computational reason, this phase is limited up to key length 3. For longer keys, only the Friedman test is performed.

2020), we store less meta-data, but some new additional information, like the nomenclator structure described with our own scheme[6]. We created a notation to describe the nomenclator scheme. Our scheme contains the sub-cipher system used in the nomenclator. The scheme also describes the graphical structure of the key[7]. We adopted the cipher symbol representation from Megyesi (2020). After the analysis of our nomenclator keys, we had to modify and extend this structure. This project is still Work in Progress, but the ultimate goal is to combine both projects to allow online encryption and decryption in a similar way with both custom nomenclators, and historical ones.

4 The Virtual Museum Module

The main goal of the whole HCPortal is to support research and education in the area of historical ciphers. By using modern information technologies we can also combine our educational goals with promoting the interest in historical ciphers among general public. The *Virtual Museum* module is based on virtual reality concept. In this way, we can present the materials collected in HCPortal in a familiar „museum" style. We use virtual reality engine for web browser. In this way, materials can be displayed online, even if the user does not have a VR device.

The main concept of the museum is based on the following ideas:

- Exposition is based on specially prepared *rooms* in the virtual reality. Different room sizes are used to present different amounts of data. Each room contains some 3D model such as a table, projector, boards, …

- The museum contains an entry point (a specific room) where the user can select and watch events. This room also works as a permanent event presenting a general description of historical ciphers.

- We can set up an *event* (exhibition) in the museum. Each event is connected with a room.

- An event is set up with a *configuration*. It contains the event description, the data, the starting and ending date of the event, author names, and similar meta-data.

- The data presented in events is represented as an *exhibition item*. It contains some meta-data such as name, description, the format of the data (text, image, video, or PDF). We are using only digital image and text document formats. It allows us to dynamically configure the prepared rooms without the need to create additional 3D models.

- The rooms can *present* the content of a PDF file or images directly on the room walls (into frames) or on a prepared table (on the placed 3D models of books and papers).

Currently, the creation of museum exhibits is manual, but we are analysing the options for open collaboration on creating the museum contents online by registered users. If the reader wants to contribute to the Virtual Museum with some interesting historical ciphers, please contact the authors.

Acknowledgments

This work was partially supported by grant VEGA 2/0072/20.

References

Eugen Antal and Pavol Zajac. 2020. HCPortal Overview. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt 2020*, pages 18 - 20. Linköping University Electronic Press

CrypTool Contributors. 2020. *CrypTool-Online*. `https://www.cryptool.org/en/cto/`

Team dCode. 2020. *dCode - The ultimate 'toolkit' to solve every games / riddles / geocaches*. `https://www.dcode.fr/en`

Fränz Friederes. 2020. *Cryptii*. `https://cryptii.com`

Beáta Megyesi. 2020. Transcription of Historical Ciphers and Keys. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt 2020*, pages 106 - 115. Linköping University Electronic Press

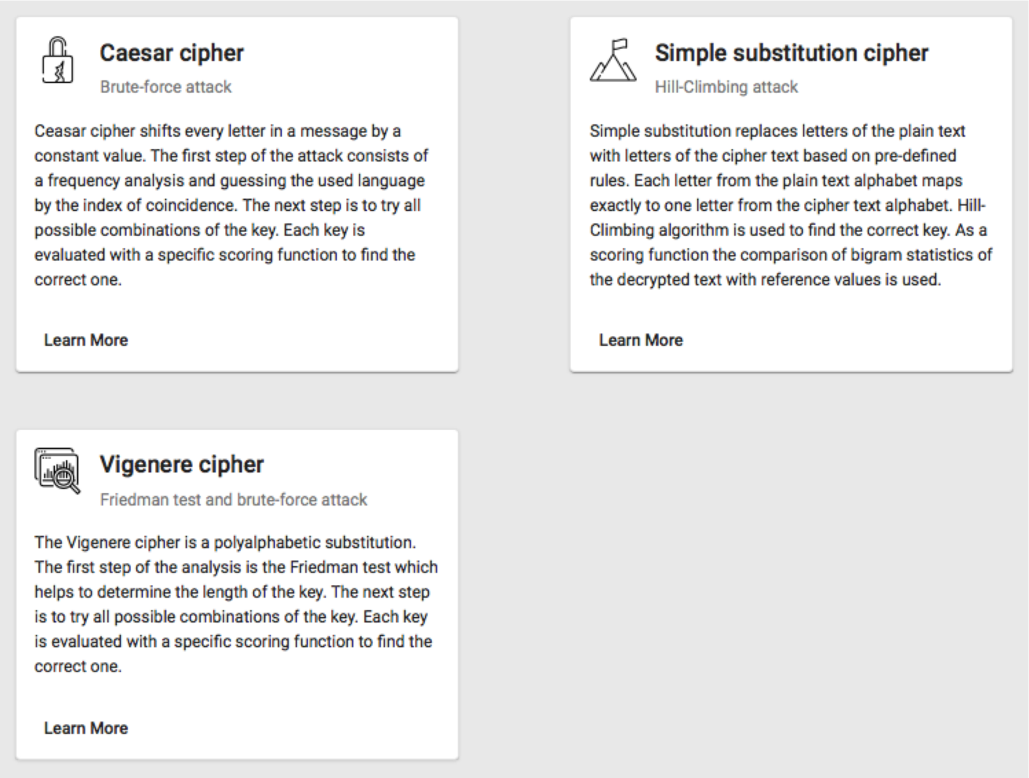
Beáta Megyesi. 2020. Transcription of Historical Ciphers and Keys: Guidelines. `https://cl.lingfil.uu.se/~bea/publ/transcription-guidelines200221.pdf`

[6] In (Megyesi, 2020) it's stated that "graphical structure of the keys cannot be represented in any simple way in the transcription".

[7] Like the order (position) of the sub-cipher types in the original key, and if the plaintext symbol is before/after/above/below the ciphertext symbol, etc.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, volume 44, number 6, pages 545-559. Taylor & Francis

Appendices



Caesar cipher

Brute-force attack

Ceasar cipher shifts every letter in a message by a constant value. The first step of the attack consists of a frequency analysis and guessing the used language by the index of coincidence. The next step is to try all possible combinations of the key. Each key is evaluated with a specific scoring function to find the correct one.

Learn More

Simple substitution cipher

Hill-Climbing attack

Simple substitution replaces letters of the plain text with letters of the cipher text based on pre-defined rules. Each letter from the plain text alphabet maps exactly to one letter from the cipher text alphabet. Hill-Climbing algorithm is used to find the correct key. As a scoring function the comparison of bigram statistics of the decrypted text with reference values is used.

Learn More

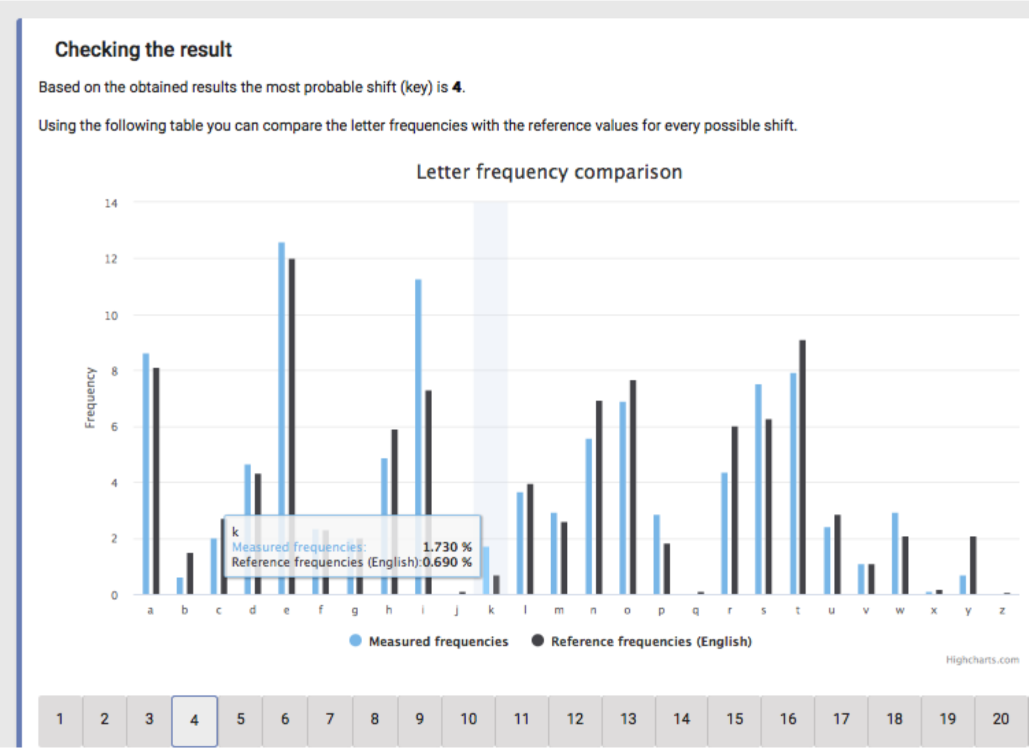
Vigenere cipher

Friedman test and brute-force attack

The Vigenere cipher is a polyalphabetic substitution. The first step of the analysis is the Friedman test which helps to determine the length of the key. The next step is to try all possible combinations of the key. Each key is evaluated with a specific scoring function to find the correct one.

Learn More

Figure 3: Education module - main screen



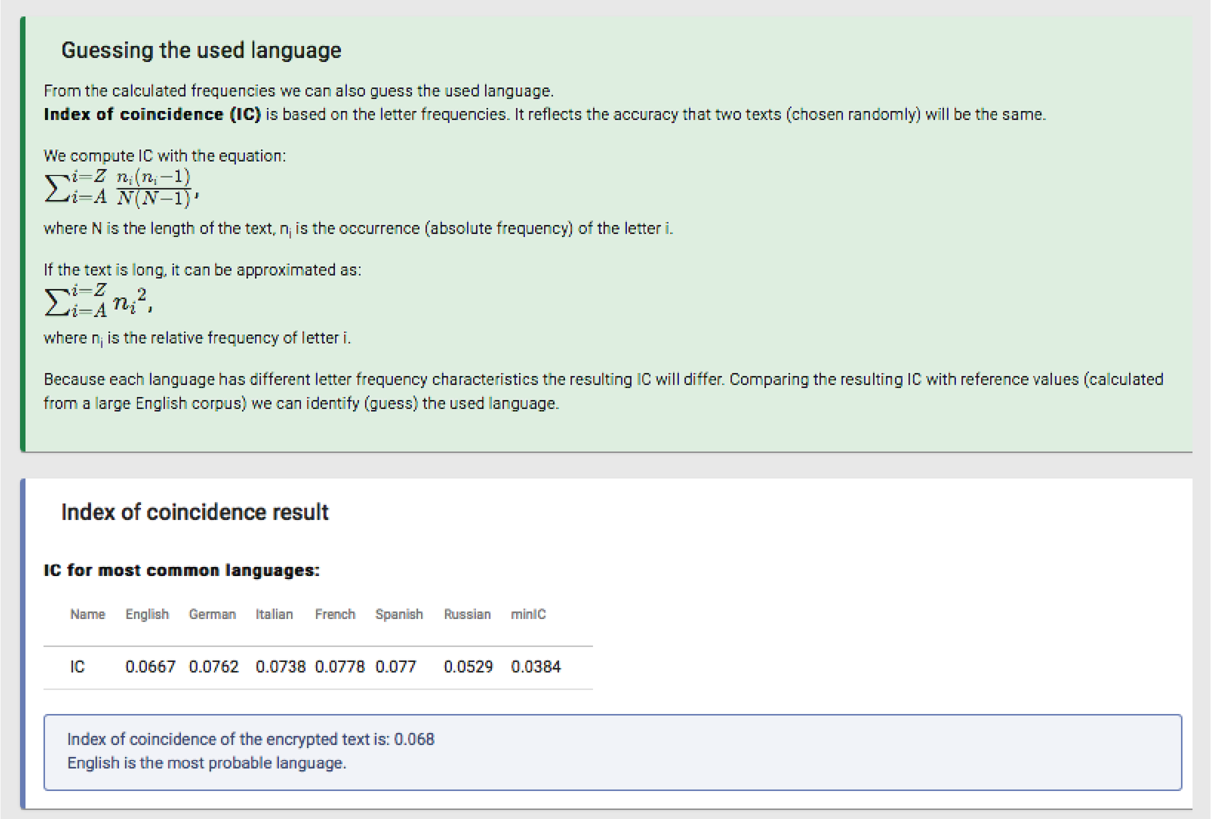
Checking the result

Based on the obtained results the most probable shift (key) is **4**.

Using the following table you can compare the letter frequencies with the reference values for every possible shift.

Letter frequency comparison
Frequency
0
2
4
6
8
10
12
14
a b c d e f g h i j k l m n o p q r s t u v w x y z
k
Measured frequencies: 1.730 %
Reference frequencies (English):0.690 %
● Measured frequencies
● Reference frequencies (English)
Highcharts.com
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20

Figure 4: Education module - result of the attack on Caesar cipher



Guessing the used language

From the calculated frequencies we can also guess the used language.
Index of coincidence (IC) is based on the letter frequencies. It reflects the accuracy that two texts (chosen randomly) will be the same.

We compute IC with the equation:

$$\sum_{i=A}^{i=Z} \frac{n_i(n_i-1)}{N(N-1)},$$

where N is the length of the text, n_i is the occurrence (absolute frequency) of the letter i.

If the text is long, it can be approximated as:

$$\sum_{i=A}^{i=Z} n_i{}^2,$$

where n_i is the relative frequency of letter i.

Because each language has different letter frequency characteristics the resulting IC will differ. Comparing the resulting IC with reference values (calculated from a large English corpus) we can identify (guess) the used language.

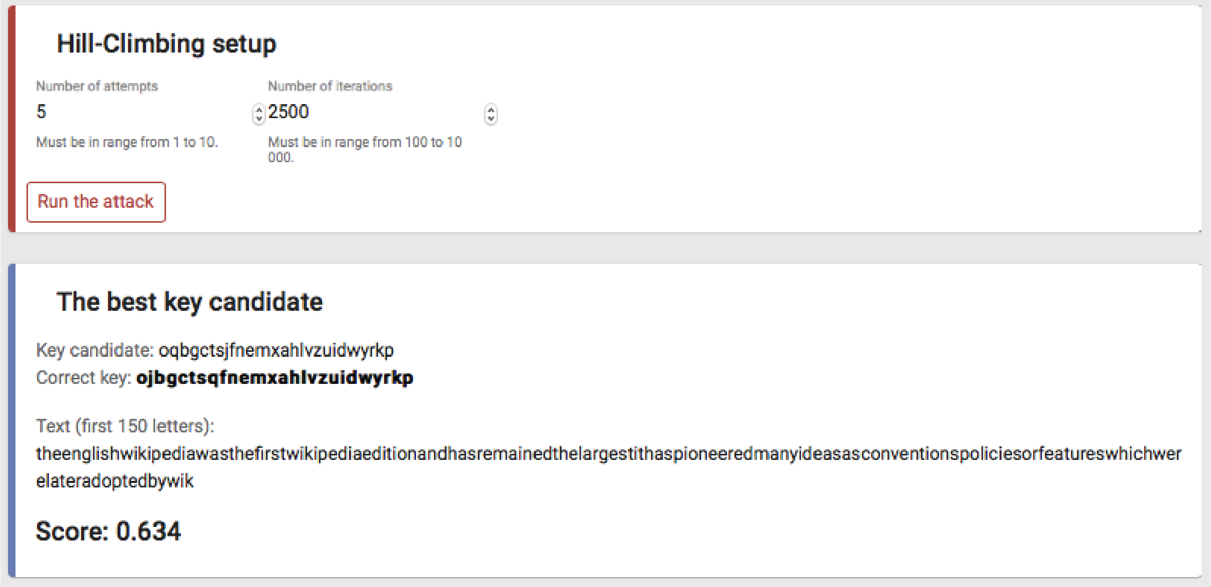
Index of coincidence result

IC for most common languages:

Name	English	German	Italian	French	Spanish	Russian	minIC
IC	0.0667	0.0762	0.0738	0.0778	0.077	0.0529	0.0384

Index of coincidence of the encrypted text is: 0.068
English is the most probable language.

Figure 5: Education module - language guessing



Hill-Climbing setup

Number of attempts
5
Must be in range from 1 to 10.

Number of iterations
2500
Must be in range from 100 to 10 000.

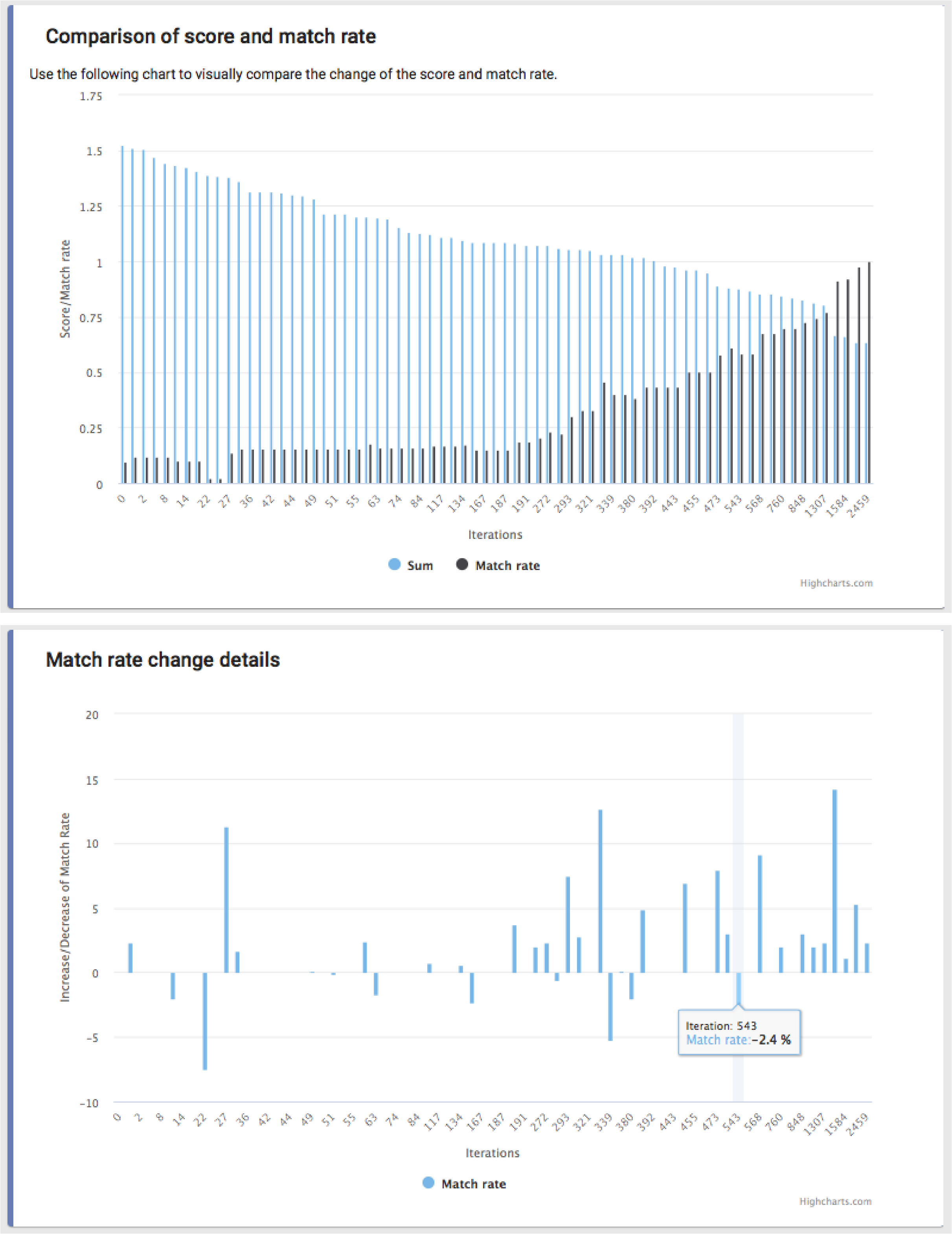
The best key candidate

Key candidate: oqbgctsjfnemxahlvzuidwyrkp
Correct key: **ojbgctsqfnemxahlvzuidwyrkp**

Text (first 150 letters):
theenglishwikipediawasthefirstwikipediaeditionandhasremainedthelargestithaspioneeredmanyideasasconventionspoliciesorfeatureswhichwerelateradoptedbywik

Score: 0.634

Figure 6: Education module - Hill-Climbing setup and the best key candidate

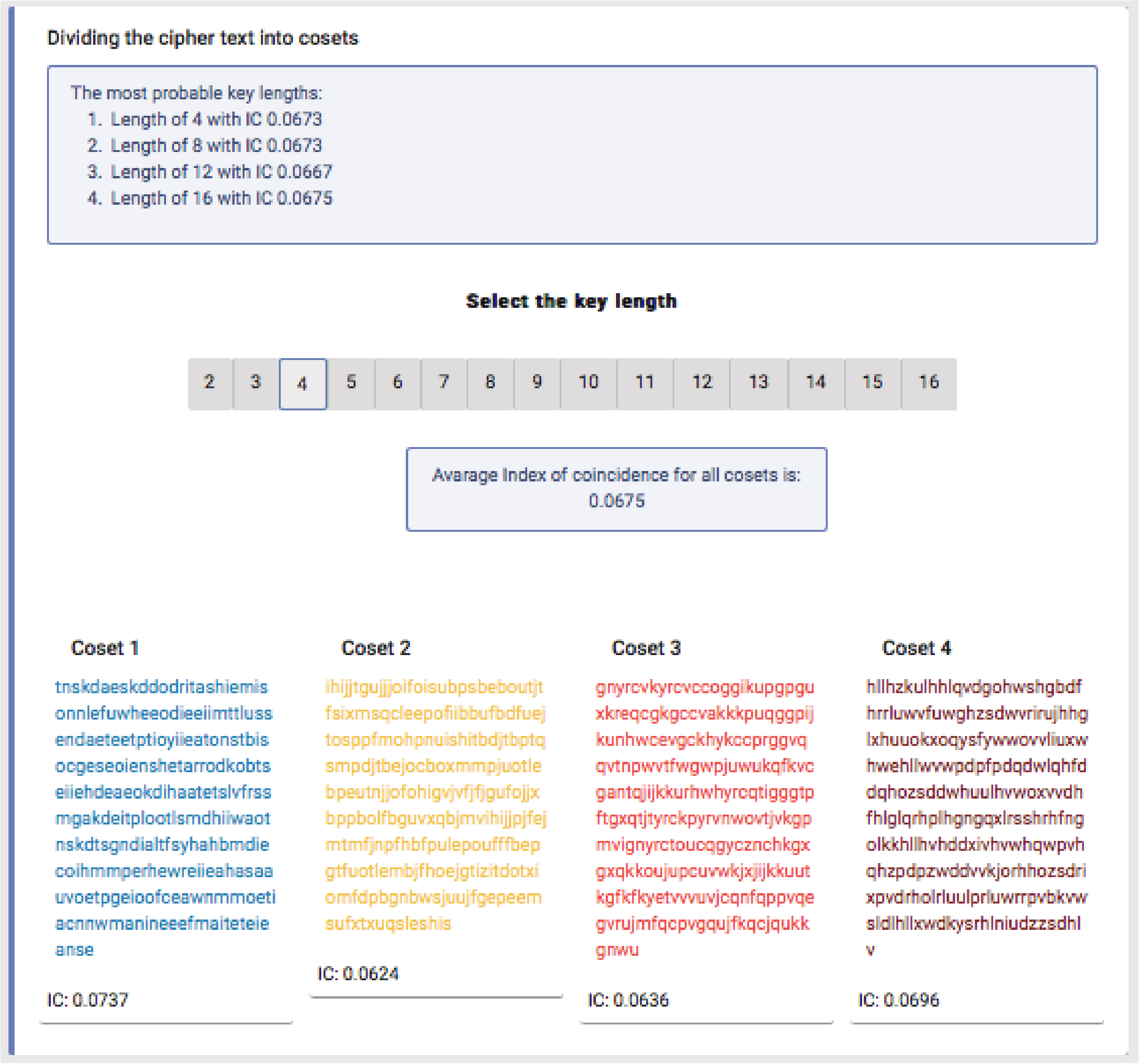


Comparison of score and match rate

Use the following chart to visually compare the change of the score and match rate.

Match rate change details

Figure 7: Education module - Hill-Climbing details



Dividing the cipher text into cosets

The most probable key lengths:

1. Length of 4 with IC 0.0673
2. Length of 8 with IC 0.0673
3. Length of 12 with IC 0.0667
4. Length of 16 with IC 0.0675

Select the key length

Avarage Index of coincidence for all cosets is:
0.0675

Coset 1

tnskdaeskddodritashiemis
onnlefuwheeodieeiimttluss
endaeteetptioyiieatonstbis
ocgeseoienshetarrodkobts
eiiehdeaeokdihaatetslvfrss
mgakdeitplootlsmdhiiwaot
nskdtsgndialtfsyhahbmdie
coihmmperhewreiieahasaa
uvoetpgeioofceawnmmoeti
acnnwmanineeefmaiteteie
anse

IC: 0.0737

Coset 2

ihijjtgujjjoifoisubpsbeboutjt
fsixmsqcleepofiibbufbdfuej
tosppfmohpnuishitbdjtbptq
smpdjtbejocboxmmpjuotle
bpeutnjjofohigvjvfjfjgufojjx
bppbolfbguvxqbjmvihijjpjfej
mtmfjnpfhbfpulepoufffbep
gtfuotlembjfhoejgtizitdotxi
omfdpbgnbwsjuujfgepeem
sufxtxuqsleshis

IC: 0.0624

Coset 3

gnyrcvkyrcvccoggikupgpgu
xkreqcgkgccvakkkpuqggpij
kunhwcevgckhykccprggvq
qvtnpwvtfwgwpjuwukqfkvc
gantqjijkkurhwhyrcqtigggtp
ftgxqtjtyrckpyrvnwovtjvkgp
mvignyrctoucqgycznchkgx
gxqkkoujupcuvwkjxjijkkuut
kgfkfkyetvvvuvjcqnfqppvqe
gvrujmfqcpvgqujfkqcjqukk
gnwu

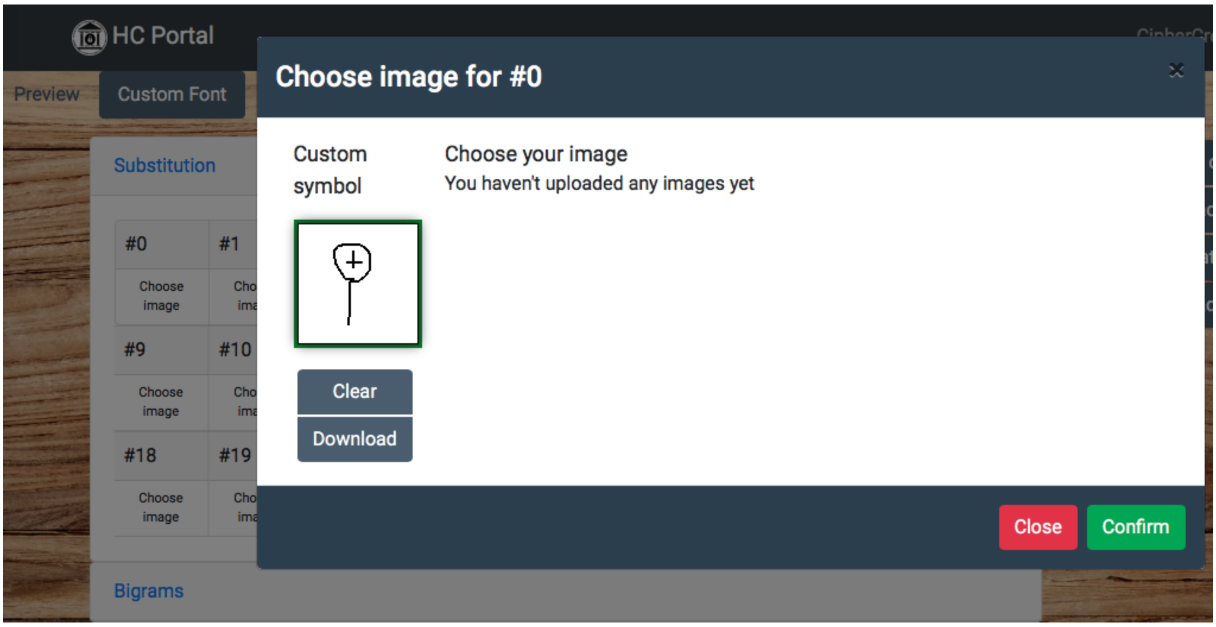
IC: 0.0636

Coset 4

hllhzkulhhlqvdgohwshgbdf
hrrluwvfuwghzsdwvrirujhhg
lxhuuokxoqysfywwovvliuxw
hwehllwvwpdpfpdqdwlqhfd
dqhozsddwhuulhvwoxvvdh
fhlglqrhplhgngqxlrsshrhfng
olkkhllhvhddxivhvwhqwpvh
qhzpdpzwddvvkjorhhozsdri
xpvdrholrluulprluwrrpvbkvw
sldlhllxwdkysrhlniudzzsdhl
v

IC: 0.0696

Figure 8: Education module - Vigenère cipher ciphertext divided to *r* cosets

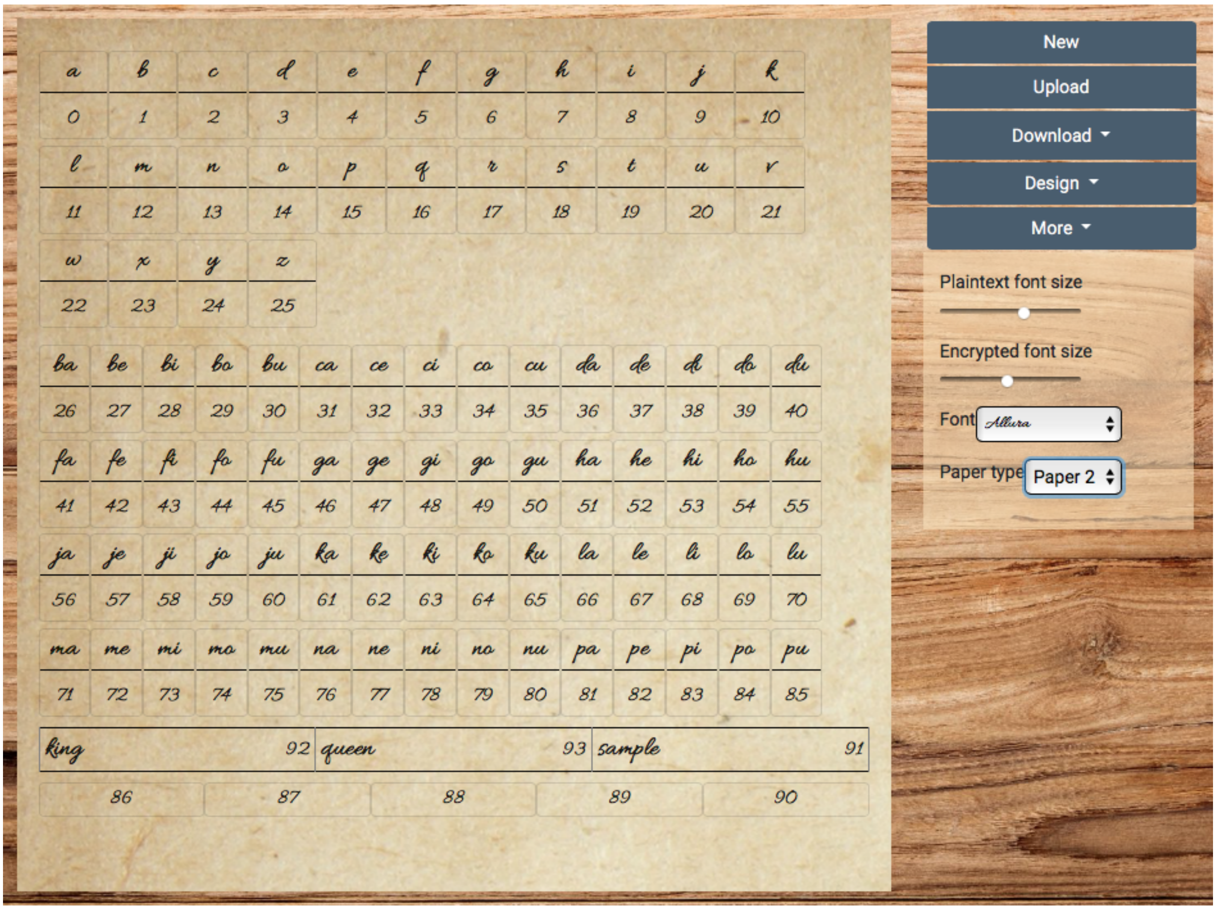


Choose image for #0

Custom symbol

Choose your image
You haven't uploaded any images yet

Figure 9: Nomenclator module - CipherCreator custom symbol drawing



a	b	c	d	e	f	g	h	i	j	k
0	1	2	3	4	5	6	7	8	9	10
l	m	n	o	p	q	r	s	t	u	v
11	12	13	14	15	16	17	18	19	20	21
w	x	y	z							
22	23	24	25							

ba	be	bi	bo	bu	ca	ce	ci	co	cu	da	de	di	do	du
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
fa	fe	fi	fo	fu	ga	ge	gi	go	gu	ha	he	hi	ho	hu
41	42	43	44	45	46	47	48	49	50	51	52	53	54	55
ja	je	ji	jo	ju	ka	ke	ki	ko	ku	la	le	li	lo	lu
56	57	58	59	60	61	62	63	64	65	66	67	68	69	70
ma	me	mi	mo	mu	na	ne	ni	no	nu	pa	pe	pi	po	pu
71	72	73	74	75	76	77	78	79	80	81	82	83	84	85

king	92	queen	93	sample	91

86	87	88	89	90

Plaintext font size

Encrypted font size

Font: Allura

Paper type: Paper 2

Figure 10: Nomenclator module - CipherCreator nomenclator key with font and paper settings

Solving a Mystery From the Thirty Years' War

Karel Rabenhaupt ze Suché's Encrypted Letter to Landgravine Amalie Elisabeth

Eugen Antal
Slovak University of
Technology in Bratislava
Slovakia
eugen.antal@stuba.sk

Pavol Zajac
Slovak University of
Technology in Bratislava
Slovakia
pavol.zajac@stuba.sk

Jakub Mírka
The State Regional Archives
in Pilsen
Czech Republic
mirka@soaplzen.cz

Abstract

In 2013, an unsolved encrypted message was presented in a Czech cryptogram solving competition. This message was sent by Karel Rabenhaupt ze Suché to Amalie Elisabeth von Hanau-Münzenberg, Landgravine of Hessen-Kassel at the end of the Thirty Years' War. During the following years, many crypto enthusiasts tried to solve the cipher without success. Finally, in 2020, the authors of this paper were able to reveal the content of the encrypted message. Here we present the decryption of the encrypted message, including the historical background of the communication. We also present an analysis of the used encryption system.

1 Introduction

The history of the European nations is deeply interconnected. European aristocratic families played crucial roles in military and international diplomacy. Through a web of correspondence, the historical mosaic can be reconstructed and presented in various aspects.

Archives of European aristocratic families still contain many unsolved secrets. In our project, we focus on historical archives of families who lived in the former Austro-Hungarian Monarchy. The nobility had strong ties in the whole Central European region. As such, the study of encrypted correspondence found in any middle-European archive is not self-contained, but instead requires additional materials from many different archives in multiple countries (Hungary, Austria, Czechia, Poland, Germany, ...).

In the present paper, we focus on a story of a specific encrypted message. We want this story to demonstrate the need for interconnected research in historical cryptography, and the need to have a platform for efficient sharing of encrypted (and decrypted) materials and keys.

2 Historical Background

Our focus is an encrypted message created during the Thirty Years' War in the Rhine region and now deposited in The State Regional Archives in Pilsen (SOA v Plzni, FA Trauttmansdorff, inv. nr. 125) in the Czech Republic.

The encrypted message is a part of the correspondence of Karel Rabenhaupt ze Suché with Amalie Elisabeth von Hanau-Münzenberg, Landgravine of Hessen-Kassel. In this section, we briefly summarize the historical background of the sender of the encrypted message.

Karel Rabenhaupt ze Suché (see Figure 1), was born in 1604 as the son of the Bohemian Protestant nobleman Zikmund Rabenhaupt (Robmháp) ze Suché, the owner of the land estates Lichnice and Třemošnice,[1] and Kateřina Žehušická z Nestajova (Engelbrecht, 1989; Genealogical-heraldic collection of Wunschvitz, inv. nr. 921). In 1620 he took part in the defense of Bautzen against the army of John George II, Elector of Saxony. After the defeat of the Bohemian Revolt on Bílá Hora he emigrated to the Netherlands because of his faith. He began to study fortress construction at the Leiden University (Woringer, 1913; Engelbrecht, 1989).

He fought in the Dutch army, and he was promoted to the rank of lieutenant in 1627 after the siege of Groenlo. Later on, he became famous in his military service. He joined William V, Landgrave of Hessen-Kassel. From 1633 he was the commander of the Schaarkopf cavalry regiment. After the death of William V in 1637, **Amalie Elisabeth** (see Figure 2) took over the government as the regent of William VI (Engelbrecht, 1989;

[1] Town in the Czech Republic near Pardubice, approx. 90 km eastwards from Prague.

CAREL RABENHAUPT Baron van Sucha, Erfheer in
Lichtenbergh en Fremesnich, Heere tot Grumbach, Generael Lieut:
en Gouverneur van Groeningen en Ommelanden Coll. over een Regim:
Infanterie Cap. van Coevorden en Drost van 't Graefschap Drenthe

Figure 1: Karel Rabenhaupt ze Suché (Wikipedia Commons, 2020)

Warlich, 2021). In the 1640s he was engaged in the fights on the left bank of Rhine.

After the Thirty Years' War Rabenhaupt stayed in Hessian service and gained high military posts there. In 1668 he left the service and moved to his estates in the Netherlands. Rabenhaupt became a national hero in the Netherlands in 1672, after defending the city of Groningen.[2] He not only defended Groningen against the Münster army, but he was also able to break through and recapture one of the most modern and powerful fortresses of its time, Coevorden. He became Coervorden's governor. In 1673 he was promoted to the status of a baron by the Emperor, now an ally of the Netherlands, for his services in defending the Netherlands. He died on August 12, 1675 (Woringer, 1913; Engelbrecht, 1989).

2.1 Rabenhaupt on the Rhine

In 1640, Count Kaspar von Eberstein became the commander-in-chief of the army of Hessen-Kassel. In the same year, Rabenhaupt, in Eberstein's presence, conquered the town of Kalkar, which lies on the lower Rhine near the border with the Netherlands. From there he made raids along the Rhine to the south (Warlich, 2021).

Figure 2: Amalie Elisabeth von Hanau-Münzenberg, Landgravine of Hessen-Kassel (Wikipedia Commons, 2020)

In 1642, the Protestants conquered Kempen, Linn, and Neuss. The city of Neuss, near Düsseldorf, fell to the Hessians. The same year, Rabenhaupt became military commander of Neuss, and made the city his base for military actions, especially on the left bank of the Rhine. On the opposite side, one of the important Imperial strongholds in this area was the fortified city of Zons, which belonged to the Electorate of Cologne, and was located about 14 km southeast of Neuss. In the following years, neither side of the rivals gained a significant advantage, and the fighting devolved into smaller skirmishes, partial raids and tactical shifts (Löhrer, 1840). Rabenhaupt was captured in April 1644 but was released again in the summer (at the latest). In the autumn of the same year, Eberstein died, and after

[2] The Dutch province of Groningen offered Rabenhaupt the position of the commander of the army.

his death, Johann von Geyso temporarily became the head of the Hessian troops (Warlich, 2021).

In 1645, Rabenhaupt tried unsuccessfully to conquer Zons (Woringer, 1913; Warlich, 2021). He was more successful in the autumn when the Hessian army managed to conquer the city of Euskirchen, which belonged to the Duchy of Jülich-Berg and was located about 60 km south of Neuss. From there, the Hessians controlled the entire adjacent area (Hofmann, 1882).

At the beginning of 1646, Colonel Rabenhaupt was appointed major general[3] (HLA-HStAM, Best. 4 h, Kriegssachen, Nr. 1810). The Hessian army on both sides of the Rhine was again engaged mainly in smaller battles. During February, Rabenhaupt managed to occupy numerous settlements in Jülich-Berg and later moved to the right bank of the Rhine for a time, as he was to be in Wipperfürth on March 2. From the west, however, the allied French army approached, led by Marshal Henri de La Tour d'Auvergne de Turenne, and from the east and north, the Swedes. The two armies tried to unite. The enemy forces were led by Imperial General Peter Melander von Holzappel, who changed sides during the war. It is interesting that until 1640 he was the commander in chief of the Hessian army. After his resignation, he was replaced by the aforementioned Count Kaspar von Eberstein. Melander's deputy was Otto Christoph von Sparr. He was also at this time the main opponent of Rabenhaupt, and also successfully prevented Marshal de Turenne from crossing the Rhine (von Schroetter, 1899; Warlich, 2021).

Rabenhaupt has been preparing for the siege of Zons since the summer or maybe has even besieged it directly for some time, as Rommel (1843) states. In early July, Rabenhaupt made a military move against Bonn to ease the pressure on Euskirchen. Rabenhaupt has described this successful operation in details in his report, which was preserved in a manuscript in the library of Wolfenbüttel (Löhrer, 1840; HAB, Cod. Guelf. 11.8 Aug. 2°, ff. 368-369).

In this situation, an encrypted letter, which is the main subject of this article, was created. Thanks to its decipherment (see Section 5), we know that Rabenhaupt's main goal at this time was indeed the conquest of Zons and that he probably hoped for the help of Marshal de Turenne (SOA in Pilsen, FA Trauttmansdorff, inv. Nr. 125). We know from other sources that de Turenne eventually refused to help Rabenhaupt with the siege of Zons. Apparently, he did so because it would keep him from the main task of uniting with Swedish troops, in which he succeeded at the turn of July and August 1646. Although he could not cross the Rhine in the south, he went north to Wesel and met the Swedes in Hessen. This operation was very swift. Sources show, that while he was supposed to be at Ahrweiler on July 12, he had managed to cross the Rhine at Wesel on July 15. Even after de Turenne's departure from the area, Rabenhaupt continued to dominate the left bank of the Rhine. General Melander, therefore, tried to get the Hessians from the Rhine by making a diversionary assault against their capital Kassel. However, at the end of September, Rabenhaupt attacked Zons again, forcing Melander to return (von Rommel, 1843; von Schroetter, 1899; Salm, 1990).

The siege of the city began on September 24, 1646, and on September 28, heavy artillery was launched, which greatly damaged the city. According to some sources, the walls had already been broken and negotiations for a possible surrender had been initiated. On October 6, however, Melander crossed the Rhine and the Hessians had to withdraw to Neuss. Melander then went on to conquer Euskirchen. Later in October, Rabenhaupt was reported to be with Geyso in Wesel. As early as 1647, Rabenhaupt operated in Neuss, but was then assigned to General Königsmarck's Swedish army and left with him. In May he was already with Königsmarck near Vechta in northern Germany (Wassenberg, 1647; von Rommel, 1843; Hofmann, 1882; Warlich, 2021).

The area of Karel Rabenhaput ze Suché's operations on the Rhine is marked in Figure 11 (in the Appendix).

3 An Unsolved Cryptogram From the Thirty Years' War

In 2013 an unsolved encrypted message was presented in a Czech journal (Mírka, 2013) as part of a cryptographic competition. Here, we summarize the most important facts known at that time:

The object of the competition was an encrypted message (see Figure 12), which is now deposited in The State Regional Archives in Pilsen (SOA v Plzni, FA Trauttmansdorff, inv. nr. 125). The encrypted message was a part of a correspondence of Karel Rabenhaupt ze Suché with Amalie Elisabeth

[3] General-Wachtmeister

von Hanau-Münzenberg, Landgravine of Hessen-Kassel, from 1646.

Beside the encrypted message, a second (unencrypted) message was found.

These two messages were intercepted near Arnsberg and were never delivered. After the interception, the encrypted message was directly investigated by the military commander Alexandre de Bournonville, but he was unable to solve it. Alexandre de Bournonville, on July 20, sent a copy of the messages from Hamm to Maximilian von Trauttmansdorff, who was the envoy of Emperor Ferdinand III to peace talks in Münster. He hoped that somehow Maximilian von Trauttmansdorff would be able to find the corresponding encryption key.

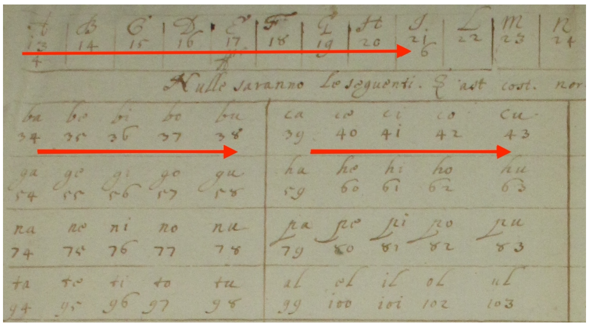
The encrypted message is dated to 11 (or 13)[4] of June 1646. The second (not encrypted) is dated July 13, 1646. Both messages were written in German language and were sent from Neuss. The date of the encrypted message is probably not correct. Due to the fact that both messages were intercepted together in the same place, it is more possible that the encrypted message was also written in July. It is also probable that Alexandre de Bournonville would not have sent the report about the encrypted message a month after the interception. So it is more probable that Bournonville's scribe made a mistake copying the message (Mírka, 2013; SOA v Plzni, FA Trauttmansdorff, inv. nr. 125).

The unencrypted letter tells about movements of Marshal de Turenne's troops, that were presumably referenced in the previous letter sent on July 11th. The unencrypted letter also references other army movements in the Rhine valley around Zons and Wesel. The encrypted letter contains cleartext parts, but the important information is encrypted with numbers and symbols. The preliminary analysis points to the possibility that a nomenclator encryption scheme was used.

4 Nomenclator Encryption System

A nomenclator is a special encryption system consisting of several different simpler encryption sys-tems used together during the encryption. A nomenclator[5] mostly contains a monoalphabetic or homophonic substitution in a combination with bigram and/or trigram substitution, code word substitutions and nulls (Mírka, 2012; von zur Gathen, 2015; Antal and Mírka, 2018). The basic encryption key of a nomenclator is frequently represented in a table. Additional codes used in a nomenclator can grow its size to several pages. The cipher text alphabet is very often represented by numbers (mostly due to the large number of possible code words and homophones).[6] In addition to digits/numbers, special symbols/glyphs were used as well[7] (Antal and Mírka, 2018; Dunin and Schmeh, 2020). This type of encryption was very popular and was used for a long time period from the fourteenth to the nineteenth century (Dunin and Schmeh, 2020; Meister, 1906; Lasry et al., 2020).

After taking a closer look at some available nomenclator constructions (Mírka and Vondruška, 2013; Antal and Mírka, 2018; Dunin and Schmeh, 2020) we can conclude that many nomenclators were poorly designed: cipher text numbers (or letters) were assigned in an alphabetical or numerical order (see Figures 3, 4 and 5). The nomenclator key could be also used incorrectly (Dunin and Schmeh, 2020). Another possible drawback of some nomenclator keys was that they have been used for a long time period, and have been reused by several different persons. On the other hand, this means that some nomenclator copies have survived in various archives.



A	B	C	D	E	F	G	H	I	L	M	N
13 4	14	15	16	17	18	19	20	21 6	22	23	24

Nulle saranno le seguenti. & ast cost. nor

ba	be	bi	bo	bu	ca	ce	ci	co	cu
34	35	36	37	38	39	40	41	42	43
ga	ge	gi	go	gu	ha	he	hi	ho	hu
54	55	56	57	58	59	60	61	62	63
na	ne	ni	no	nu	ſa	ſe	ſi	ſo	ſu
74	75	76	77	78	79	80	81	82	83
ta	te	ti	to	tu	al	el	il	ol	ul
94	95	96	97	98	99	100	101	102	103

Figure 3: Poor nomenclator design. SOA v Plzni, FA Windischgrätz, inv. nr. 1403.

[4] 11 is rewritten to 13 in the document. Supporting evidence for July 13 is also that according to sources, de Turenne was at Ahrweiler on July 10, and thus could not have been at Euskirchen, as mentioned in the letter.

[5] Based on the authors' experience from research in archives.

[6] In many cases, a special separator is required such as dot, comma, or space between cipher text units. This separator is used to correctly split the digits into ciphertext numbers.

[7] Mostly for the monoalphabetic/homophonic substitution parts, but there are also examples of nomenclators with glyphs representing code words.

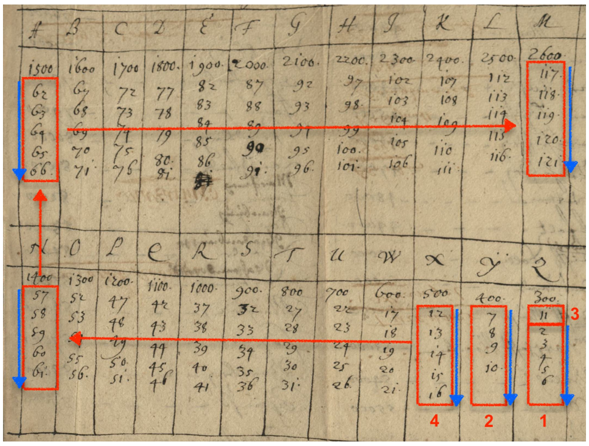


Figure 4: Poor nomenclator design (homophonic substitution part). HLA-HStAM Best. 4d Nr. 1218.

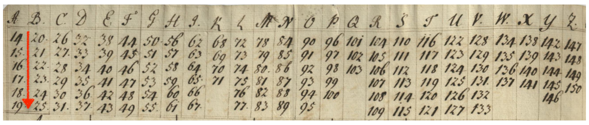


Figure 5: Poor nomenclator design (homophonic substitution part). HLA-HStAM Best. 4d Nr. 1227.

5 The Solution of the Cryptogram

The cryptogram was made publicly available in 2013 as a part of a Czech cryptogram solving competition. Antal and Zajac (2013) tried to solve it within the competition but without success. Despite to additional popularization of the encrypted message in conferences and blogs, nobody was able to directly solve this mystery. In fact, a correctly used nomenclator scheme can remain unbreakable even with modern computing power, especially if the analyzed cipher text is short. However, there was still a hope that the nomenclator key may have been preserved in some archive.

We have searched for additional documents related to Karel Rabenhaupt in the Hessian State Archives in Marburg. Unfortunately, the studied collections did not contain nomenclator keys. After the cipher challenge was published on Klaus Schmeh's site (Klausis Krypto Kolumne, 2020), one of his readers (using the Thomas nickname) directed us to the HSTaM 4d collection in this archive. The collection contains a lot of different nomenclator keys from the seventeenth and eighteenth century, and as we found out, it also contains the key that was used to encrypt Rabenhaupt's letter.

In fact, there are three almost identical nomenclator keys, which fit our cryptogram (one of the keys in Figure 13). Two of them have also an inverse key.[8]

To date the keys, we use the fact that they contain notes with names. Probably the first of the series is from 1641 and was created as a new key to communicate with Lieutenant-General Count von Eberstein. In original: "Neewer (=neuer) Clavis Mit dem Herrn General Lieutenant Graven von Eberstein des 4./14. t[en] Aprilis a[nn]o 1641 ufgerichtet (=aufgerichtet)". The second key is probably the updated version of the previous one, where additional "users" of this nomenclator were gradually added (different names written with different handwriting). The first name is von Eberstein ("Clavis mitt H[errn] General Lieut[enant] Graven von Eberstein"), then the added ones: "Herrn Wicqueforten, Herrn Obrist Lieut[enant] von Kroßieg, Obersten Karpffen, Herrn Gen[eral] Wachtmeister Geyso". On the third version of the key there is a different name mentioned - Hans Heinrich Günterode, Court Marshal and Obrist of Hessen-Kassel ("Clavis ahn Herrn Obristen Günterode").

Interestingly, the name of Karel Rabenhaupt is not present on these keys.[9] There is also a possibility that there were even more users of this nomenclator than mentioned on the nomenclator itself. Note that the latter story of Rabenhaupt's letter shows that the Emperor's agents were unable to read the encrypted parts of the letter in 1646, and perhaps later. This means that even with multiple users of the key, the key management was relatively secure.

Having found the (correct or related?) nomenclator key, we used it to decipher the encrypted parts of the letter. The obtained result contains dozens of typos/mistakes. Such mistakes can also make potential decryption harder. These mistakes could have been caused either by Rabenhaupt or more probably by a scribe while copying the original document.

After the corrections, the final version of our solution is the following (the deciphered parts in the

[8] We also found another inverse key that fits our cryptogram and that does not belong to the previous ones. So there are six versions of these keys preserved in total.

[9] We have found a different cipher key from 1646 with the name Rabanhaupt, but this one does not fit our cryptogram.

text are marked in bold):

Copia

Durchleuchtige, Hochgeborne Fürstin, Gnädige Frau. Dießen Morgen empfang ich Andtwortschreiben von ***Mareschall de Turenne****, so gestern in* ***Eusskirchen gewesen*** *undt* ***daherumb gelegen, dass er heutte marchiren*** *undt also* ***morgen vor Zoons sich setzen wolte****. Alß mache ich meine Rechnung* ***fünffzehen Hundert Mann Fuessvolck nebst ein Batterie Stückgeschütz und einen Fewermörsel***[10] *undt negst Gottes Hülff* ***einen kurtzen Proces damitt zu machen****, undt weil[e]n ich gute Hoffnung habe, daß die Aliirten in ihren Vortheil lenger alß andere werden können stehen pleiben, undt vielleicht waß geshehen solle, ehe* ***der Mareschall dahin kommen kan wird, geschehen sein****. So hette Eu[er] Fürst[lichen] Gn[aden] underthenig zu bitten, die schleunige Verordnung zu thuen, daß* ***di[e] Ostfrieslandtsche commendirte Völcker****, auch die* ***gelehnete Stückgeschütz zu Wesel mir ausgefolget werden möchten****. Vielleicht möchte* ***ich etwas wichtiges verrichten und gutte Winterquartier machen****. Eu[er] Fürst[lichen] Gn[aden] hi[e]rmit [Gedelicher] Obhuet empfehlend verpleib*
Eu[er] Fürst[lichen] Gnaden underthänig, gehorsahmer, pflichtschuldig Diener Rabenhaubt
Neuß den 13 Junÿ 1646, ahn die Landtgravin zu Hessen Amalie Elisabet

We have created a rough translation of the deciphered message to English, presented in Appendix B. The original text is difficult t o t ranslate, and without further context, some passages can be interpreted in various ways. We tried to preserve the nuances of the original text to avoid potential shifts in the meaning of the text.

6 Analysis of the Cryptogram

In this section, we revise to and extend the preliminary analysis of the cryptogram from Antal and Zajac (2013), and point out some weaknesses of the used encryption method.

The encrypted message contains both encrypted and unencrypted parts. The whole text consists of 34 rows: 13 rows are fully encrypted, 13 rows are encrypted partially and 8 rows contain only plain text (not encrypted) parts.

Rows containing both encrypted and unencrypted passages are very valuable. This property can be useful to guess the content of the encrypted part. As an example, the second row starts with the German sentence ”*Dießen Morgen empfang ich Andtwortschreiben von* ...” (*This morning I received a letter of reply from* ...) and continues with cipher symbols. From the meaning of this passage, it is clear, that the text should continue with a name of a person. From section 2.1 we know that Rabenhaupt was in Neuss at the time of the writing of the document (in July 1646) and the army of Vicomte de Turenne operated nearby Rabenhaupt. His name is also present in the second (not encrypted document). In fact, the solution of the first encrypted passage is directly ”MARESCHALL DE TURENNE” (see section 5).

The encrypted part of the document consists of numbers, letters (uppercase and lowercase), and special symbols/glyphs.[11] These units are mostly divided by dots.[12] Dots were commonly used in nomenclators as separator chars. In the document, a dot is the most frequent symbol and really serves as a separator char. The transcription of the cipher text is in Appendix A. After splitting the text by the separator char,[13] the encrypted part of the document contains 118 unique cipher text units and is 369 symbols/letters/numbers long.[14] There are:

- 145 numbers (51 unique),
- 105 symbols (30 unique),
- 72 lower case and 24 upper case letters (24 unique),
- and 23 double letters (13 unique).

Based on a large number of cipher text units and on the relatively flat frequencies (Figure 6), Antal and Zajac suggested (2013) that the nomenclator consists of a homophonic substitution, bigrams, codes, and nulls. The possible way of solving such

[10] = Feuermörsel

[11] The first row contains the word *Copia*, the document is probably only a copy of the original document. This is the reason why many transcription errors (typos) occurred during the copy, see section 5.

[12] In some cases the dot separator is missing, but there is a wide space between the cryptogram units so they can be distinguished.

[13] We added several dots in case it was not present to unify the structure of the cipher text.

[14] Numbers and double letters are counted as one cipher text unit, e.g. 110, pp.

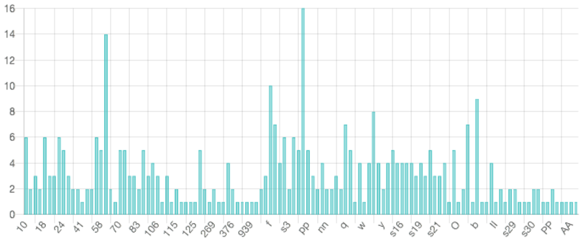


Figure 6: Frequency characteristic of the cipher text

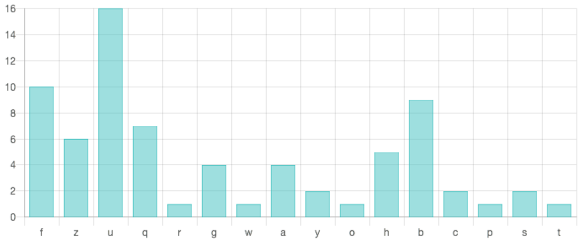


Figure 7: Frequency characteristic of lower case single letters

a complicated nomenclator is to guess/separate the nomenclator sub-ciphers, guess the most frequent cipher text units, or try to find something that was incorrectly used during the encryption.

If we examine the frequency characteristic separately for numbers, symbols, and letters we can find some interesting properties. The frequency characteristics of the lower case letters (Figure 7) is similar to a frequency distribution of a simple substitution (the index of coincidence of these letters is also high). This means it can also be a row in a homophonic substitution table. The letter *u* has the highest frequency and can stand for the plain text letter *e*.

The number frequencies (Figure 8) are relatively flat, only the number *59* has a relatively high frequency. If *59* is also obtained in the homophonic cipher, it can stand for a frequent letter such as *e*. From the symbol frequencies (Figure 9) we were unable to find any useful details.

In (Antal and Zajac, 2013) the authors also investigated the cyclic structure of homophones.

Although the used nomenclator is relatively strong, we can see that there are some mistakes that allowed Antal and Zajac (2013) to correctly guess that

- one row in the homophonic substitution table consist of lower case single letters (Figure 10),
- letter *u* stands for plain text letter *e*,
- number *5* (that is in fact a symbol very similar to number 5) stands for plain text letter *n*,
- number *59* also stands for plain text letter *e*.

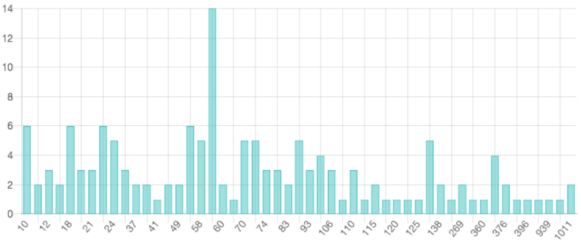


Figure 8: Frequency characteristic of numbers in the cipher text

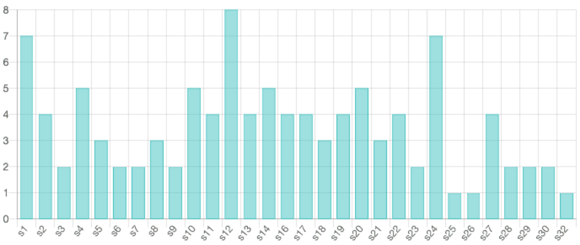


Figure 9: Frequency characteristic of symbols in the cipher text

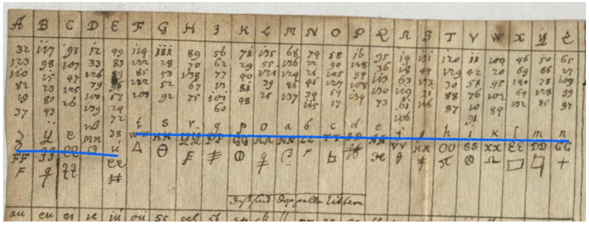


Figure 10: Letters in the homophonic substitution table. HLA-HStAM Best. 4d Nr. 1218.

Unfortunately, these suggestions are not enough to solve the cipher by analysis. The used nomenclator key is complicated and was designed carefully. The letters are used as homophones and nulls as well. The numbers are used to encrypt homophones, code words, and nulls. The symbols are used both as homophones and to encrypt common bigrams.

The used homophonic substitution is very strong - each set of homophones (assigned to a plain text letter) consist of 8 to 11 elements and consist of numbers, lower case single letter, upper case double letters, and symbol. The frequency characteristic of the used cipher text elements is flat (except for few elements from the homophones). We expect that to solve such a homophonic substitution an impractically large number of cryptograms would be required.

7 Conclusions

Nomenclator encryption systems were used extensively in European warfare and diplomacy. Some of them were used and designed incorrectly (see section 4), so they could be solved. On the other hand, correctly designed (and used) nomenclators provide strong encryption which is almost impossible to solve correctly.

Our case study shows that historians and historical cryptography enthusiasts have an alternative method of solving encrypted historical documents. We can try to ”connect the dots” and search for original keys (or their copies and related documents) in historical archives.

In our present paper, we focused on the story of a specific encrypted message, that resisted classical cryptanalytic attempts. As we have shown in the paper, the nomenclator used in the communication between Karel Rabenhaupt ze Suché and Amalie Elisabeth von Hanau-Münzenberg, Landgravine of Hessen-Kassel was designed carefully (see sections 5 and 6). It contains a strong homophonic cipher, common bigram encryption, code words, and a large selection of nulls. Despite the presence of unencrypted parts between the encrypted parts of the document, the cipher text does not contain enough information for relevant cryptanalysis.

The search for the decryption key was quite complicated as well. The encrypted document was found in the State Regional Archives in Pilsen, Czech Republic. However, the correct nomenclator key was preserved in the Hessian State Archives in Marburg, Germany. This particular situation was caused by the fact that the encrypted message was intercepted during the Hessian war and sent for analysis to Emperor's envoy von Trauttmansdorff, who has stored the (unsolved) letter in his family archive. This situation might not be unique, and archives of one side of a war can contain intercepted messages of the other side, with keys remaining in the opposite archives. Furthermore, documents in family archives were passed down during centuries and moved to different locations.

The goal of our story was to demonstrate the need for interconnected research in historical cryptography. To efficiently analyze and solve historical cryptograms, researchers need to have a platform for efficient sharing of encrypted (and decrypted) materials and keys. We hope that collaboration efforts such as the *DECODE*[15] (Megyesi et al., 2020) database and the *HCPortal Cryptograms*[16] (Antal and Zajac, 2020) database can result in further joint projects and connected open data platforms and tools available for both researchers and crypto and history enthusiast all over the world.

Acknowledgments

This work was supported by grant VEGA 2/0072/20.

We thank Ingo Niebel for his help with the initial search in the Hessian State Archives in Marburg, Klaus Schmeh for publishing the challenge on his blog, Thomas for the pointers, Jiří Kubeš for his help with the translation, Dušan Kováč for his help with the documents from the Herzog August Bibliothek, and Pavel Vondruška, who has initiated the cryptographic challenge leading to this article.

References

Eugen Antal and Jakub Mírka. 2018. Selected encrypted messages found in Slovak and Czech archives. In *HistoCrypt 2018 Workshop: Solving codes rather than ciphers. Is there a software challenge?*

Eugen Antal and Pavol Zajac. 2013. Analýza Rabenhauptovho zašifrovaného dopisu (Analysis of Rabenhaupt's encrypted message). In *Crypto-World*, 11-12.

Eugen Antal and Pavol Zajac. 2020. HCPortal Overview. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt 2020*, pages 18 - 20. Linköping University Electronic Press.

Elonka Dunin and Klaus Schmeh. 2020. Codebreaking: A Practical Guide. Robinson. Great Britain.

Wilken Engelbrecht. 1989. Flüchtling im fremden Lande. Weißenberger Exulanten in niederländischen Quellen. In *Sborník příspěvků. IV. setkání genealogů a heraldiků*, Ostrava.

[15] https://cl.lingfil.uu.se/decode/database

[16] https://cryptograms.hcportal.eu

Wilhelm Hofmann. 1882. *Peter Melander Reichsgraf zu Holzappel. Ein Charakterbild aus der Zeit des dreißigjährigen Krieges.* Bibliographisch-Artistisches Institut, München.

George Lasry, Beáta Megyesi and Nils Kopal. 2020. Deciphering papal ciphers from the 16th to the 18th Century. In *Cryptologia*. Taylor & Francis.

Friedrich J. Löhrer. 1840. *Geschichte der Stadt Neuß von ihrer Gründung an bis jetzt, nach gedruckten und handschriftlichen Quellen verfaßt.* Druck u. Verlag von L. Schwann, Neuß: 314–325.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. In *Cryptologia*, volume 44, number 6, pages 545-559. Taylor & Francis.

Aloys Meister. 1906. *Die geheimschrift im dienste der Päpstlichen kurie von ihren anfängen bis zum ende des XVI. Jahrhunderts.*

Jakub Mírka. 2013. Dosud nevyluštěný dopis českého pobělohorského emigranta Karla Rabenhaupta ze Suché z doby třicetileté války. (Unsolved message of Czech emigrant Karel Rabenhaupt ze Suché from the Thirty Years' War). In *Crypto-World*, 7-8.

Jakub Mírka. 2012. Raně novověká šifrovaná korespondence ve fondech šlechtických rodinných archivů Státního oblastního archivu v Plzni (Early modern encrypted correspondence in the fonds of the aristocratic family archives in the State Regional Archive in Pilsen). In *Západočeká archivy* 2012, also published in three parts in *Crypto-World*, 11-12/2012, 1-2/2013 and 3-4/2013.

Jakub Mírka and Pavel Vondruška. 2013. Nomenklátory 17. a 18. století (Nomenclators from the 17. and 18. century). In *Crypto-World*, 11-12.

Christoph von Rommel. 1843. *Neuere Geschichte von Hessen. Vierter Band.* Verlage von Friedrich und Andreas Perthes zu Hamburg und Gotha, Cassel: 701, 704.

Hubert Salm. 1990. *Armeefinanzierung im Dreissigjährigen Krieg Der Niederrheinisch-Westfälische Reichskreis 1635-1650.* Aschendorff. Münster: p. 71.

Klaus Schmeh. 2020. *Klausis Krypto Kolumne (Cipherbrain)* `http://scienceblogs.de/klausis-krypto-kolumne`.

Friedrich von Schroetter. 1899. Otto Christoph von Sparr, der erste brandenburgische Generalfeldmarschall. *Hohenzollern-Jahrbuch. Forschungen und Abbildungen zur Geschichte der Hohenzollern in Brandenburg-Preußen, 3*: 163–187.

Joachim von zur Gathen. 2015. CryptoSchool. Springer.

Bernard Warlich. Der Dreißigjährige Krieg in Selbstzeugnissen, Chroniken und Berichten. Entries: Rabenhaupt, Karl; Sparr zu Trampe, Ernst Georg; Eberstein, Kaspar. Available online (accessed 27. 1. 2021), `https://www.30jaehrigerkrieg.de/`.

Eberhard Wassenberg. 1647. Der Ernewerder Teutsche Florus Wassenberg: *mit Animadversionem Addition und Correctionen deren in vorigen eingeruckten ungleichen Hystorien verbessert der Warheit restituirt und biss Anno 1647 continuirt.* Bey Antoni Hummen, Frankfurt am Main: 692.

August Woringer. 1913. Ausländer als Offiziere im hessischen Heere. Hessenland, 27: 60–61.

Herzog August Bibliothek /HAB/ (Herzog August Library in Wolfenbüttel), manuscript Cod. Guelf. 11.8 Aug. 2°, ff. 368-369.

Hessisches Landesarchiv – Hessisches Staatsarchiv Marburg /HLA-HStAM/ (The Hessian State Archives – Hessisches Staatsarchiv Marburg), archival fond Politische Akten nach Philipp dem Großmütigen: Kanzlei- und Geheimeratskorrespondenz (4d), Nr. 1218; Kriegssachen (4h), Nr. 1810.

Národní archiv /NA/ (National Archives in Prague), Sbírka genealologicko-heraldická Wunschwitzova (Genalogical-heraldic collection of Wunschwitz), inv. nr. 951.

SOA v Plzni. Státní oblastní archiv v Plzni /SOA v Plzni/ (State Regional Archive in Pilsen), archival fond Rodinný archiv Trauttmansdorffů (Family Archive Trauttmansdorff), inv. nr. 125.

Wikipedia Commons. 2020.

Appendices

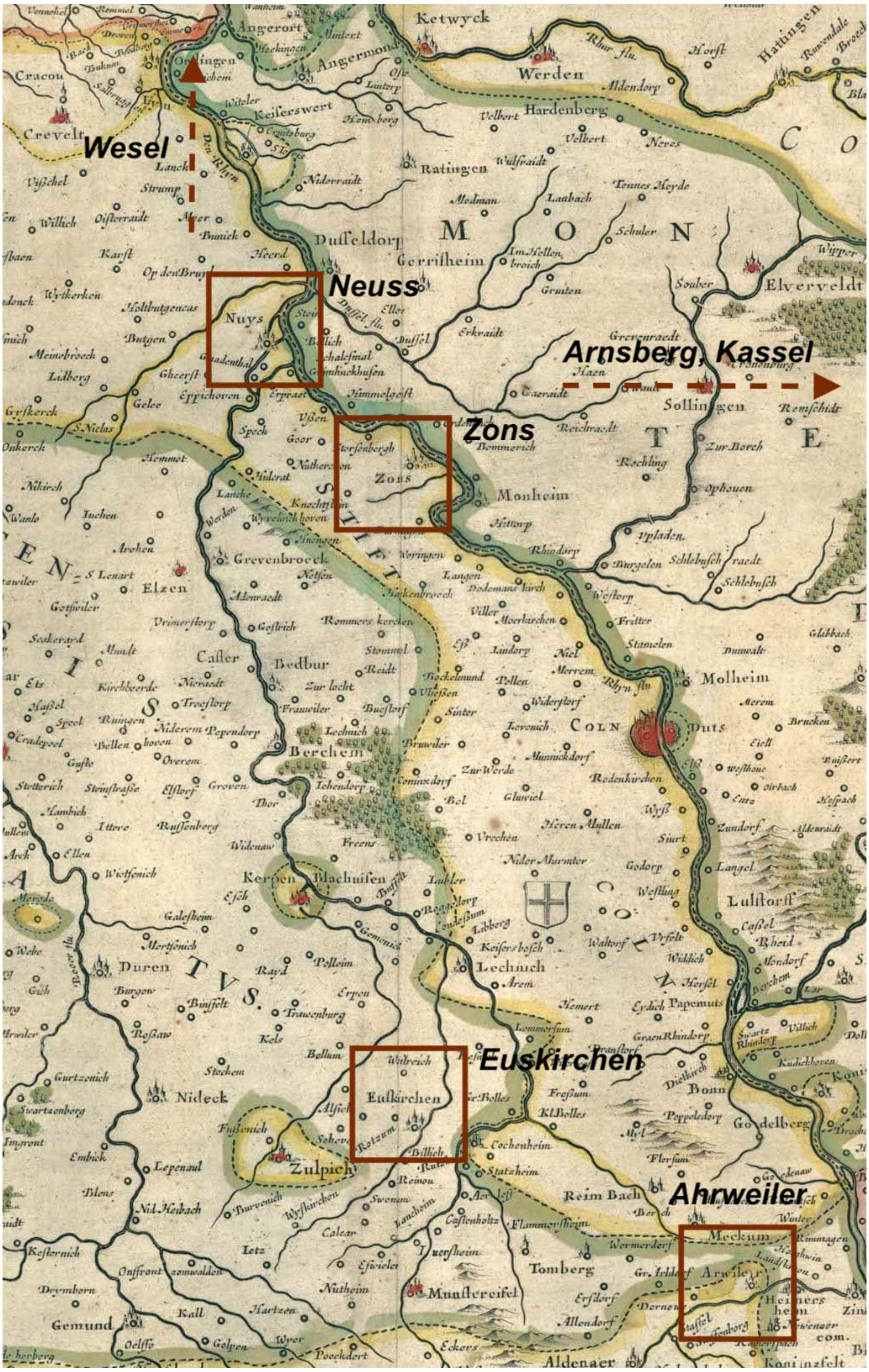
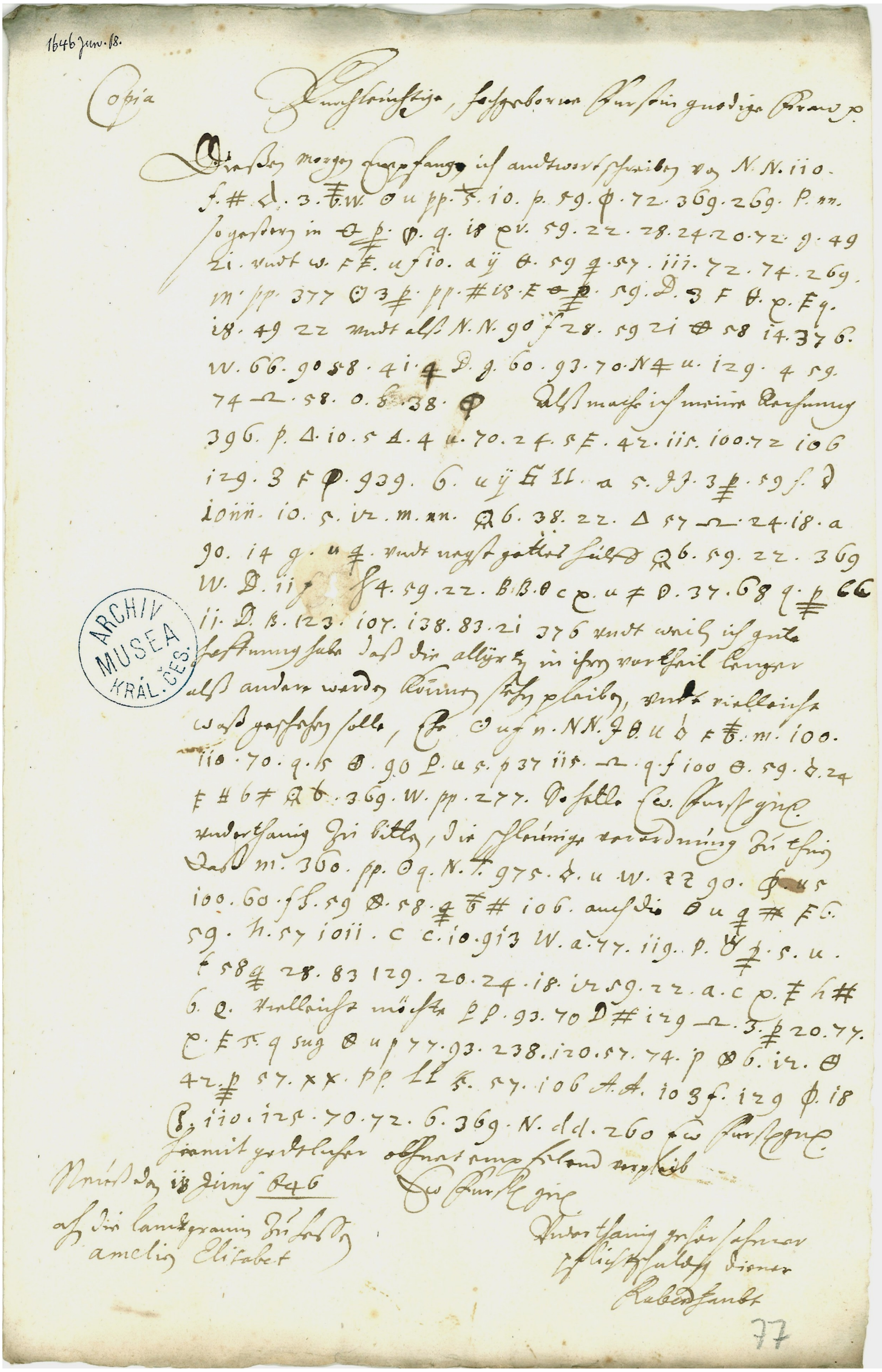


Figure 11: Area of Karel Rabenhaput ze Suché’s operations on the Rhine (Wikipedia Commons, 2020)



1646 Jun. 18.

Copia

Durchleüchtige, hochgeborne Fürstin gnedige Frau p.

[illegible]

Hiemit göttlicher obhut uns befehlend verbleib

E. Fürstl. gn.

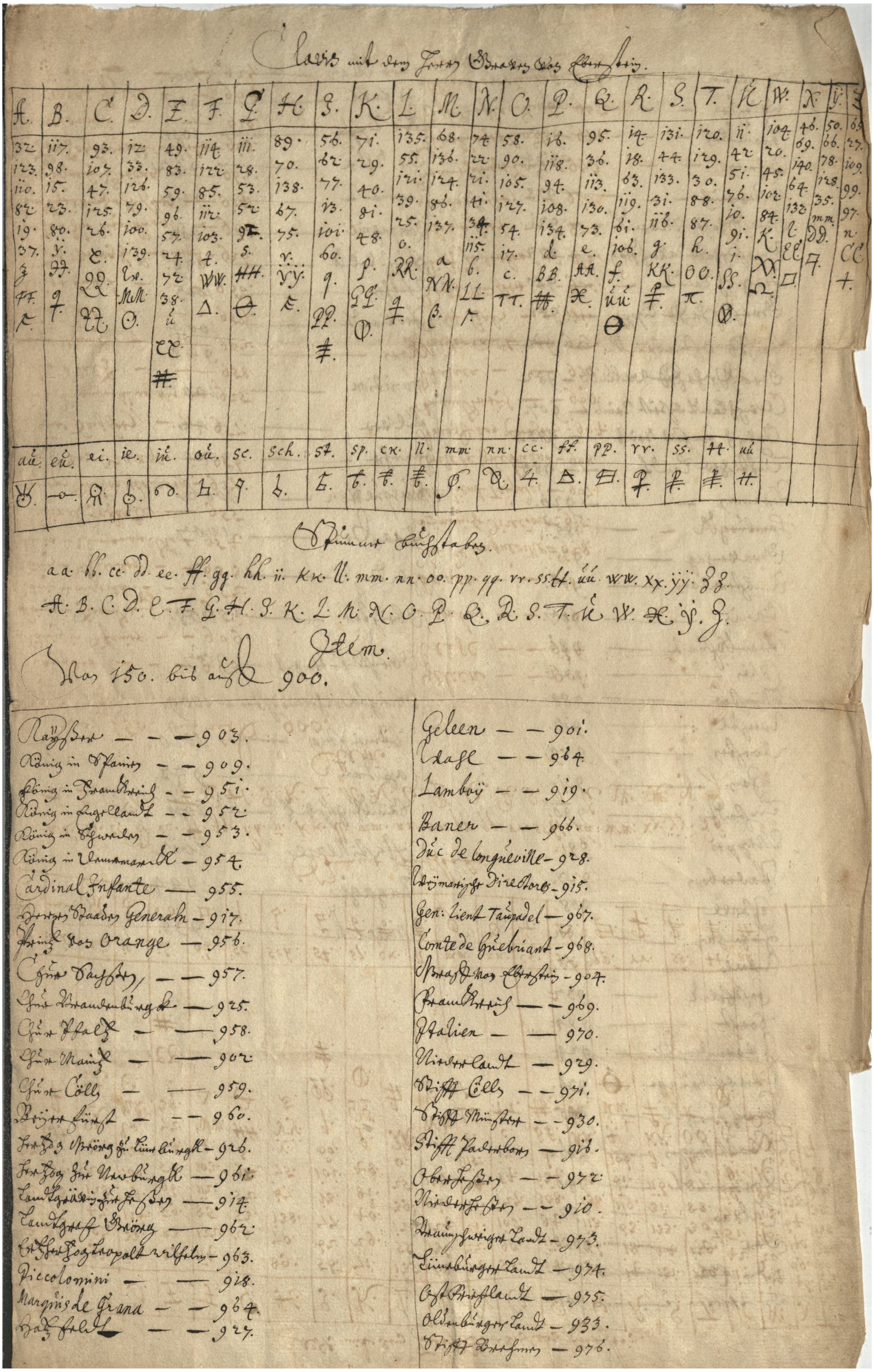
Underthenig gehorsamer pflichtschuldigster diener

Rabenhaupt

an die landtgrävin zu hessen amelia Elisabet

77

Figure 12: Encrypted message. SOA v Plzni, FA Trauttmansdorff, inv. nr. 125.



Clavis mit dem Herrn Obristen von Eberstein.

A.	B.	C.	D.	E.	F.	G.	H.	I.	K.	L.	M.	N.	O.	P.	Q.	R.	S.	T.	U.	W.	X.	Y.	Z
32	117.	93.	12	49.	114	111.	89.	56.	71.	135.	68.	74	58.	16.	95.	14.	131.	120.	11.	104	46.	50.	65
123.	98.	107.	33.	83.	122	28.	70.	62	29.	55.	136.	22	90.	118.	36.	18.	44.	129.	42	20.	69.	66.	27.
110.	15.	47.	126.	59.	85.	53.	138.	77.	40.	121.	124.	21.	105.	94.	113.	63.	133.	30.	51.	45.	140.	78.	109.
82	23.	125.	79.	96.	112.	52	67.	13.	81.	39.	86.	41.	127.	108.	130.	119.	31.	88.	76.	102	64.	128.	99.
19.	80.	26.	100.	57.	103.	92.	75.	101.	48.	25.	137.	34.	54.	134.	73.	61.	116.	87.	10.	84.	132	35.	97.
37.	ÿ.	ε.	139.	24.	4.	5.	v.	60.	p.	0.	a	115.	17.	d.	e.	106.	g.	h.	91.	K.	l.	mm.	n.
ʒ	ʒʒ.	QQ.	ix.	72	WW.	HH.	ÿÿ.	q.	GG.	RR.	NN.	6.	c.	BB.	AA.	f.	KK.	OO.	i.	W.	EE.	DD.	CC
FF.	q.	ZZ.	MM.	38.	Δ.	θ.	E.	PP.	Φ.	ǂ.	ℭ.	LL.	TT.	⋕.	ϰ.	üü.	ǂ.	π.	SS.	℧.	◇.	7.	+.
r.	ƒ.		⊙.	ü.				ǂ.				r.				Θ			⊗.				
				CC.																			
				⋕.																			

aü.	eü.	ei.	ie.	iü.	oü.	sc.	sch.	st.	sp.	ck.	ll.	mm.	nn.	cc.	ff.	pp.	rr.	ss.	tt.	uu
♅.	-o-.	♀.	⚲.	6ꝺ.	b.	q.	b.	E.	b.	ƀ.	ƀ.	S.	Q.	4.	Δ.	▱.	ꝑ.	ꝑ.	ǂ.	H.

Stummen Buchstaben.

aa. bb. cc. dd. ee. ff. gg. hh. ii. kk. ll. mm. nn. oo. pp. qq. rr. ss. tt. üü. ww. xx. ÿÿ. zz.

A. B. C. D. E. F. G. H. I. K. L. M. N. O. P. Q. R. S. T. U. W. X. Y. Z.

Item.

Von 150. bis auff 900.

Kayser	903.
König in Spanien	909.
König in Franckreich	951.
König in Engellandt	952.
König in Schweden	953.
König in Dennemarck	954.
Cardinal Infante	955.
Herren Staaden Generaln	917.
Printz von Orange	956.
Chur Sachsen	957.
Chur Brandenburgk	925.
Chur Pfaltz	958.
Chur Maintz	902.
Chur Cölln	959.
Beyern fürst	960.
Hertzog Georg zu Lüneburgk	926.
Hertzog zu Newburgk	961.
Landgräffin zu Hessen	914.
Landgraff Georg	962.
Ertzhertzog Leopolt Wilhelm	963.
Piccolomini	918.
Marquis de Grana	964.
Hatzfeldt	927.

Geleen	901.
Haße	964.
Lamboÿ	919.
Baner	966.
Duc de longueville	928.
Weymarische Directores	915.
Gen: Lieut Taupadel	967.
Comte de Guebriant	968.
Obristen von Eberstein	904.
Franckreich	969.
Italien	970.
Niederlandt	929.
Stifft Cölln	971.
Stifft Münster	930.
Stifft Paderborn	916.
Oberhessen	972.
Niederhessen	910.
Braunschweiger Landt	973.
Lüneburger Landt	974.
Ost Frießlandt	975.
Oldenburger Landt	933.
Stifft Bremen	976.

Figure 13: The key we used to decrypt the cryptogram. HLA-HStAM Best. 4d Nr. 1218.

A Cryptogram Transcription

NN.110.f.s1.s2.z.s3.W.s4.u.pp.s5.10.f.59.s6.72.369.269.P.nn.s7.s8.s9.q.18.s10.r.59.22.28.24.20.72.g.49.21.w.s11.s12.u.f.10.a.y.s13.59.s14.57.111.72.74.269.M.pp.377.s4.z.s8.pp.s1.18.s12.s7.s16.59.D.s17.s11.s18.s10.s12.q.18.49.22.NN.90.f.28.59.21.s19.58.14.376.W.CC.90.58.41.s20.D.g.60.93.70.N.s20.u.129.s21.59.74.s22.58.o.h.38.O.396.P.s23.10.s24.s25.s21.u.70.24.s24.s12.42.115.100.72.106.129.s17.s11.s6.939.b.u.y.s26.ll.s27.s24.II.z.s16.59.f.s28.1011.10.s24.12.M.nn.s27.b.38.22.s23.57.s22.24.18.a.90.14.g.u.s14.s27.b.59.22.369.W.s9.11.f.h.s21.59.22.BB.s18.c.s10.u.s20.s4.37.68.q.s16.CC.11.D.s17.123.107.138.83.21.376.s4.u.f.NN.z.s18.u.s2.s11.s3.M.100.110.70.q.s24.59.90.s29.u.s24.p.37.115.s22.q.f.100.s13.59.s2.24.s12.s1.b.s20.s27.b.369.W.pp.277.M.360.pp.s4.q.N.T.975.s2.u.W.ZZ.90.s29.u.s24.100.60.f.h.59.s19.58.s14.s30.s1.106.s13.u.s14.s1.s12.b.59.h.57.1011.CC.10.913.W.a.77.119.P.s30.s20.s.u.t.58.s14.28.83.129.20.24.18.12.59.22.a.c.s10.s12.h.s1.b.Q.PP.93.70.D.s1.129.s22.z.s8.20.77.s10.s12.s5.q.s.u.g.s19.u.s32.77.93.138.120.57.74.P.s19.b.12.s13.42.s16.57.XX.PP.LL.s5.57.106.AA.10.z.f.129.s28.18.s17.110.125.70.72.b.369.N.dd.260

Note: Elements starting with letter *s* in a combination with numbers (starting with s1) represents the symbols.

B English Translation of the Deciphered Message

Copia

Enlightened, highly noble Princess, a gracious Lady. This morning I received an answer from Marshal de Turenne, who was in Euskirchen yesterday and had a camp nearby. He wanted to march today and settle in front of the Zons tomorrow. So I do my math, I have 1,500 infantry with one battery of cannons and one mortar. With God's help, we'll do a short process with that [Zons]. And because I firmly believe that it is an advantage of the Allies that they endure [in the siege] longer than others, perhaps what is to happen will happen before the Marshal gets there. Therefore, I would like to ask your Princely Grace to order immediately that the levies from East Frisia and the borrowed cannons be sent to Wesel. I would like to do something important and arrange [for the army] good winter quarters. I hereby commend myself to the beneficial protection of Your Princely Grace and remain

a subject to Your Princely Grace, an obedient servant Rabenhaubt

Neuss, June 13th, 1646, to Landgravine of Hessen, Amalia Elizabeth

Cryptodiagnosis of “Kryptos K4”

Richard Bean
School of Information Technology and Electrical Engineering
University of Queensland, Australia 4072
r.bean1@uq.edu.au

Abstract

The sculpture “Kryptos” at the Central Intelligence Agency in Langley, Virginia, contains four encrypted passages. The last, known as “K4” and consisting of 97 letters, remains unsolved.

In this work, we look at unusual statistical properties of the K4 ciphertext, together with the known plaintext, using Monte Carlo sampling to perform permutation testing. This provides evidence strongly indicating a definite “one-to-one” relationship between corresponding plaintext and ciphertext letters. It also points toward a possible encryption method which could account for most or all of the observed properties. This is the “Gromark” cipher invented by Hall (1969, 1975) and analyzed by Blackman (1989).

1 Introduction

The “Kryptos” sculpture was installed at the Central Intelligence Agency (CIA) in Langley, Virginia in November 1990. The sculptor was Jim Sanborn and the cryptographic consultant was Ed Scheidt, who retired from the CIA in December 1989. The sculpture contains four encrypted messages totalling 865 letters plus 4 question marks.

Scheidt has indicated that the codes were designed to be solved in five, seven or ten years.

The first three sections were solved independently by three different teams or individuals: an NSA team in 1992, David Stein from the CIA in 1998, and Jim Gillogly in 1999. The fourth section, “K4”, consisting of 97 letters remains unsolved and its encryption method remains publicly unknown. During the period 2010 to 2020, four parts of the K4 plaintext with locations were released by the sculptor, totalling 24 letters. Further details may be found in Dunin and Schmeh (2020).

Callimahos (1977) and Lewis (1992) describe the process of diagnosis of an unknown cipher type. Callimahos, in a chapter entitled “Principles of Cryptodiagnosis”, sets out a process of hypothesis formulation and hypothesis testing. This involves arrangement and rearrangement of the data to disclose nonrandom characteristics, followed by recognition and explanation of these characteristics. The chapter headers are: manipulating the data, recognizing the phenomena, and interpreting the phenomena.

Lewis states that the task of an analyst is finding, measuring, explaining, and exploiting a phenomenon (or phenomena). Writing about cipher type diagnosis, he describes the search for “something funny” or “finding the phenomena”.

Since these publications, Mason (2012) prepared a table of cipher statistics for many American Cryptogram Association (ACA) types, with associated random forest (2013) and neural net (2016) classifiers. Nuhn and Knight (2014) also developed a classifier for ACA cipher types using a support vector machine approach.

In this paper we attempt to measure some of the interesting phenomena seen in K4 and provide possible explanations. We perform statistical testing using Monte Carlo sampling and describe one possible encryption method, the Gromark cipher of the ACA, and its variants. Finally we conduct an extensive search of the Gromark key space for various bases and key primer lengths before discussing our conclusions.

2 Analysis

Good (1983) commented on the practice of looking at a sample ciphertext and deciding on a test of significance based on the observed data, instead of running a standard series of tests. The passage is worth quoting in full to describe the risks and rewards of such an approach.

> *... it is sometimes sensible to decide on a significance test after looking at a sample. As I've said elsewhere this practice is dangerous, useful, and often done. It is especially useful in cryptanalysis, but one needs good detached judgment to estimate the initial probability of a hypothesis that is suggested by the data. Cryptanalysts even invented a special name for a very far-fetched hypothesis formulated after looking at the data, namely a "kinkus" (plural: "kinkera"). It is not easy to judge the prior probability of a kinkus after it has been observed.*

2.1 Ciphertext analysis

One of K4's most prominent unusual features is the number of repeated bigrams when the ciphertext is written at width 21 (Hannon, 2010; LaTurner, 2016; Kirchner, 2003); see Table 1.

O B K R U O X	O G H U L B S	O L I F **B B** W
F L R V **Q Q** P	R N G K **S S** O	T W T Q S J Q
S S E K **Z Z** W	A T J K L U D	I A W I N F B
N Y P V **T T** M	Z F P K W G D	K Z X T J C D
I G K U H U A	U E K C A R	

Table 1: K4 ciphertext written at width 21

If we consider the 76 bigrams formed vertically (starting with OF and finishing with GR), there are 11 repeated bigrams (AZ BS IT LS LW PK QZ SN WA ZT KK). This value is in line with the expected number of repeated bigrams if a typical English plaintext was written out at width 21; for example, testing all 97-letter contiguous subsets of the King James Bible gives an average value of 9.7 repeated bigrams at width 21.

If we perform Monte Carlo sampling and take a large number of permutations of the ciphertext (Good, 2013), we can estimate the proportion of permutations of the ciphertext which would have at least this number of repeated bigrams. In this case, this proportion is approximately one in 6,750. Programs written in C to calculate values in this paper are provided via Github.[1]

The recently solved (Oranchak et al., 2020) "Zodiac 340" cipher also had a similar property (Daikon, 2015; Van Eycke, 2015). A relatively high number of repeated bigrams was seen at width 19 in the ciphertext. The cipher was ultimately found to be a combination of transposition and homophonic substitution. The width 19 property can thus, after the fact, be deemed "causal" as the enciphering process caused this property to appear.

2.2 Seriated ciphers

The "seriated Playfair" cipher of the ACA might provide a partial aesthetic explanation for the width 21 patterns. This cipher is digraphic and works by performing Playfair encryption on vertical pairs of letters. That is, any given pair of letters in plaintext (p_1, p_2) maps to another pair of letters (c_1, c_2) in a one-to-one fashion. Thus, numbering the positions 0 to 96, the repeated "BS" bigrams at positions 12/33 and 18/39 would reflect the same underlying plaintext, or "AZ" at positions 49/70 and 57/78. Similarly, the "double box cipher" or Doppelkastenschlüssel, sometimes referred to as "double Playfair", described by David (1996) is a digraphic cipher which required seriation at width 21.

There are also several arguments against this as a method:

- according to the "ACA Cipher Statistics" webpage of Mason (2012), the index of coincidence (IC) of a typical "seriated Playfair" ciphertext is 0.048 with standard deviation 0.003 versus K4's IC of 0.036
- 26 letters occur in the ciphertext, while the most common Playfair variant uses only 25 from a 5x5 square
- the doublet "KK" is present, which cannot occur in standard Playfair
- a plain interpretation is that there are 97 ciphertext letters, an odd number, while Playfair works on pairs of letters. As 97 is also prime, this is also an argument against the Hill cipher suggestion of Bauer et al (2016).

The original description of the Playfair cipher by Wheatstone entitled "Specimen of a Rectangular Cipher", seen in Kahn (1996, p. 199) uses

[1] https://github.com/RichardBean/k4testing

a 9x3 rectangle, and enciphers doubled letters, which would account for the last three observations. We could take the question mark before "OBKR" on the sculpture as the initial character, with 27 different characters and 98 ciphertext characters. The low IC could then be accounted for by careful selection of the key.

However, these theories all became moot after the "BERLIN" plaintext clue was released in November 2010. This corresponds to the ciphertext "NYPVTT". Thus, if a seriated digraphic cipher at width 21 had been used to encipher the plaintext, we would have two different plaintext bigrams ending in "I" and "N" both mapping to "ZT", which is impossible. As the letter "K" in the 2014 plaintext clue of "CLOCK" also enciphered to "K" this ruled out the use of standard Playfair for the vertical bigrams.

We might also wish to check a width of seven, based on a purely aesthetic argument, since 98 characters is seven pairs of rows with seven characters per row. Similarly, the "NORTHEAST" plaintext clue was released in January 2020, which corresponded to letters 26-34 in the ciphertext, "QQPRNGKSS". If a seriated digraphic cipher had been used to encipher the plaintext at a width of seven, we would have two different plaintext bigrams ending in "N" and "O" both mapping to "BQ", again impossible.

2.3 Other observations

Many other statistical anomalies have been noted by others. Previous solvers of Kryptos have noted the repeated doublets (BB, QQ, SS, ZZ and TT) in the same columns when the ciphertext is written at width seven. These letters are shown in bold in Table 1. An NSA analyst (Redacted, 2007) and Gillogly (Gillogly, 1999a) suggested this property could be due to combined polyalphabetic substitution and transposition. The width 21 property could also be used to argue for combined transposition and substitution, as with the Zodiac 340 cipher; however, this paper argues against a transposition step.

Stehle (2000) noted that the ciphertext segment "DIAWINFBN" has the property that when converted to numbers (from the standard alphabet), 0 to 25, the difference between the first five letters and the corresponding letters four positions right is 5 (modulo 26). Thus *I* minus *D* corresponds to 8 minus 3, *N* minus *I* to 13 minus 8, and so on.

These observations are unusual, and may well be causal, but were ultimately considered harder to measure, explain or exploit than the observations discussed here.

2.4 Known plaintext analysis

The 24 known plaintext letters are as follows: "FLRVQQPRNGKSS" in cipher corresponds to "EASTNORTHEAST" in plain and "NYPVTTMZFPK" in cipher corresponds to "BERLINCLOCK" in plain.

Materna (2020) noted that for the known K4 plaintext, where the plaintext letters are in the set $\{K,R,Y,P,T,O,S\}$ the corresponding ciphertext letters are very close in the standard alphabet to the plaintext letters. Thus, the 10 shortest distances (the so-called "minor differences" (Friedman, 1954)) sum to 21, as shown in Table 2, for a mean of 2.1.

Plaintext letter	S T O R T S T R O K
Ciphertext letter	R V Q P R S S P F K
Distance	1 2 2 2 2 0 1 2 9 0

Table 2: Minor differences between plain and ciphertext letters

Monte Carlo sampling by permuting the ciphertext letters of K4 demonstrates this occurs only in about one in 5,520 permutations of K4 ciphertext letters.

With the release of 24 known plaintext characters, we can create a table showing, for each repeated plaintext letter, what the corresponding ciphertext letter set is, and then measure the shortest distance between each of the ciphertext letters. The repeated known plaintext letters are A, C, E, L, N, O, R, S and T. Table 3 measures the minor differences between the ciphertext letters corresponding to each, producing 13 values.

We note that the mean is 3.6 and 10 of 13 values are less than five. Performing Monte Carlo sampling and permuting the ciphertext randomly, we found that about one in 240 permutations have a mean less than or equal to 3.6, while about one in 310 permutations have at least 10 values less than five.[2]

[2] Looking at the distances in the Kryptos alphabet, 7 of 13 values are multiples of 5. If the plaintext letters are numbered 0-25 from the standard alphabet and the ciphertext letters are numbered 0-25 from the Kryptos alphabet reversed, then the sequence plain minus cipher modulo 26 is calculated, 13 of 24 values are multiples of 5; randomly permuting the cipher-

Plain	Cipher	Distances in standard alphabet	Distances in "KRYPTOS" alphabet
A	KL	1	9
C	MP	3	11
E	FGY	1,7,8	1,10,11
L	VZ	4	3
N	QT	3	10
O	QF	11	8
R	PP	0	0
S	RS	1	5
T	RSV	1,3,4	5,5,10

Table 3: K4 distances between cipher letters corresponding to repeated plaintext letters

These observations are unusual and strongly suggest that "one-to-one" encryption of single letters to single letters is occurring; that is, there is no transposition involved in the encryption process. It is of course possible that a new encryption algorithm never before seen is in use, but this would render solution very unlikely.

2.5 The Gromark Cipher

Instead, we suggest that these observations are most compatible with the "Gromark" cipher (Hall, 1969; Rogot, 1975a; American Cryptogram Association, 2016). This was also a suggestion of Gillogly (1999a; 1999b; 2004b; 2004a) before known plaintext was made available in 2010.

Gromark as described by Hall (1969) operates by using a "primer" of five digits, which is expanded to form a key of the length of the plaintext, using a "lagged Fibonacci generator" by continually adding the first two available digits, starting with and including the primer, to get the next key digit (modulo 10).

A plain and cipher alphabet are used; in ACA puzzles, the standard alphabet is used for the plain (A to Z) and the primer is given. The plain and cipher alphabets are written in rows with the plain on the top. The key digit corresponding to a particular plaintext letter is then used to count that many steps right from the corresponding letter in the cipher alphabet to produce each ciphertext letter.

Rogot (1975b) pointed out that, analogously to the "Quagmire" cipher types, various kinds of plain and cipher alphabets can be used. They can be standard or keyword-based. Lewis (1992, p. 116) wrote about using the same alphabet for plain and cipher, or using a superadditive numeric key.

Thus, one explanation for the "minor differences" observations in the "Analysis" section above could be that the cipher alphabet is "near" the standard A-Z alphabet, perhaps based on a keyword, and then the minor differences between ciphertext letters corresponding to repeated plaintext letters are small numbers.

Blackman (1989) considered further variations, such as using a non-decimal base, varying the length of the primer, or as in Barker (1984), using a different rule for building up the key.

Holden (2018) used Gromark as an illustration of the concept of the "linear feedback shift register" (LFSR) which is more fully described in Barker (1984).

Rogot (1975a), Deavours (1977a) and Blackman (1989) all noted that with an even base and a five digit primer, there is an underlying structure of length 21 in the key and ciphertext. For example, Deavours remarked that writing such Gromark ciphertext out at width 21, each column is encrypted by either all even or all odd key digits, and with enough ciphertext, the underlying structure of the primer is revealed. Blackman extended this approach to recovery of the base and length of the primer by examination of the index of coincidence, although typically a ciphertext of length much greater than 100 letters is required.

The more general concept of inferring a sequence generated by a pseudo-random number generator (PRNG)[3] from known terms is dealt with in more detail in Reeds (1977), Plumstead (1982), Knuth (1985), and Boyar (1989). For instance, Boyar wrote about a linear congruential recurrence with n terms:

$X_i = a_1 X_{i-1} + a_2 X_{i-2} + \ldots + a_n X_{i-n} + a_{n+1}$ (mod m)

In the case of the standard ACA Gromark cipher, we have $m = 10$, $n = 5$, $a_4 = a_5 = 1$, and $a_1 = a_2 = a_3 = a_6 = 0$.

If a base two, five digit Gromark cipher is used, with standard English plaintext taken from the King James Bible, simulations indicate that for any given key, about one in 10 ciphertexts will have the property of at least 11 repeated vertical

text, this is a 1 in 1,470 result. This might imply a method involving 5x5 Polybius squares, such as a conjugated matrix bifid; but nothing was found.

[3] It is curious, but probably not relevant, that this is the only 4-letter sequence occurring twice in the ciphertext part of the sculpture.

bigrams at width 21. By way of explanation, Table 4 shows the key expansion beginning with the primer 00001. Ten of the values in each complete row are the digit 1, and the vertical bigrams are enciphered with either 00 or 11; thus, enciphering will tend to preserve existing patterns of vertical bigrams present in the plaintext.

0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1
0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1
0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1
0	0	0	0	1	0	0	0	1	1	0	0	1	0	1	0	1	1	1	1	1
0	0	0	0	1	0	0	0	1	1	0	0	1								

Table 4: Gromark binary key

This "one in 10" proportion is very different from the "one in 6,750" result obtained from the Monte Carlo sampling above. Similarly, for a particular plaintext and sufficiently large base, it is generally simple to find a five digit primer which results in the ciphertext having the property of a large number of repeated vertical bigrams.

The known K4 plaintext now indicates the base must be at least three, because some plaintext letters encipher to at least three different ciphertext letters.

Additional arguments for the use of the Gromark cipher include:

- Gromark was described by Blackman (1989) as a "pencil-and-paper field cipher". Similarly, Scheidt has been quoted as stating: "K4 cryptography is similar to what would be provided agents or pilots in case of capture" (Hannon, 2011);

- Gromark is definitely "more than one stage" as the primer must be expanded to the complete key. Scheidt stated in 2015 that "[he] would consider [K4 encryption] [to be] more than one stage". (Schmeh, 2015);

- Gromark does not involve transposition and enciphers letters to letters. Sanborn has been quoted as stating: "BERLINCLOCK in plain matches directly with NYPVTTMZFPK. It is a one-to-one match with plain B taken, has the encipherment done to it, and out pops a cipher N, plain E is then enciphered to a cipher Y" (Bogart, 2019);

- Gromark is one of the few ACA ciphers in Mason's table to result in a "flat" index of coincidence, that is, one close to the value 1/26 = 0.03846. The IC value of K4 is 336/97/96 = 0.03608;

- The unique feature of the Berlin Clock is that it uses base 5 or 12 arithmetic (Schridde, 2020) and Sanborn has stated "you'd better delve into that particular clock" (Schwartz, 2014);

- Scheidt hints about base arithmetic in the 2015 interview above and also in 2020: "if you can change the language base then it becomes in my favor and not your favor of trying to break it. It becomes more of a challenge now, when it was used as the mask it was current, 2020 secret." (Jacobs, 2020). This may refer to Blackman (1989);

- The raised letters on the sculpture stylized as "$D^{YA}H^{R}$" may refer to a Gromark five-digit primer and are reminiscent of binary.[4] Indeed, the "Vimark" cipher (Dickerhoof, 1971) is just Gromark at base 26 using numeric values of letters.

Arguments against use of the Gromark cipher are:

- The initial ACA experience showed that Gromark encryption is error-prone and all ACA challenges are now provided with a check digit. However, the most error-prone aspect is the key expansion stage; this could be checked by a third party without revealing the plaintext.

- Sanborn has stated that he is an "anathemath" on several occasions (Allsop, 2010);

- At the 2011 "Kryptos Dinner" at the Zola Restaurant in Washington DC, Scheidt stated "[K4 cryptography] is not mathematical (although this does not preclude it being modeled mathematically), it is simple, can be remembered, and executed years later when used with the correct key word/s." (Hannon, 2011)

[4] Alternatively, this may be a reference to historical codes. Telegraph and telex messages were charged per word sent. To reduce costs, large international companies (mostly banks) developed and used five letter codes. Codes such as Acme had error correction features which in time were replaced by binary error correction systems.

- Typically, the ACA version of the cipher uses a "standard" plain alphabet, that is, A-Z in order. With the release of the "CLOCK" crib, C and R in the plaintext alphabet would both need to map to P in the ciphertext alphabet but are more than 10 places apart in the standard alphabet.

- Assuming Gromark is in use, there is a tension between the observations concerning the IC, the ciphertext bigrams at width 21, and the base chosen. A low base means that the number of different vertical bigrams in the key may be low; but on the other hand, very few ciphertext outputs will have an IC as low as 0.036. If, however, the encipherer wants to deliberately insert the width 21 property, with a higher base, they have many primers to choose from to achieve that property.

The given plaintext maps plain T to cipher V at position 24, and L to V at position 66 (numbering the positions 0 to 96). If Gromark was used as the cipher, this implies the key is not repeating at period 21 or 42. Perhaps the period is 63 or 84. A period of 63 is reminiscent of an *m*-sequence or "maximal length sequence" as seen in a particular example of Golomb and Gong (2005). In this example, Golomb and Gong produced an *m*-sequence over $\mathbb{F}_{2^2}$ of degree 3 with period 63 using the irreducible polynomial x^2+x+1.

This sequence (extended to 84 entries) is shown in Table 5, with the field elements in $\mathbb{F}_{2^2}$ $\{0,1,\beta,\beta^2\}$ replaced by $\{0,1,2,3\}$. As with the binary Gromark key, the number of distinct vertical bigrams is quite low; only four: $00,12,23$ and 31.

1	1	1	0	1	3	0	0	1	1	2	3	0	1	0	3	1	3	1	1	3
2	2	2	0	2	1	0	0	2	2	3	1	0	2	0	1	2	1	2	2	1
3	3	3	0	3	2	0	0	3	3	1	2	0	3	0	2	3	2	3	3	2
1	1	1	0	1	3	0	0	1	1	2	3	0	1	0	3	1	3	1	1	3

Table 5: Golomb and Gong *m*-sequence of period 63

Meanwhile, a period of 84 is often seen with base eight and primer length five, further explained below.

3 Search

Given the 24 known plaintext letters, we discovered that a simulated annealing search (see for instance Lasry (2018) using hexagram statistics for scoring as in Bean (2020)) for the plain and cipher alphabets would eventually converge for a given key, when the alphabets were allowed to vary.

A set of inequalities and equalities was developed to narrow down the possible primers for base and primer length possibilities. By means of this reduction, the entire search space for base 10, length 5 was examined thoroughly.

If we number the numeric key from 0 to 96, so that each key digit is denoted by $k_0,\dots,k_{96}$, we can write out the relationships between the 24 known plaintext and ciphertext letters. The "p" and "c" functions here calculate the numerical equivalent of a given letter in the plaintext and ciphertext alphabets (0 to 25). Then, pairs of these relationships imply relationships between digits of the key.

- $p(T)+k_{24}=c(V), p(T)+k_{28}=c(R), p(T)+k_{33}=c(S) \implies k_{24}\neq k_{28}, k_{28}\neq k_{33}, k_{24}\neq k_{33}$

- $p(E)+k_{21}=c(F), p(E)+k_{30}=c(G), p(E)+k_{64}=c(Y) \implies k_{21}\neq k_{30}, k_{21}\neq k_{64}, k_{30}\neq k_{64}$

- $p(R)+k_{27}=c(P), p(R)+k_{65}=c(P) \implies k_{27}=k_{65}$

- $p(N)+k_{68}=c(T), p(N)+k_{25}=c(Q) \implies k_{68}\neq k_{25}$

- $p(A)+k_{22}=c(L), p(A)+k_{31}=c(K) \implies k_{22}\neq k_{31}$

- $p(L)+k_{66}=c(Q), p(L)+k_{70}=c(Z) \implies k_{66}\neq k_{70}$

- $p(O)+k_{26}=c(Q), p(O)+k_{71}=c(F) \implies k_{26}\neq k_{71}$

- $p(C)+k_{69}=c(M), p(C)+k_{72}=c(P) \implies k_{69}\neq k_{72}$

- $p(S)+k_{23}=c(R), p(S)+k_{32}=c(S) \implies k_{23}\neq k_{32}$

- $p(O)+k_{71}=p(E)+k_{21}=c(F) \implies k_{71}\neq k_{21}$

- $p(N)+k_{25}=p(O)+k_{26}=c(Q) \implies k_{25}\neq k_{26}$

- $p(T)+k_{24}=p(L)+k_{66}=c(V) \implies k_{24}\neq k_{66}$

- $p(A) + k_{31} = p(K) + k_{73} = c(K) \implies k_{31} \neq k_{73}$
- $p(H) + k_{29} = p(B) + k_{63} = c(N) \implies k_{29} \neq k_{63}$
- $p(S) + k_{32} = p(T) + k_{33} = c(S) \implies k_{32} \neq k_{33}$
- $p(I) + k_{67} = p(N) + k_{68} = c(T) \implies k_{67} \neq k_{68}$
- $p(R) + k_{27} = p(C) + k_{72} = c(P) \implies k_{27} \neq k_{72}$
- $p(S) + k_{23} = p(T) + k_{28} = c(R) \implies k_{23} \neq k_{28}$

For base 10, primer length 5, out of the initial 99,999 possible non-zero keys, this left 1,040 remaining. If the digits in each key were randomly chosen and uncorrelated within each key, we have 21 inequalities and one equality at base 10; the proportion of keys satisfying all these would be $(\frac{9}{10})^{21}(\frac{1}{10})$ = 0.01094, which implies in some sense that the Gromark key digits are approximately "random".

After this, we can apply further restrictions. The SageMath software (Stein, 2007) allows us to compute the Gröbner basis for the set of equations showing the relationship between the 24 plaintext and ciphertext letters. This leads to another set of 14 inequalities and one equality which each have either four or six terms. The full set may be found in the Github source.

This process ultimately showed that only 39 different primers (for the base 10 five digit case) could lead to the 24 letters of known plaintext in the correct positions.

Two of these primers, 26717 and 84393, are equivalent, up to length 97, using a variation of an observation of Blackman (1989): the keys are inverses of each other (modulo 10). So, for any given plain and cipher alphabet P and C, the result of encrypting with 26717 is equal to the result after encrypting with 84393, with the original alphabets P and C reversed. See Table 6.

These are the only two of the 39 keys which use only nine different digits. Of the 99,999 keys of length 97, only 88 keys do not contain the zero digit anywhere.

One of Blackman's ideas is that, for a given numeric key, the index of coincidence can be calculated for the ciphertext letters corresponding to each digit, and the average taken. This is the method used to determine the most likely key primers. In this case, starting with the key 98800 gives an index of coincidence of 0.0625, which is the highest of any of the keys and closest to the index of coincidence of typical English plaintext.

2 6 7 1 7 8 3 8 8 5 1 1 6 3 6 2 7 9 9 8 9
6 8 7 7 5 4 5 4 2 9 9 9 6 1 8 8 5 7 9 6 3
2 6 5 9 5 8 1 4 4 3 9 5 8 7 2 4 3 5 9 6 7
8 4 5 3 5 2 9 8 8 7 1 7 6 5 8 8 3 1 3 6 1
4 4 9 7 5 8 3 6 2 3 1 9 8 5

Table 6: Expansion of 26717 primer at base 10

The restrictions above can be applied to primers of other bases and key lengths: for instance, the only possible base 10, four digit primers are 3301[5], 6740, and 9903, and the four possible base eight, five digit primers include 00351 and 00537. As seen in Table 7 the expansions of these base eight keys have period 84 and the extra property that all columns at width seven, as well as at width 21, have either all odd or all even key digits.

0 0 3 5 1 0 3	0 6 1 3 3 6 7	4 6 1 5 3 2 7
6 0 5 1 5 6 5	6 6 3 3 3 4 1	6 6 7 5 7 4 5
4 4 3 1 1 0 7	4 2 1 7 3 6 3	0 2 1 1 3 2 3
2 4 5 5 5 6 1	2 2 3 7 3 4 5	2 2 7 1 7 4 1
0 0 3 5 1 0 3	0 6 1 3 3 6 7	

Table 7: Expansion of 00351 primer at base eight

After this, different key expansion rules can be tried, perhaps inspired by the raised letters on the sculpture. We restricted ourselves to rules where the first digit in the primer (shift register) is used in the generation function, as explained in Beker and Piper (1982, p. 183).

Although many plaintexts close to ordinary English were discovered, none were entirely convincing. If a Gromark variant was indeed used in the K4 encryption process, with a more general key expansion rule, it becomes difficult to test all the possibilities. Instead, it may be worth considering implications of the other observations in this paper.

4 Conclusion

With the use of Monte Carlo sampling analysis, the known plaintext released by Sanborn pro-

[5] Which reminds one of the Internet mystery "Cicada 3301"

vides strong indications that transposition is not involved in the K4 encryption process.

If the "Gromark" cipher of the ACA was used as the encryption method, this would explain many of the observed properties of the ciphertext and known plaintext. The "unicity distance" (Deavours, 1977b) of the Gromark cipher is approximately 48 letters, not accounting for the numeric primer, which means the solution would be unique at a ciphertext length of 97 letters.

As the Gromark cipher is the inspiration for another high-security cipher of Rubin (1996) such a cipher may be quite difficult to solve, fulfilling Sanborn's stated intention of it "going on for a century, hopefully long after my death." (Sanborn, 2009)

Acknowledgments

The author wishes to thank Ed Hannon for his extensive correspondence on this subject, and also Jim Gillogly and Eleanor Joyner (SCRYER and HONEYBEE of the ACA).

References

Laura Allsop. 2010. Kryptos sculpture inspires hope in weary code-breakers. `http://edition.cnn.com/2010/SHOWBIZ/11/26/CIA.sculpture.clue/index.html`.

American Cryptogram Association. 2016. Gromark: Gronsfeld with mixed alphabet and running key. `https://www.cryptogram.org/downloads/aca.info/ciphers/Gromark.pdf`.

Wayne G Barker. 1984. *Cryptanalysis of Shift-Register Generated Stream Cipher Systems*, volume 39. Aegean Park Press.

Craig Bauer, Gregory Link, and Dante Molle. 2016. James Sanborn's Kryptos and the matrix encryption conjecture. *Cryptologia*, 40(6):541–552.

Richard Bean. 2020. The use of Project Gutenberg and hexagram statistics to help solve famous unsolved ciphers. In *Proceedings of the 3rd International Conference on Historical Cryptology HistoCrypt 2020*, number 171, pages 31–35. Linköping University Electronic Press.

Henry Beker and Fred Piper. 1982. *Cipher systems: the protection of communications*. Northwood Books, London.

Deane R Blackman. 1989. The Gromark Cipher, and Some Relatives. *Cryptologia*, 13(3):273–282.

Bob Bogart. 2019. CNN Documentary about Kryptos a making of report with many photographs. `https://scienceblogs.de/klausis-krypto-kolumne/2019/03/15/cnn-documentary-about-kryptos-a-making-of-report-with-many-photographs/`.

Joan Boyar. 1989. Inferring sequences produced by pseudo-random number generators. *Journal of the ACM (JACM)*, 36(1):129–141.

Lambros D. Callimahos. 1977. *Military Cryptanalytics Part III*. National Security Agency. `https://www.governmentattic.org/39docs/NSAmilitaryCryptalyticsPt3_1977.pdf`.

Daikon. 2015. Things I noticed about Z340. `http://www.zodiackillersite.com/viewtopic.php?f=81&t=2625`.

Charles David. 1996. A World War II German army field cipher and how we broke it. *Cryptologia*, 20(1):55–76.

Cipher A Deavours. 1977a. The kappa test. *Cryptologia*, 1(3):223–231.

Cipher A Deavours. 1977b. Unicity points in cryptanalysis. *Cryptologia*, 1(1):46–68.

Dean W. Dickerhoof. 1971. The Vimark Cipher. *The Cryptogram*, 37(4).

Elonka Dunin and Klaus Schmeh. 2020. *Codebreaking: A Practical Guide*. Hachette UK.

William F. Friedman. 1954. Basic cryptologic glossary. `https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/friedman-documents/publications/FOLDER_234/41761109080025.pdf`.

Jim Gillogly. 1999a. another news article on Kryptos. `https://groups.google.com/g/sci.crypt/c/HTQqcW9XDAI/m/yC_JQZYxBPUJ`.

Jim Gillogly. 1999b. Kryptos Morse code. `https://groups.google.com/g/sci.crypt/c/d3SNKxTYsBA/m/BoQwYMWdWQIJ`.

Jim Gillogly. 2004a. Non-periodic polyalphabetic substitutions. `http://kryptos.yak.net/63`.

Jim Gillogly. 2004b. re: new member. `https://kryptos.groups.io/g/main/message/1236`.

Solomon W Golomb and Guang Gong, 2005. *Signal design for good correlation: for wireless communication, cryptography, and radar*, pages 134–135. Cambridge University Press.

Irving John Good. 1983. *Good thinking: The foundations of probability and its applications*. U of Minnesota Press.

Phillip Good. 2013. *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media.

WJ Hall. 1969. The Gromark cipher (Part 1). *The Cryptogram*, 35(2):25.

Edward Hannon. 2010. Novel K4 Results. https://kryptos.groups.io/g/main/message/10213.

Edward Hannon. 2011. Oct 8th DC Meet with Sanborn and Scheidt. https://kryptos.groups.io/g/main/message/12611.

Joshua Holden. 2018. *The mathematics of secrets: cryptography from Caesar ciphers to digital encryption*. Princeton University Press.

AJ Jacobs. 2020. Ed Scheidt Kryptos Transcript. https://kryptos.groups.io/g/main/files/PersonalFolders/AJJacobs/EdScheidtKryptosTranscript.pdf.

David Kahn. 1996. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon and Schuster.

Tim Kirchner. 2003. Don't know what this means, but... . https://kryptos.groups.io/g/main/message/232.

Donald Knuth. 1985. Deciphering a linear congruential encryption. *IEEE Transactions on Information Theory*, 31(1):49–52.

George Lasry. 2018. *A methodology for the cryptanalysis of classical ciphers with search metaheuristics*. kassel university press GmbH.

Geoffrey LaTurner. 2016. New member introductory message. https://kryptos.groups.io/g/main/message/18246.

Frank W. Lewis. 1992. *Solving Cipher Problems: Cryptanalysis, Probabilities and Diagnostics*. Aegean Park Press, Laguna Hills, CA.

William Mason. 2012. ACA Reference Statistics. https://bionsgadgets.appspot.com/gadget_forms/acarefstats.html.

William Mason. 2013. Compare unknown cipher against ACA cipher types (extended). http://bionsgadgets.appspot.com/gadget_forms/refscore_extended.html.

William Mason. 2016. Neural net ID test collection. http://bionsgadgets.appspot.com/gadget_forms/nnet_id_test_collection.html.

Greg Materna. 2020. Keyword for K4. https://aivirai.com/2020/08/24/4-3-kryptos-aivirai-muko-series-and-the-keyword-for-k4/.

Malte Nuhn and Kevin Knight. 2014. Cipher type detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1769–1773.

David Oranchak, Sam Blake, and Jarl van Eycke. 2020. Z340 has been solved! http://www.zodiackillersite.com/viewtopic.php?f=23&t=5079.

Joan B Plumstead. 1982. Inferring a sequence generated by a linear congruence. In *23rd Annual Symposium on Foundations of Computer Science (sfcs 1982)*, pages 153–159. IEEE.

Redacted. 2007. The CIA Kryptos Sculpture: A summary of previous work and new revelations in working toward its complete solution. https://www.nsa.gov/Portals/70/documents/news-features/declassified-documents/cia-kryptos-sculpture/KRYPTOS_Summary.pdf.

James Reeds. 1977. "Cracking" a random number generator. *Cryptologia*, 1(1):20–26.

Eugene Rogot. 1975a. Cycles for the Gromark Running Key. *The Cryptogram*, 41(1).

Eugene Rogot. 1975b. Gromarks 1-4. *The Cryptogram*, 41(5).

Frank Rubin. 1996. Designing a high-security cipher. *Cryptologia*, 20(3):247–257.

Jim Sanborn. 2009. Oral History interview with Jim Sanborn, 2009 July 14-16. https://www.aaa.si.edu/collections/interviews/oral-history-interview-jim-sanborn-15700.

Klaus Schmeh. 2015. 2015-10-25-Kryptos Workshop. https://youtu.be/25YFYKKKkDo?t=2704.

Christian Schridde. 2020. Kryptos The Cipher Part 4. http://numberworld.blogspot.com/2020/07/kryptos-cipher-part-4.html.

John Schwartz. 2014. Another Kryptos Clue is offered in a 24 year old mystery at the CIA. https://www.nytimes.com/2014/11/21/us/another-kryptos-clue-is-offered-in-a-24-year-old-mystery-at-the-cia.html.

Ferdinando Stehle. 2000. help needed to break KRYPTOS. https://groups.google.com/g/sci.crypt/c/EYd9673EihM.

William Stein. 2007. Sage mathematics software. *http://www.sagemath.org/*.

Jarl Van Eycke. 2015. Schemes 340. http://www.zodiackillersite.com/viewtopic.php?p=38542.

Unsupervised Alphabet Matching in Historical Encrypted Manuscript Images

Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés
Computer Vision Center
Computer Science Department
Universitat Autònoma de Barcelona
{jchen,msouibgui,afornes}@cvc.uab.es

Beáta Megyesi
Dept. of Linguistics and Philology
Uppsala University, Sweden
beata.megyesi@lingfil.uu.se

Abstract

Historical ciphers contain a wide range of symbols from various symbol sets. Identifying the cipher alphabet is a prerequisite before decryption can take place and is a time-consuming process. In this work we explore the use of image processing for identifying the underlying alphabet in cipher images, and to compare alphabets between ciphers. The experiments show that ciphers with similar alphabets can be successfully discovered through clustering.

1 Introduction

Historical ciphers contain many different symbols from various types of symbol sets. Although digits are the most popular types of symbols, we find alphabetical characters such as Latin or Greek letters, punctuation marks, diacritics, along with various types of graphic signs, such as Zodiac symbols or alchemical signs.

The first step in attacking a cipher is to digitize it and transcribe it by identifying each unique type of symbol that was used (namely, the 'cipher alphabet'). This is not easy if the cipher contains symbols from various symbol sets. The task is even more challenging when the symbols are touching or connected where individual symbols in the hand-writing are hard to segment.

Automatic methods using a kind of "AI-in-the-loop" strategy might help in the identification of symbol types, and assist the transcription process. This leads us to image processing, which has been shown its usefulness for handwritten recognition in historical manuscripts, including ciphers, see e.g. Fornés et al. (2017), Baró et al. (2019), Souibgui et al. (2020). However, as far as we know, there are no methods for searching and grouping ciphers with similar symbol sets. We believe that such a tool could help experts to identify the 'cipher alphabet' of any incoming new cipher, and also to retrieve similar ciphers that may help in the subsequent analysis and decryption stages. Thus, in this work, we explore the use of unsupervised clustering for the automatic identification and comparison of symbol types in ciphers. This process shall be done without the need of any transcribed, or annotated datasets.

2 Related Work

Encrypted manuscripts contain a wide range of symbols, especially those from Early Modern Times. An investigation of 700 historical cipher keys shows that the usage of digits, Latin characters, and graphic signs were evenly distributed in keys from the 15th and 16th centuries, as illustrated in Figure 1 (Megyesi et al., 2021). In fact, 30% of the symbols were graphic signs representing a large variety of symbols taken from symbol sets including not only the Zodiac or alchemical signs, but also various unknown, fancy symbols.

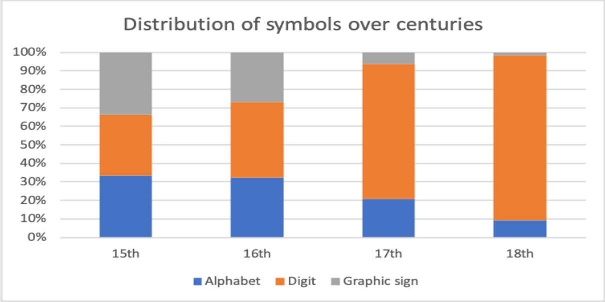


Figure 1: The usage of symbol types in cipher keys from the 15th to 18th centuries.

Image processing has proven to be useful for recognizing handwritten ciphers. Fornés et al. (2017) compared manual transcription versus automatic transcription with Recurrent Neural Networks with manual post-correction, showing that manual transcription was 15% slower if the model's accuracy was over 90%. Since then, more cipher transcription methods were proposed, using Siamese Neural Network and Gaussian Mixture Models (Yin et al., 2019), clustering (Baró et al., 2019; Chen et al., 2020), and few-shot learning (Souibgui et al., 2020).

As stated before, we are not aware of any existing image processing method for comparing and retrieving similar ciphers according to their symbol set. Thus, unsupervised clustering techniques (Gupta et al., 2019; Baró et al., 2019) are worth to explore since they can directly be applied to manuscript images without any transcription.

3 Methodology

The proposed method consists of three steps: a preprocessing stage consisting of binarization and segmentation into isolated symbols, a clustering phase where similar symbols are grouped together, and the analysis of the obtained clusters.

Image Preprocessing: The preprocessing stage starts by binarizing (Sauvola et al., 1997) the document image to facilitate the succeeding segmentation. Then, symbols are segmented using two different approaches. In case symbols are easy to segment because they are mostly isolated (i.e. there are very few touching symbols), we opt for a connected component analysis to obtain the segmented symbols. Contrary, if symbols are frequently touching, the symbol segmentation becomes difficult. Therefore, we opt for a more sophisticated method based on deep learning and proposed in Axler and Wolf (2018). Although the method was designed for word segmentation, we have adapted it for symbol segmentation. For this purpose, we have re-trained the model on 7000 synthetically generated document pages, which have been created by concatenating Omniglot symbols (Lake et al., 2015) and adding some random transformations to make them look similar to real ciphers. An example of a training page is shown in Fig. 2-A, and a segmentation example of a real cipher using the trained model is shown in Fig. 2-B.

Clustering: Once symbols are segmented, we compute the SIFT descriptor for each symbol and we apply the k-means clustering algorithm. Clustering consists in grouping those visually similar symbols in sets, named clusters. Since we are interested in comparing the different 'cipher alphabets', it is important to avoid unbalanced data. Thus, we take the same amount of symbols from each encrypted document to balance the data for a fair comparison in the clustering analysis stage.

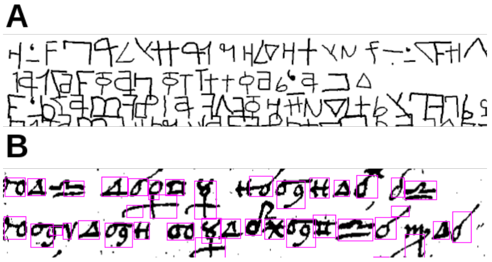


Figure 2: A: An example of a synthetic page created from Omniglot symbols. B: The segmentation output on the Borg cipher.

Clusters Analysis: Once we obtain the clusters from the two ciphers to compare, namely Cipher A and Cipher B, we analyze the similarity of their symbol elements. The goal is to analyze each cluster and verify the origin of its elements, whether they belong to Cipher A or B, or both. A cluster can have different levels of mixing, as shown in Figure 3. Depending on the frequency of each type of cluster, two ciphers will be considered more or less similar:

- If the 'cipher alphabets' are different, most clusters will contain symbols belonging to the same cipher (many clusters of type 1, 2 or 3, see Fig. 3).
- It the 'cipher alphabets' are similar, most clusters will contain symbols belonging to both ciphers (e.g. many clusters of type 4, see Fig. 3).

Being C_{mix} the number of clusters with mixed symbols (belonging to both Ciphers A and B) and C_{total} the total amount of clusters, the alphabet similarity is computed as follows:

$$Similarty(Cipher_A, Cipher_B) = \frac{C_{mix} \times 100}{C_{total}} \quad (1)$$

In this similarity computation we omit those clusters with very few elements (probably they are infrequent symbols). It is worth to observe that this analysis is sensitive to the symbol segmentation and the handwriting styles. For example, ciphers with different alphabets but similar handwriting styles could produce mixed clusters.

4 Experimental Results

We have evaluated our approach on encrypted manuscripts, most of them from the Decode

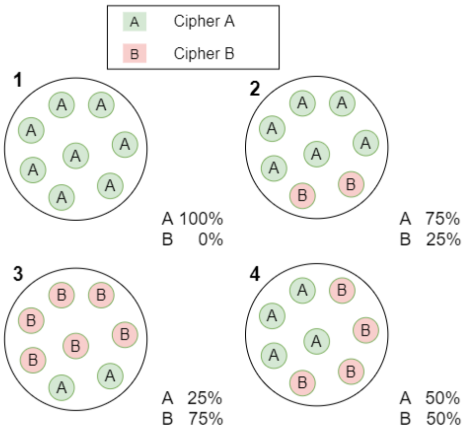
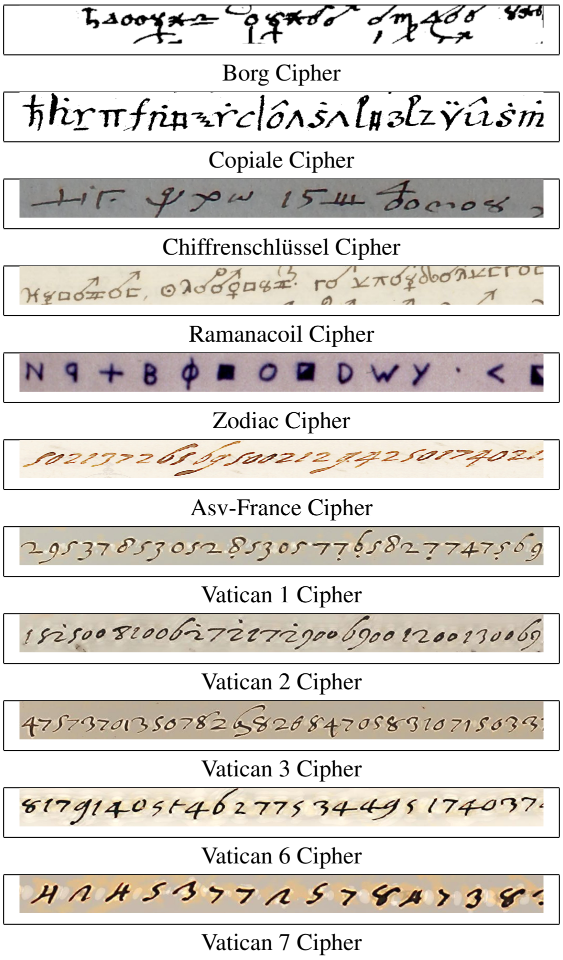


Figure 3: Cluster analysis. Cluster 1: All elements are from Cipher A. Cluster 2: There are more elements from Cipher A than from B. Cluster 3: There are more elements from Cipher B than from cipher A. Cluster 4: There is the same amount of elements from cipher A and B.


Borg Cipher

Copiale Cipher

Chiffrenschlüssel Cipher

Ramanacoil Cipher

Zodiac Cipher

Asv-France Cipher

Vatican 1 Cipher

Vatican 2 Cipher

Vatican 3 Cipher

Vatican 6 Cipher

Vatican 7 Cipher

Figure 4: Samples from the evaluated ciphers.

database (Megyesi et al., 2019). Figure 4 shows some examples. As it can be seen, some documents contain similar symbols, especially for the Vatican ciphers, with Arabic digits. However, these have different handwriting styles. During experiments, we took 5 pages from each cipher.

The obtained results are presented in Table 4. As can be seen, the similarity percentages range between 2.77% and 62.91%. Note that we are not reaching a higher similarity score probably because all the compared ciphers are different from each other in hand-writing style. The first observation is that ciphers with similar alphabets, such as the Vatican ones, are getting the highest similarity scores, compared to the rest of the ciphers. However, as we said before, the alphabet similarity can be easily affected by the writing styles. This is indeed the case: We obtain the highest score (62.91 %) when the writer style is similar, such as in the case of Vatican 3 and Vatican 6 with similar writing style of the digits ”2”, ”4” and ”7”, as shown in Figure 4). In the case of different writing styles, like Vatican 1 and Vatican 7, or between Asv-France and all the Vatican ciphers, we obtain a low similarity (20.94 %) though they all share the same cipher alphabet, digits.

We also observe a low similarity between the Zodiac and the rest of ciphers because Zodiac's cipher alphabet does not share overlapping symbols with the other cipher's alphabets. The other ciphers mainly use well-known graphic signs and digits and their similarity is medium to the rest of ciphers, without being too high or too low, indicating that these alphabets contain more or less overlapping symbols (e.g. digits) and are similar to each other.

Figure 5 illustrates some obtained clusters where symbols from different ciphers are grouped together if their shape appearance is similar.

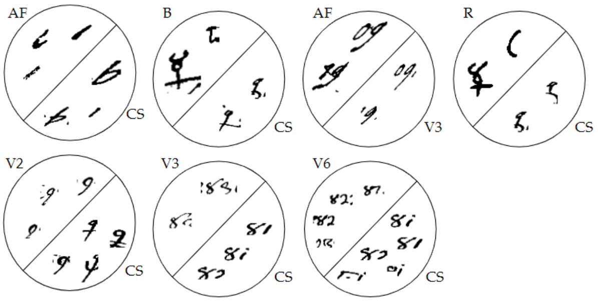


Figure 5: Results. Examples of mixed clusters.

From the different quantitative and qualitative results, we note that it is hard to assess the perfor-

Table 1: Results. Percentage of similarity between different pairs of ciphers. AF: Asv-France, B: Borg, CS: Chiffrenschlüssel, C: Copiale, R: Ramanacoil, V*n*: Vatican *n*, Z: Zodiac.

%	B	CS	C	R	V1	V2	V3	V6	V7	Z
AF	11.00	20.94	05.95	07.73	07.41	11.01	18.59	09.95	07.33	04.55
B	—	21.46	19.11	13.27	14.15	20.18	25.91	23.81	08.66	05.20
CS	—	—	14.74	18.13	17.48	37.04	43.81	35.21	14.90	12.20
C	—	—	—	10.33	21.07	14.62	21.08	20.37	09.39	07.14
R	—	—	—	—	08.89	05.43	08.07	07.56	03.71	08.83
V1	—	—	—	—	—	32.21	39.61	39.00	20.94	06.12
V2	—	—	—	—	—	—	54.78	46.17	24.70	07.85
V3	—	—	—	—	—	—	—	62.91	25.00	07.66
V6	—	—	—	—	—	—	—	—	24.05	05.00
V7	—	—	—	—	—	—	—	—	—	02.77

mance of the proposed method without any access to the ground-truth. Thus, we opted for visually checking the manuscripts. A thorough evaluation would be necessary, preferably by an expert in paleography who could establish the ground truth to set the similarity degree between ciphers and unify symbol sets across different ciphers.

5 Conclusion

We have presented an unsupervised method for identifying the symbol set in cipher images, avoiding the need of manual transcription or human intervention. The experiments show that it can provide an intuition of the underlying symbol set, and group ciphers with similar cipher alphabets. The presented results are promising and encourage us to further explore image processing for automatic alphabet recovery and transcription of ciphers.

Acknowledgement

We would like to thank all master's students in language technology at Uppsala University who helped us producing the manual transcriptions. This work has been supported by the Swedish Research Council, grant 2018-06074, DECRYPT – Decryption of Historical Manuscripts, the Spanish project RTI2018-095645-B-C21 and the CERCA Program / Generalitat de Catalunya.

References

Gregory Axler and Lior Wolf. 2018. Toward a dataset-agnostic word segmentation method. In *ICIP*.

Arnau Baró, Jialuo Chen, Alicia Fornés, and Beáta Megyesi. 2019. Towards a generic unsupervised method for transcription of encoded manuscripts. In *DATECH*, pages 73–78.

Jialuo Chen, Mohamed Ali Souibgui, Alicia Fornés, and Beáta Megyesi. 2020. A web-based interactive transcription tool for encrypted manuscripts. In *HistoCrypt 2020*, pages 52–59.

Alicia Fornés, Beáta Megyesi, and Joan Mas. 2017. Transcription of encoded manuscripts with image processing techniques. In *Digital Humanities*.

Divam Gupta, Ramachandran Ramjee, Nipun Kwatra, and Muthian Sivathanu. 2019. Unsupervised clustering using pseudo-semi-supervised learning. In *ICLR*.

Brenden M Lake, Ruslan Salakhutdinov, and Joshua B Tenenbaum. 2015. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338.

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The decode database: Collection of historical ciphers and keys. In *HistoCrypt*, pages 69–78.

Beáta Megyesi, Crina Tudor, Benedek Láng, and Anna Lehofer. 2021. Key Design in the Early Modern Era in Europe. In *HistoCrypt*.

J. Sauvola, T. Seppanen, S. Haapakoski, and M. Pietikainen. 1997. Adaptive document binarization. In *ICDAR*, pages 147–152.

Mohamed Ali Souibgui, Alicia Fornés, Yousri Kessentini, and Crina Tudor. 2020. A few-shot learning approach for historical ciphered manuscript recognition. In *ICPR*.

Xusen Yin, Nada Aldarrab, Beáta Megyesi, and Kevin Knight. 2019. Decipherment of historical manuscript images. In *ICDAR*, pages 78–85.

The Imperial Japanese Navy IKA Cipher Machine

Chris Christensen
Department of Mathematics and Statistics
Northern Kentucky University
Highland Heights, KY 41099, USA
christensen@nku.edu

Abstract

The Imperial Japanese Navy IKA cipher machine was a predecessor to the more familiar Japanese cipher machines of the 1931 and 1937 series. Nothing is known about the machine itself, but the cryptography of the machine is known. What follows describes the cryptography of the IKA machine and discusses that machine in the context of the 1931 and 1937 series of Japanese cipher machines that followed it.

1 IJN Cipher Machines

In the 1930s and 1940s, the Imperial Japanese Navy used a succession of three cipher machines for secret administrative matters among shore stations. The US Navy's codebreaking section OP-20-G referred to the ciphers as the "dockyard ciphers." Until 21 July 1933, the machine that was used was called IKA[1] (and the cipher was designated JN 111). It was followed by a machine designated M-1, or ORANGE (JN 141). M-1 was one of a series of three Japanese cipher machines that are referred to as the 1931 series: M-1, M-2 (naval attaché cipher), and RED (diplomatic cipher). These cipher machines were replaced by the three machines of the 1937 series: JADE, CORAL, and PURPLE, respectively. JADE (JN 157) was the last in the succession of dockyard cipher machines.

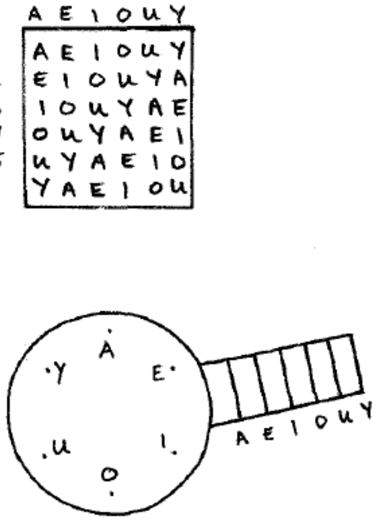
The Japanese language can be written in three different sets of characters. Kanji uses Chinese characters, and each character represents a word or phrase. Kanji precisely expresses language. An alternative method of expressing Japanese is kana. There are two versions of kana – hiragana and katakana -- which consist of 46 basic symbols plus some additional symbols and diacritical marks. Kana is syllabic, and each character corresponds to a sound. It is not unusual that several kanji have the same expression as kana. Reading kana is similar to deciphering a polyphonic cipher. Japanese has a special "kana Morse" code, and that code was used by the Imperial Japanese Navy to transmit codes and ciphers that used kana characters. The third method of expressing Japanese is romaji. Romaji uses Roman letters to transliterate kana. The three dockyard ciphers mentioned above used katakana characters.

The cryptography of the 1931 series machines was based on the Damm half-rotor and a 47-pin break wheel that staggered the motion of the half-rotor.

Unlike a full-rotor, a half-rotor has contacts on only one side of the rotor. The Damm half-rotor consisted of a rod with a disk on one end (Figure 1). Along the rod were slip rings through which the electrical charge from the plaintext typewriter keys entered. The slip rings were wired to outputs on the half-rotor's disk. Consider the enciphering of the six-letter alphabet shown in Figure 1. Letters on the slip rings are wired to the same letters on the half-rotor disk. The half-rotor disk makes contact with an output disk that is connected to the ciphertext typewriter. The output disk is labeled in the same manner as the disk on the half-rotor. Corresponding to Figure 1,

[1] In some documents the name appears as I KA. It appears to be the romanization of two kana characters.

if the letter A were typed on the plaintext typewriter, an electrical charge would enter the half-rotor by means of the slip ring corresponding to A. The charge would pass to the A-position on the half-rotor disk. If the A-position on the half-rotor disk were in contact with the A-position on the output disk, then the ciphertext typewriter would type an A. Corresponding to Figure 1, when a letter was typed on the plaintext typewriter, the half-rotor would step one position -- in this case, clockwise. If plaintext A were typed in the initial position of the half-rotor, the corresponding ciphertext letter would be A. If the half-rotor stepped one position and the plaintext A were typed again, ciphertext E would be typed. Then I, O, U, Y, A, etc. If the half-rotor stepped one position for each letter, the rows of the enciphering table correspond to the six successive encipherings of the plaintext letters which are shown in the table. The cipher is polyalphabetic; the cipher shown cycles through the six alphabets that are the rows of the enciphering table. The enciphering table has the pattern of the classical Vigenère cipher.



	A	E	I	O	U	Y
1	A	E	I	O	U	Y
2	E	I	O	U	Y	A
3	I	O	U	Y	A	E
4	O	U	Y	A	E	I
5	U	Y	A	E	I	O
6	Y	A	E	I	O	U

Figure 1. Damm half-rotor. (Raven)

The 1937 machines had regular stepping and composed ciphers that were implemented with 25-point telephone stepping switches.

IKA was the simplest of the machines and preceded the 1931 series. What follows is a description of the IKA cipher and what is known about the machine. No IKA cipher machine was captured or seen by US Navy codebreakers. The description that follows is based primarily on RIP 28A[2] "The M-1 Machine System," (CNO 1946) which reflects what the Navy knew about the IKA machine and its successor M-1 in April 1946. What the Navy knew about the IKA machine was based on their analysis of ciphertext messages. RIP 28A notes that there was little early data on either machine and that the data that existed was contradictory and confusing. It also notes that until 1936 IKA might have been known as M-1 and ORANGE as M-2.

2 Cryptography

Two stories describe trips to Europe by Japanese representatives to examine cipher machines. One describes a trip "commencing in approximately 1927" during which the Japanese acquired several commercial cipher machines including Enigma and Kryha. (Wenger, 286) The other tells of a trip in the early 1930s to Aktiebolaget Cryptograph, which had at that time been acquired by Boris Hagelin. Hagelin suspected that the Japanese wished to purchase a few machines to copy and, therefore, told the representatives that he had none to sell. Noticing two of Arvid Damm's obsolete cipher machines, the Japanese purchased them. (Raven, 1) Principles from the Damm machines were included in the 1931 series of Japanese cipher machines.

Much is known about the cryptography of the IKA machine, but essentially nothing is known about the actual machine.

IKA enciphered katakana characters. 49 characters were used in the system, and they were split into a minor sequence of seven characters and a major sequence of 42 characters. The reason for the split is not known.

The seven characters in the minor sequence (kana characters will be shown in romaji) were:

[2] SRH 355, page 61, which was written in 1971 by Captain Jack S. Holtwick, Jr., describes Holtwick as the author of RIP 28A, which was issued on 1 September 1935.

Ciphertext	Plaintext
RO	Parenthesis
WI	RO
SO	Nigori
NU	Hannigori
O	SO
WE	NU
X	Stop

This was a monoalphabetic substitution cipher; it was consistent throughout all messages. Existing records do not indicate how ciphertext X was transmitted. Kana characters were transmitted using kana Morse code. (Nigori and hannigori are diacritical symbols.)

The sequence (i.e., the ordering) of the 42 other characters changed monthly. Here is the sequence for September 1932:

KI	RU	TE	MO	MI	TU	WA	MU	RA	SE	ME	RE	KE	HI
A	U	HE	YA	HA	NE	E	N	YO	HO	I	YU	SA	KU
KA	TA	MA	SU	NI	TO	FU	KO	TI	NA	WO	SI	RI	NO

The IKA machine enciphered the major sequence by sliding a copy of the sequence against a copy of itself by one, two, or three places as each character was enciphered. This staggered motion was produced by a 47-pin break wheel. Active pins on the break wheel caused the machine to step one position. One inactive pin caused the machine to step two positions, and two inactive pins in sequence caused the machine to step three positions. It was not possible to have more than two inactive pins in sequence. Depending on the key, between 12 and 15 pins were removed. The machine stepped even when the minor sequence was used.

3 Settings

There were 50 message keys numbered 01, 02, …, 50. The key for a message was the last two digits of the originator's serial number – subtracting 50, if necessary. In addition to determining the status of the pins, the key also determined the starting position in the sequence. Here are keys 01, 02, and 03 from the key list for 1 January – 20 July 1933:

Key	Positions of inactive pins	Starting point
01	1 6 11 15 19 22 25 29 33 36 39 43 45	10
02	2 5 8 11 15 16 21 24 27 31 34 37 41 45 46	11
03	3 4 10 13 14 18 23 27 28 32 35 40 41 45	23

IKA messages appear as a, possibly incomplete, rectangle of ten columns. RIP 28A includes message number 608 from 22 October 1932. What appears below is the message with the heading removed.

```
6   0   8   0   8   0   2   2   1   0

SA  NO  TI  NO  SE  RE  KE  KI  WO  RU

NA  HE  RE  WA  E   TA  MA  TA  KU  SA

A   TE  KO  NE  SI  A   NI  FU  SU  A

MA  YU  YO  SE  KE  SE  SA  RA  TU  SI

HI  YA  MA  YO  FU  YA  YO  NA  HO  WA

MU  HO  MO  MU  KI  MU  WA  U   YA  TO

ME  TE  HE  NO  RU  SE  NE  MI  SO  SA

TA  I   KA  HI  NO  YU  ME  KU  RA  KE

WA  ME  ME  N   HI  X   KE  SI  X   TO

ME  NA  HO  YO  KO  SA  A   NU  KE  ME

TI  KU  ME  YU  NE  KI  TI  O   HA  NI

TE  MA  WA  MI  TI  KA  HA  RE  HI  MA

MO  TU  MO  SI  NO  SE  SU  KO  MU  TE

E   YU  HE  TA  KE  SA  SE  TA  NI  RA

A   SI  RA  SE  SA  MO  TA  KU  N   TO

HI  FU  KU  MU  YU  HO  YU  SA  MA  TE

TO  KA  RI  O   TE  YU  YA  YU  MA  ME

A   TI  KO  HO  SI  KA  YO  MU  SI  HA

WA  YO  MA  HO  YA  YO  TU  NE  N
```

The separators that would appear at the end of each line are not shown. RIP 28A describes the separator as “unknown,” but SRH 355 (Appendix 8, 90F) suggests that it was // (and, furthermore, that the Japanese character // was called “IKA,” which was used for the name of the machine).

The ten reference numbers above the message include information that determines the key.

```
6   0   8   0   8   0   2   2   1   0
```

The first three digits are the message originator’s serial number 608. The last two digits of that number 08 is the key. Because it is important that the key be transmitted ungarbled, those two digits are repeated as digits four and five of the reference number. Digit six 0 is the part number; this is a one-part message. Digits seven and eight are the day of origin 22. Digits nine and 10 are the hour of origin 10.

Key 08 on 22 October 1932 is a special key that appeared at the end of the keylist:

Key	Positions of inactive pins	Starting point
08	5 8 11 17 22 25 26 29 34 37 41 44 46	10

Because the major sequence is periodic, there is no true "starting point;" the term "starting point" refers to the offset of the plain and cipher components of the major sequence. A starting point of 0 would have plain and cipher components aligned with no shift. The other offsets could be as small as 1 or as large as 41.

4 Deciphering

Navy codebreakers took "KI" as the first letter in the sequence. For October 1932, the sequence was:

KI	HI	WO	ME	KE	TO	SA	KO	SE	HO	MU	NO	YU	RU
KA	FU	RE	RA	YA	SI	HA	TU	N	U	A	I	MA	TE
WA	MO	ME	KU	E	NA	TA	NI	YU	TI	RI	NE	HE	SU

Starting point 10, means an offset of 10 between the cipher (on top) and plain alphabets:

KI	HI	WO	MI	KE	TO	SA	KO	SE	HO	MU	NO	YO	RU
MU	NO	YO	RU	KA	FU	RE	RA	YA	SI	HA	TU	N	U

KA	FU	RE	RA	YA	SI	HA	TU	N	U	A	I	MA	TE
A	I	MA	TE	WA	MO	ME	KU	E	NA	TA	NI	YU	TI

WA	MO	ME	KU	E	NA	TA	NI	YU	TI	RI	NE	HE	SU
RI	NE	HE	SU	KI	HI	WO	MI	KE	TO	SA	KO	SE	HO

The 22 October 1932 intercept has starting point 10, therefore, the first character SA deciphers to RE. Following the Navy's procedure, the plaintext alphabet (i.e., the bottom row above) slides to the left. To decipher the second ciphertext character, the plaintext alphabet slides one position to the left, and ciphertext NO deciphers to N. To decipher the third ciphertext character, the plaintext alphabet slide one more position to the left, and TI deciphers to KO. Next, NO deciphers to A, which in plaintext represents the diacritical symbol nigori. The nigori signals that the K is to be voiced; so KO becomes GO.

Pin 5 is inactive; therefore, the next slide is two positions, and SE deciphers to U. And, the deciphering continues.

Deciphering can be done by hand with sliding alphabets while giving attention to whether pins are active or inactive.

Cryptographically IKA was similar to the Kryha machine. It also is cryptographically similar to a Damm half-rotor with staggered motion.

There does not seem to be any description of the physical machine. In particular, there is no record of how the staggered motion of the cipher alphabet against itself was produced nor how plaintext was entered and ciphertext outputted.

There also does not seem to be any record of how the machine was attacked. RIP 28A contains the comment, which seems to refer to both IKA and M-1, that:

> Unfortunately there are no longer any records available of the cryptanalytic attacks used in recovery of the machine. Whatever the methods, the solution must be recognized as a cryptanalytic masterpiece in the pioneering of machine solutions. (CNO 1946, 1)

5 Relationship to Other Japanese Cipher Machines

The IKA machine was replaced on 21 July 1933 by M-1. RIP 28A notes that there was a slight change in the machines. (CNO 1946, III-1) The major sequence of 42 characters was identical with the major sequence of IKA; however, the minor sequence consisted of 14 characters rather than the 7 characters of IKA's minor sequence. The minor sequence of the M-1 was the minor sequence of IKA with the X character removed and the digits – excluding 2 and 8 – added to the sequence. The plaintext for characters in minor sequence mostly consisted of diacritical marks, punctuation, and "stop."

M-1 output consisted of 10 x 10 blocks of kana characters.

Although the sequence of the characters of the major sequence was part of the machine setting, the characters in the minor sequence were always in the same order.

The polyalphabetic nature of the M-1 was caused by a Damm half-rotor. The minor and major sequences stepped together. Effectively, the enciphering consisted of a 42-position half-rotor with one of the 42 characters of the major sequence in each position on, say, the edge of the disk and the 14 characters of the minor sequence repeated three times, say, around the disk just inside the positions of the major sequence. Because a Damm half-rotor produces Vigenère tables, M-1 would produce two Vigenère tables – one with period 14 for the minor sequence and one with period 42 for the major sequence. The total period produced by the half-rotor is 42.

It is not known how IKA physically stepped the major sequence – whether, for example, it stepped with a half-rotor or with sliding disks like Kryha.

Similar to IKA, a 47-pin break wheel was used to extend the period of the M-1.

Two M-1 machines were captured at Rashin, Korea, at the end of the war. One of those machines (Figure 2) is on display at the National Cryptologic Museum, which is located next to NSA Headquarters in Fort George G. Meade, Maryland.

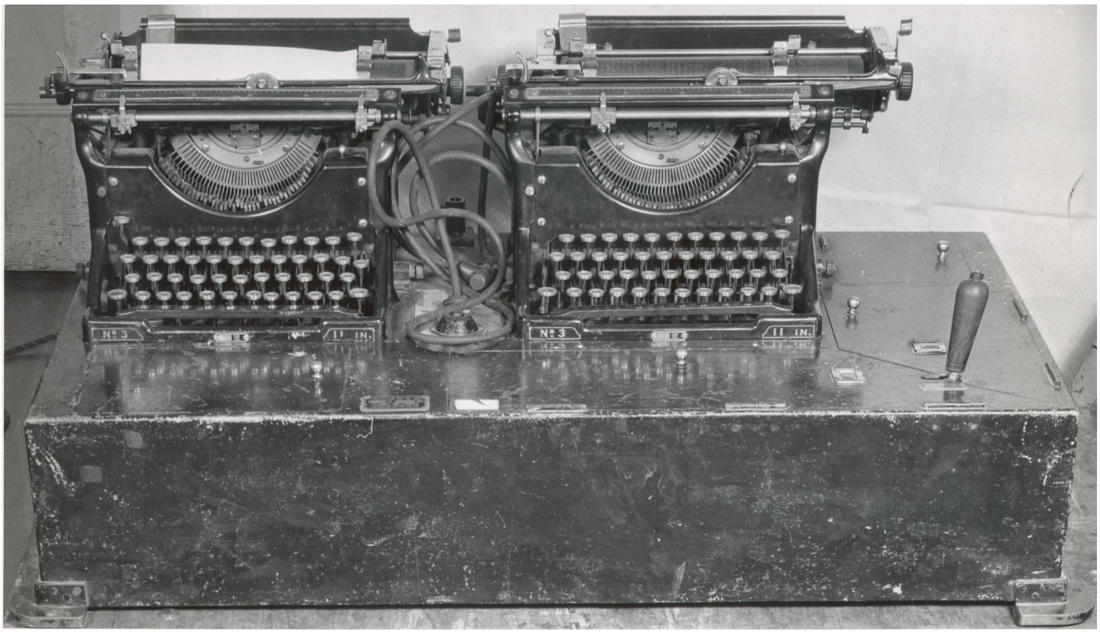


Figure 2. An M-1 cipher machine that was captured at Rashin, Korea after World War II. Courtesy of the National Security Agency.

All three of the 1931 series of Japanese cipher machines had a minor and major sequence, used a Damm half-rotor, and implemented irregular stepping by means of a 47-pin break wheel.

In the 1937 series of cipher machines, JADE replaced M-1. JADE had a 25-character minor sequence and a 25-character major sequence – low frequency and high frequency kana, respectively. The keyboard had 25 keys and a shift key to shift between sequences. Enciphering was implemented using 25-point telephone stepping switches. The period, which was length 25 for each switch was extended by composing the switches. JADE had 5 switches. The first three switches stepped; the last two switches were set to a given position but did not step. There was a plugboard only on the ciphertext side. The first three switches stepped regularly – one switch was fast, one was medium, and one was slow. Only three of the possible six orders of motions were available: fast-slow-medium, slow-medium-fast, and medium-fast-slow.

JADE came into use in 1942. It did not received much use. There was a break in its use for two weeks at the end of April 1943. Kwajalein was one of the heavy users, and after the Allied invasion (31 January – 3 February 1944) the shutting down of that station reduced the use of JADE significantly. Use was further reduced after a bombing of Rabaul. The last JADE intercept was 30 August 1944.

6 Analog

There is considerable confusion about an analog – or analogs -- designed by Holtwick. He is variously credited with designing an analog for IKA or an analog for M-1. Captain Laurance Safford refers to an analog designed by Holtwick about 1937. Although that occurred after the 1931 series of cipher machines, Safford notes that

> The cipher system was considerably older and had been solved for several years. … [We] had always read the messages by paper and pencil methods and rotating disks. Holtwick made an arrangement in which a kana typewriter was mounted on a box and inside of this box as a cylinder with pins. The pins were pluggable on the rotating cylinder. I do not recall whether its action was electrical or mechanical. The effect of the pins was chiefly to give an irregular stepping cycle. Two machine were built: one was retained at the Navy Department and the other sent to [Pearl Harbor]. Just about the time we got the machine to

> [Pearl Harbor] the Japanese abandoned the system so the machine actually never did any particular good. (Safford 3 February 1944, paragraph 2)

SRH 355, which was written by Holtwick, notes that:

> [Holtwick] assumed (incorrectly, as it turned out) that the M-1 was but the initial one of an era of cipher machines that would supplant codes and manual ciphers in Japanese Naval communications … .
>
> With this possibility in mind, he designed and roughly sketched a mechanical device, including interlocking gears and pin-controlled stepping sequences, which would not only duplicate the stepped slidings of the cipher sequence recovered in the M-1 machine, but included methods for coping with more complicated variations of the sequence, some of which he assumed the Japanese cryptographers might introduce in their next version of the machine. (SRH 355, 161)

As noted, however, the Japanese moved away from irregular stepping with the 1937 series of machines.

SRH 355 states that the M-1 analog was designed in 1935 but not completed until 1937. A 13 May 1937 letter to Holtwick from Lieutenant Joseph Wenger mentions that an M-1 analog and other material were being shipped to Pearl Harbor so that that station might take over deciphering of M-1. However, the last definite M-1 intercept was from March 1937.

SRH 355 (i.e., Holtwick) states that "Occasional reference to Holtwick's machine as the IKA may be encountered; these are not valid." (SRH 355, Appendix VIII, 90F) That seems to confirm that Holtwick's analog was designed for M-1 – not for IKA. Furthermore, SRH 355 assigns the designation RIP 41 to M-1. (SRH 355, Appendix VIII, 90G)

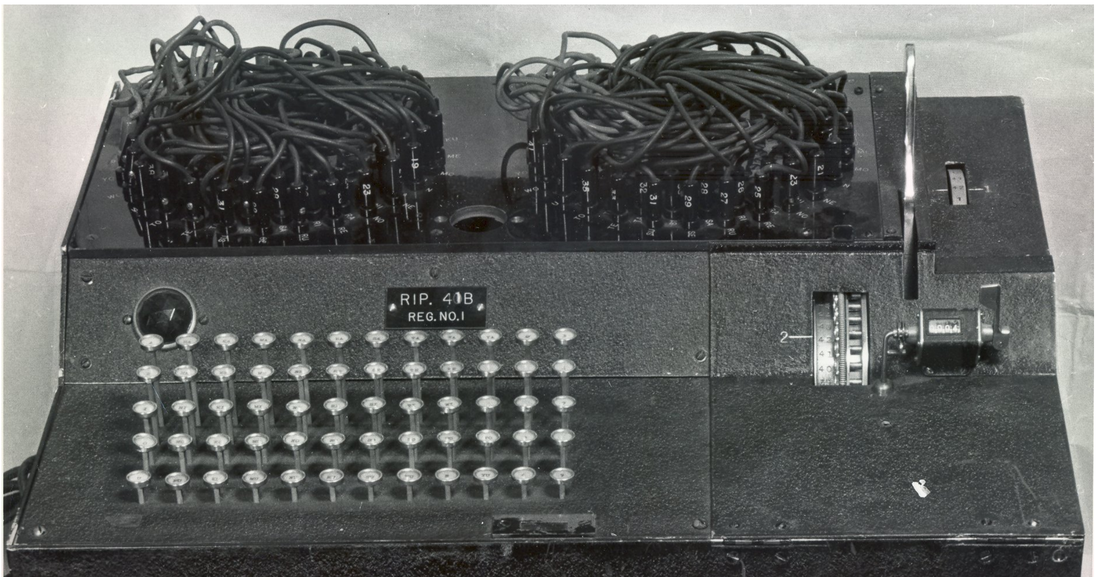


Figure 3. Holtwick's analog for the M-1 cipher machine. Courtesy of the National Security Agency.

Figure 3 is a photograph from NSA files of a machine designated RIP 41B, which therefore should be Holtwick's M-1 analog. The keyboard and plugs are labeled with kana characters. The wheel on the top right has 42 positions and should set the starting position. The wheel to the left of the counter appears to be a 47-pin break wheel.

7 Conclusion

The IKA and M-1 cipher machines are cryptographically similar. Although M-1 machines were captured and, therefore, the nature of the physical machine is known, nothing is known about the physical IKA. It could have had a design like the Kryha – a machine with which the Japanese were familiar. IKA was in service only briefly before it was replaced by M-1. Holtwick designed an analog for M-1, but by the time his analog was constructed M-1 was no longer in use and the irregular stepping that Holtwick had expected to continue to evolve in use with subsequent Japanese cipher machines was replaced by regular stepping with composed switches.

References

Chief of Naval Operations. April 1946. “The M-1 Machine System: RIP-28A.” National Archives and Records Administration College Park RG 38 Box 16.

Holtwick, Captain Jack S., Jr. 1971. SRH 355 “History of the Naval Security Group to World War II.” Reprinted by U.S. Naval Cryptologic Veterans Association, Pensacola, FL.

“History of JNA-20 – Coral, Volume II.” National Archives and Records Administration College Park RG 457, Box 1387.

Raven, Francis A. Undated. “Some Notes on Early Japanese Naval/Diplomatic Cipher Machines,” CCH Series Files IV.W.III.23.

Safford, Captain L. F. 3 February 1944. “Memorandum for Lieutenant Commander Raven, Subject: History of Japanese Cipher Machines.” National Archives and Records Administration College Park RG 457 Box 808.

Wenger, Jeffrey Joseph. Unpublished manuscript. “RADM Joseph N. Wenger, USN. Biography & Autobiography. Communication Technology in World War II to Computer Technology. ”National Cryptologic Museum Library.

Appendix A

Information about Japanese Cipher Machines

IKA

Enciphering device	Unknown Sequencing of major alphabet
How period was extended	Staggered stepping
U.S. intercept dates	Late 1931/early 1932 – 21 July 1933
User	Major IJN stations
Alphabet	Kana
Split of alphabet	42/7

1931 Series

Enciphering devices	Damm half-rotor Input and output plugging
How period was extended	Staggered stepping caused by 47-pin break wheel

Machine	RED (or M-3 or Type A)	ORANGE (or M-1)	M-2
User	Diplomats	Major IJN stations	Naval attachés
Alphabet	Romaji	Kana	Romaji
Split of alphabet	6/20	42/14	Perhaps identical with RED
U.S. intercept dates		20 July 1932 – March 1937	
Replaced by	PURPLE	JADE	CORAL

1937 Series

Enciphering devices	25-point telephone stepping switches Input and output plugging except JADE (output only)
How period was extended	Composition of switches

Machine	PURPLE (or M-5 or Type B)	JADE	CORAL
User	Diplomats	Major IJN stations	Naval attachés
Alphabet	Romaji	Kana	Romaji
Split of alphabet	6/20	25/25	None
U.S. intercept dates	20 February 1939 – end of World War II	1 December 1942 – 30 August 1944	8 September 1939 – end of World War II
Number of composed switches	3 for 20s 1 for 6s	3 stepping 2 stators	3

Island Ramanacoil a Bridge too Far. A Dutch Ciphertext from 1674

Jörgen Dinnissen
Historian and
consultant business analytics,
The Netherlands
dinnissen.jorgen@gmail.com

Nils Kopal
University of Siegen,
Germany
nils.kopal@uni-siegen.de

Abstract

The deciphered Ramanacoil ciphertext reveals two Dutch East India Company letters, from 1674 that are, in retrospect, already known in the National Archives as plaintext letters. The letters are written in Dutch. We have been able to relate them. The first letter, from Van Goens senior from Sri Lanka to the Lords Seventeen in The Netherlands, was most important to the sender. He sent his personal secretary Leeuwenson overland with the ciphertext in his pocket and its key in his head. And with additional oral information that had to be delivered in person. Van Goens senior requested to conquer the whole of Sri Lanka, the island Ramanacoil and coastal area around it along with 1,000 more soldiers. This paper shows that by sending Leeuwenson, Van Goens senior wanted to repeat his most successful 'Vertoog' from 1655. Substantiate his strategic goals and get approval for them from the Lords Seventeen. In 1655 he got a 'Go!' and twenty years later in 1675 he got a 'No!'. The zeitgeist of expansion had changed. Ramanacoil was a bridge too far.

1 Introduction

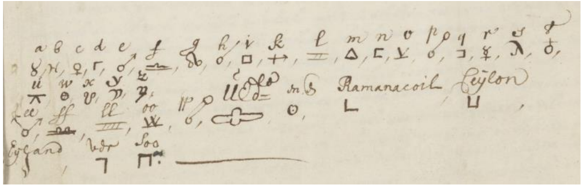

Ramanacoil
Ceylon

Figure 1: Key from the Ramanacoil ciphertext.

The Ramanacoil ciphertext is a 46 pages manuscript – with 39 pages of ciphertext and one page with the key – of the Dutch East India Company (VOC) located at the National Archives, The Hague, The Netherlands (Ramanacoil, 1674). The description of this manuscript reads: "piece in unknown language without Dutch heading; encrypted text with key probably related to Ramanacoil." This is based on the words *Ramanacoil* and *Ceylon* in the key (see Figure 1).

From an initial inventory the ciphertext did not appear to be deciphered. In retrospect it turned out that it was deciphered by Van Meersbergen (2009). This only became clear after conducting follow-up research with the name of the man, Joannes Leeuwenson, who encrypted (Dutch: *in caracters gebragt*) the plaintext in 1674. Van Meersbergen didn't make a transcription. He deciphered the ciphertext of all pages directly into plaintext with the straightforward key in his head. (Notice: Plaintext can also be a transcription. But for us here, we define that transcription is a digitized text that can be used for further (crypt-)analysis.) Only the first revealed plaintext page of the ciphertext is published in Van Meersbergen (2009).

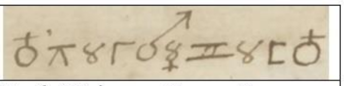
Ciphertext	
Manual transcription	Earth ' Quincunx Taurus Gamma Male Mercury Gemini Taurus SquareRight Earth
Revealed plaintext as CT2 output	T'UADERLANT
Translation from Dutch into English	The Fatherland

Table 1: First word of the ciphertext deciphered.

Crina Tudor and her team of five students from Uppsala University, Sweden made a digital transcription of all pages manually. They assigned each ciphertext symbol to a transcription word. This transcription was used as input in the software CrypTool 2 (CT2). Using the digitized nomenclature CT2 generated the plaintext used for this paper. The Table 1 shows the very first word of the document in ciphertext, transcription, revealed plaintext in Dutch, and its translation into English.

Next to the ciphertext we found six manuscripts in the National Archives that are relevant for interpretating it. One of them is a daily register (Dutch: *Daghregister*) that the encrypter Leeuwenson (1675) made during his overland journey from Ceylon (present day Sri Lanka) to Amsterdam.

This paper is structured as follows: Section 2 gives a summary of the two letters that are hidden in the ciphertext and provides background information. Section 3 takes a closer look at the register of Leeuwenson and two other letters in which he is mentioned. Section 4 shows two identical plaintext copies of these letters found in the National Archives. Section 5 discusses the importance of the ciphertext and whether the requests were granted. The technical matters of the used cipher are discussed in Section 6, and we give recommendations regarding transcriptions done manually. Finally, in Section 7, we draw three conclusions.

2 Content of the two letters

The ciphertext consists of two letters that bring us at a decisive period of the VOC in which, in retrospect, its expansion reached its limits. What strategic-military choices do we have to make and which ones are crucial for trading? Van Goens senior was one of the protagonists in these discussions.[1]

2.1 VOC

The business concept of the VOC (1602-1800) was: cashing in price differences for products from the East Indies (read: Asia and not the south of present day country India) in Europe. They wanted to achieve this with a ‘Grand Strategy’ of three main goals:

1. Establishing a monopoly on spices on the Moluccas (present day Maluku Islands, Indonesia). This goal was achieved in 1666.
2. Taking control of the pepper and textile trade in the south of India and establishing a monopoly on cinnamon on Ceylon. This goal was achieved in 1663.
3. Taking control of the silk trade in China. This goal was never achieved. After the surrender of Fort Zeelandia on Formosa (present day Taiwan) in 1662 the company had to admit that this goal had failed.

The VOC’s power was exercised through fortifications, spread throughout Asia, on the landward side of the sea. At sea, armed merchant ships usually sailed back to the Netherlands in convoy with warships for protection. In times of tension, expansion, or punitive expeditions, additional ships were deployed in fleets with a lot of firepower and soldiers. The company was above all a maritime power with a strong fleet. They were mainly interested in trade and not in territorial expansion.

Around 1674 the VOC had 200 to 250 fortifications. The headquarters of Asia were in Batavia (present day Jakarta, Indonesia). Formally, the governor of the Ceylon government was subordinate to the governor general in Batavia. With the appointment of Van Goens senior in 1655, Batavia faced serious competition from Ceylon as a second return port.

A few numbers: in 1687, the company in Asia employed 12,000 European employees of whom 8,000 were soldiers; 3,000 employees were employed both in Batavia and Ceylon of which 2,000 were soldiers. In addition to the staff from Europe, there were 8,000 local employees and slaves at work. Of which 2,000 were working at Batavia and 3,000 at Ceylon.

[1] The background information of Subsections 2.1 and 2.2 are based on: Gaastra (2012), Knaap and Teitler (2002), Knaap et al. (2015), and Van Meersbergen (2009 and 2011).

2.2 Superintendent and former governor of Ceylon Van Goens senior

Until 1670 Rijckloff van Goens senior (1619-1682) was a successful diplomat and warrior whose great Ceylon project seemed to be realizable. He had been a successful diplomat in present day Indonesia and belonged, at the age of 37, to the top in Batavia before he travelled to The Netherlands.

In 1655 he was allowed to unfold his plans (Dutch: *Vertoog*) for Asia to the Lords Seventeen (board of the VOC) in Amsterdam. After this he got a 'Go!' to personally lead the conquests of northwest India (Diu), island Ceylon, and south India (Tutucorin). They were realized in 1658 with the exception of Diu. In 1663 he also conquered Cochin on the southwest coast of India (Malabar), the heartland of the pepper trade.

Van Goens senior was governor for the Ceylon government from 1662-1663 and 1665-1671. In 1671 he was succeeded by his son Rijckloff van Goens junior (1642-1686). Senior became superintendent (Dutch: *opperkoopman*) but it was clear to everyone that senior was still in charge.

2.3 Island Ramanacoil and Ceylon

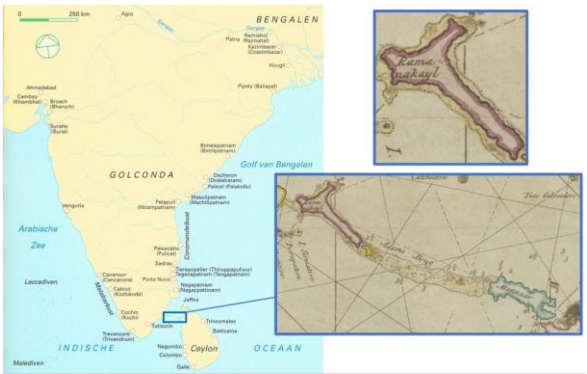


Figure 2: India and Ceylon. Ramanacoil, Adam's Bridge, and Manaar enlarged. The 'Map India' is cropped and published with permission from Bert Stamkot and taken from Gaastra (2012: 53).

Ramanacoil (present day Rameswaram) is an island against the mainland of south India (see Figure 2). Across Adam's Bridge (Dutch: *Adams brug*) on the other side is the island of Manaar, which in turn lies against the island of Ceylon. In 1674 the coast of Ceylon was in the possession of the VOC and the interior was in the hands of the King of Kandy.

2.4 The two letters of the ciphertext

The ciphertext contains two letters written in Dutch. Letter One from Van Goens senior to the Lords Seventeen in The Netherlands consists of six pages ciphertext. Letter Two, which has 33 pages, is addressed to the governor general in Indonesia. See Table 2, for more details about the two letters: sender, encrypter, and receiver.

	Letter one	**Letter two**
Number of pages	6	33
Key (same key for both letters)	available	available
Date	1674-01-24	1674-01-23
Plaintext language	Dutch	Dutch
Sender name	Van Goens senior, Van Goens junior and board of Ceylon government	Van Goens senior, Van Goens junior and board of Ceylon government
Sender place	VOC, Ceylon	VOC, Ceylon
Receiver name	VOC, Lords Seventeen	VOC, governor general and board of Asia
Receiver place	Amsterdam, The Netherlands	Batavia, Indonesia
Encrypter	Leeuwenson	Leeuwenson
Content	Strategical information about the future of VOC in Asia. Request for expansion and 1,000 more soldiers	Operational information of the ins and outs of the Ceylon government

Table 2: The two letters of the ciphertext.

2.5 Letter One: from Van Goens senior on Ceylon to the Lords Seventeen in Amsterdam

Summary of the content: The company has expelled the French from city Trincomali, Ceylon (Dutch: *principale haven deses eylants*). We must avoid that a European competitor takes possession of a part of Ceylon. The King of Kandy cooperates with the English, Portuguese, and French. He cannot be trusted. Only after learning this, we took actions to occupy all of Ceylon and not only a part. We must take possession of all of Ceylon! Ceylon is a better place than Batavia to protect Asia. It is in the heart of Asia. Without Ceylon, Asia is in danger (Dutch: *los*). On Ceylon is everything we want.

We have to take possession of the island Ramanacoil and the region below Adams Bridge (Madura) and above (land of Tanjore). With 200-300 men, Ramanacoil can be defended against

European and local competitors. Without Ramanacoil, everything on Ceylon is in danger. Dutch: *eylandt Rammenecoyl, sonder t'welck alles op Ceylon los is*.

Make whole Malabar including Cananur and a part of Coromondel (from Nagapatnam to Cranganoor) subordinate to Ceylon. Unity is strength. Then Ceylon gets stronger, we will overcome the costs (Dutch: *lasten*) in a few years and earnings will double.

Last sentence before signing: Send at least 1,000 soldiers directly to us. We are so weak that a few soldiers will not help. Ceylon's large fleet has consumed so much that all supplies have been devoured.

See Figure 3 for the wish list of expansions of Van Goens senior in 1674.

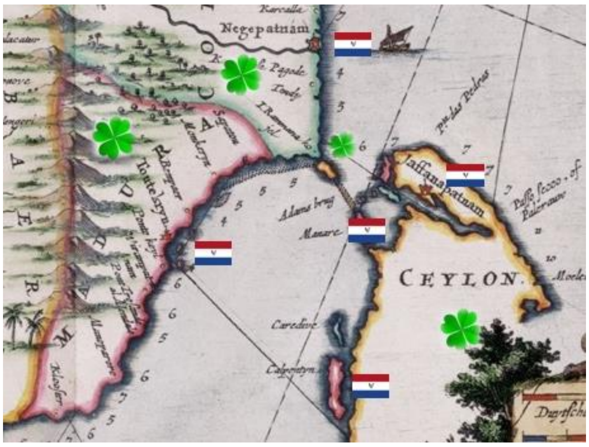


Figure 3: The wish list of expansions ('clover') of Van Goens senior in 1674 on a Dutch map from 1682. The Dutch fortifications have a 'VOC flag'.

2.6 Letter Two: from Van Goens senior on Ceylon to governor general in Batavia

Summary of the content: Trade in products: what comes from where in what quantities. Information about employees: appointments, deceased, requests for salary increases, employees to be penalized. Ships repaired.

Due to looting, seven of the ten English ships at Masulipatnam on the Coromandel coast have not been captured. Otherwise, the entire English fleet would have fallen into our hands and we could have conquered St. Thomé in 5-6 days. We hope for peace with England. We are going to conquer St. Thomé together with the Moors. Please, send an additional armed force around 1674-04-01. What should we do with St. Thomé once it is conquered? Trade it for the smelly Palleacatte? We await further orders.

Permission to take Ramanacoil and the coast of Madura in possession. They are important to keep Ceylon into our possession.

We have "absolutely" overcome the attacks from the men of the King of Kandy, Raja Sinha. They burned our brown rice (Dutch: *nely*) and knocked off the heads of four soldiers. Raja Sinha is a horrible tyrant. He slept with the only princess in the country: his father's sister. A daughter was born from that relationship.

Request for more capacity of European employees. Soldiers from Ambon do a good job, we wish we had 2,000-3,000 more of them to build a militia. Request of 100 horses more to do patrols with.

The Portuguese language in schools and churches has been abolished. Only our national language is used.

3 Encrypter Leeuwenson and his overland journey

In 1674 secretary Leeuwenson was ordered by his boss Van Goens senior to travel overland from Ceylon to Amsterdam with a soldier as company. The overland route can theoretically be covered in less than four months. Over sea with a sailing ship would take him seven months from Ceylon to Amsterdam. But the Dutch and the VOC were at war with France, England, and a few other countries during the Franco-Dutch War (1672-1678) and Van Goens senior wanted to be sure (Dutch: *apparentie om de seckerheijt*) that his most important letter was delivered swiftly in Amsterdam. In June 1673 most ships with letter books sent to The Netherlands were confiscated by the English near Saint Helena and Van Goens senior did not trust the route over the seas.

3.1 Overland journey from Ceylon to Amsterdam

During his journey Leeuwenson (1675) kept a register.[2] In this, we read that he encrypted two copy letters handed over to him by Van Goens senior and that he sent them back to his boss with the key. Van Goens senior ordered Leeuwenson to encrypt these two most important letters to ensure that the scope of these letters would be hidden to their enemies in case they would be intercepted. In Dutch: *dat ick de twee voorgemelde importante brieven in caracters zoude stellen, opdat (indien deselve onderschept wierden) de teneur voor onse vianden verborgen.*

In the register we read that Leeuwenson had to consult VOC employees De Hase in Gamron and Repelaer in Basra how and where to cross overland exactly from Basra (present day Iraq) to Aleppo (present day Syria). This stretch was the only part of the journey the company couldn't provide protection for. A part of the journey where no difficulties were expected (Dutch: *reijse waar geen swaerigheijt in gelegen is*). In Basra they changed their Dutch into modest "Turkish" cloths and their hair was cut. To avoid suspicion, they had to pretend to be ordinary and poor traders and not employees of the VOC. With letters of introduction, guides, interpreters, paying tolls, and paying a "reasonable gift" (Dutch: *redelijcke schenkagie*) to three Sheikhs who were the heads of four groups of raiders, Leeuwenson and soldier Van Daelen were able to cross the desert. A crossing that was not without danger, but that was justifiable with the right precautions and the willingness of paying money. The fact that Leeuwenson kept a diary, in his luggage, for the VOC, in cleartext, in which he writes that he encrypted the letters and for whom, indicates that they did not expect to be intercepted. The encryption was a precaution.

Van Meersbergen (2011) writes that for VOC employees the landroute from Basra to Aleppo was forbidden since 1624. The Dutch and all other countries in Asia used Armenian traders and French clergymen for postal delivery, back and forth, between Aleppo and Basra. During the war with France in 1674 they were not trusted with these most important letters. Leeuwenson had to deliver them in person.

On 1675-1-5 Leeuwenson arrived in Amsterdam and he delivered the letters the same day to the Lords Seventeen. The letters were handed over to him almost one year before. In his register Leeuwenson never mentions when he decrypted his ciphertext. He must have done this somewhere between Aleppo and Amsterdam. Or was it done after his delivery in Amsterdam? Was his additional oral information sufficient? We cannot say this with certainty because we don't have the revealed plaintext from 1675. Next to that, no interview report is known of the content of Leeuwenson's meeting with the Lords Seventeen.

3.2 Letters with additional information

What additional information can we gather from other sources about the ciphertext and the key? In Van Goens' (1674d) letter from 1674-2-10 to the Lords Seventeen we read that Leeuwenson will orally provide (Dutch: *bij monde*) additional information to the Lords Seventeen about "many matters that should not be entrusted to paper". Van Goens senior adds that Leeuwenson speaks Latin, French, and Portuguese well.

In the letter from 1674-5-2, De Hase (1674) writes that Leeuwenson told him that the "important letters" were encrypted and couldn't be helpful to anyone (Dutch: *niemand sich soude connen dienen*) even if they had them in their hands. Without his presence, the Lords Seventeen can't do anything with these letters (Dutch: *sonder sijne presentie niet gedient conden sijn*).

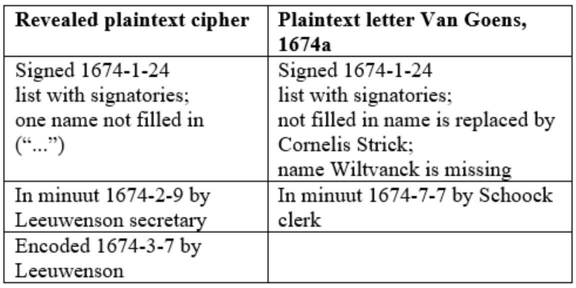
From the above we draw the conclusion that Leeuwenson was sent from Gamron – where De Hase wrote his letter – to The Netherlands with the ciphertext in his pocket and the key in his head. Without him neither the Lords Seventeen, nor the enemy, nor anyone else would be able to

[2] Leupe (1863) transcribed and published the handwritten journal Leeuwenson (1675) kept of his overland journey.

read these most important letters. Next to the letters, he had additional information in his head that could not even be entrusted to paper. It had to be told in person, face to face, to the Lords Seventeen.

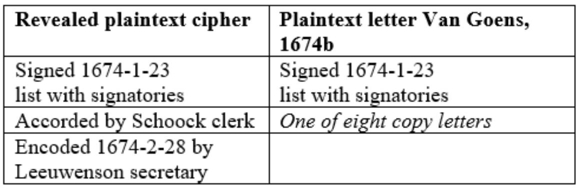
4 Plaintext copies of the two letters in the National Archives

In the National Archives there is a plaintext copy of the 1674-1-24 letter sent from Van Goens senior to the Lords Seventeen in Amsterdam (Van Goens, 1674a) and of the 1674-1-23 letter sent to the governor general in Batavia (Van Goens, 1674b).[3] In Table 3 and Table 4 they are compared. Leeuwenson (1675) tells that the ciphertext was based on two copy letters. The plaintext letter from 1674-1-24 is neither the revealed plaintext from the ciphertext nor the original or copy letter it was based on. For the 1674-1-23 letter, we must draw the same conclusion as for the 1674-1-24 letter.



Revealed plaintext cipher	Plaintext letter Van Goens, 1674a
Signed 1674-1-24 list with signatories; one name not filled in ("...")	Signed 1674-1-24 list with signatories; not filled in name is replaced by Cornelis Strick; name Wiltvanck is missing
In minuut 1674-2-9 by Leeuwenson secretary	In minuut 1674-7-7 by Schoock clerk
Encoded 1674-3-7 by Leeuwenson	

Table 3: Two plaintexts of letter 1674-1-24 to the Lords Seventeen in Amsterdam compared.

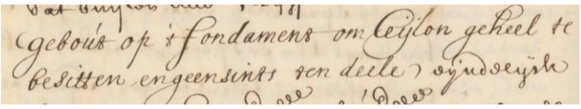


Revealed plaintext cipher	Plaintext letter Van Goens, 1674b
Signed 1674-1-23 list with signatories	Signed 1674-1-23 list with signatories
Accorded by Schoock clerk	*One of eight copy letters*
Encoded 1674-2-28 by Leeuwenson secretary	

Table 4: Two plaintexts of letter 1674-1-23 to the governor general in Batavia compared.

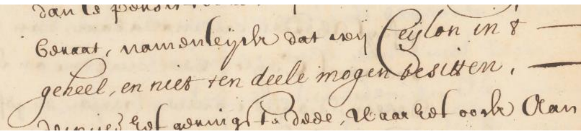
A closer look shows that the content of the plaintext letters are identical. The differences are minor. We may assume that the original and copy letters are identical. We can learn from this the following for the encrypting and decrypting process:

- Catchwords as page numbers are used both in the ciphertext and the plaintext.
- Symbol meaning *full* or *whole*, see Table 5, is used both in the cipher- and the plaintext.
- Abbreviations are used with the same words in the ciphertext and the plaintext.
- Year *1674* as number in the plaintext appears *anno 74* or *seventy and four* in the ciphertext. Particularly striking is that the place name and date of signing and encrypting are completely converted into words and then encrypted. As a result, a first look at the ciphertext will give the enemy at the end of the letter no indication of the sender or date of shipment.
- In the plaintext of Letter One – in Letter Two we do not find this – there are three places where the text is written bigger and in calligraphy. This is not reflected in the ciphertext. These sentences were more important and certainly have been an important part of Leeuwenson's additional oral information (see Subsection 3.2).



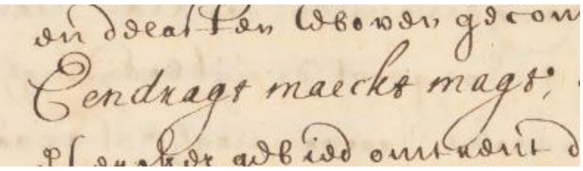
gebout op 't fondament om Ceylon geheel te
besitten en geensints ten deele) sijnde eÿgh

Figure 4: First of three sentences written bigger and in calligraphy in the plaintext (Van Goens, 1674a).



Gouvernt, namentlijck dat wij Ceylon in 't
geheel, en niet ten deele mogen besitten,

Figure 5: Second sentence written bigger and in calligraphy in the plaintext (Van Goens, 1674a).



Eendragt maeckt magt;

Figure 6: Third sentence that is written bigger and in calligraphy in the plaintext (Van Goens, 1674a).

[3] We also found an 'Appendix' dated 1674-1-24 from Van Goens senior to the Lords Seventeen in plaintext (Van Goens, 1674c). This 'Appendix' was not included in the ciphertext. We have analyzed its content but it gives no additional information about the ciphertext or key. List with signatories: Van Goens junior and board and secretary Faa. Colombo, 1674-2-13. Mark, Van Goens senior is missing in the list of signatories but he is mentioned in the heading.

Sentences:

1. Dutch: *Gebout op ‘t fondament om Ceijlon geheel te besitten en geensints ten deele*. (English: built on the foundation to possess all of Ceylon and not only a part). See Figure 4.
2. Dutch: *Ceijlon in ‘t geheel, en niet ten deele mogen besitten*. (English: owning Ceylon in its entirety and not a part). See Figure 5.
3. Dutch: *Eendragt maeckt magt*. (English: unity is strength). See Figure 6.

5 How important was sending the ciphertext?

What Van Goens senior didn’t tell in Letter One nor in Letter Two of the ciphertext is that the conquest of the interior of Ceylon didn’t go as planned. In fact, he lied when he wrote in Letter Two that he “absolutely” overcome the attacks from the men of the King of Kandy. The King waged a guerilla war since 1670 and Van Goens senior and his soldiers didn’t have an adequate answer to that (Arasaratnam, 1956).

5.1 An echo in 1675 of his ‘Vertoog’ from 1655

For Van Goens senior an expansion was the only military-strategic solution to solve the threat of the French and other European competitors in the Ceylon region. Next to that, Arasaratnam (1956) shows that the Ceylon government had, between 1666-1674, serious financial problems. The expenses were significantly higher than its income.

From 1665 onwards, Van Goens senior sent many letters and reports to the governor general in Batavia and the Lords Seventeen in The Netherlands, pleading for his great Ceylon project. To quote Arasaratnam (1956: 80), starting from 1673: “There was a sense of urgency in Van Goens’ efforts, for he realised that if his schemes were not adopted then, they would never be put into operation.” With the French trapped, in St. Thomé on the Coromandel coast since 1672, he made another bold (Dutch: *recklige stoute*) move. An echo of his ‘Vertoog’ from 1655 were he got an approval for his strategic plans after presenting them personally. To enforce his strategic plans from 1674 he sent his secretary Leeuwenson in person to The Netherlands. That was most important for Van Goens senior.

5.2 Did Van Goens senior get what he asked for?

Did Van Goens senior get a ‘Go!’ for his 1674 plans? The answer is: ‘No!’ The Lords Seventeen changed their expansive strategy in the fall of 1673 – one and a half year before Leeuwenson was able to deliver the letter and even before Van Goens senior wrote it – to a defensive strategy: spend less money as a company by reducing the number of soldiers and addressing the sprawl of fortifications.

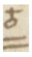
Van Goens senior’s request to conquer the interior of Ceylon and the island Ramanacoil and the area around it and to obtain 1,000 more soldiers was denied. Ramanacoil was a bridge too far.

6 Ciphertext and key

This section shows technical details about the used cipher and the key.

6.1 Technical analysis

The cipher is a monoalphabetic substitution cipher where every plaintext letter is always replaced with the same symbol of the ciphertext alphabet. Of the 26 letters of the Latin alphabet 24 are being used. The letters *V* and *J* are missing. There are symbols for the following five double letters: *EE*, *FF*, *LL*, *OO*, and *PP*. There are seven words that have a separate nomenclature element (code symbol), for example one is used for *Ramanacoil*.

Only numbers occur as inline cleartext. Catchwords are used as page numbers. Abbreviations and punctation marks are also used. An abbreviation is not always written in the same way. The way abbreviation marks should be interpreted and expanded depends on their context in the sentence. For example  is

OORT in *ANTW[OORT]* and *EIT* in *SWARIGH[EIT]*.

Plaintext symbol(s)	**Number of ciphertext symbols**	**Notes**
Latin letters A to Z	24	V and J are missing.
Double Latin letters	5	EE, FF, LL, OO, and PP
Words	7	UEDLE, ENDE, RAMANACOIL, CEYLON, EYLAND, VDR, and SOO
Total code symbols according to key	**36**	
Code symbols not in key	1	Meaning *full* or *whole*. Example: for 40 *whole* guilders
Abbrevations	9	:3, =3, =R, =S, =T, =A, =M, =O, and =L
Punctuation marks	9	**:** (colon), ---- (line), **.** (stop), **,** (comma), **‘** (apostrophe), (space), // (forward slashes), **…** (ellipsis), and **(: :)** (round brackets)
Total code symbols	**55**	

Table 5: Symbols used in the cipher.

The key consists of 36 graphical signs, as shown in Table 5. (See Figure 1 for a facsimile of the key). Only one symbol in the ciphertext is not accounted for in the key. This is the symbol for *full* or *whole*.

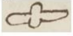
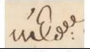
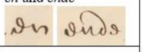
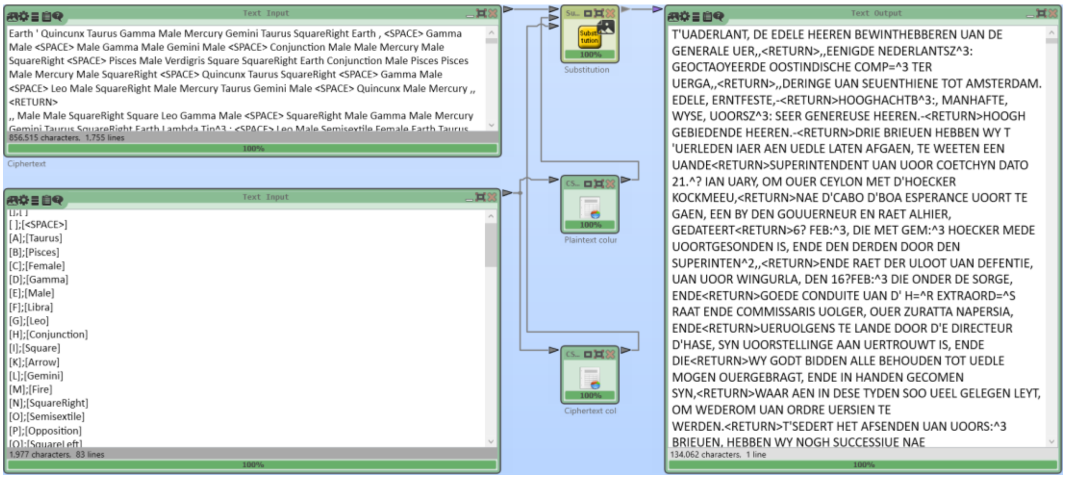
Key symbol	**Key definition**	**Transcription**	**Count in cipher**	**In plaintext Van Goens 1674a**
	UEDLE	<Equivalent>	187	
	EN(DE)	<Sun>	829	*en* and *ende* 
	RAMANACOIL	<L>	0	
	CEYLON	<Cup>	0	
	EYLAND	<Leaf>	0	
	VDR	<Corner>	1 *Probably Leeuwenson read number 7 instead of letter T*	
	SOO	<Intersection>	0	

Table 6: Counts of occurrences of nomenclature elements within the ciphertext.

Only two of the seven nomenclature elements appear in the ciphertext, *UEDLE* (English: *Your Lordship*) and *ENDE* (English: *and*), as shown in Table 6. The other nomenclature elements are not used in all of the 39 pages of ciphertext.



Earth ' Quincunx Taurus Gamma Male Mercury Gemini Taurus SquareRight Earth , <SPACE> Gamma Male <SPACE> Male Gamma Male Gemini Male <SPACE> Conjunction Male Male Mercury Male SquareRight <SPACE> Pisces Male Verdigris Square SquareRight Earth Conjunction Male Pisces Pisces Male Mercury Male SquareRight <SPACE> Quincunx Taurus SquareRight <SPACE> Gamma Male <SPACE> Leo Male SquareRight Male Mercury Taurus Gemini Male <SPACE> Quincunx Male Mercury ,, <RETURN>
,, Male Male SquareRight Square Leo Gamma Male <SPACE> SquareRight Male Gamma Male Mercury Gemini Taurus SquareRight Earth Lambda Tin^3 : <SPACE> Leo Male Semisextile Female Earth Taurus

856,515 characters, 1,755 lines

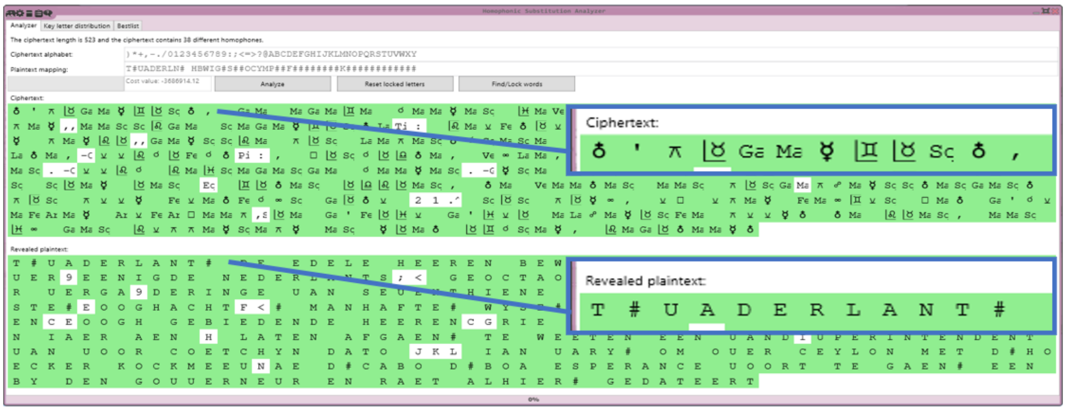
```
[];[ ]
[ ];[<SPACE>]
[A];[Taurus]
[B];[Pisces]
[C];[Female]
[D];[Gamma]
[E];[Male]
[F];[Libra]
[G];[Leo]
[H];[Conjunction]
[I];[Square]
[K];[Arrow]
[L];[Gemini]
[M];[Fire]
[N];[SquareRight]
[O];[Semisextile]
[P];[Opposition]
[Q];[SquareLeft]
```

1,977 characters, 83 lines

T'UADERLANT, DE EDELE HEEREN BEWINTHEBBEREN UAN DE GENERALE UER,,<RETURN>,,EENIGDE NEDERLANTSZ^3: GEOCTAOYEERDE OOSTINDISCHE COMP=^3 TER UERGA,,<RETURN>,,DERINGE UAN SEUENTHIENE TOT AMSTERDAM. EDELE, ERNTFESTE,-<RETURN>HOOGHACHTB^3:, MANHAFTE, WYSE, UOORSZ^3: SEER GENEREUSE HEEREN.-<RETURN>HOOGH GEBIEDENDE HEEREN.-<RETURN>DRIE BRIEUEN HEBBEN WY T 'UERLEDEN IAER AEN UEDLE LATEN AFGAEN, TE WEETEN EEN UANDE<RETURN>SUPERINTENDENT UAN UOOR COETCHYN DATO 21.^? IAN UARY, OM OUER CEYLON MET D'HOECKER KOCKMEEU,<RETURN>NAE D'CABO D'BOA ESPERANCE UOORT TE GAEN, EEN BY DEN GOUUERNEUR EN RAET ALHIER, GEDATEERT<RETURN>6? FEB:^3, DIE MET GEM:^3 HOECKER MEDE UOORTGESONDEN IS, ENDE DEN DERDEN DOOR DEN SUPERINTEN^2,,<RETURN>ENDE RAET DER ULOOT UAN DEFENTIE, UAN UOOR WINGURLA, DEN 16?FEB:^3 DIE ONDER DE SORGE, ENDE<RETURN>GOEDE CONDUITE UAN D' H=^R EXTRAORD=^S RAAT ENDE COMMISSARIS UOLGER, OUER ZURATTA NAPERSIA, ENDE<RETURN>UERUOLGENS TE LANDE DOOR D'E DIRECTEUR D'HASE, SYN UOORSTELLINGE AAN UERTROUWT IS, ENDE DIE<RETURN>WY GODT BIDDEN ALLE BEHOUDEN TOT UEDLE MOGEN OUERGEBRAGT, ENDE IN HANDEN GECOMEN SYN,<RETURN>WAAR AEN IN DESE TYDEN SOO UEEL GELEGEN LEYT, OM WEDEROM UAN ORDRE UERSIEN TE WERDEN.<RETURN>T'SEDERT HET AFSENDEN UAN UOORS:^3 BRIEUEN, HEBBEN WY NOGH SUCCESSIUE NAE

134,062 characters, 1 line

Figure 7: CT2 Substitution component decrypting the ciphertext using the digitized key.



The ciphertext length is 523 and the ciphertext contains 38 different homophones.

Ciphertext alphabet:)*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXY

Plaintext mapping: T#UADERLN# HBWIG#S##OCYMP##F########K###########

Cost value: -3686914.12

Ciphertext:

Ciphertext:

♁ ' ⊼ |♉ Ga Ma ☿ |♊ |♉ Sc ♁ ,

Revealed plaintext:

T # U A D E R L A N T # D E E D E L E H E E R E N B E W
U E R 9 E E N I G D E N E D E R L A N T S ; < G E O C T A O
R U E R G A 9 D E R I N G E U A N S E U E N T H I E N E
S T E # E O O G H A C H T F < # M A N H A F T E # W Y S E #
E N C E O O G H G E B I E D E N D E H E E R E N C G R I E
N I A E R A E N H L A T E N A F G A E N # T E W E E T E N E E N U A N D I U P E R I N T E N D E N T
U A N U O O R C O E T C H Y N D A T O J K L I A N U A R Y # O M O U E R C E Y L O N M E T D # H O
E C K E R K O C K M E E U N A E D # C A B O D # B O A E S P E R A N C E U O O R T T E G A E N # E E N
B Y D E N G O U U E R N E U R E N R A E T A L H I E R # G E D A T E E R T

Revealed plaintext:

T # U A D E R L A N T #

Figure 8: CT2 Homophonic Substitution Analyzer component. The upper rectangular text part of the screenshot shows the transcribed ciphertext. The lower rectangular text part shows the deciphered plaintext.

The word *CEYLON* occurs 42 times in Letter One and 23 times in Letter Two of the revealed plaintext as separate alphabetic letters instead of using the corresponding nomenclature element. The word *RAMANACOIL* occurs respectively three times in Letter One and once in Letter Two, also without using its nomenclature element. This seems to indicate that Leeuwenson did not make the key himself when he had to encrypt the letters in 1674. It makes no sense to add symbols to a key that are not being used in a ciphertext. It also seems to indicate that this is not the key that he wrote down before he arrived in Amsterdam on 1675-1-5. He should still have known very well, after his overland journey, which symbol represents which letter. Encrypting the two letters in 1674 must have taken him a few days of work.

6.2 Cryptanalysis with CrypTool 2

We employed our open-source software CT2 to perform automatic as well as semi-automatic cryptanalysis. At first, CT2 can be used to identify the used type of cipher. After the identification, special components for cryptanalyzing and breaking the cipher can be applied. CT2 implements a graphical programming language, which allows combining different ciphers as well as cryptanalysis methods, implemented in components. CT2 contains, for example, special components for cryptanalyzing monoalphabetic, polyalphabetic, and homophonic ciphers. See Kopal (2018) for a more detailed introduction to CT2 and its components. Since we were in possession of the original key, which appears in the document, there was no need to perform a cipher type analysis. From the start, we assumed that the cipher is a monoalphabetic substitution cipher with some nomenclature elements. Therefore, we entered the key manually into CT2. With the help of the substitution component, we were able to decrypt most parts of the ciphertext correctly (see Figure 7). Additionally, we used the Homophonic Substitution Analyzer of CT2, since it allows viewing the plaintext and ciphertext below each other (see Figure 8). In addition, the Homophonic Substitution Analyzer is able to visualize some of the original ciphertext symbols using UTF-8 characters (but this feature is still work in progress).

6.3 Tips and tricks for digital transcriptions done manually

As the CT2 software worked easy and flawlessly it can be a meaningful tool for historians, too.

We have four tips and tricks for digital transcriptions that are done manually:

1) Use a tool for **counting unique words**. The biggest constraint was to get a digital transcription without duplications and errors in the list of used transcription symbols. The symbol is, by the different members of the transcription team, transcribed as *Earth*. But in the early versions there were also variants that were apparently a typo, for example *eArth*, *Eerth*, or *earth*. CT2 will not recognize the typos and the result is that symbols, that are not represented in the CT2 key, will not be decrypted. We have overcome this by using *Unique Words Count*[4] on the digital transcribed ciphertext and then cleaned up the errors before entering it into CT2.

2) End every transcribed ciphertext line with a ***hard return***. When the number of pages is large or the lines of the ciphertext are close together, it is useful to have each line in the transcription on a separate line by using a *hard return*. This will result in separate lines in the revealed plaintext too. This makes it easier to compare the ciphertext with the plaintext, line by line.

3) The DECODE database gives uploaded images of ciphertexts a unique name. This name differs from the original name. To avoid post-processing or rework in the plaintext, every time after generating a new output, you can add the **name** of the **image** in DECODE and original in the **nomenclature** of CT2, for example *[DC6955_RAM003];[6955]*. *6955* is the name of the image in DECODE and *RAM003* refers to the third scan of the Ramanacoil ciphertext. Having them both automatically makes it easy to navigate between the pages in both sources. One could also add the name of the folio, for example *f544r*. *[DC6955_RAM003_f544r];[6955]* is then the corresponding nomenclature.

[4] Unique words count at https://planetcalc.com/3205/.

4) The transcription team did identify two symbols in the ciphertext, which in retrospect were differently written variants of existing symbols. If in doubt, one should **create** a **new symbol** and don't smuggle them away. In the output one can analyze these new symbols and resolve them in the key of CT2.

7 Conclusions

Our main findings have been:

1) The plaintext of the deciphered ciphertext reveals two letters that, in retrospect, were already known in the National Archives as plaintext letters (Van Goens, 1674a and 1674b). We have been able to relate them.

2) The ciphertext and the six additional letters bring to light that for Van Goens senior the letter from 1674-1-24 was of utter importance. His personal secretary Leeuwenson had to encrypt them and deliver the letters in person with additional oral information. This paper shows that Van Goens senior wanted to repeat his most successful 'Vertoog' from 1655. Substantiate his goals and get approval for them from the Lords Seventeen. While in 1655 he got a 'Go!', twenty years later in 1675 he got a 'No!'.

3) The encryption process consisted of using the key but also of additional steps, which are not described, to make cryptanalysis more difficult. For example, the year *1674* in the signature is converted into written words before encryption.

Acknowledgments

We would like to thank the following persons. Karl de Leeuw for introducing Dinnissen to the DECRYPT team. Guido van Meersbergen for his revealed plaintext and pointing us at Van Goens, 1674a and 1674b. Crina Tudor and her team of students for the digital transcription of the ciphertext. This work has been supported by the Swedish Research Council, grant 2018-06074: DECRYPT-Decryption of historical manuscripts.

A book is planned for the ciphertext with an introduction and annotations as a facsimile of Ramanacoil (1674), the revealed plaintext in Dutch, and with a translation into English. This publication will also include Van Goens, 1674a, 1674b, 1674c, 1674d, and De Hase, 1674.

References

Sinnappah Arasaratnam. 1958. *Dutch Power in Ceylon 1658-1687*.

Femme Gaastra. 2012. *Geschiedenis van de VOC*.

Rijckloff van Goens, 1674a. 1674-1-24. NL-HaNA, VOC, 1.04.02, inv.nr. *1298*: ff246-256.

Rijckloff van Goens, 1674b. 1674-1-23. NL-HaNA, VOC, 1.04.02, inv.nr. *1298*: ff300-340.

Rijckloff van Goens, 1674c. 1674-1-24. NL-HaNA, VOC, 1.04.02, inv.nr. *1303*: unfoiled, scans 191-198.

Rijckloff van Goens, 1674d. 1674-2-10. NL-HaNA, VOC, 1.04.02, inv.nr. *1292*: ff539-543.

De Hase, 1674. 1674-5-2. NL-HaNA, VOC, 1.04.02, inv.nr. *1302*: ff727-731.

Nils Kopal. 2018. Solving Classical Ciphers with CrypTool 2. *Proceedings of the 1st Conference on Historical Cryptology*, HistoCrypt 2018: 29-38.

Gerrit Knaap and Ger Teitler editors. 2002. *De Verenigde Oost-Indisch Compagnie: Tussen Oorlog en Diplomatie*.

Gerrit Knaap, Henk den Heijer and Michiel de Jong. 2015. *Oorlogen Overzee: Militair optreden door compagnie en staat buiten Europa 1595-1814*: 85-122; 430-432.

Joannes Leeuwenson, 1675. 1675. NL-HaNA, VOC, 1.04.02, inv.nr. *4894*: unfoiled, scans 0001-0280.

Leupe. 1863. Daghregister van de landreijs, gedaen bij mij Joannes Leeuwenson… beginnende anno 1674. *Bijdragen tot de Taal- Land- en Volkenkunde van Nederlandsch Indië* VI: 94-112.

Guido van Meersbergen. 2009. *'In goede en vertroude handen': Communicatie en beleid bij de VOC tijdens de Hollandse Oorlog (1672-1678).*

Guido van Meersbergen. 2011. 'In goede en vertroude handen': Communicatie en beleid bij de VOC tijdens de Hollandse Oorlog (1672-1678). *De Zeventiende Eeuw*, 27 (1): 80-101.

Beáta Megyesi, Nils Blomqvist and Eva Pettersson. 2019. The DECODE Database. Collection of Historical Ciphers and Keys. *Proceedings of the 2nd Conference on Historical* Cryptology, HistoCrypt 2019: 69-78.

Ramanacoil, 1674. 1674-1-23 and 1674-1-24. NL-HaNA, VOC, 1.04.02, inv.nr. *1292*: ff544-563.

3D Digitalization of historical cipher machines using computed tomography

Matthias Göggerle
Deutsches Museum / München
m.goeggerle@deutsches-museum.de

Carola Dahlke
Deutsches Museum / München
c.dahlke@deutsches-museum.de

Abstract

Being already an established method for non-destructive examination of cultural heritage objects from a conservational perspective, computed tomography is getting more and more popular for answering historical questions. As part of the three-year project *3D-Cipher*, the technology will be applied to scan and digitize 61 historical cipher machines ranging from the late 19th century to the 1990s. The German Federal Ministry of Education and Research funds the project in the eHeritage program[1], which has the goal of supporting the digitalization of cultural heritage objects and making them accessible for researchers.

The aim of this contribution is to introduce the museum's collection as well as the project's idea and relevance to cryptologic researchers. Since 3D scans are able to provide non-destructive insights into our rare exhibits, we can thus hopefully contribute by making our devices available to scientists.

1 Introduction

The Deutsches Museum has a large collection of historical cipher machines ranging from the late 19th century to the 1990s. As part of the three-year 3D-Cipher project, 60+1[2] objects of the cryptologic collection will be scanned and digitalized using computed tomography technology. The scan data and 3D-models will then be made available online in an open access format for international researchers.

The main goal of this project is to enable further research with the newly generated 3D-CT digitalization data of the cipher machines. Therefore, an adequate presentation of the project results is a necessity and an integral part of the project. The enrichment of the CT data with technical and scientific information is an important step in this process. To learn more about the requirements and wishes of the researchers, we want to use this contribution to present the project in its early stage to the crypto community and start a knowledge exchange.

2 Computed tomography and other 3D scanning technologies

2.1 Surface scanning technologies

As part of the digital surge of museum collections, 3D scanning techniques are increasingly being used in addition to 2D photographic recordings. The purpose and the outcome of 3D scans vary greatly with the different techniques applied. The most common are surface scanning technologies like photogrammetry, structured light scanning or laser scanning, which capture the surface of objects.[3]

2.2 CT – functional principle

The great advantage of the CT technology in comparison with the above-mentioned techniques is the possibility to scan the interior of objects. Using X-ray measuring from various angles, CT is not only able to show inside layers of objects, but can also be used to build digital 3D models. There are two main forms of CT technology, medical and industrial CT scanning. One of the main differences is the lower voltage in medical CT scans to minimize the radiation on the human body. Furthermore, the X-ray source will move around the body in medical CT scanners, whereas the object itself will usually move

[1] URL: https://www.geistes-und-sozialwissenschaften-bmbf.de/de/eHeritage-1736.html (26.05.2021)

[2] The project includes a rare and well-preserved SG-41 device from WWII owned by a private collector.

[3] A good overview with advantages and disadvantages of the different technologies can be found here: URL: https://bitfab.io/blog/types-of-3d-scanning/ (26.05.2021).

in the X-ray beam in industrial scanners.[4] Industrial scanners also have a higher resolution, depending on the size of the scanner and of the object, ranging from five to 150 µm Voxel size in Macro and Micro-CT scanners up to 0.5 µm in Nano-CT scanners. The best resolution of medical scanners is around 70 µm (Hanke, 2010; Du Plessis 2016).

All CT scanners use the penetration of the X-ray beams to measure the density of different materials. Different materials absorb the radiation to different degrees; the picture on a detector panel appears darker or lighter depending on the absorption. The rotation of the source/ the object and the repeated measurements from different angles lead to a large number of X-ray scans in the x-, y- and z-axis that can then be reconstructed to a 3D-CT-model (Luccichenti, 2005).

2.3 3D-Reconstruction

Since the absorption of the X-rays is measured, the free space (i.e. air) is identified as a material with very low opacity. The result of the CT-scan is a digital cube, consisting of materials with different absorption rates. With the appropriate software, the parameters can be adjusted to see the different materials in a 3D reconstruction.

For further measurements or segmentation of different parts, e.g. rotors, gears, screws, the different materials, including the free spaces, have to be separated from each other. Automated processes exist, but for high quality results, elaborate manual editing is still required (Luccichenti, 2005).

To complicate things even more, visual artefacts appear in the scanning process where materials with high variance in density meet, i.e. metal artefacts e.g. beam hardening and scatter that result in black and white streaks (Boas, 2012).

2.4 Usefulness of CT technology

Despite these challenges, the possibility of non-destructive inspection of the interior of objects makes the CT technology an invaluable tool for researching unknown features of fragile objects. These include changes or small fractures that otherwise would not be noticed.

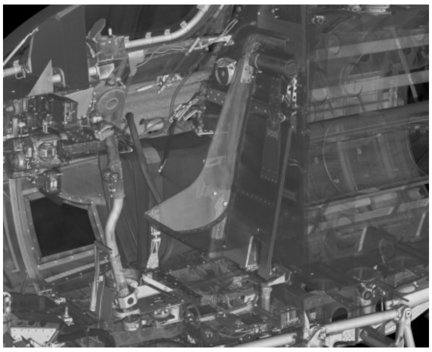


Fig. 1: 3D-CT-rendering of a WWII airplane cockpit (Fraunhofer IIS, EZRT/ Deutsches Museum, CC BY-SA 4.0).[5]

Depending on the quality of the scan data and the extent and accuracy of the segmentation process, even completely reverse engineered 3D-models are possible that include the details of mechanisms.[6] These features are very useful concerning the cryptologic collection researched in this project.

3 Collection overview

The museum's collection of cipher machines is very extensive, and depicts the variance of 120 years of Central European mechanical cryptology. The collection can be categorized into five chronological periods:

3.1 The beginning of mechanical encryption around 1900

The oldest cipher machines in the collection date from before 1900 and do not yet contain any complicated techniques. They are particularly interesting because some of them are completely unknown; e.g. a very early prototype of a cipher machine was donated to the museum by the Danish inventor Alexis Køhl himself. Other machines from the 1900s to the 1920s follow, e.g. devices from Friedrich Rehmann and

[4] There are further subdivisions, which would go beyond the scope of this brief introduction. A good summary can be found in Hanke (2010).

[5] Similar quality can be expected for the cipher devices. Link to the project, URL: https://www.deutsches-museum.de/presse/presse-2019/me-163/#c137993 (26.05.2021)

[6] The team from Prof. Philip Withers at the University of Manchester scanned and segmented an Enigma in 2018. URL: https://www.manchester.ac.uk/discover/news/x-ray-imaging-reveals-the-secrets-inside-the-enigma-machine/ (26.05.2021).

Alexander von Kryha. CT scans of these rare devices will complete the existing data available so far.

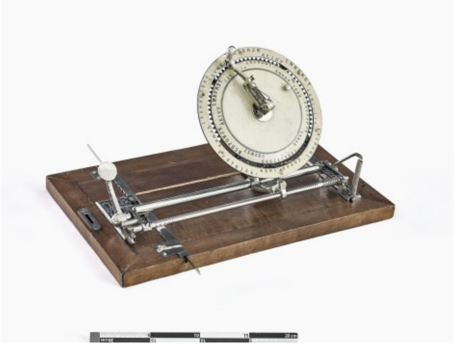


Fig. 2: Index Typewriter „Diskret" by Friedrich Rehmann, (Deutsches Museum/ Konrad Rainer, Inv.-No. 67624, CC BY-SA 4.0).

3.2 Cipher machines of World War II

The collection contains mainly German cipher machines from Heimsoeth & Rinke, Siemens, Lorenz and Wanderer Werke AG, but also includes a variety of Hagelin machines from A. B. Cryptoteknik and L. C. Smith. Some types are already very well analyzed (i.e. army and naval Enigma models). Others, e.g. the German Siemens secret teleprinter T52, are much rarer and their interesting history is less present. For this very reason, scan data give the opportunity for the not-so-well-known devices to be examined more closely.

3.3 Cipher machines of the post-war period

Shortly after the end of the Second World War, a large number of different cipher machines were developed. Transmission was still mainly by radio or telegraph cable. Many of these devices are still not well known.

This period is represented in the collection by some successors to Enigma models, such as NeMa and Fialka, and Hagelin devices from Crypto AG and Rudolph Hell. Other items of a different kind by the Swedish company Transvertex and the company Stenographic Machines, Inc. add to the picture.

So-called mixers, i.e. devices meant to encrypt with random sequences on punched tape, are represented by devices of the companies Crypto-AG and Siemens & Halske.

In the context of the gentlemen's agreement between Boris Hagelin and William Friedman (see e.g. the declassified report of W. Friedman, 1955), the comparison of scan data from various Hagelin devices of the C- and CX-series in the collection is planned. Particular interest lies in a possible detection of differences in the mechanics that are usually not accessible without opening and thus destroying the rare devices.

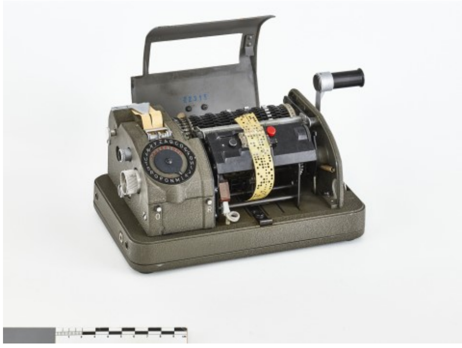


Fig. 3: Hagelin CX-52/RT (Deutsches Museum/ Konrad Rainer, Inv.-No. 2017-389, CC BY-SA 4.0).

3.4 Beginning of the computer age from 1965

From 1965, encryption algorithms are no longer mechanically driven, but executed on circuit boards. Almost none of the devices from this period have been investigated yet, and for a comprehensive study of the algorithms, contemporary documents are required.

The collection is rather extensive in this section and contains devices of the companies Mils Electronic, Crypto AG, Tele-Security Timmann, Telta. Telefunken, Philips and ANT Nachrichtentechnik GmbH.

Since the electronic components from this period are still easily recognizable, the CT scans will enable to clarify how specific functions and algorithms have been implemented electronically.

3.5 Computer age from 1980

From the 1980s onwards, almost all devices are completely unknown, and had been kept top secret until a few years ago. Most are grey or black tightly welded boxes and from the outside it is difficult to see what might be inside or what they were used for. Above all, documentation is hard to come by.

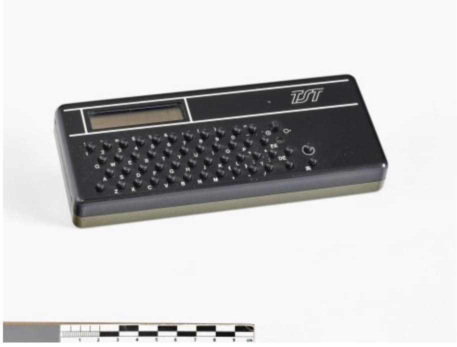


Fig. 4: TST 3226 by Tele Security Timmann (Deutsches Museum/ Konrad Rainer, Inv.-No. 2017-410, CC BY-SA 4.0).

The devices were mostly designed for data and voice encryption and transmitted via telephone. Random number generators, modems and faxes are part of the exhibits from this period. CT scans are essential to uncover this part of our collection. This way, one can at least narrow down what the device did. Apart from studying the components inside and identifying their modes of operation, we are in particular interested in revealing the object histories of some devices. Tele-Security Timmann devices from this period are said to contain a special copy protection, i.e. a substance that fills the device and destroys the inner parts if it is opened by force. We are curious to see whether CT scans will reveal any information.

4 Conclusion

4.1 CT & the collection

The CT technology has the possible features to answer open research issues across the spectrum of the collection. At least two WWII machines will be processed via segmentation of important functional parts to enable a direct comparison between them. From every other object, we intend to create a 3D-CT-model that can be used for further editing in the future by international researchers. The Open Access Policy is crucial at this point and we hope to engage in collaboration with various experts as part of the project.

4.2 Researching the data

The CT data can be analyzed with proprietary software. Although the operation of the software requires a certain knowledge and the handling of the data makes a high-end computer, e.g. with lots of RAM, a necessity, we will freely share the data with international researchers for further enquiry.

4.3 Presentation of the data

The above-mentioned process is important for research purposes, but is hidden from a broader audience due to the technological requirements. Therefore, we plan to show the 3D-CT models in an online web viewer with the possibility to download these models for further use.[7] This two-way approach with research data and online exhibition will hopefully path a way to uncover the last secrets of the cipher machines.

4.4 Upcoming roadmap

The first images and 3D reconstructions are expected in the summer 2021. Following this, we will enrich and research the CT data in the museum and prepare the publications. We aim to submit a long paper in the following HistoCrypt proceedings, including first research results.

References

Anton Du Plessis et al, 2016: Laboratory X-ray micro-computed tomography: a user guideline for biological samples. In *GigaScience* 2017, 6(6), p. 1–11.

Franz Edward Boas and Dominik Fleischmann: CT artifacts: Causes and reduction techniques. In *Imaging in Medicine* 2012, 4 (2), p. 229–240.

Giacomo Luccichenti et al.: 3D reconstruction techniques made easy: Know-how and pictures. In *European Radiology* 2005, 15 (10), p. 2146–2156.

Randolph Hanke: *Computertomographie in der Materialprüfung. Stand der Technik und aktuelle Entwicklungen*. Fürth 2010.

William Friedman, 1955, National Security Archive REF: A2436259. Report of Visit to Crypto A.G. (Hagelin) by William F. Friedman, Special Assistant to the Director, National Security Agency. March 15, 1955. Top Secret.

Webpages

https://bitfab.io/blog/types-of-3d-scanning/ (26.05.2021).

https://www.deutsches-museum.de/presse/presse-2019/me-163/#c137993 (26.05.2021).

https://www.geistes-und-sozialwissenschaften-bmbf.de/de/eHeritage-1736.html (26.05.2021).

https://www.manchester.ac.uk/discover/news/x-ray-imaging-reveals-the-secrets-inside-the-enigma-machine/ (26.05.2021).

https://musices.gnm.de/ (26.05.2021)

[7] A possible solution was used by the Germanisches National Museum in cooperation with the Fraunhofer EZRT in a project that CT scanned historical musical instruments. URL: https://musices.gnm.de/ (26.05.2021).

Documents of Polish-Soviet War of 1919-1920 Codebreaking

Marek Grajek
Independent researcher
mjg@interia.eu

Abstract

Codebreaking during the Polish-Soviet war of 1919-1920 not only assured Polish victory in this conflict but also provided the foundations for the future triumph of the Cipher Bureau over Enigma. Original documents from that period not only survived several storms of history, but have been digitized and are now available for the researchers. This paper is divided into three parts. The first one drafts the historical context of the documents, the second presents their structure and contents, and the final one offers some remarks regarding errors committed by Soviet cipher clerks which had facilitated Polish victory.

1 Introduction

Having been reborn in 1918, after 123 years of partitions, Poland had no tradition in the cryptology or the codebreaking. Its international situation did not place either of them in the center of attention. Immediately after its resurrection the new state had to fight five wars on its only vaguely defined borders. It was natural for its leaders to focus on the number of available bayonets and sabers rather than on arcane and mysterious discipline – the codebreaking. But in spite of this understandable tendency its was the codebreaking that provided the cornerstone for Polish victory in the most deadly conflict of that period – war with the Soviet Russia in 1919-1920.

Only few documents from that period survived the storms of history that kept rolling over Poland through the next decades. Files referring to the cryptology and the codebreaking operations are usually well guarded and protected from falling into foreign hands. Historians knew quite a bit about the scale of Polish success with the Soviet ciphers from the indiscretions of the participants of the events (Wyżeł-Ścieżyński, 1928). However, it seemed unlikely that the original documents of this operation might have survived and reemerge in rather surprising circumstances.

After the collapse of communism in Poland most archives of the former secret police were transferred to the civilian institutions. Historians were surprised to find among them the presumably complete archive of the Polish Army codebreaking operation from the period of the Polish-Soviet war. The stamps and inventory numbers on the files witnessed its long and complicated journey to its final destination – country's Central Military Archive. Over a period of more than ten years the files have been catalogued, digitized and made available for the researchers. Finally, in 2017, the entire archive was included into the UNESCO Memory of the World register.

2 Historical background

This paper is not intended to introduce the reader into the history of the Polish-Soviet war of 1919-1920. Interested reader will find its more extensive coverage in (Davies, 2003) and (Zamoyski, 2008). Minimal historical background provided below is addressed mostly to the readers interested mainly in cryptography.

Polish-Soviet war of 1919-1920 broke out undeclared. On 5 February 1919 Poland and Germany had signed an agreement concerning the evacuation of German troops stationed at the former eastern front of WWI. Their gradual transfer to Germany was leaving vacuum in the previously occupied Polish and Russian territories. That vacuum was being gradually filled in by the troops of the neighboring states: Soviet Russia, Poland, Ukraine and the Baltic countries. Considering the collapse of the Tsarist Russia, replaced by the aggressive Soviet regime, emergence of the successor states and lack of the defined and recognized borders between them, peaceful solution seemed unlikely.

Polish soldiers first clashed with the advancing Soviet troops on 14 February 1919 near Mosty, stopping the Soviet advance and then gradually pushing Bolsheviks back to the east, reaching in August of the same year Minsk, Bobruisk and Borisov. During the following period of lull, in July and August 1919, a lucky coincidence facilitated Polish breakthrough with the Soviet ciphers. One of the officers of the emerging cipher service of the Polish Army wished to dance at his sister's wedding and asked a colleague for replacement at the night duty. Lieutenant Jan Kowalewski had no previous experience with the ciphers or the codebreaking, but his perfect knowledge of Russian language plus common sense permitted him to break the cipher before the morning. Kowalewski was immediately transferred to the cipher section of the General Staff, where during the following months he managed to organize an effective and efficient codebreaking service.

Polish codebreakers permitted the Polish Army HQ an almost complete penetration of enemy's communications and played a crucial role in pivotal episodes of the war. More or less at the same time when Kowalewski was breaking the first Soviet message, Polish-Soviet peace talks started in Mikaszewicze. Bolsheviks, fighting at the same time desperately against Denikin's white Russians, were offering considerable territorial concessions for the peace at the Polish front. Some historians describe Polish operation in Ukraine in April 1920 as an unprovoked aggression. Two facts contradict this opinion. Polish Army was entering Ukraine in alliance with the Directorate of People's Republic of Ukraine. Kowalewski and his service provided the second critical element of decision. Immediately after decisive Soviet victory over Denikin, Polish codebreakers were able to detect a fast buildup of the Soviet forces at the Polish front, indicating clearly Soviet aggressive intentions; escalation of the conflict was unavoidable.

During the following operations the codebreakers managed to play the decisive role. Their precise information about Soviet forces in Ukraine assured a complete Polish victory in this theater of operations. The codebreakers were also able to provide a timely warning about the Budionny's First Cavalry Army being transferred from Caucasus to the Polish front, changing thus the strategic situation in Ukraine. During the operations following Soviet attack in the northern front sector on 4 July, information provided by the codebreakers was of utmost importance for the Polish Army HQ. Warfare took highly mobile character. Polish troops were forced to execute the strategic retreat of over 600 kilometers, ending in mid-August at the gates of Warsaw. During that period Polish forces at the front line and beyond it were instructed to damage existing wire networks, forcing the advancing Soviets to go wireless.

When the Soviet divisions were approaching the central Poland, hundreds of thousands of Poles volunteered for military service. Among them were three mathematics professors of the Warsaw University, Stanisław Leśniewski, Stefan Mazurkiewicz and Wacław Sierpiński. Attached to Kowalewski's service they played a critical role during the events of the next few weeks. It was Sierpiński, who in early August had broken the new Soviet cipher key basing on just the single intercepted message. This message, however, presented the complete Marshal Tuchachevski's plan of the decisive Warsaw operation. Precise knowledge of enemy's intentions delivered the foundations for the Polish victory in the ensuing Battle of Warsaw and the entire war. Role played by the mathematicians in this victory was well remembered and provided a cornerstone of the future Cipher Bureau's triumph over Enigma.

3 Fates of Kowalewski's archive

Soon after the victory Jan Kowalewski was transferred to other duties in Polish intelligence service. For some time in 1921/1922 he was teaching cryptology at the Japanese Military Academy. The archive of his service was deposited at the Central Military Archive, where is rested undisturbed until September 1939.

During the Polish campaign in 1939 the Cipher Bureau, successor of Kowalewski's service, managed to evacuate or destroy all the traces of its operation, including in particular its success over Enigma. However, part of the its historical records stored at Central Military Archive fell into the German hands after Warsaw surrender. From the German sources (Reile, 1963) we know that it took six trucks to transfer captured documents to the military archive in Danzig-Oliva, were they were thoroughly examined by the Abwehr staff. This blunder brought tragic consequences for Polish intelligence service; over 100 of its agents in Germany have been identified, captured and mostly executed. But it was probably the same blunder that we owe the preservation of the codebreakers' archive.

Sometime in 1945 Soviet Army captured Danzig, where the entire archive was stored. The documents were transferred in bulk again, this time to the Soviet State Archive. We do know nothing about their fates there, judging however by the results they were considered redundant by their Russian holders and, at time and circumstances unknown, returned to Poland.

There they landed in the archive of the communist secret service, inaccessible to outsiders. Paranoia of secrecy common for the communist regimes, plus the character of the files, witnessing one of the major Polish triumphs over current forced ally, determined their fate for as long as communists ruled the country. It was only after the collapse of communism in Poland, that during the review of the archives files have been discovered, and transferred back to the place of their origin, i.e. Central Military Archive.

Their reappearance sparked considerable sensation among the military historians, catalyzing some reinterpretations of the conflict of 1920 (Nowik, 2004, 2010). This interest led to the digitization of the complete archive, comprising over 20 thousand pages, which is now accessible at:

https://wbh.wp.mil.pl/pl/pages/zdigitalizowane-teczki-polskiego-radiowywiadu-wojskowego-z-1920-roku-wpisanego-na-swiatowa-liste-unesco-pamiec-swiata-2020-06-17-kaf5/

In 2018 entire archive of Polish signals intelligence in 1920 has been added to the UNESCO Memory of the World register. This decision finalized recognition of the Battle of Warsaw as one of the decisive battles in the world history and the decisive role of the codebreakers and codebreaking therein.

4 Structure of the archive

Structure of the digitized archive is slightly chaotic and seems to reflect grouping of the documents adopted originally by the codebreakers in 1919/1920. Although the documents have been fully digitized, PDF files comprising the contents of the original folders have been placed in the directories titled after the their names in Polish language, which does not facilitate the research. This section provides brief notes concerning the contents of every directory in the collection. Names of folders in Polish language appear as the subsection titles.

4.1 Depesze nadesłane z Dowództwa Frontu Południowo-Wschodniego, Dowództwa 1 Armii oraz Dowództwa Poleskiej Grupy do Sekcji RTGt

Original Soviet cipher messages (partially deciphered inline) of messages intercepted by the listening stations of Polish South-Eastern Front Command, 1st Army, and Polesie Group (243 pages).

4.2 Depesze szyfrowane z dowództwa armii i frontów przesłane do NDWP

Continuation of the previous directory: Soviet cipher messages (mostly deciphered inline) of messages intercepted by the listening stations of Front and Army commands (329 pages).

4.3 Depesze szyfrowe z Dowództwa 2, 4 i 6 Armii, Grupy Bieniakonie i D.O.K. Lwów

Original Soviet cipher messages (mostly undeciphered) intercepted by the listening stations of 2nd, 4th and 6th Armies, Bieniakonie Group and Lwów Military District (493 pages).

4.4 Depesze szyfrowe ze Stacji RTG. Telegramy nadesłane z dowództwa armii i frontów

Cipher messages intercepted by the listening stations of Army and Front commands; mostly traffic of foreign diplomatic representations in Soviet Russia (Turkey, possibly other countries), Soviet diplomatic traffic (456 pages).

4.5 Dziennik stacji telegraficznej przy Polskiej Misji Wojskowej w Rydze. Szyfrogramy do Stacji RTG nadesłane z Dowództwa 4 Armii

Station log of Polish Military Mission in Riga. Covers the period of the peace talks between Poland and Soviet Russia (227 pages).

4.6 Komplet tłumaczeń szyfrogramów dotyczących oddziałów Armii Czerwonej

Translations into Polish of the deciphered Soviet messages (260 pages).

4.7 Kopie radiotelegramów Naczelnego Dowództwa WP i podległych oddziałów

Directory name suggests the copies of Polish Army HQ messages, however most of its content represents original Soviet cipher messages, partially deciphered (461 pages).

4.8 Korespondencja dla Delegata Łącznikowego 6 Armii, zestawienie dyslokacji nieprzyjacielskiej, zaszyfrowane depesze Oddziału II

Directory name suggests the copies messages by the liaison officer at the 6th Army, enemy's OdB, and cipher messages of Polish 2nd Dept. (Military Intelligence). Most of the content represents the translations into Polish of the broken Soviet messages (XII and XVI Armies) (574 pages).

4.9 Księgi rozwiązanych szyfrów Armii Czerwonej, oddziałów armii gen. Wrangla i gen. Denikina

One of the most interesting parts of the collection; keys to the Soviet, Wrangel's and Denikin's ciphers (485 pages).

4.10 Materiały Biura Szyfrowego-radiogramy szyfrowe i depesze radiowe nadesłane ze Stacji RTG Grudziądz i Toruń

Cipher and coded messages intercepted by the listening stations in Toruń and Grudziądz. Assortment of various ciphers and codes, mostly of diplomatic nature, some open text messages. Message headers suggest diplomatic traffic between Berlin and Moscow (1095 pages).

4.11 Materiały szyfrowe Sekcji Szyfrowej nadesłane ze Stacji RDT Lwów i Toruń

Cipher and coded messages intercepted by the listening stations in Lwów and Toruń. Assortment of various ciphers and codes, mostly of diplomatic nature. Message headers suggest diplomatic traffic between Turkey and Soviet Russia (781 pages).

4.12 Meldunki bolszewickie w tym zestawienie dyslokacji wojsk i wykaz sygnałów radiostacji sowieckich, a także rad

Deciphered and translated Soviet cipher messages, reports regarding dislocation of the Soviet troops based thereupon. Original texts of messages to and from the Soviet diplomatic representation in Warsaw (1159 pages).

4.13 Radiotelegramy dotyczące sytuacji w Rosji bolszewickiej i Anglii, projektowanej pożyczki dla Polski, sytuacji

Open text messages, mostly by news agencies of several European countries (610 pages).

4.14 Radiotelegramy przejęte przez Stację RTG

Open text messages in several languages, mostly diplomatic and news agency (395 pages).

4.15 Radiotelegramy przejęte przez Stację RTG Toruń i Poznań

Open text messages in several languages, mostly of diplomatic and agency nature (606 pages).

4.16 Radiotelegramy Stacji RTG Wilno i Lwów

Open text messages, official releases of the Red Army HQ and Soviet diplomatic sources, relating mostly to operations against Wrangel's and Denikin's forces (689 pages).

4.17 Radiotelegramy szyfrowe oraz depesze szyfrowe do NDWP wysłane z podległych oddziałów

Original Soviet cipher messages (mostly undeciphered) intercepted by various listening stations (820 pages).

4.18 Radiotelegramy zaszyfrowane nadesłane ze Stacji RTG

Soviet cipher messages, most probably in diplomatic code or cipher, addressed to the head of Soviet delegation for the peace talks in Riga (306 pages).

4.19 Radiotelegramy zawierające komunikaty dotyczące sytuacji politycznej i gospodarczej w krajach europejskich

Open text messages in several languages, mostly of diplomatic nature and news agencies (580 pages).

4.20 Radiotelegramy zawierające komunikaty ze Stacji RTG

Open text messages in several languages, mostly diplomatic and news agencies (450 pages).

4.21 Radiotelegramy ze Stacji Radiotelegraficznej RTG Warszawa, Przemyśl i Lwów

Open text messages in several languages, mostly official Soviet diplomatic messages and news agency releases (341 pages).

4.22 Sprawozdania z toczących się spraw w Referacie Śledczym oraz depesze szyfrowe z podległych oddziałów

Original Soviet military cipher messages (some deciphered inline) intercepted by various Polish listening stations (286 pages).

4.23 Sprawy szyfrów i kodów w Naczelnym Dowództwie. Szyfry nieprzyjacielskie

Polish Cipher Bureau's administrative documents (306 pages).

4.24 Sprawy szyfrów i kodów wraz z tłumaczeniem szyfrów

Polish Cipher Bureau's administrative documents, some news agency releases (252 pages).

4.25 Sprawy szyfrów i kodów. Opinia Sekcji Szyfrowej

Polish Cipher Bureau's administrative documents (12 pages).

4.26 Szyfrogramy nadesłane do Oddziału II NDWP z Dowództwa Grupy Południowej i Dowództwa 2 i 3 Armii

Translations of the decrypted Soviet military messages (490 pages).

4.27 Szyfry nadane przez attaché wojskowych

Cipher messages from Polish military attachés in several European countries (907 pages).

4.28 Szyfry nadesłane do NDWP z Grupy Bieniakonie, Ekspozytury MSWojsk, i Dowództwa 2 i 3 Armii

Soviet military cipher messages, mostly deciphered inline and transcribed (669 pages).

4.29 Telegramy dotyczące sytuacji na froncie nadesłane z Dowództwa 3, 6 i 7 Armii, Dowództwa Grupy Poleskiej

Folder name does not reflect its content: open text messages and orders directed from the HQ of Polish intelligence service to various units of Polish Army (446 pages).

4.30 Tłumaczenia nadesłanych szyfrogramów

Translations of Soviet military cipher messages, mostly relating to the critical phase of 1920 campaign directly preceding the Battle of Warsaw (190 pages).

4.31 Tłumaczenia szyfrogramów sowieckich

Translations of Soviet military cipher messages, mostly relating to the critical phase of 1920 campaign, directly preceding the Battle of Warsaw (2.269 pages).

4.32 Tłumaczenia szyfrów przejętych przez Stację RTG Kraków

Open text releases by Rosta (Russian telegraphic news agency) (581 pages).

4.33 Wyciągi z przechowywanych depesz bolszewickich, wykazy ewidencji personelu armii sowieckiej i tłumaczenia szyfrogramów

Translations of broken Soviet messages, mostly from or to the 1st Cavalry Army. Extracts from various deciphered messages, mostly unrelated to the Polish campaign (Black Sea, Caucasus) (1.217 pages).

4.34 Wykazy depesz szyfrowych nadesłane z Poselstwa Polskiego w Wiedniu i z Dowództwa Frontu gen. Szeptyckiego

Inventory of messages from Polish Military Attaché in Vienna. Soviet messages in various codes and ciphers (384 pages).

4.35 Zaszyfrowane depesze Oddziału II nadesłane przez attaché wojskowych

Messages in cipher from and to Polish military attachés in several European countries and White Russian commands in the South of Russia (294 pages).

4.36 Zaszyfrowane dokumenty i radiotelegramy nadesłane z Dowództwa 7 Armii, Dywizji Legionów i Frontu gen. Szeptyckiego

Soviet coded messages, mostly in 6-letter code and 5-digit codes (329 pages).

4.37 Zaszyfrowane meldunki z podległych oddziałów do NDWP Oddział II

Soviet coded messages, mostly in 6-letter and 5-digit codes (440 pages).

4.38 Zestawienia telegramów wysłanych przez attaché wojskowego w Brukseli do NDWP

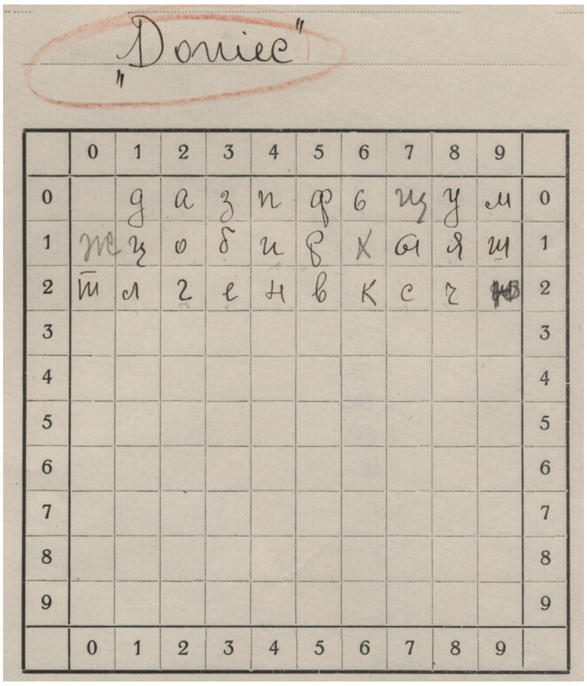
Soviet military cipher messages, some deciphered inline (folder name misleading) (175 pages).

5 Basic features of Soviet military ciphers of 1920 campaign

Discussed archive contains examples of many codes and ciphers used in the period of 1919-1920 by several European and non-European countries. It was natural that Polish signals intelligence was heavily focused on the Soviet Russia, representing the most serious threat to Poland's freshly regained independence.

The archive contains many examples of Soviet codes and ciphers, both military and diplomatic. It seems that Polish codebreakers have not undertaken a serious attack at the Soviet diplomatic codes. After all, the codes became important only after the victory, but considering the scale of the Soviet defeat in war against Poland and Bolsheviks' problems in other parts of their nascent empire stimulated the peace talks in Riga, reducing the need for the codebreaking. Therefore Soviet military ciphers, and their solutions, represent much more interesting part of the archive.

Most Soviet military messages of the period were transmitted in numeric groups, each consisting of 5 digits. In their basic form virtually all ciphers were representing a monoalphabetic substitution based on Polybius square extended to 10x10 fields (Fig. 1).

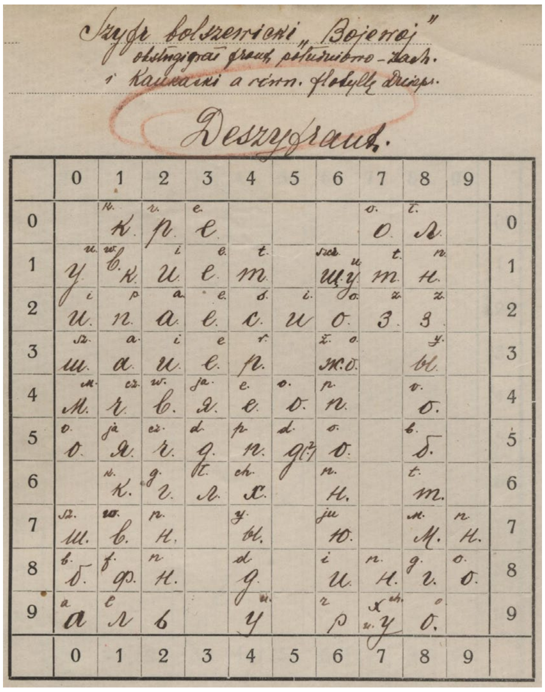


„Doniec"

	0	1	2	3	4	5	6	7	8	9	
0		д	а	з	п	ф	б	щ	у	м	0
1	ж	ц	о	б	и	в	х	ы	я	ш	1
2	т	л	г	е	н	в	к	с	ч	ю	2
3											3
4											4
5											5
6											6
7											7
8											8
9											9
	0	1	2	3	4	5	6	7	8	9	

Figure 1. Key "*Donets*"

In spite of their construction permitting many homophones, only some cipher keys were using them, "Boievoi" (Fig. 2) being one of their examples.



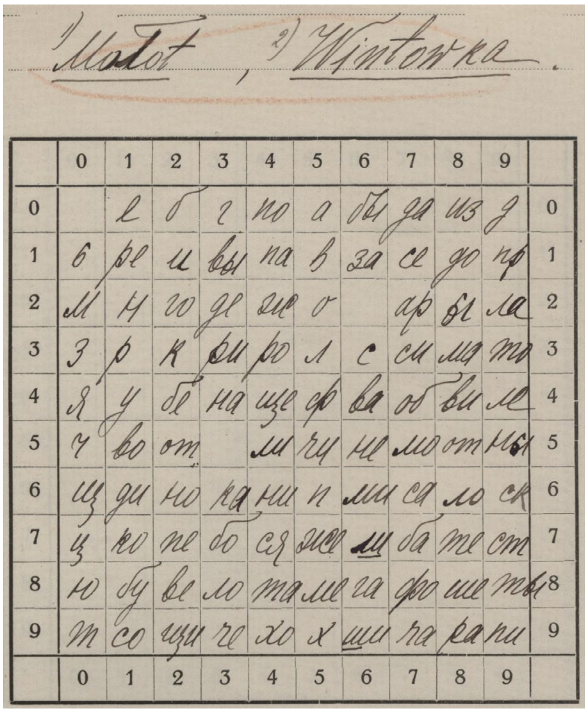
Szyfr bolszewicki „Bojewoj" obsługiwał grupę południowo-zach. i Kaukaski a równ. flotyllę dniepr.

Deszyfrant.

	0	1	2	3	4	5	6	7	8	9	
0		k. к.	r. р.	e. е				o. о.	ł. л.		0
1	u. у	w. в. к.	i и.	e. е	t. т.		szcz. u щу.	t. т.	n. н.		1
2	i и.	p п.	a. а.	e. е.	s. с.	i. и	o. о.	z. з.	z. з.		2
3	sz. ш.	a. а.	i и.	e е.	r. р.		ż. o. ж.о.		y. ы.		3
4	m. м.	cz. ч.	w. в.	ja. я.	e. е.	o. о.	p. п.		o. о.		4
5	o. о.	ja я.	cz. ч.	d. д.	p. п.	d. д(?)	o. о.		b. б.		5
6		k. к.	g. г.	ł. л.	ch. х.		n. н.		t. т.		6
7	sz. ш.	w. в.	n. н.		y. ы.		ju ю.		m. м.	n. н.	7
8	b. б.	f. ф.	n н.		d. д.		i и.	n. н.	g. г.	o. о.	8
9	a а	e л	ь		u. у		r р	ch. x u. у	o о.		9
	0	1	2	3	4	5	6	7	8	9	

Figure 2. Key *“Boievoi"*

Some cipher keys took syllabic nature, where pair of digits represented single letters and/or their pairs, key “Vintovka/Molot” being a good example of this group (Fig.3).



1) Molot, 2) Wintowka.

	0	1	2	3	4	5	6	7	8	9	
0		е	б	г	но	а	бы	да	из	д	0
1	в	ре	и	вы	на	в	за	се	до	пр	1
2	м	н	го	ге	же	о		ар	бг	ла	2
3	з	р	к	ви	ро	л	с	си	ма	то	3
4	я	у	бе	на	ще	ф	ва	об	ви	ле	4
5	ч	во	от		м	чу	не	мо	от	ны	5
6	щ	ди	но	ка	ни	п	ми	са	ло	ск	6
7	ц	ко	пе	бо	ся	же	ш	ба	те	ст	7
8	ю	бу	ве	ло	та	ме	га	фо	ше	ты	8
9	т	со	щи	ге	хо	х	ши	на	ра	пи	9
	0	1	2	3	4	5	6	7	8	9	

Figure 3. Key *"Vintovka/Molot"*

In the simplest scenario, when no superencipherment was used, result of the character substitution was simply combined into 5-digit groups and transmitted in this form. However, many keys were using additional layer of protection, inserting dummy digits into every group. In the simplest case a dummy digit was inserted into the constant (usually middle) position of the group. In more elaborate examples dummy digits were not only changing their positions, but were used to transform the pairs of digits representing letters within the same group. For example, in the cipher key “Сюртук” (frock coat) dummy digit was inserted in the first position of the first group, second position of the second, and so on until the fifth group. Moreover, the value of the dummy digit was subtracted (arithmetically) from both pairs representing the letters in a given group.

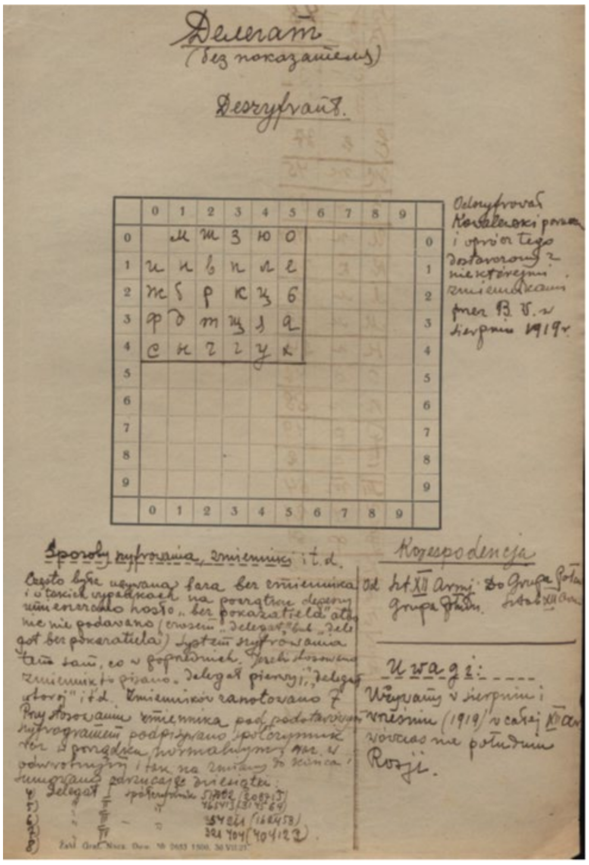
Most keys were utilizing superencipherment in a rather basic form. After the open text has been transformed into the numerical groups using the Polybius square, a superencipherment key was added or subtracted (without carry or borrow). Superencipherment key usually represented a number from three to six digits long used twice; in its normal and then reverse order. For instance the basic superencipherment factor for “Centralnyi” key was 234571, but was used as a sequence of 234571175432.

In some keys an additional transposition layer was added in the form of switching the positions of digits within a group.

Cipher elements described above were combined in the real keys in various scope, resulting in a variety of ciphers ranging from the basic monoalphabetic substitution to more elaborate examples, combining substitution with transposition and superencipherment.

6 Some Soviet crypto blunders

Most probably the largest Soviet blunder facilitating Kowalewski’s and his section’s job was having lost one of their keys, using most common features, to the enemy. According to Kowalewski’s notes at the margins of “Delegate” key (Fig.4) he has broken the key cryptanalytically, but its copy had been also seized by Polish intelligence service.



Делегат
(без показателя)

Deszyfrant.

	0	1	2	3	4	5	6	7	8	9	
0		м	ш	з	ю	о					0
1	и	н	ь	п	л	е					1
2	ж	в	р	к	ц	б					2
3	ф	д	т	щ	я	а					3
4	с	к	г	ч	у	х					4
5											5
6											6
7											7
8											8
9											9
	0	1	2	3	4	5	6	7	8	9	

Odszyfrował Kowalewski porucz. i oprócz tego dostarczony z niektóremi zmieniaczami przez B. V. w sierpniu 1919 r.

Sposoby szyfrowania, zmienniki i t. d.

Często była używana tara bez zmiennika i w takich wypadkach na początku depeszy umieszczało hasło „bez pokazatiela" albo nic nie podawano (zwłaszcza „Delegat" lub „Delegat bez pokazatiela") System szyfrowania ten sam, co w poprzednich. Przy stosowaniu zmiennika pisano „Delegat pierwszy", „Delegat wtoryj" i t. d. Zmienników zanotowano 7. Przy stosowaniu zmiennika pod podstawowym szyfrogramem podpisywano współczynnik raz z porządku normalnym, raz w odwrotnym i tak na zmianę do końca i sumowano porzucając dziesiątki:

4) Delegat I współczynnik 57802 (308715)
5) " II " 765413 (314564)
6) " III " 354021 (168458)
7) 8) " IV " 321704 (404122)

Korespodencja

Od Szt. XII Armji Do Grupa Gol. Grupa Gol. Sztab XII Arm.

Uwagi:
Używany w sierpniu i wrześniu (1919) w całej XII arm. wówczas na południu Rosji.

Figure 4. Key *"Delegate"*

This key represented a good example of the principles of cipher construction and Soviet cipher procedures.

Some other blunders, of a more conceptual nature, included:

- frequent use of a monoalphabetic substitution without any other form of complexity,
- generating new key tables using the circular shifts of the old ones,
- making only a half of the superencipherment key independent and reusing it in the reverse order,
- frequent reuse of the same superencipherment key by various keys,
- transmitting the reference to the superencipherment key in open text,
- even number of digits in both letter representation and superencipherment resulting in auto resynchronization of both streams,
- inserting punctuation in open text into the cipher messages and restarting the superencipherment from every punctuation mark, facilitating setting the message parts in depth,
- extensive use of the Soviet military jargon ("komdiv" - officer commanding the division, "kavkor" – cavalry corps, etc.) providing reliable cribs.

All of these blunders, and some more, were used by Polish codebreakers with good effect. Desperate state of the army and the country in the summer of 1920, and the sudden reversal of fortunes during the Battle at the Vistula river caused Polish victory to be traditionally described as the "Miracle at the Vistula". Now, that we are all able to access and study the digitized archives of Polish codebreaking service, we have to admit there was nothing supernatural in Polish victory. The newly established codebreaking service of Polish Army provided a solid foundation for the victory. Polish soldiers and their commanders managed to make good use of this advantage. And the role of the mathematicians in this cryptologic adventure provided foundations for the future triumph of Polish Cipher Bureau over Enigma.

References

Davies Norman. 2003 (1972). *White Eagle, Red Star: the Polish-Soviet War, 1919–20* (New ed.). New York.

Nowik Grzegorz. 2004. *Zanim złamano Enigmę*, Warszawa.

Nowik Grzegorz. 2010. *Zanim złamano „Enigmę" rozszyfrowano Rewolucję. Polski radiowywiad podczas wojny z bolszewicką Rosją 1918–1920,* Warszawa.

Reile Oskar. 1963. *Geheime Ostfront. Die deutsche Abwehr 1921–1945, München/Wels.*

Wyżeł-Ścieżyński Mieczysław. 1928. *Radiotelegrafia jako źródło informacji o nieprzyjacielu,* Przemyśl.

Zamoyski Adam. 2008. *Warsaw 1920: Lenin's Failed Conquest of Europe*, UK.

Experimental Analysis of the Dorabella Cipher with Statistical Language Models

Bradley Hauer†, Colin Choi†, Anirudh S Sundar†† [*]
Abram Hindle†, Scott Smallwood‡, Grzegorz Kondrak†
† AMII, Department of Computing Science, University of Alberta, Edmonton, Canada
‡ Department of Music University of Alberta, Edmonton, Canada
†† Dept. of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, USA
†, ‡ {bmhauer,cechoi,hindle1,scott.smallwood,gkondrak}@ualberta.ca
†† asundar34@gatech.edu

Abstract

The Dorabella cipher is a symbolic message written in 1897 by English composer Edward Elgar. We analyze the cipher using modern computational and statistical techniques. We consider several open questions: Is the underlying message natural language text or music? If it is language, what is the most likely language? Is Dorabella a simple substitution cipher? If so, why has nobody managed to produce a plausible decipherment? Are some unusual-looking patterns in the cipher likely to occur by chance? Can state-of-the-art algorithmic solvers decipher at least some words of the message? This work is intended as a contribution towards finding answers to these questions.

1 Introduction

The Dorabella cipher (henceforth, *Dorabella*) is a cipher sent by Edward Elgar to his acquaintance Dora Penny (Figure 1). Elgar was an English composer, best known for works such as *Pomp and Circumstance*, and the *Enigma Variations*. He also had an interest in cryptography, which was an inspiration for some of his compositions.

Prior decipherment attempts have adopted various assumptions. Arguably the most popular assumption is that it is a monoalphabetic substitution cipher (MASC) encoding an English text (Sams, 1970). Given that there was no known key exchange between Elgar and Penny, it is reasonable to assume that the cipher was not intended to be complicated; likewise, given that Elgar was an English composer, it is reasonable to assume that English is the language of the cipher. Another hypothesis is that it is enciphered music (Santa and Santa, 2010). However, no plausible systematic decipherment has been proposed to date, nor a convincing demonstration that it is a hoax.

In this paper, we investigate several hypotheses using modern computational techniques. Our methods are based on *statistical n-gram language models*, which are induced over characters or words from large collections of texts (*corpora*). We apply a state-of-the-art ciphertext language identification algorithm to identify the underlying language of the cipher. We also apply automated decipherment algorithms developed for monoalphabetic substitution ciphers in an attempt to obtain at least partial decipherment. To test the music hypothesis, we develop a transcription encoding scheme that is restricted to 24 distinct musical notes. Finally, we consider whether some statistical properties of the ciphertext support the hoax hypothesis.

Our experiments demonstrate that highly-accurate algorithmic solvers fail to produce any readable decipherment, providing evidence against the hypothesis that Dorabella is a simple MASC encrypting an English text. We do, however, find new evidence that English is one of the most likely languages behind Dorabella. Furthermore, experiments with musical transcriptions suggest that the cipher is unlikely to encode music. Finally, we find evidence of non-random patterns in the ciphertext, which we interpret as evidence against the hoax hypothesis.

This paper is structured as follows: We describe the properties of the Dorabella cipher in Section 2. In Section 3, we summarize prior publications on the topic. The methods and data are described in Sections 4 and 5, respectively. The experimental results are discussed in Section 6.

[*] Sundar's work while at the University of Alberta.

2 Dorabella Symbols

Figure 1 shows the cipher in its entirety. It contains 87 characters, each consisting of one, two, or three semicircles, in one of eight distinct orientations (in increments of 45 degrees), yielding a total of 24 possible symbol types. The orientation of some of the symbols is ambiguous. In our transcription, 20 distinct symbols appear in the cipher, with four hypothetical symbols being unused. The symbols follow a highly non-uniform distribution, with one symbol appearing 11 times, while some appear only once.

This distribution of the number of semicircles is relatively uniform, while the distribution of orientations is not. In our transcription, 29 tokens have one semicircle, 33 have two semicircles, and 25 have three semicircles. On the other hand, one orientation (with the bottom of each semicircle directed at 315 degrees) occurs 23 times in our transcription, while another (with the bottom of each semicircle directed at 0 degrees, e.g. right) occurs only 4 times. In this paper, we make no assumptions or deductions about the meaning of the number and orientation of semicircles; rather, we arbitrarily map each symbol type to an arbitrary lowercase English letter, and treat the resulting transcription as a straightforward substitution cipher.

3 Related Work

In an early decipherment attempt, Sams (1970) analyzes Dorabella using frequency analysis, contact charts, and brute force methods. This work assumes that that the message is partly phonemicized, but not strictly monoalphabetic. The result of this analysis is a decipherment which is not systematic, verifiable, or falsifiable.

Santa and Santa (2010) analyze Dorabella in the broader context of Elgar's work, particularly his *Enigma Variations*. They speculate that Elgar may have used the mathematical constant π, approximated as 3.142, to encipher scale degrees. However, they note that a plausible solution to Dorabella, whether in the form of natural language text or musical notation, is yet to be found.

Schmeh (2018) explores several established techniques for identifying vowels and consonants in monoalphabetic substitution ciphers. The result is a transcription of Dorabella, with some symbols identified as vowels or as consonants. These results are supported by independent analysis on a sample cipher of the same length.

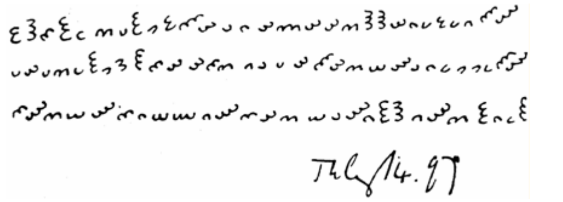


Figure 1: The Dorabella cipher.

The task of computational decipherment of monoalphabetic substitution ciphers is well-studied. Most recent work involves character and/or word language models (Norvig, 2009; Nuhn et al., 2013; Hauer et al., 2014) as well as other techniques, such as electronic dictionaries (Olson, 2007), integer programming (Ravi and Knight, 2008), and Bayesian inference (Ravi and Knight, 2011).

4 Methods

In this section, we describe the cryptographic tools, both previously published and original to this work, which we employ in our analysis of Dorabella.

4.1 Language Models

Our methods are based on statistical n-gram language models, which are induced over characters or words. Language models guide decipherment algorithms by computing the probabilities of various possible decipherments, allowing algorithms to favour decisions which result in more probable solutions. An n-gram language model can be used to compute the probability of a token given the $n-1$ previous tokens. A 3-gram, or *trigram*, character language model, for example, is able to predict that, given the previous characters 'aq', the letter 'u' is more likely to follow than 'e', despite 'e' generally being more common than 'u'. To create language model for our experiments, we use KenLM.[1]

Language models can be applied over a sequence of characters to measure their *perplexity*, which quantifies the extent to which a language model is "surprised" by the text in question. A high perplexity indicates that the sequence of tokens has a correspondingly low probability under the model.

[1] https://github.com/kpu/kenlm

4.2 Computational Decipherment

We experiment with three previously published methods for deciphering monoalphabetic substitution ciphers, which are based on statistical *n*-gram language models.

HILLCLIMB (Norvig, 2009), is a solver that performs a hill-climbing search with multiple random restarts to maximize the probability of the decipherment under a character language model. The best decipherment is selected according to a word language model. We use the implementation provided by the author.[2] Since word language models require word boundaries, we also experiment with HILLCLIMBC, a variant that instead uses a character language model to identify the best decipherment.

TREESEARCH (Hauer et al., 2014) uses a tree search algorithm to find the highest-scoring key. A key scoring function combines word and character n-gram language models of various orders. An initial decipherment based on unigram character frequencies serves as the root of the tree. A key mutation function leverages character repetition patterns to generate a set of children for each key. The solver is reported to have decipherment accuracy on ciphers without spaces (i.e., without word boundaries) of over 92% for length 64, and over 99% for length 128, which represents the state-of-the-art for monoalphabetic substitution decipherment.

UNRAVEL (Nuhn et al., 2015) searches for a mapping of letters that maximize the probability of the decipherment under an *n*-gram character language model. Partial key mappings are structured into a search tree, and a beam search is used to traverse the tree and find the most promising candidates. Unlike TREESEARCH, UNRAVEL does not constrain every node of the search tree to contain a complete decipherment; not all nodes decipher all symbol types. Rather, initially incomplete keys are iteratively expanded, with heuristic search used to guide the expansion until a complete solution is found. We use the version of UNRAVEL that is applicable to deterministic rather than probabilistic ciphers. The experiments presented by the authors focus on word-level decipherment (e.g. identification of lexical translations), without any claims regarding the efficacy of the solver on character-level monoalphabetic substitution ciphers.

We also developed a novel greedy search algorithm with random restarts, which we refer to as GREEDY. Starting with a random key, possible successors are generated by sequentially swapping letters in the current key. Each successor key is assigned a probability using a character trigram language model. The successor which produces the most probable decipherment becomes the new key, provided that its decipherment is more probable than the current key. The key that produces the most probable decipherment over multiple random restarts is returned as the solution.

4.3 Ciphertext Language Identification

Identification of the underlying language of a cipher is crucial for a successful decipherment. For this task, we apply two methods presented by Hauer and Kondrak (2016): UNIGRAM and TRIAL. Both methods are applicable to monoalphabetic substitution ciphers without word boundaries, and require a set of sample texts, each representing one of the candidate languages. Each method iterates over the set of sample texts, computing a score function on each sample. The language of the sample text which maximizes this score function is returned as the identified language of the ciphertext.

The first method, UNIGRAM, leverages the observation that a monoalphabetic substitution does not alter the relative frequencies of characters: the frequency of the i-th most frequent character before encipherment is equal to the frequency of the i-th most frequent character after encipherment. Given the ciphertext and a sample text, UNIGRAM computes the *sorted symbol distribution* of each. This is a probability distribution over characters $1,\ldots,k$ where k is the length of the longer of the two symbol alphabets, and $P(i)$ is the probability of a randomly selected character being the i-th most frequent character in the text. For each language, we compute its score as the distance metric of Bhattacharyya (1943) between the unigram probability distributions of the sample text and the ciphertext.

The second method, TRIAL, is based on the intuition that attempting to decipher a ciphertext into the incorrect language (e.g., deciphering enciphered English into French) will almost certainly not yield a probable text in that language. The method learns a bigram character language model for each language using the corresponding sample

[2] http://norvig.com/ngrams

text. It then applies a hill-climbing decipherment algorithm which seeks to maximize the probability of the decipherment. This algorithm terminates quickly in practice, allowing hundreds of candidate decipherments to be tried. The probability of the best decipherment is returned as the score. It is important to note that Hauer and Kondrak (2016) developed and tested the TRIAL method on ciphers with spaces included, as it was originally designed for the Voynich manuscript.

4.4 Music Decipherment

The algorithms described in the previous section were designed to be applied to natural language texts. Since we wish to test the hypotheses that Dorabella is enciphered music, we seek to apply these algorithms to music as well. This presents multiple challenges, which we discuss here.

Most western music is presented as pitches with duration over time with dynamics, phrasing, and articulations. In terms of pitch, multiple pitches can sound at the same time, resulting in chords, homophony (a primary melody with accompanying chordal notes), or polyphony (simultaneous melodic lines that have independent characteristics, but also outline harmonic motion). A piano is an example of a polyphonic instrument, as with multiple fingers one can play many piano keys at the same time, and each piano string will sound a distinct separate pitch. Thus much music is written and composed in a polyphonic manner. There is no analogue to this phenomenon in natural language text. We therefore need to first serialize the notes and choose an order. To this end, we work with single lines of music rather than polyphonic passages. For example, we would consider only the melodic line of a four-part piece.

Further, music differs from written language in several key ways. Notes do not refer to specific real-world concepts, as words do, and have different intents or meaning. Furthermore, music can be transformed (such as by changing octave or transposing the key of the music) in ways whereby musicians will still understand the music or its origin. Finally, there is no clear equivalent of a sentence or punctuation in music; if such equivalents exist, it is not clear if they can be ignored for the purposes of encipherment and decipherment, as is the case with natural language. [3]

For music to be enciphered it must be first represented as symbols, such as western music notation. Then, we must serialize them, such that one note comes after another, as described above. An example encoding could be the note name, which ignores octave and duration, expressed as space separated notes: `E D C D E E E` (*Mary had a little lamb*). Alternatively we could add duration: `Eq Dq Cq Dq Eq Eq Eq Eq` where `q` would indicate a quarter note. We might also include octave: `E4q D4q C4q D4q E4q E4q E4q E4q` where `C4` is middle C, and `C5` is an octave above that, and so on. Such an encoding would allow us to treat each note (a tuple of pitch and duration) as a symbol. These symbols could then be enciphered or deciphered, just as can be done with the sequence of symbols in a natural language text.

For our experiments we start with music encoded as MIDI files (a digital music communication protocol), which we then pre-process into simpler serial formats. MIDI for our purposes presents notes as pitches that are turned on and off at specified times. In terms of duration, notes are normalized into sixteenth notes, eighth notes, quarter notes, half notes, and whole notes. In terms of pitch, we can decide to look for any 12 notes of the octave, or confine ourselves to a diatonic scale (7 of the 12 notes).

In order to convert these MIDI files into a sequence of symbols, as described above, the files are transposed to the key of C major, and only a single octave of notes is used. Rests, accents, and other symbols that do not signify notes are removed from the sequences; chords are decomposed to their roots. The representation is composed of notes with their respective duration. Three different durations are used for the notes. A duration of 0.5 represents anything shorter than a quarter note, a duration of 1 represents a quarter note, and a duration of 2 represents anything longer than a quarter note.

Our encoding uses 24 unique symbols, the same number of unique symbols that can be made using the Dorabella cipher system. This encoding only uses the eight most frequent notes A, B, C, D, E, F, F♯, and G along with the three durations described above. Notes not among the 8 most frequent notes are moved half a step up or down. For example, a D♯ would be changed to a D and A♭ would be changed to A.

[3] There does exist a musical term of "sentence," which refers to a complete statement that is bigger than a motive or phrase, but shorter than a theme.

5 Data

This section is devoted to the language and music corpora used in our experiments (Hauer et al., 2021). Our natural language corpora include literary prose, newspaper texts, movie subtitles, and multilingual documents. To generate ciphertexts with known solutions for testing purposes, we extract samples from the 19th century fiction works in Project Gutenberg[4], including *The Adventures of Sherlock Holmes*, and *The Letters of Jane Austen*. We chose *Dangerous Connections*, an English translation of an epistolary novel, for deriving character-level language models; and a much larger *New York Times Corpus*[5] for deriving word-level language models. For our language identification experiments, we use a dataset constructed from 380 translations of the *Universal Declaration of Human Rights* (UDHR) (Emerson et al., 2014), and the multilingual *OpenSubtitles* corpus of movie subtitles (Lison and Tiedemann, 2016).

To create test samples used in our experiments, we first normalize the natural language corpora, by removing punctuation, digits, and other non-alphabetic characters, and lower-casing all letters. The test samples we use are 87 letters long, the same length as Dorabella. They are created by first randomly selecting a word in the corpus, and then appending subsequent words until the length of exactly 87 letters is reached. Samples that end with partial words are discarded, and no duplicate samples are admitted. This process ensures that each generated test cipher begins and ends at a word boundary, and contains exactly 87 characters, with no spaces.

Our music corpora consist of monophonic tracts extracted from collections of Elgar and Bach MIDI files.[6] For each composer, we split the collection of MIDI files into testing and training sets.[7] For Bach, the training corpus is composed of 295 MIDI files concatenated together (3.7M notes) with 3 MIDI files (174K notes) held out for testing. For Elgar, the training corpus is composed of 29 concatenated MIDI files (1.2M notes), with 3 MIDI files (24K notes) held out for testing. Samples 87 notes in length are extracted from the test data and enciphered to create sets of music ciphers with known solutions for our experiments.

6 Experiments

In this section we present our applications of the methods described in Section 4, using the data described in Section 5, with the goal of testing several hypothesis regarding the Dorabella cipher. Throughout our experiments, we make the assumption that Dorabella is a monoalphabetic substitution cipher (MASC), which is based on the number and relative frequencies of the characters. For the evaluation of MASC solvers, we compute both *key accuracy*, the proportion of cipher character types which are correctly mapped to the corresponding plain-text character type, and *decipherment accuracy*, the proportion of cipher character tokens which are correctly deciphered.

As a precursor to these experiments, we applied the BION classical cipher type classification programs[8] as used by Nuhn and Knight (2014), to our transcription of Dorabella. Both programs classify the text as a "patristocrat" cipher, which is equivalent to our definition of a monoalphabetic substitution cipher without word divisions. This supports our assumption that Dorabella is a MASC.

6.1 Ciphertext Language Identification

In this section, we apply the ciphertext language identification methods described in Section 4.3 to analyze Dorabella. This includes empirically assessing the reliability of these methods on short ciphers without spaces, as well as examining the output of the state-of-the-art method when applied to Dorabella.

Given an output which induces a ranking of possible classes, the *reciprocal rank* for a given instance is the multiplicative inverse of the position of the correct class, with the highest-ranked class being rank 1. For example, if the correct class is assigned rank 4, the reciprocal rank for that instance is $1/4 = 0.25$. The *mean reciprocal rank* (MRR) is the average of the reciprocal ranks over all instances. A high MRR indicates that the correct class is consistently placed near the top. Closely related to MRR is *average rank* (AvgR), which is simply the mean position of the correct class over all instances (i.e. MRR, without the reciprocal operation). Top-1 accuracy, or simply accuracy (Acc), is the proportion of instances

[4] http://www.gutenberg.org/ebooks/

[5] https://catalog.ldc.upenn.edu/LDC2003T05

[6] https://www.classicalmidi.co.uk/elgar.htm, http://bestclassicaltunes.com, http://dardel.info/musique/Bach.html

[7] https://archive.org/download/midi-sources

[8] http://bionsgadgets.appspot.com

Method	Length	Spaces	MRR	AvgR
UNIGRAM	2000	No	0.18	15.7
TRIAL	2000	No	0.94	1.2
TRIAL	2000	Yes	0.96	1.1
UNIGRAM	87	No	0.02	120.0
TRIAL	87	No	0.13	52.6
TRIAL	87	Yes	0.25	33.4

Table 1: Results of the ciphertext language identification methods.

for which the correct class is placed in the first position. For MRR and Acc, higher is better; for AvgR, lower is better. If and only if a method always places the correct class in the first position, its MRR, Acc, and AvgR will all be 1, the maximum/minimum values.

6.1.1 Validation on Synthetic Ciphers

In this experiment, we aim to establish the effectiveness of the current state-of-the-art method of Hauer and Kondrak (2016) on synthetic cipher samples from multiple languages. They report that the TRIAL method achieves over 97% top-1 accuracy; however, their results are on ciphers longer than a thousand characters, which include word boundaries. In contrast, Dorabella is only 87 characters long, and contains no spaces.

We begin by assessing the impact of the cipher length and the presence of word boundaries on ciphertext language identification accuracy. We test 4 cipher variants: long (2000 characters) vs. short (87 characters), with and without spaces. As our data, we use the UDHR dataset (Section 5) for training language models, and the OpenSubtitles corpus for generating test ciphers.

Table 1 shows the results of the experiment. We report the mean reciprocal rank (MRR) and average rank (AvgR) of the correct ciphertext language evaluated over a set of 500 ciphers in 5 distinct languages: English, French, Polish, German, and Italian. The results indicate that even for short ciphers without spaces, the TRIAL method is able to rank the language of the ciphertext much more highly than the UNIGRAM method. Even for Dorabella-like 87-character ciphers without spaces, the TRIAL method consistently assigns a relatively high rank to the correct language. For comparison, a random baseline yields MRR of 0.017, and an average rank of 190.5. From this we conclude that the TRIAL method provides useful information about the language of short ciphers without spaces.

	En	Fr	Pl	De	It	**Avg**
MRR	.68	.68	.72	.69	.87	**.73**
Acc	.49	.48	.55	.50	.78	**.56**

Table 2: MRR and top-1 language identification accuracy on 87-character ciphers The MRR and Acc for each language are the averages over all ciphers for that language.

6.1.2 Impact of Language Sample Size

In our second set of language identification experiments, we investigate whether there is a substantial benefit to increasing the size of the texts used by TRIAL to create language models. Due to the greater difficulty in acquiring larger texts for training language models, we only test on English, French, Polish, German, and Italian, so there are only five possible classifications, rather than 380. For each language we obtain 100M characters from the OpenSubtitles corpus for inducing the language model, and another 20M to create test ciphers. We create 1000 ciphers without spaces for each of the five languages.

The results in Table 2 indicate that TRIAL is able to correctly select, from English, French, Polish, German, and Italian, the language of a ciphertext from one of those languages more than half the time. The MRR values for each language are all well above 0.5, which indicates that the correct language is usually among the top two candidates. We conclude that, given sufficient training data for inducing language models, the TRIAL method can be used to analyze short ciphers without spaces.

6.1.3 Is Dorabella English?

We now explore the hypothesis that the Dorabella cipher represents enciphered English. This hypothesis is based on the observation that the remainder of Elgar's letter, in which the Dorabella cipher is embedded, is written in English. To maximize the number of candidate languages we consider, we again use the UDHR data as a source of language samples. We then apply TRIAL, the more accurate of the two language identification methods, to Dorabella, inducing a ranking of the 380 samples.

Table 3 shows the five highest-scoring languages. The numerical values are the log-probabilities of the best decipherment for each

Rank	Language	LM Score
1	Latin	-217.34
2	Aceh	-221.05
3	English	-221.24
4	Toksave	-222.19
5	Scots	-222.62

Table 3: The highest-scoring Dorabella candidate languages with the TRIAL method.

language, estimated using the corresponding language model. It is notable that this method places English as the third-best choice for the language of the cipher, and the closely related Scots language as the fifth choice. Latin, which is a major source of the English lexicon and its orthography, is ranked first. Given the accuracy of the TRIAL decipherment method, and given the context in which the Dorabella cipher was produced, we conclude that English is the most likely natural language candidate for Dorabella.

6.2 Is Dorabella Music?

Using the music representation described in Section 4.4, and inducing character "language" models over the music corpora described in Section 5, we investigate the hypothesis that Dorabella enciphers music, rather than natural language. To determine the accuracy of our solvers on music, we test two different decipherment programs. The HILLCLIMBC solver and GREEDY solver are chosen for this test because our text decipherment experiments show that these two solvers perform well on short ciphers without spaces, and without a large training corpus.

We created Elgar and Bach language models from the corpora of their music, described in Section 5. The test samples were randomly enciphered with a substitution cipher. Since the accuracy of both solvers on short samples was very low, we instead used very long samples of around 20,000 notes each.

Table 4 shows the results on long ciphers. The best key and decipherment accuracies are only 26.6% and 32.7% respectively, both obtained using our GREEDY method. This indicates that approximately one-third of the notes in each cipher are deciphered correctly, on average. We conclude that deciphering music, in our minimalist representation, is much more difficult than deciphering natural language.

Music	**Solver**	**Key Acc**	**Dec Acc**
Elgar	GREEDY	4.8%	6.4%
Elgar	HILLCLIMBC	7.0%	12.0%
Bach	GREEDY	26.6%	32.7%
Bach	HILLCLIMBC	26.5%	32.0%

Table 4: Key and decipherment accuracy on long music ciphers.

One of the authors of this paper analyzed the notes in the highest-scoring decipherment obtained with the Elgar language model. The notes appear and sound random, containing no clues that would point to an expected tonal center and harmonic progression. Further, no recognizable motives, phrasing, or repetition can be identified. It has nothing to do with Elgar's music, which was more complex and chromatic. We hypothesize that Elgar may have instead enciphered a simple folk-like melody, rather than something comparable to his more mature work. We intend to investigate this direction in future work.

6.2.1 Impact of Perplexity

In this section, we investigate a hypothesis that music has a less predictable structure than natural language, which would make it more difficult to decipher, explaining the results in the previous section. We calculate the relative perplexity of samples of texts vs. samples of music notation encoded using a simple scheme. Both types of encodings have a similar number of distinct symbols: 26 letters vs. the 24 symbols in our encoding of musical notes.

We create language models for Bach music, Elgar music, and English text as in the preceding sections. We then create 100 samples of 87 characters for each of Bach, Elgar, and English. The samples are not included in the training corpora. The English LM is derived from *Dangerous Connections*, while the samples are from *Letters of Jane Austin*. The average perplexity is then calculated for all three sets of samples against the three language models.

We find that English is much more predictable than music, even under our highly simplified encoding scheme. Averaged across the 100 samples, the music of Elgar and Bach have perplexities of 24.40 and 24.52 respectively, while English has a perplexity of only 16.18. We propose this as an explanation of our finding that decipherment algorithms are much less effective on enciphered music compared to enciphered English.

6.2.2 Classifying Text vs. Music

Since rank-based attempts showed some promise in determining the ciphertext language (Sections 6.1.1 and 6.1.3), we decided to create a classifier to determine whether a cipher encodes English text or music. In this section, we describe the classifier, test it on synthetic ciphers, and finally apply it to Dorabella.

We use TRIAL as our classifier, with English and music as candidate languages. We trained the necessary bigram language models on 1M characters of *Dangerous Connections* for English, and *either* 1M symbols of Bach, or 1M symbols of Elgar. This yields two distinct experiments: (1) distinguishing English and Bach music, and (2) distinguishing English and Elgar music.

When tested on the 300 test ciphers from *Letters of Jane Austen*, and 300 samples each of Bach music and Elgar music, we found that TRIAL was able to distinguish between English and Elgar ciphers with 82% accuracy, and between English and Bach with 88% accuracy. These results demonstrate that TRIAL can reliably distinguish between enciphered English and enciphered music.

That established, we applied our classifier to our transcription of Dorabella. We found that TRIAL classifies the cipher as English, compared to both Bach and Elgar music. In the first case, language model log-probabilities of -228.8 and -244.5 are assigned to English and Bach, respectively. The variances on these mean log-probabilities (averaged over ten independently randomized runs) are 11.1 and 13.5, respectively. In the second case, the corresponding average log-probabilities are -226.8 and -248.6, with the variances of 5.0 and 2.0, respectively. We interpret these results as evidence that Dorabella is much more likely to represent English than music.

6.3 Decipherment of English Texts

In this section, we perform validation experiments on several substitution cipher solvers. We compare their accuracy on English MASCs, and attempt to decipher Dorabella with the best-performing solver. Note that *we do not claim to have produced a correct decipherment of or solution to the Dorabella cipher.*

We test five decipherment methods which are described in Section 4.2: TREESEARCH, HILLCLIMB, HILLCLIMBC, GREEDY, and UNRAVEL. To establish the reliability of each of these methods, we measure their accuracy on 87-character ciphertexts without spaces. We use the same set of 300 English ciphers and English training corpus as in Section 6.2.2.

Solver	Key Acc	Dec Acc
TREESEARCH	43.1%	44.9%
UNRAVEL	42.8%	47.8%
GREEDY	69.0%	79.1%
HILLCLIMB	75.8%	84.5%
HILLCLIMBC	78.3%	88.1%

Table 5: Accuracy of substitution cipher solvers on short English ciphers without spaces.

Table 5 shows the average key and decipherment accuracy of the five solvers on the set of 300 test samples. The relatively low accuracy of TREESEARCH is likely due to the small size of the training corpus.[9] Similarly, UNRAVEL did not perform very well on short ciphers without spaces, regardless of the size of the corpus. The remaining three solvers were much more effective. HILLCLIMBC, the variant of HILLCLIMB which is based entirely on a character language model, performed best, reaching nearly 90% average decipherment accuracy.

However, applying HILLCLIMBC to Dorabella does not produce a readable decipherment. The highest-scoring decipherment is as follows:

```
ychswamsopledieveeacceirprult
memarsofsheehaudmeleantdiroorlt
htanthingutheandtuscutasirs
```

Since the other solvers likewise failed to produce any partial decipherment, we conclude that the Dorabella cipher is unlikely to represent English text enciphered with a simple MASC.

6.4 Ciphertext Characteristics

The experiments in this section are aimed at the statistical analysis of two observations made in a video by Keith Massey.[10] The first is that the number of two-symbol sequences in Dorabella which are reflections of one another is greater than chance would allow. The second is that there are

[9] In a separate experiment, we were able to replicate the high decipherment accuracy reported by Hauer et al. (2014), given a larger (but out-of-domain) text corpus.

[10] Keith Massey, *The Dorabella Cipher: Proven to be a Friendly Joke*, 2017-05-29

long series of symbols in Dorabella with no repeat of a symbol with the same number of semicircles. Based on those two observations, it is claimed that the Dorabella cipher is a nonsensical message constructed as a playful joke.

6.4.1 Mirrored Symbols

The Dorabella cipher contains 13 pairs of mirrored symbols. A mirrored pair consists of 2 consecutive symbols that have the same number of semicircles but are facing in opposite directions. (For example, the final pairs of symbols in line 1 and 2 in Figure 1.) How likely is it for a ciphertext of 87 symbols to contain 13 mirrored pairs?

Our procedure is as follows. We randomly extract 100,000 samples of length 87 from *The Adventures of Sherlock Holmes* using the procedure described in Section 5. Given the large number of samples relative to the length of the corpus, there is some overlap between distinct ciphers, however, each starts at a distinct character in the corpus. For each sample, we generate a random key that maps each letter in the sample to a Dorabella symbol. Since there are 26 letters in the alphabet but only 24 Dorabella symbols, up to 2 pairs of letters may share a single symbol. We encode each of the 100,000 samples with Dorabella symbols, and count the number of mirrored symbols that occur in each sample.

The results show that, an English text of length 87 encoded with the Dorabella symbols contains an average 3.64 mirrored pairs. Out of the 100,000 samples, only 123 contained 13 or more mirrored pairs, which implies that a text with 13 mirrored pairs, like Dorabella, has only about a 0.1% chance of occurring by accident.

While these results support Keith Massey's observation, we disagree with the implication that Dorabella is a hoax. Instead, we posit that the mirrored pairs in Dorabella may have some special interpretation, which would support our earlier conclusion that Dorabella is not a simple MASC. For example, the mirrored symbols could have been used by Elgar to represent double letters, such as "ee", in a less conspicuous way.

6.4.2 Longest Non-Repeating Sequence

Each symbol in Dorabella has 1, 2, or 3 semicircles. In each of the three lines of Dorabella, there are sequences of symbols without two consecutive symbols containing the same number of semicircles. The longest such sequence is of length 12. The claim made in the video is that the occurrence of such long sequences with no two adjacent symbols having the same number of semicircles is highly improbable, indicating the Dorabella is a hoax.

We test this claim by applying a similar procedure as in the previous experiment: We encipher 100,000 samples of English with Dorabella symbols using randomly generated keys and count the longest sequence of symbols without repeated semicircles in each sample.

The results show that the average length of the longest sequence of consecutive symbols with different number of semicircles is approximately 10.23. Specifically, 27,472 out of the 100,000 samples contained sequences of 12 or more symbols where there were no repeated semicircles. We conclude that the probability a single occurrence of a sequence of length 12 in Dorabella is about 27.4%. Therefore, while the sequences observed in Dorabella are surprising, they are not sufficiently improbable to dismiss the cipher as a joke. In sum, our investigation of the claim made in this video provide no evidence for the hoax hypothesis.

7 Conclusion

While the short length and lack of word boundaries in the Dorabella cipher present a formidable cryptographic challenge, we have been able to provide evidence for and against various hypotheses via experimental analysis. The failure of several substitution solvers to produce any partially readable decipherment suggests that the cipher is not a simple monoalphabetic substitution cipher that encodes an English text. Our application of a state-of-the-art method for ciphertext language identification provides new evidence for English as the language of the cipher. Furthermore, application of a classifier based on character language models suggests that the underlying message of Dorabella is more likely to be natural language than music. Finally, the occurrence of several pairs of mirrored symbols is unlikely to be due to chance, suggesting that Dorabella is not a hoax.

Acknowledgements

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Alberta Machine Intelligence Institute (Amii).

References

Anil Bhattacharyya. 1943. On a measure of divergence between two statistical populations defined by their probability distributions. *Bull. Calcutta Math. Soc.*, 35:99–109.

Guy Emerson, Liling Tan, Susanne Fertmann, Alexis Palmer, and Michaela Regneri. 2014. Seedling: Building and using a seed corpus for the human language project. In *Workshop on the Use of Computational Methods in the Study of Endangered Languages*, pages 77–85.

Bradley Hauer and Grzegorz Kondrak. 2016. Decoding Anagrammed Texts Written in an Unknown Language and Script. *Transactions of the Association for Computational Linguistics*, 4:75–86.

Bradley Hauer, Ryan Hayward, and Grzegorz Kondrak. 2014. Solving substitution ciphers with combined language models. In *Proceedings of COLING, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2314–2325, Dublin, Ireland.

Bradley Hauer, Colin Choi, Anirudh S Sundar, Abram Hindle, Scott Smallwood, and Grzegorz Kondrak. 2021. Zenodo: Code and Data for "Experimental Analysis of the Dorabella Cipher with Statistical Language Models", HistoCrypt 2021, May. https://doi.org/10.5281/zenodo.4819086.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929.

Peter Norvig. 2009. Natural language corpus data. In Toby Segaran and Jeff Hammerbacher, editors, *Beautiful Data: The Stories Behind Elegant Data Solution*, pages 219–242. O'Reilly Media.

Malte Nuhn and Kevin Knight. 2014. Cipher Type Detection. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1769–1773, Doha, Qatar.

Malte Nuhn, Julian Schamper, and Hermann Ney. 2013. Beam search for solving substitution ciphers. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1568–1576.

Malte Nuhn, Julian Schamper, and Hermann Ney. 2015. UNRAVEL—A Decipherment Toolkit. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL-IJCNLP)*, pages 549–553, Beijing, China.

Edwin Olson. 2007. Robust dictionary attack of short simple substitution ciphers. *Cryptologia*, 31(4):332–342.

Sujith Ravi and Kevin Knight. 2008. Attacking decipherment problems optimally with low-order n-gram models. In *Empirical Methods in Natural Language Processing*, pages 812–819. Association for Computational Linguistics.

Sujith Ravi and Kevin Knight. 2011. Bayesian Inference for Zodiac and Other Homophonic Ciphers. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 239–247.

Eric Sams. 1970. Elgar's Cipher Letter to Dorabella. *The Musical Times*, 111(1524):151–154.

Charles Richard Santa and Matthew Santa. 2010. Solving Elgar's Enigma. *Current Musicology*.

Klaus Schmeh. 2018. Examining the Dorabella Cipher with three lesser-known cryptanalysis methods. In *Proceedings of the 1st International Conference on Historical Cryptology (HistoCrypt 2018)*, pages 145–152.

Two Encrypted Diplomatic Letters Sent by Jan Chodkiewicz to Emperor Maximilian II in 1574-1575

Nils Kopal
University of Siegen
Germany
nils.kopal@uni-siegen.de

Michelle Waldispühl
University of Gothenburg
Sweden
michelle.waldispuhl@sprak.gu.se

Abstract

This paper presents the work on two encrypted diplomatic letters sent by the Lithuanian nobleman Jan Chodkiewicz to emperor Maximilian II in 1574 and 1575. It describes the decipherment process as well as the content and the context of the letters. Furthermore, it provides linguistic aspects of the used plaintext language. It continues our previous work on Habsburg ciphers where we analyzed and contextualized three diplomatic letters sent by Maximilian II. All presented and analyzed letters relate to the Polish-Lithuanian election in 1575, where Maximilian II, his son Ernst, and his brother Ferdinand were amongst the candidates. The deciphered German plaintexts of all five letters can be accessed via the DECODE database, a storage for historical encrypted manuscripts, which is maintained by members of the DECRYPT project.

1 Introduction

This paper presents a new direct outcome of the DECRYPT project which collects, transcribes, and analyzes historical original encrypted manuscripts in an international and interdisciplinary team of researchers. The final project goal is to research and develop methods and tools which can be used by any researcher, e.g. historians, for free to decipher encrypted material they found in archives all over the world.

In early 2020 three Austrian diplomatic encrypted letters caught our attention for being cryptanalyzed. Photos of the letters were made previously by our project colleagues Anna Lehofer and Benedek Láng in the ”Haus-, Hof- und Staatsarchiv – Österreichisches Staatsarchiv” (HHStA), a unit of the Austrian State Archive, in Vienna. They uploaded the photos into the DECODE database, a storage infrastructure for encrypted historical manuscripts. In the course of the year, we managed to decipher all three letters. The letters were written in German and sent in the 16th century. The sender was Maximilian II, a Habsburg emperor. The letters were sent in July and December 1575. Receivers were delegates of Maximilian II in Poland and Lithuania. The content of the letters related the Polish-Lithuanian royal election in 1575, where Maximilian II was among the candidates. He gave direct orders in favor of his position. Despite all his effort, Maximilian II did not succeed in obtaining the Polish-Lithuanian crown and died soon after in October 1576.

After finishing the cryptanalytical work on the three diplomatic letters by Maximilian II, we turned our attention at the end of 2020 to two other encrypted manuscripts, which equally had been collected in the HHStA by Benedek Láng and Anna Lehofer. The visual writing style is similar to the three encrypted letters previously deciphered. Upon request, student assistants who work for DECRYPT at the University of Uppsala provided transcriptions of the two ”new” letters. After that, we started analyzing the two letters. The experiences we obtained analyzing the first three letters helped us cryptanalyzing the additional letters that turned out to be sent likewise in the time of Maximilian II. While writing this paper in early 2021, we are still in the process of cryptanalyzing and contextualizing the two additional letters. Nevertheless, this paper here gives an overview of the findings which could be of interest to the HistoCrypt audience.

The rest of the paper is structured as follows: Section 2 gives a brief summary of the previously deciphered letters sent by Maximilian II in 1575. After that, Section 3 contains preliminary results of the cryptanalysis of the two recently found letters. Then, Section 4 gives a brief overview of the historical context as well as of the content of the letters. Section 5 presents some aspects of the plaintext language. Finally, Section 6 concludes this paper.

2 Deciphering three diplomatic letters sent by Maximilian II in 1575

This section briefly summarizes the previous findings on the three Habsburg letters sent by Maximilian II in 1575 (letters A-C). While letters A and B both consist of eight full pages of ciphertext symbols and a ninth page with only a few lines of ciphertext, letter C only consists of two full pages and a third page with a few lines of ciphertext symbols.

We published a detailed article about the decipherment, content, and linguistic analysis of the letters A-C in (Kopal and Waldispühl, 2021). In the same article, we also present the historical background of the happenings referred to in the letters in more detail and give an overview on previous work on Habsburg cryptography (e.g. Láng, 2020). Table 1 shows an overview of the meta data of all five letters.

In early 2020, when the work started on letter A, we were not aware of the fact that a set of three letters share the same encryption key. Not knowing the original key used for encryption, we firstly had to perform a ciphertext-only attack on the cipher to reconstruct the used key as well as to decipher the plaintext.

While working on the decipherment of letter A, we found a photo of the original key from 1572, named ”Cyffra nova ad Poloniam” in the DECODE database. Shortly after finding the original key, we found two other letters (B and C) sharing the same key. Finally, we found a second copy of the original key in the DECODE database. Having both copies of the original key helped to decrypt the first line of letters A and B, which contained a multitude of null symbols making the decipherment challenging. These lines contain in plaintext ”MAXIMILIAN”, thus, identify him as the sender. Moreover, the sending dates were similarly ”hidden” between nulls at the end of the letters. Only after having obtained the original key these sending dates could be identified.

The used cipher is a homophonic substitution cipher with nomenclature elements and null symbols. Used symbols are a mixture of astrological signs, Greek letters, and esoteric symbols.

The letters contain a total of about 80 distinct ciphertext symbols (homophones). These encrypt single letters and multiple letters. ”I” and ”J” and ”U” and ”V” share the same ciphertext symbol, respectively. There are symbols for duplices (bigrams) like ”NN” or ”ST”, as well as a homophone for the frequently used word ”UND” (English: ’and’). For each of these, at most two different homophones are used. Embedded in the ciphertext are Latin words written in clear, e.g. ”Benigni” or ”Ater”, which turned out to be nomenclature elements (code words) of the cipher. For example ”Benignus” encrypts Archduke Ernest of Austria, which we could only find out by finding the original key.

To decipher letter A, we firstly used the Homophonic Substitution Analyzer component implemented in our open-source software CrypTool 2 (Kopal, 2018). For details on the analyzer, we suggest reading Kopal (2019). In short, the analyzer uses hill climbing and simulated annealing to incrementally improve the decryption key (mapping of homophones to plaintext letters). The user is able to manually improve the automatically generated results of the analyzer. Thus, it was possible to decipher 80% of the letter without having the original key. After finding the original key and entering it into CrypTool 2, it was easily possible to decrypt letters A-C by 95%. Only the code for a few nomenclature elements is still unknown, since these are not described in the original key. Therefore, we assume that there has to be another original key which we have not been able to find so far. Until then, only assumptions can be made about the meaning of these nomenclature elements, e.g. by their usage within the plaintext.

3 Cryptanalysis of the two additional letters

Encouraged by the success with deciphering the first three Maximilian II letters, we started in December 2020 (crypt-)analyzing the two additional letters, which are also stored in the DECODE database. Letter D consists of five full pages of ciphertext symbols and eight lines of ciphertext symbols on a ninth page. Letter E consists of three and a half pages of ciphertext letters as well as three lines of ciphertext on a fifth page.

At first, we compared the ciphertext symbols

Letter	Key	Sender	Receiver(s)	DECODE database (name)	Sending date	Sent from
A	Cyffra Nova Ad Poloniam	Maximilian II	Johan Kochtitzky	Chiffrenschlüssel_fasc 20_kt_14_200-204	7 July 1575	Prague
B	Cyffra Nova Ad Poloniam	Maximilian II	Ambassadors	Chiffrenschlüssel_fasc 20_kt_14_194-198	24 December 1575	Vienna
C	Cyffra Nova Ad Poloniam	Maximilian II	unknown receivers	Chiffrenschlüssel_fasc 20_kt_14_174	23 December 1575	(probably) Vienna
D	unknown name	Johan Chodkiewicz	Maximilian II	Chiffrenschlüssel_fasc 20_kt_205-208	15 November 1574	Vilnius
E	unknown name	Johan Chodkiewicz	Maximilian II	Chiffrenschlüssel_fasc 20_kt_210-212	22 February 1575	Sklow

Table 1: Metadata of all five letters. The column ”Key” contains the names as written on the original key. The column ”DECODE database (name)” is the name used in the database based on the location in the HHStA.

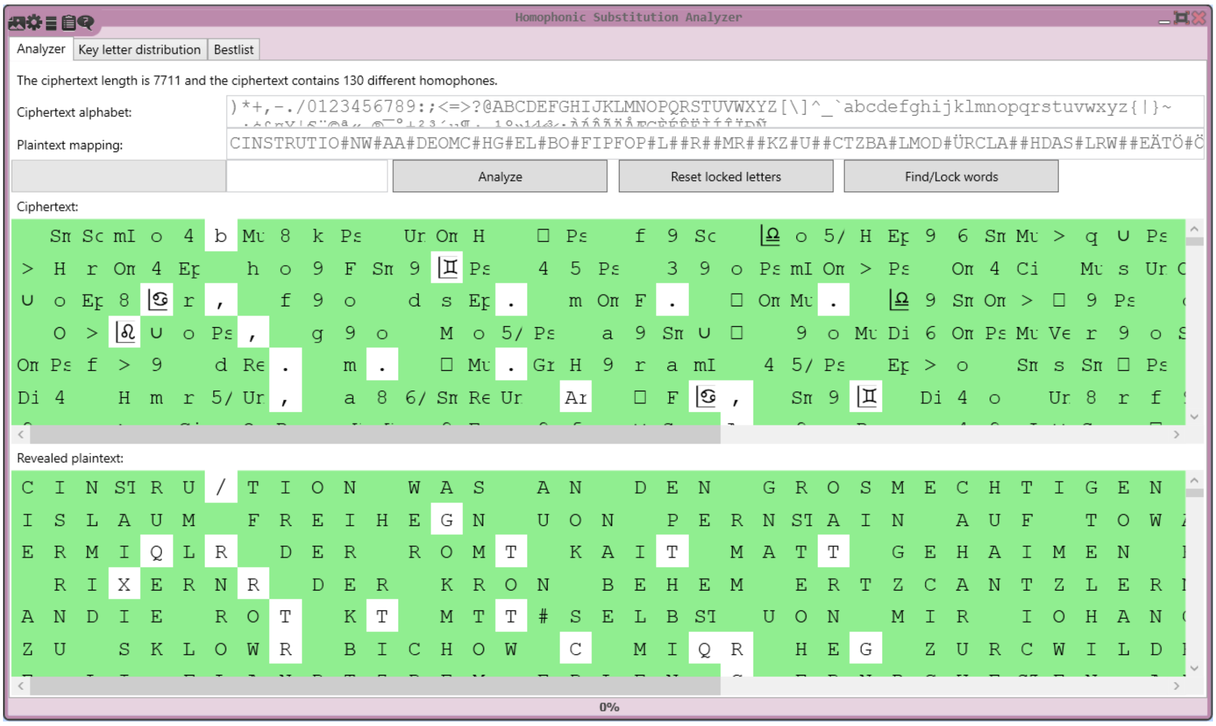
used in the letters D and E with the symbols of the already analyzed letters A-C. Despite of a few ciphertext symbols looking familiar, the used key turned out to be a different one. Therefore, we also searched the key records in the DECODE database for a possible original key. But to our regret we could not find any key suitable for deciphering letters D and E. Therefore, we started to perform a ciphertext-only attack on letter D. We used the Homophonic Substitution Analyzer component implemented in CrypTool 2 to semi-automatically decipher letter D. Figure 1 shows this process. After that, it turned out that letter E was encrypted using the same key. The cryptanalysis of letters D and E was more difficult than the cryptanalysis of the first three letters. Here, we give a short summary of challenges we had in deciphering as well as helpful properties of the two additional letters:

Spaces between words are clearly visible As in letters A-C, spaces between words are (mostly) clearly visible. This eased the decipherment work, since frequently used short German words, like ”DER/DIE/DAS” (English: ’the’) could be spotted and deciphered easily.

Usage of nulls similar to usage in letters A-C Null symbols are rarely used within the ciphertexts. Exceptions are the endings and beginnings of the letters, where multiple nulls are used to confuse an attacker trying to decipher these. The same practice was employed in letters A-C. In addition, in the beginning of letter E, nulls are used between different words in the salutation formula. Moreover, the digits of the sending year (1574) were written as plaintext digits embedded in null symbols. Since we also saw this usage in the first three letters, spotting the sending dates in the additional letters was quite easy.

Usage of Latin words as nomenclature elements Similar to letters A-C, nomenclature elements (code words) used for enciphering persons and places are cleartext Latin words embedded in the ciphertext. Nomenclature elements are, however, only used in letter D. We found a key similar to the two keys used in letters A-E in a collection of letters issued by Andreas Dudithius in 1575 edited in Dudith and Kotońska (1998). In the introduction to the edition, the key of the cipher Dudithius used in his correspondence with the Habsburg is given. However, the source for this key is not indicated. Thus, it remains unclear if it was reconstructed by the author on the basis of the edited letters only, or if it represents a transcription of an original key document. Further investigation of key documents in archives are needed to clear this question. Luckily for us, the published key contains all nomenclature elements used in letter D which facilitated their encoding. This case shows that two keys used in different geographical places and by different persons shared the same nomenclature elements in that time. While that practice opens security issues for keys, it facilitates and accelerates the practicability of keys for encoding and decoding, cf. (Ernst, 1992).

Homophones encoding frequently used words In letters A-C the frequently used German word ”UND” (English: ’and’) was encrypted using its own homophone. Also, we assume that homophones for the words ”Poland” and ”Lithuania” were used. In the additional letters, we found another homophone that encrypts a frequently used word. Based on its positions and usages in the plaintext, we assume that this homophone



The ciphertext length is 7711 and the ciphertext contains 130 different homophones.

Ciphertext alphabet:)*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~

Plaintext mapping: CINSTRUTIO#NW#AA#DEOMC#HG#EL#BO#FIPFOP#L##R##MR##KZ#U##CTZBA#LMOD#ÜRCLA##HDAS#LRW##EÄTÖ#C

Ciphertext:

```
 Sn Sc mI o  4  b  Mu 8  k  Ps    Ur On H     □  Ps    f  9  Sc    ♌  o  5/ H  Ep 9  6  Sn Mu >  q  ∪  Ps
>  H  r  On 4  Ep    h  o  9  F  Sn 9  ♊  Ps    4  5  Ps    3  9  o  Ps mI On >  Ps    On 4  Ci    Mu s  Ur C
∪  o  Ep 8  ♋  r  ,     f  9  o     d  s  Ep .     m  On F  .     □  On Mu .     ♌  9  Sn On >  □  9  Ps
   O  >  ♌  ∪  o  Ps ,     g  9  o     M  o  5/ Ps    a  9  Sn ∪  □     9  o  Mu Di 6  On Ps Mu Ve r  9  o  S
On Ps f  >  9     d  Re .     m  .     □  Mu .  Gr H  9  r  a  mI    4  5/ Ps    Ep >  o     Sn s  Sn □  Ps
Di 4     H  m  r  5/ Ur ,     a  8  6/ Sn Re Ur    Ar    □  F  ♋  ,     Sn 9  ♊     Di 4  o     Ur 8  r  f
```

Revealed plaintext:

```
C I N SI R U / T I O N   W A S   A N   D E N   G R O S M E C H T I G E N
I S L A U M   F R E I H E G N   U O N   P E R N SI A I N   A U F   T O W
E R M I Q L R   D E R   R O M T   K A I T   M A T T   G E H A I M E N
  R I X E R N R   D E R   K R O N   B E H E M   E R T Z C A N T Z L E R
A N D I E   R O T   K T   M T T # S E L B SI   U O N   M I R   I O H A N
Z U   S K L O W R   B I C H O W   C   M I Q R   H E G   Z U R C W I L D
```

Figure 1: Letter D being analyzed using the Homophonic Substitution Analyzer component of CrypTool 2. The top of the analyzer displays the encrypted ciphertext. The bottom of the analyzer displays the deciphered plaintext.

encrypts a royal title, e.g. ”Majestät” (English: ’Majesty’). Additionally, the homophone for encrypting ”UND” in German plaintext parts is also used to encrypt ”ET” in Latin plaintext parts.

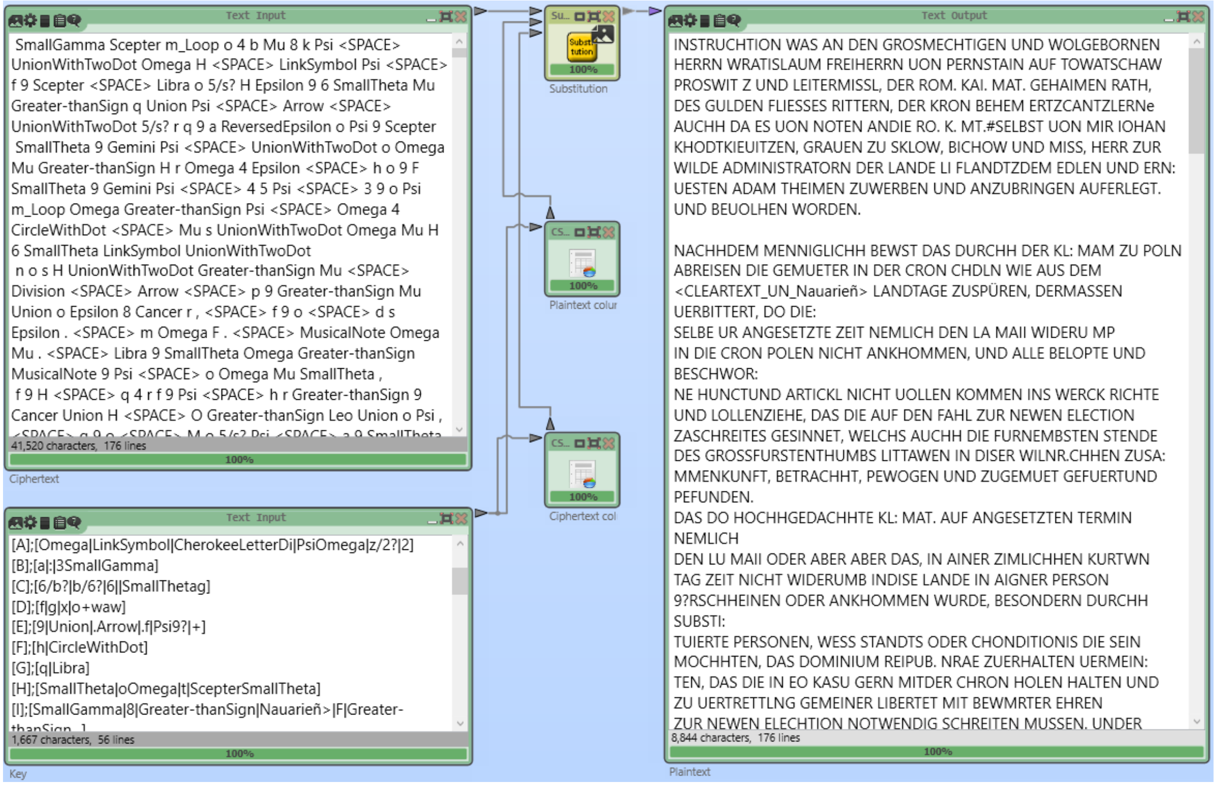
Usage of abbreviations in the plaintext We found several constructions in the plaintext that are abbreviations. For example, we found ”KAY.” (= ”kaiserliche”, English: ’imperial’) and ”E.” (= ”Eure”, English: ’Your’). Such abbreviations can be easily spotted already in the ciphertext, since the used dots are not encrypted.

Usage of interpunction The interpunction, as already shown above, is (mostly) clearly visible in the ciphertexts. Endings of sentences are marked with a dot. Enumerations and abbreviations are also constructed with dots.

Encryption of umlauts The German umlauts are also encrypted in the same manner as in the first three letters. At many positions (but not at all), the homophones for A, O, and U have two small dots on top, meaning, these are the German umlauts Ä, Ö, and Ü.

Non-encrypted cleartext digits When dates are given (such as typically at the end of the letters, but also within the texts) the numbers of the day are presented in non-encrypted digits, e.g. the numbers ”12” on page 1, line 13 in letter E in the date ”12 MAII”. Since the digits might possibly also function as homophones or nomenclature elements, we could only definitely decipher them and disambiguate their meaning in the context of the plaintext. However, at the end of the letters it was easier to spot the digits and identify them as numbers indicating the sending year (1574 and 1575, respectively) since we saw the same usage in letters A-C.

After reconstruction the mappings of homophones to plaintext letters using the Homophonic Substitution Analyzer, the complete ciphertexts could be decrypted easily using the Monoalphabetic Substitution component of CrypTool 2. Figure 2 shows the decryption of letter D using CrypTool 2. As an example decipherment, Figure 3 presents the first paragraph of letter D. Above each line of ciphertext the corresponding deciphered line of German plaintext is shown in red letters. We were able to decipher both letters (D and E) completely. Table 2 contains all homophones used in the first paragraph of letter D.



SmallGamma Scepter m_Loop o 4 b Mu 8 k Psi <SPACE>
UnionWithTwoDot Omega H <SPACE> LinkSymbol Psi <SPACE>
f 9 Scepter <SPACE> Libra o 5/s? H Epsilon 9 6 SmallTheta Mu
Greater-thanSign q Union Psi <SPACE> Arrow <SPACE>
UnionWithTwoDot 5/s? r q 9 a ReversedEpsilon o Psi 9 Scepter
SmallTheta 9 Gemini Psi <SPACE> UnionWithTwoDot o Omega
Mu Greater-thanSign H r Omega 4 Epsilon <SPACE> h o 9 F
SmallTheta 9 Gemini Psi <SPACE> 4 5 Psi <SPACE> 3 9 o Psi
m_Loop Omega Greater-thanSign Psi <SPACE> Omega 4
CircleWithDot <SPACE> Mu s UnionWithTwoDot Omega Mu H
6 SmallTheta LinkSymbol UnionWithTwoDot
n o s H UnionWithTwoDot Greater-thanSign Mu <SPACE>
Division <SPACE> Arrow <SPACE> p 9 Greater-thanSign Mu
Union o Epsilon 8 Cancer r , <SPACE> f 9 o <SPACE> d s
Epsilon . <SPACE> m Omega F . <SPACE> MusicalNote Omega
Mu . <SPACE> Libra 9 SmallTheta Omega Greater-thanSign
MusicalNote 9 Psi <SPACE> o Omega Mu SmallTheta ,
f 9 H <SPACE> q 4 r f 9 Psi <SPACE> h r Greater-thanSign 9
Cancer Union H <SPACE> O Greater-thanSign Leo Union o Psi ,
<SPACE> q 9 o <SPACE> M o 5/s? Psi <SPACE> a 9 SmallTheta

41,520 characters, 176 lines

Ciphertext

[A];[Omega|LinkSymbol|CherokeeLetterDi|PsiOmega|z/2?|2]
[B];[a|:|3SmallGamma]
[C];[6/b?|b/6?|6||SmallThetag]
[D];[f|g|x|o+waw]
[E];[9|Union|.Arrow|.f|Psi9?|+]
[F];[h|CircleWithDot]
[G];[q|Libra]
[H];[SmallTheta|oOmega|t|ScepterSmallTheta]
[I];[SmallGamma|8|Greater-thanSign|Nauarień>|F|Greater-thanSign]

1,667 characters, 56 lines

Key

INSTRUCHTION WAS AN DEN GROSMECHTIGEN UND WOLGEBORNEN
HERRN WRATISLAUM FREIHERRN UON PERNSTAIN AUF TOWATSCHAW
PROSWIT Z UND LEITERMISSL, DER ROM. KAI. MAT. GEHAIMEN RATH,
DES GULDEN FLIESSES RITTERN, DER KRON BEHEM ERTZCANTZLERNe
AUCHH DA ES UON NOTEN ANDIE RO. K. MT.#SELBST UON MIR IOHAN
KHODTKIEUITZEN, GRAUEN ZU SKLOW, BICHOW UND MISS, HERR ZUR
WILDE ADMINISTRATORN DER LANDE LI FLANDTZDEM EDLEN UND ERN:
UESTEN ADAM THEIMEN ZUWERBEN UND ANZUBRINGEN AUFERLEGT.
UND BEUOLHEN WORDEN.

NACHHDEM MENNIGLICHH BEWST DAS DURCHH DER KL: MAM ZU POLN
ABREISEN DIE GEMUETER IN DER CRON CHDLN WIE AUS DEM
<CLEARTEXT_UN_Nauarień> LANDTAGE ZUSPÜREN, DERMASSEN
UERBITTERT, DO DIE:
SELBE UR ANGESETZTE ZEIT NEMLICH DEN LA MAII WIDERU MP
IN DIE CRON POLEN NICHT ANKHOMMEN, UND ALLE BELOPTE UND
BESCHWOR:
NE HUNCTUND ARTICKL NICHT UOLLEN KOMMEN INS WERCK RICHTE
UND LOLLENZIEHE, DAS DIE AUF DEN FAHL ZUR NEWEN ELECTION
ZASCHREITES GESINNET, WELCHS AUCHH DIE FURNEMBSTEN STENDE
DES GROSSFURSTENTHUMBS LITTAWEN IN DISER WILNR.CHHEN ZUSA:
MMENKUNFT, BETRACHHT, PEWOGEN UND ZUGEMUET GEFUERTUND
PEFUNDEN.
DAS DO HOCHHGEDACHHTE KL: MAT. AUF ANGESETZTEN TERMIN
NEMLICH
DEN LU MAII ODER ABER ABER DAS, IN AINER ZIMLICHHEN KURTWN
TAG ZEIT NICHT WIDERUMB INDISE LANDE IN AIGNER PERSON
9?RSCHHEINEN ODER ANKHOMMEN WURDE, BESONDERN DURCHH
SUBSTI:
TUIERTE PERSONEN, WESS STANDTS ODER CHONDITIONIS DIE SEIN
MOCHHTEN, DAS DOMINIUM REIPUB. NRAE ZUERHALTEN UERMEIN:
TEN, DAS DIE IN EO KASU GERN MITDER CHRON HOLEN HALTEN UND
ZU UERTRETTLNG GEMEINER LIBERTET MIT BEWMRTER EHREN
ZUR NEWEN ELECHTION NOTWENDIG SCHREITEN MUSSEN. UNDER

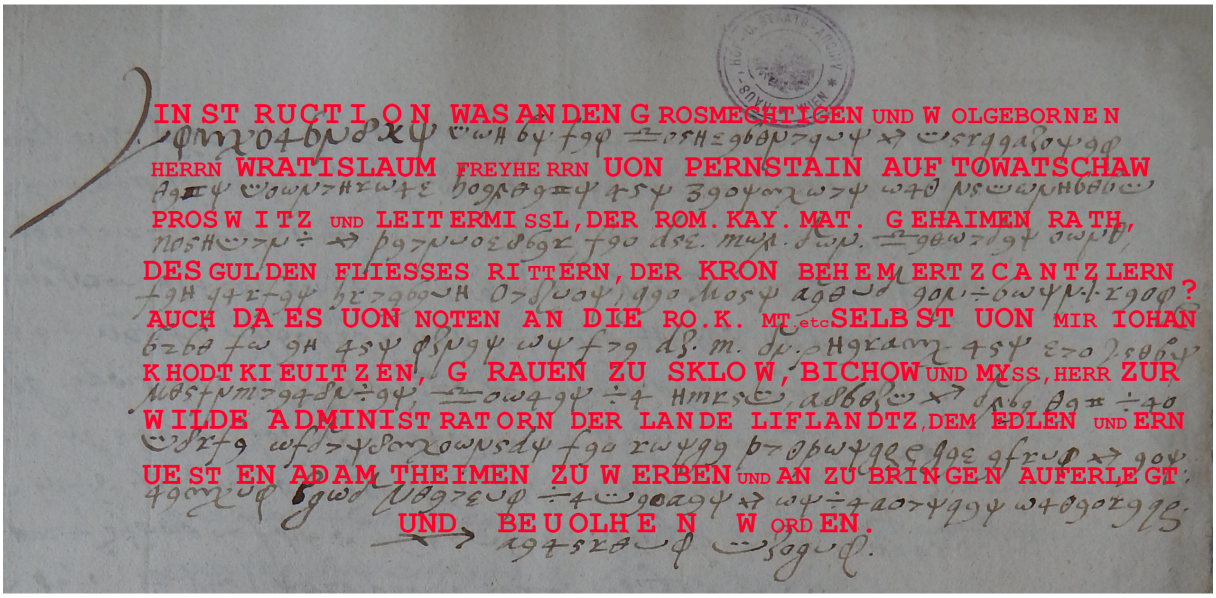
8,844 characters, 176 lines

Plaintext

Figure 2: Letter D decrypted using the Monoalphabetic Substitution component of CrypTool 2 and the reconstructed key.

Plaintext symbol(s)	Ciphertext symbol(s)	Plaintext symbol(s)	Ciphertext symbol(s)
A	ω ♂	R	o d
B	a	S	H
C	6b	T	ʍ 5
D	fg	U / V	4 ≥
E	9 U	W	◡̈
F	⊙h	Y	ʎ
G	♎ ꝗ	Z	÷
H	θ		
I / J	> 8 ʔ	UND/ET	↗
K	m M		
L	т p	ST	♍
M	ε ♪		
N	Φ Ψ	RR	♊
O	3 S	SS	♋
P	3 n	TT	♌

Table 2: Partially reconstructed key (showing only homophones used in the first paragraph of letter D).



INSTRUCTION WAS AN DEN GROSMECHTIGEN UND WOLGEBORNEN
HERRN WRATISLAUM FREYHERRN UON PERNSTAIN AUF TOWATSCHAW
PROSWITZ UND LEITERMISSL, DER ROM. KAY. MAT. GEHAIMEN RATH,
DES GULDEN FLIESSES RITTERN, DER KRON BEHEM ERTZCANTZLERN ?
AUCH DA ES UON NOTEN AN DIE RO.K. MT.etc SELBST UON MIR IOHAN
KHODTKIEUITZEN, GRAUEN ZU SKLOW, BICHOW UND MYSS, HERR ZUR
WILDE ADMINISTRATORN DER LANDE LIFLANDTZ, DEM EDLEN UND ERN
UESTEN ADAM THEIMEN ZU WERBEN UND AN ZU BRINGEN AUFERLEGT
UND, BEUOLHEN WORDEN.

Figure 3: Original first paragraph of letter D with deciphered German plaintext shown in red letters above each line of ciphertext. English translation: ”Instruction about what I, Johan Khodtkievitz, Count of Shklow, Bichow and Miss, Castellan at Vilnius [and] governor of the land Livonia have imposed and ordered to Adam Theim that should be advertised and promoted to the mighty and honorable baron Vratislav (II.) baron z Pernštejna of Tovačov, Prostějov and Litomyšl, the imperial Roman Majesty‘s privy counsellor, knight of the Golden Fleece, archchancellor of the Crown Bohemia, and also, because there is need, to the imperial Roman Majesty etc. himself.”

4 Historical context and content of the letters

Both letters form part of the same broad historical context as the previously presented letters A-C: the election of the ruler of the Polish-Lithuanian Commonwealth in 1575. The Commonwealth (originally ”Crown of the Kingdom of Poland and the Grand Duchy of Lithuania”) included areas of today’s Poland, Ukraine, Estonia, Latvia, Lithuania, and Belarus. The Polish-Lithuanian crown had been vacant from June 1574 and the election of a successor started in November 1575. In the Polish-Lithuanian Commonwealth, the monarch ruler was elected by the nobility which in 1574-1575 included more than 50,000 persons. The Habsburg were interested in gaining the crown and nominated several candidates, among them the emperor Maximilian II himself and his son Ernst (cf. Rhode (1997), Augustynowicz (2001), Roşu (2017). In the *interregnum* period when the crown was vacant, the Habsburg put intensive efforts into diplomatic correspondence to make campaign for its candidacy. The main supporters of the Habsburg were the members of the Lithuanian higher nobility and the clerics while the nobility of Lesser Poland had an anti-Habsburg attitude and favored a local candidate. This divide eventually led to a double election in December 1575 of both Maximillian II. and the Polish princess Anna Jagiellon, giving her Stefan Báthory, the Prince of Transylvania, for husband (cf. ibid.). Eventually, the latter candidates succeeded in claiming the throne and got married and crowned in May 1576.

While letters A-C presented in Kopal and Waldispühl (2021) were sent by Maximilian II to his Polish and Lithuanian delegates in July 1575 (letter A) and on 23 and 24 December 1575 (letter C and B), respectively, the current letters D and E are dated earlier and were sent by the Lithuanian nobleman, Grand Marshal of Lithuania, Johan Chodkiewicz. They show the perspective and interests of the Lithuanian higher nobility in late 1574 and early 1575 as presented to Maximilian II.

In the following is a short summary of the content of the two letters and report on open problems regarding source criticism and the historical contextualization.

Letter D: LEGATIO IOANNI KHODTKIEUITI[1], sent from Vilnius, 15 November 1574 This letter dated on 15 November 1574 is indicated as a message (LEGATIO) by Johan Chodkiewicz and was addressed to Maximilian II. The letter was sent from Vilnius (ZUR WILDE[2]) where Johan Chodkiewicz was castellan.

In the first paragraph, Johan Chodkiewicz makes himself known and says that he gives an instruction on what he has ordered to his servant Adam Theim to report to Vratislav (II.) z Pernštejna and also to the emperor himself. In the following he advises Maximilian to win supporters and prepare for the election of a new king of Poland and Grand Duke of Lithuania before the gathering on 12 May. He informs the emperor about the divide between the Lithuanian and Prussian senators who back the Habsburg candidacy on the one side and the Polish senators who refuse to take Archduke Ernst as their king and ruler on the other side. Chodkiewicz then recommends Maximilian to mobilize his allies in Hungary, Moravia, and other places. He expresses his wish that, if ”in the lucky case” Ernst will gain the thrown, the privileges of the Lithuanian Grand Duchy will be restored. He advises Maximilian to send envoys to Lithuania at the latest by March to negotiate certain privileges with the local nobility and make a resolution. In the last two paragraphs he gives his allegiance and loyalty and concludes the letter.

Letter E: Copy of Johan Chodkiewicz' letter to Maximilian II, sent from Sklow (Sjkloŭ), 22 February 1575 This letter was filed as ”ABSCHRIFT IOHANN CHODTKIEUIZ SCHREIPENS AN DIE MAJESTÄT” ('copy of Johan Chodtkieviz letter to the Majesty') and was sent from Sjkloŭ (Chodkiewicz' Duchy in today's Belarus) on 22 February 1575.

In contrast to letter D, this exemplar is introduced with a salutation formula addressing the emperor. Chodkiewicz then confirms the receipt of Maximilian's message and reassures his loyalty to the emperor. In a humble tone he utters his doubts about what Maximilian mentioned in his earlier letter. Unfortunately, we lack the whole context to understand what Maximilian's suggestions exactly were. However, Chodkiewicz fears these matters might lead to ”all sorts of repulsive thoughts [...] all kinds of confusions and splits”. He suggests to send out his own envoy, Adam Theim, in order to bring the Lithuanian electorate on the emperor's side and he thinks that in the following, it would equally be possible to attract several Polish nobles as supporters of the Habsburg candidacy. Uttering his loyalty to the emperor, he reassures that it would be impossible for the emperor to achieve common consensus among the voters without supporters like him. He suggests Maximilian to make a contract first with the higher nobility only who then would communicate it further to other electors. The letter concludes with a declaration of loyalty and a humble excuse for bringing up a suggestion that might annoy Maximilian.

Open problems In the course of research on the historical background, we found a reference to the content of letter D in Augustynowicz (2001) who even cites parts of the letter in note 106 on page 50. The text given corresponds to five lines in letter D, however, it shows some deviations in orthography, e.g in the use of more double consonants (*auff, dessenn* vs. AUF, DESSEN in letter D) or *meinung* instead of MAINUNG. In addition, for this passage, Augustynowicz refers to two documents with the shelf marks ”HHStA Wien, Polen I, 23, D, 44r” and ”ebenda, Ungarn, 105, C, 20r-v”. These documents are different from the ones we are dealing with here. Thus, the same text content seems to be represented in different physical documents in the State Archive of Vienna. It is the task of future investigation to compare these three documents and determine their relation both with regard to content and textual representation. For instance, there might even be the possibility that one of these documents cited in Augustynowicz (2001) contains the plaintext of the here analyzed ciphertext.

The content of letter E, on the other hand, seems not to form part of (Augustynowicz, 2001)'s work. However, it is filed as a copy, which implies that there must be an original. In future work, the possible transmission of this letter in other documents likewise has to be clarified.

5 Language

The plaintext language of letters D and E is German with short passages in Latin. In 16th cen-

[1] Here and in the following, we use capital letters to represent plaintext passages of the two deciphered documents.

[2] The place name ”Wilde” for Vilnius was used in historical German from 14th century onwards.

tury Lithuania, many languages were used simultaneously. German was one of the languages for communication with foreigners (next to Latin and Church Slavonic) while Polish (for nobleman) and Lithuanian (for peasants) were the main means of spoken communication. For written correspondence within Lithuania a local chancery language labeled ”Old White Russian” was used (Niendorf, 2006). With regard to this diversity of different languages and the German speaking addressee, Johan Chodkiewicz’ use of German is not surprising.

Written dialect The main linguistic characteristics of the written dialect can be defined as very similar to the Austrian-German office language we found in the letters A-C sent by Maximilian II’s chancery. There is, for instance, the differentiation of the spelling <ai> for an old Germanic diphthong _*ai_ (e.g. AINER ’one’, BAIDE ’both’) and the spelling <ei> for a younger diphthong from an older long vowel _*ī_ (e.g. ZEIT ’time’, FLEISS ’diligence’, SCHREIBEN ’letter, writing’). However, this use is less consistent in letters D and E than in letters A-C. Letter D, for instance, shows variation of <ei>- and <ai>-spellings in some instances (MEINUNG and MAINUNG ’opinion’ or GEMEIN and GEMAIN ’general’). Additionally, the use of <p> for an older _*b_ is more common in the two current letters than in the letters sent by Maximilian and also found in the prefix *be-* (e.g. PEWOGEN, PEFUNDEN in letter D) which is usually not the case in Austrian German (Wiesinger, 2012). The syntax is typical of the chancery style used in the 16th century and the sentences show a similar complexity to what we found in letters A-C. Additionally, some main features, such as the dropping of auxiliary verbs in subordinate clauses, are equally present in the letters sent from Lithuania.

German-Latin code switching The plaintexts of both letters include some smaller parts in Latin embedded into the German text. This is a linguistic feature we did not observe in letters A-C. The code switching includes both shorter passages, such as example 1 and longer passages, such as example 2.

1. ”EO KASU” in the sentence DAS DIE IN *EO KASU* GERN MIT DER CRON POLEN HALTEN UND ZU VERTRETUNG GEMEINER LIBERTET
2. DAS ALLE DENEN SO E. [EURE] M [MAIESTET] GEWOGEN, AUCH MEINER PERSON *SINE ISTIS MEDIIS IMPOSSIBILE [EST], OMNIUM ANIMOS IN EODEM KONSENSU* ZU ERHALTEN

It is interesting to note that the Latin passages are always embedded syntactically into German sentences. There is no entire Latin sentence standing on its own. Moreover, the Latin parts involve both formulaic language (e.g. example 1) but also more freely formulated passages (example 2).

The homophone used for the German word UND (English: ’and’) is also used in the Latin passages which means it performs its semantic function irrespective of the language-specific context. One example is the passage DE OMNIBUS [ET] SINGULIS in letter E, where [ET] is represented by a homophone (see Table 2).

Punctuation marks In contrast to the earlier analyzed letters A-C, the punctuation marks were included in the transcriptions of letters D and E which allows for some observations of the use of punctuation. Generally it can be noted that punctuation marks are not encrypted and function exactly in the way they would be employed in a cleartext. This concerns, for instance, dots that were used consistently in abbreviations (e.g. ROM. KAY. MAT.), as already mentioned in Section 3, but also commas that separate clauses and dots at the end of sentences and paragraphs. Since dots are easily visible in the ciphertext they imply a security flaw. This is equally the case for the use of colons at the end of a line as a separator when the word continues on the next line (e.g. DIE:SELBE or ZUSA:MMENKUNFT in letter D).

These observations clearly show that it is worth transcribing punctuation marks in ciphertexts since they give information about linguistic structures and may not only facilitate the decryption of the ciphertext but also the comprehension of the content.

6 Conclusion and future work

This paper is a new direct outcome of the DECRYPT project. It describes how we transcribed, analyzed, and deciphered two additional diplomatic letters sent in the time of Maximilian II in the years 1574 and 1575. The work on these letters is the continuation of the previous work on the decipherment of letters sent by Maximilian. The letters presented here were sent by the nobleman Johan Chodkiewicz to the emperor in November 1574 and February 1575. The letters were encrypted with a different key as the one used for encrypting Maximilian's letters. In contrast to Maximilian's letters, where we were able to find the original key in the DECODE database, we currently do not have knowledge about the original key used in Chodkiewicz' letters.

However, in the course of working on this paper in early 2021, we found an edition of letters of Andreas Dudith, a Hungarian nobleman, bishop, humanist, and ambassador of Maximilian II in Kraków where the key used in Dudith's correspondence is presented (Dudith and Kotońska, 1998). This key contains, besides additional homophones, the same homophones as used in the letters A-C sent by Maximilian II. Moreover, it contains nomenclature elements that fit for letter D written by Chodkiewicz. Therefore, in future work, we will compare the Dudith nomenclature to the keys stored in the DECODE database and to the key which we reconstructed for the decipherment of the Chodkiewicz letter. As a preliminary result, we can say that it seems that the same nomenclature elements were used among different cryptographic keys at that time. Clearly, this introduced a potential threat since being in possession of one key enables an adversary to also decipher nomenclature elements of other (similar) keys. On the other hand, this practice facilitated the work for the encoders and decoders because they probably knew the code words by heart.

Review of previous literature has likewise shown that the HHStA holds other documents that have a close relation to some of the cryptographic letters presented here. Hence, another future task is to visit the HHStA and gather material in the folders "Polen I" for further analyses and comparison.

The main new cryptographic findings of and differences between the Maximilian's letters (Letters A, B, and C) and Chodkiewicz' letters (Letters D and E) are:

- Letters D and E are encrypted using the same key, but it was a different key than the one used in the Maximilian letters (A, B, and C).
- However, in general it can be said that knowledge of text structure and cryptographic practice from other letters written in the same historical context are useful for deciphering newly found encrypted manuscripts. In our case, the comparison of the use of nulls, the placing of names (of sender and addressee), and the placing and execution of dates we have seen in letters A-C facilitated the decipherment of letters D-E.
- Chodkiewicz used abbreviations in the ciphertext, while in the Maximilian letters no abbreviations can be found.
- In contrast to the Maximilian letters, where a lot of nulls were used, these can only rarly be found in Chodkiewicz' letters. Only the dates at the endings are embedded in nulls similar to the practice in Maximilian's letters.
- As described above, nomenclature elements were shared among different keys at that time in the Habsburg Empire.
- Chodkiewicz switches between German and Latin (code switching) in the plaintext. Latin was not recognized in the automatic cryptanalysis and therefore, transcription and deciphering errors were assumed in the beginning of the cryptanalysis. After Latin had been identified in the linguistic analysis, the decipherment could be verified.
- Interestingly, the same ciphertext symbol (homophone) was used for the conjunction "UND"/"ET" in the ciphertext, irrespective of the plaintext language German or Latin.

The decipherment of the five Maximilian II letters using the homophonic substitution analyzer in CrypTool 2 helped us to further enhance our cryptanalytical algorithms as well as to improve the general handling of our tools. Furthermore, having all letters as transcriptions that follow the DECRYPT transcription guidelines, proved to ease and speed up the cryptanalysis. This confirms that the common standards developed within the DECRYPT project and the cooperation between experts from different scientific fields can be very helpful and fruitful.

Additionally to performing cryptanalysis to decipher the Chodkiewicz letters, we analyzed linguistic aspects of the letters. Our main findings here are that the written dialect is similar to the one employed by Maximilian's chanceries detected in letters A-C. However, the letters sent from Lithuania seem to show more orthographic variation and make use of German-Latin code switching. Moreover, since punctuation is not encrypted, it brings linguistic structures to the fore and facilitates both decipherment and text comprehension.

The decipherments of the three letters of Maximilian II to his chamberlain Johann Kochtitzky and ambassadors (letters A-C) and the letters from the Lithuanian nobleman Jan Chodkiewicz to Maximilian II (letters D and E) provide insight in the (secret) Habsburg views and actions before the free election in 1575. They show the deep division between the Polish and Lithuanian noblemen, the Lithuanian side pro and the Polish side contra the Habsburg empire's candidates. Because of that division, Maximilian made efforts to achieve his goal of convincing the Polish electors to vote for his position. Besides the offer of money and rights he even considers war efforts in case his wishes are not fulfilled. Clearly, a deeper and more profound historical analysis and contextualization of the revealed content in the diplomatic letters by historians is needed in future work. To allow this, we uploaded the complete decipherments of all of the discussed letters to the DECODE database.

Acknowledgments

This work was supported by the Swedish Research Council, grant 2018-06074. We would like to thank our DECRYPT project colleagues Crina Tudor, for managing the work on the transcriptions, and Bernhard Esslinger for valuable feedback on our work. Moreover, we are very grateful to Leopold Auer for important references on the historical contextualization he shared with us. Lastly, we would like to thank the experts from the State archive of Vienna for their fruitful comments on the decipherment.

References

Christoph Augustynowicz. 2001. *Die Kandidaten und Interessen des Hauses Habsburg in Polen-Litauen während des Zweiten Interregnums 1574-1576.* WUV-Univ.-Verl., Wien.

András Dudith and edited by Catharina Kotońska. 1998. *Epistulae 4: 1575*, volume 13,4 of *Bibliotheca scriptorum medii recentisque aevorum: Series Nova.* Akadémiai Kiadó.

Hildegard Ernst. 1992. Geheimschriften im diplomatischen Briefwechsel zwischen Wien, Madrid und Brüssel 1635–1642. *Mitteilungen des Österreichischen Staatsarchivs*, 42:102–126.

Nils Kopal and Michelle Waldispühl. 2021. Deciphering three diplomatic letters sent by Maximilian II in 1575. *Cryptologia*, pages 1–25.

Nils Kopal. 2018. Solving Classical Ciphers with CrypTool 2. In *Proceedings of the 1st International Conference on Historical Cryptology HistoCrypt 2018*, pages 29–38. Linköping University Electronic Press.

Nils Kopal. 2019. Cryptanalysis of Homophonic Substitution Ciphers Using Simulated Annealing with Fixed Temperature. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt*, pages 107–16.

Benedek Láng. 2020. Was it a Sudden Shift in Professionalization? Austrian Cryptology and a Description of the Staatskanzlei Key Collection in the Haus-, Hof-und Staatsarchiv of Vienna. In *Proceedings of the 3rd International Conference on Historical Cryptology HistoCrypt 2020*, pages 87–95. Linköping University Electronic Press.

Mathias Niendorf. 2006. *Das Großfürstentum Litauen. Studien zur Nationsbildung in der Frühen Neuzeit (1569-1795).* Harrassowitz Verlag, Wiesbaden.

Maria Rhode. 1997. *Ein Königreich ohne König. Der kleinpolnische Adel in sieben Interregna.* Deutsches Historisches Institut Warschau. Quellen und Studien. Harrassowitz, Wiesbaden.

Felicia Roşu. 2017. *Elective monarchy in Transylvania and Poland-Lithuania, 1569-1587.* Oxford University Press, Oxford, 1st edition.

Peter Wiesinger. 2012. Bairisch-österreichisch – Die Wiener Stadtkanzlei und die habsburgischen Kanzleien. In Albrecht Greule, Jörg Meier, and Arne Ziegler, editors, *Kanzleisprachenforschung. Ein internationales Handbuch*, pages 415–439. de Gruyter, Berlin.

An alchemical cipher in a shared notebook of John and Arthur Dee (Sloane MS 1902) [Work In Progress]

Sarah Lang
University of Graz
Elisabethstraße 59/III
8042 Graz, Austria
sarah.lang@uni-graz.at

Megan Piorko
Science History Institute
315 Chestnut St
Philadelphia, PA 19106
mpiorko@sciencehistory.org

Abstract

Alchemy, while being known for its secrecy, cryptographical and stylistic devices, isn't known for its ciphers in particular. However, ciphers can sometimes be found in alchemists' and chymists' (laboratory) notebooks. This paper discusses a ciphertext and cipher table found in a shared notebook by John and Arthur Dee (Sloane MS 1902). It presents a bibliographical description as well as context for interpretation. However, thus far it has not been possible to solve the cipher.

1 Introduction: Ciphers in the context of alchemical secrecy

The secret is paradigmatic of alchemy (Principe, 2013). It is a topos in secondary literature about alchemy as well as the alchemical tradition itself. Over the last decades, secrecy studies have contributed important new insights on early modern secrecy, its contents (such as recipes), its media (such as books of secrets) and its plethora of related practices, especially with regard to scientific secrecy (Vermeir, 2012). While the discussions around the 'New Historiography of Alchemy' led by W. Newman and L. Principe have greatly improved the methodology for the discussion of alchemical language and its secrets, studies on secrecy more specific to alchemy are yet lacking (Principe and Newman, 2001). Much has been written about the cultural and practical significance of secrecy in alchemy (Principe, 2013), its proclivity for playful encipherment (Bilak, 2020) but also its rhetoric of secrecy in the 'economy of secrets' (Jütte, 2011) which serves as the marketplace for 'entrepreneurial alchemy' (Nummedal, 2007) and the circulation of crafts knowledge.

Alchemy and chymistry, for the most part, are known for their cryptographic devices which are metaphorical and qualitative in nature, such as anagrams or *Decknamen* (Newman, 1996).[1] Stylistic devices, so to say, rather than actual ciphers based on mathematical principles and letter substitutions. The result is a somewhat special status of alchemical secretive devices which are mostly non-mathematical but rather qualitative in nature, compared to the rest of the cryptological landscape of their contemporaries. Agnieszka Rec laments that alchemical ciphers remain a seriously understudied topic, especially given the abundance, even omnipresence of such devices in alchemical literature (Rec, 2014). Consequently, alchemy thus far lacks contextualization in strictly cryptological contexts: David Kahns *The Codebreakers*, the classic work of cryptography studies, only mentions alchemy in passing (Kahn, 1996).

However, chymical laboratory notebooks have been known to contain ciphers (Newman and Principe, 2003). It is along those lines that we can locate the topic of the present paper: We discuss an alchemical cipher found in a shared notebook of John and Arthur Dee, Sloane MS 1902 (reproduced in figures 1–3).

2 An alchemical cipher by Arthur Dee? Sloane MS 1902

Sloane MS 1902 is a small Paracelsian astrological medical notebook containing notes from father and son, John Dee (1527–1608) and Arthur Dee (1579–1651). While his father John has been a popular subject of historical studies, studies on Arthur Dee remain scarce (Piorko, 2019). Arthur's handwriting is similar to his father's but can be differentiated with a careful eye.[2] The pages that contain John's notes are exclusively parchment

[1] The historiography of the term 'chymistry' has been studied in detail (Principe and Newman, 2001).

[2] We conclude by a handwriting analysis comparing the relevant pages to Arthur Dee's manuscripts and his father's handwriting in the same medical notebook that the cipher and table are in Arthur's handwriting. Also, a material analysis of

while Arthur wrote either on the verso of John's notes or on a separate sheet of paper, subsequently combined to create this notebook.[3]

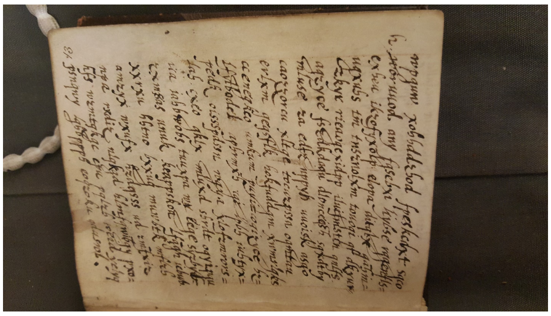


Figure 1: Sloane MS 1902, Folio 13r, British Library

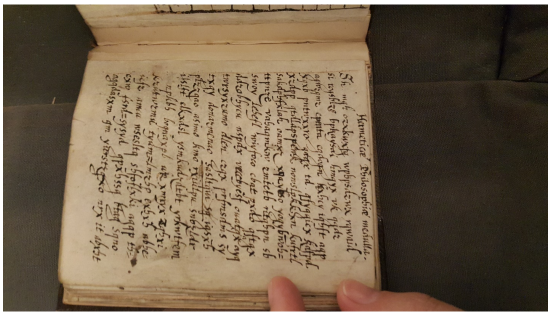


Figure 2: Sloane MS 1902, Folio 13v, British Library

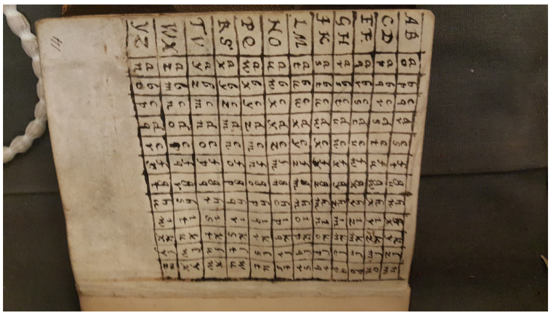


Figure 3: Sloane MS 1902, Folio 14r, British Library

2.1 On the contents of the handbook in general

To the modern reader, the organization of this commonplace medical text appears random and contradictory. However, this would have been a working notebook for Arthur, as alchemical and astrological ideas about the human body directly influenced his medical practice. The themes found in this text are astrology, alchemy, coded language, and Paracelsian iatrochemical treatment. Medicine and alchemy/astrology were not mutually exclusive to the early modern alchemical physician, but provided explanations for the inner workings of the human body.

This commonplace medical notebook provides historians with a primary source account of early modern knowledge creation through scribal speculation. In this notebook, John Dee worked through the relationship between medicine and astrology in a micro-macrocosmic universe as is evident by the drawing of the human form and corresponding alchemical and astrological symbols. Arthur subsequently worked through his father's textual and visual conclusions and added his family's horoscopes and general medical astrological observations, sometimes in the margins or even in a small blank space left by his father, as is the case with his own horoscope. In this way, Arthur is taking his father's medical philosophy and using it as a basis from which to build his own knowledge through the scribal medium. This type of material evidence of early modern speculation is invaluable as it allows the historian to be privy to a seventeenth-century physicians' knowledge-making process on paper.

The following section contains an original bibliographical description of the notebook, followed by an analysis.

2.2 Bibliographical Description of British Library, Sloane MS 1902

Paper and parchment, small manuscript bound in leather, 10cm x 12.

31 folios numbered with Arabic numerals throughout.

- Folios 11v-14r, part of 27v, 28r, 29v are oriented upside-down from the rest of the codex.

- Fols. 5r, 9v, 10r, 11r, 27v, 28r: Natal horoscopes and lifetime predictions.

- Fols. 1v, 4r/v, 6r-8r, 14v, 15r-22v, 23r-27r, 29r/v: Astrological medical projection.

- Fols. 2r, 3r/3v, 9r, 10v: Astrological symbols and corresponding body parts.

the paper quality and of how the book is compiled supports this. The relevant pages being upside down, the cipher is likely to have been added by Arthur after he compiled his father's notes.

[3] The following leaves are paper: 1-2, 5-8, 31.

- Fols. 13r/13v-14r: Ciphertexts and cipher table.

- Fols. 11v-12v, 28v, 30r/v, 31r/v: References to alchemical authors and processes.

The leaves of this tiny square commonplace book are taped together, rather than sewn, to create a codex. After the loose leaves were assembled into codex form, an owner wrote page numbers on the top right on the recto of each leaf. It is bound in a Sloane collection binding with a gold gilt Sloane library stamp on the front and "BRIT. MUS.—S.L. 1902/ASTROLOGICAL NOTES" on the spine. Five types of alchemical-medical knowledge making categories can be gleaned from this manuscript. Sometimes the leaves of this notebook are written on both recto and verso sides on related topics, when that is the case they will be referred to as unit (example: 4r/v). As this manuscript is a collection of John's loose notes filled in later by Arthur, it is more fruitful to examine its pages as two sides of a single leaf which may have corresponding information on the recto and verso rather than as a codex with continuous information from left to right, which modern readers are inclined to do. Evidence such as the later additions to John's notes on parchment, the matching size of the paper that Arthur used, and the corresponding relationship between the folios indicate that Arthur created the codex and added to it in response to his father's notes.

2.3 The ciphertext and code table (folios 13r/13v-14r)

Folio 13 is bound upside-down in the notebook. Both the recto and verso are filled with prose written in a ciphertext, with the Latin title *Hermeticæ Philosophiæ medulla* ('Marrow of the Hermetic Philosophy'). Folio 14 recto is also upside-down in the context of the majority of the codex and contains a grid cipher for the ciphertext on folio 13. The pages that are written upside down correspond to Arthur Dee's handwriting, and are written on the reverse side of a correctly oriented leaf written in the hand of his father. The code is not a simple monoalphabetic substitution cipher (for example, 'n' represents 'a'). Digital cryptanalysis algorithms commonly available on the web yielded no meaningful results.

3 Conjectures on the context of the cipher

All of the pages with ciphers are pasted upside down into the booklet. Referring to this notebook specifically, there is just one publication (Appleby, 1977). However, it doesn't analyze it or give further information. The approximate dating is 1610, assuming the upside-down cipher parts were written by Arthur. Somewhat similar tables are also to be found in the Book of Soyga (*Aldaraia sive Soyga vocor*), a 16th century Latin treatise owned by John Dee. Among other content on magic stemming from the context of the Christian Cabalistic tradition, there are several so-called Magic Tables (Reeds, 2006). However, it seems unlikely that there is a relationship. Furthermore, René Zandbergen and Rafał T. Prinke demonstrate that the evidence that John Dee ever owned the Voynich MS (and that Arthur Dee saw it as a child) is very thin and hardly reliable (Zandbergen and Prinke, 2016), so a connection to the Voynich isn't likely either.

The 'medulla' (*marrow*) mentioned in the plaintext heading could possibly be a reference to the text "Benjamin Lock his Picklock to Riply his Castle" which Arthur Dee copied as a manuscript. Furthermore, Lock was a student of John Dee's.[4] *Medulla* could also be a reference to Ripley's *Medulla* (*Georgii Riplei Angli Medvlla Philosophiae Chemicae*, 1614) which "is a Latin re-translation of the English *Marrow*" (Rampling, 2012).

Given that the main languages used by both John and Arthur Dee are English and Latin but the plaintext heading is in Latin, we assume that the language of the ciphertext must be either Latin or English.[5] It is not a simple Caesar cipher or other monoalphabetic substitution cipher, since a frequency analysis shows no spikes for vowels and an overall too uniform distribution for a simple substitution cipher. Substitutions based on the table reproduced as Figure 3 didn't even yield partial results. Either the correct usage of this table eludes the authors of this paper (which is very possible) or the table might have been a try by the Dees themselves to crack the cipher. The key table visually resembles a *tabula recta*, so it's likely

[4] This can be gathered from the Wellcome MS 436.

[5] The notebook has Latin and English texts to equal parts with John writing mostly in Latin and Arthur writing mostly in English with some cross-over.

a Vigenère-type cipher, however, the solution has thus far eluded the present authors. A set of likely keywords was tried out but none yielded any results.

The cipher table from Figure 3 matches those of the Bellaso/Della Porta ciphers which are polyalphabetic substitution ciphers similar to the Vigenère (Buonafalce, 2006). However, while Vigenère ciphers use 26 alphabets, Bellaso/Della Porta ciphers only use 13 reversible alphabets, each being associated with two letters from the alphabet (like the row indices 'AB', 'CD', etc. in Figure 3).[6]

While John Dee was a mathematician well versed in ciphering techniques, his son Arthur was not. Albeit it is likely he was exposed to the subject area through his father. Since neither the exact context nor author of this cipher table and ciphertext are known, it is possible that, for example, the ciphertext was copied into this notebook by either John or Arthur Dee from an external source. The table could have been used to encode the ciphertext but it could just the same have been a (possibly unsuccessful) attempt at solving the ciphertext from figures 1 and 2.

References

John H. Appleby. 1977. Arthur Dee and Johannes Banfi Hunyades: Further information on their alchemical and professional activities. In *Ambix*, volume 24/2, pages 96–109.

Donna Bilak. 2020. Chasing Atalanta. Maier, Steganography, and the secrets of nature. In *Furnace and Fugue. A Digital Edition of Michael Maier's Atalanta fugiens (1618) with Scholarly Commentary.*

Augusto Buonafalce. 2006. Bellaso's reciprocal ciphers. In *Cryptologia*, volume 30, pages 39–51.

Deborah E. Harkness. 1999. *John Dee's Conversations with Angels: Cabala, Alchemy, and the End of Nature.* CUP, Cambridge.

Daniel Jütte. 2011. *Das Zeitalter des Geheimnisses. Juden, Christen und die Ökonomie des Geheimen (1400–1800).* Vandenhoeck & Ruprecht, Göttingen.

David Kahn. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Scribner, NY.

William R. Newman and Lawrence M. Principe. 2003. The chymical laboratory notebooks of George Starkey. In Frederic L. Holmes, Jürgen Renn, and Hans-Jörg Rheinberger, editors, *Reworking the Bench. Research Notebooks in the History of Science*, Archimedes. New Studies in the History and Philosophy of Science and Technology 7, pages 25–42, Dordrecht. Kluwer Academic Publishers.

William R. Newman. 1996. "*Decknamen* or pseudochemical language?" Eirenaeus Philalethes and Carl Jung. In *Revue d'histoire des sciences*, volume 49, pages 159–188.

Tara E. Nummedal. 2007. *Alchemy and Authority in the Holy Roman Empire.* University of Chicago Press, Chicago.

Megan Piorko. 2019. Seventeenth-century chymical collections: A study of unique copies of fasciculus chemicus. In *Papers of the Bibliographical Society of America*, volume 113/4, pages 409–446.

Lawrence M. Principe and William R. Newman. 2001. Some problems with the historiography of alchemy. In William R. Newman and Anthony Grafton, editors, *Secrets of Nature: Astrology and Alchemy in Early Modern Europe*, pages 385–432, Cambridge/Massachusetts. MIT Press.

Lawrence M. Principe. 2013. *The Secrets of Alchemy.* The University of Chicago Press, Chicago.

Jennifer M. Rampling. 2012. Transmission and transmutation: George Ripley and the place of english alchemy in early modern Europe. In *Early Science and Medicine: Alchemy on the Fringes: Communication and Practice at the Peripheries of Early Modern Europe*, volume 17/5, pages 477–499.

Agnieszka Rec. 2014. Ciphers and secrecy among the alchemists: A preliminary report. In *Societas Magica Newsletter*, volume 31 (Fall), pages 1–6.

Jim Reeds. 2006. John Dee and the magic tables in the book of Soyga. In Stephen Clucas, editor, *John Dee: Interdisciplinary Studies in English Renaissance Thought*, International Archives of the History of Ideas/Archives internationales d'histoire des idées 193, pages 177–204, Dordrecht. Springer.

Koen Vermeir. 2012. Openness versus secrecy? Historical and historiographical remarks. In *The British Journal for the History of Science. Special Issue: States of Secrecy*, volume 45/2, pages 165–188.

René Zandbergen and Rafał T. Prinke. 2016. The Voynich ms in Rudolfine Prague. In Ivo Purš and Vladimír Karpenko, editors, *Alchemy and Rudolf II. Exploring the Secrets of Nature in Central Europe in the 16th and 17th centuries*, pages 279–314, Prag. Artefactum.

[6] According to the `DCODE` online tool, the Friedman index of coincidence is: 0.04506 which would indicate a probable key of length of 2,3, 4, 6, 7, 13, 19 or 24 (`https://www.dcode.fr/index-coincidence`). Since the ciphertext isn't very long, this test cannot be expected to be particularly successful. However, it is to be expected from the cryptographical habits of that time period that the key can indeed be found within the same notebook or even the document itself. A quick survey of likely candidate words ('monas', 'medulla', 'adam', 'ripley', 'riply' and so forth) did not yield any results.

Deciphering a Letter to Louis XIV from his Ambassador to the Dutch Republic, le Comte d'Avaux, 1684

George Lasry
The CrypTool Team
george.lasry@cryptool.org

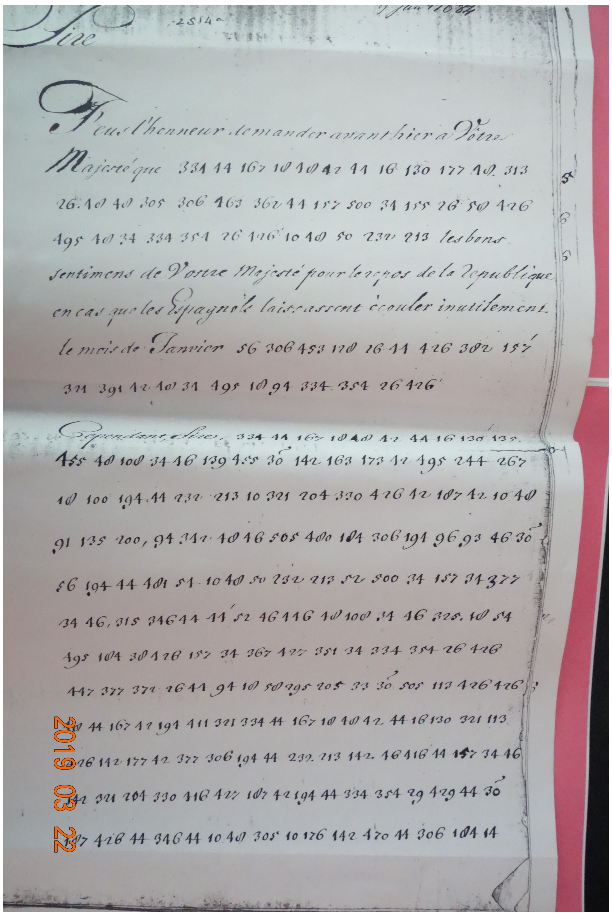
Abstract

The Dutch Royal Archives (Koninklijk Huisarchief - KHA) at The Hague holds a number of enciphered letters written by French diplomats in Holland in the 17th and 18th centuries, including a letter, from January 9, 1684, from Jean-Antoine de Mesmes (1640 – 1709), Comte d'Avaux, the French ambassador at The Hague from 1678 to 1689, to Louis XIV, King of France from 1643 to 1715.

In this article, we show how we deciphered the letter, and identified the historical plaintext as a letter intercepted and deciphered by the Spanish authorities in the Southern Netherlands. The letter was also published by the Prince of Orange, to expose d'Avaux secret contacts with deputies of the city of Amsterdam. D'Avaux claimed that the decryption was intentionally modified to harm France's image, but the modern decipherment demonstrates that the historical decryption was in fact fully accurate.

1 The Document

The document is held in the Dutch Royal Archives, Koninklijk Huisarchief (KHA), under reference Prins Willem III, inv.nr. XIII-I (d'Avaux, 1684b). The first page is shown in Figure 1. The document contains a mix of cleartext and encoded parts. The cleartext indicates that it is dated January 9, 1684, and that it was sent by the Comte d'Avaux to Louis XIV ("Votre Majeste"). The digit codes are separated by spaces, although the separation is not always clear. There are some punctuation marks, like a comma. On the top of some digits, an accent appears, or a Tilda sign. Their meaning could be identified only after deciphering most parts of the ciphertext. The codes contain either two digits (e.g., 34, 14) or three digits (e.g., 295, 505).



Sire -2514- 9 Jan 1684

J'eus l'honneur de mander avant hier a Votre Majesté que 334 44 167 10 10 42 44 16 130 177 10. 313 26. 10 40 305 306 163 362 44 157 500 34 155 26 50 426 495 10 34 334 354 26 126 10 40 50 232 213 les bons sentimens de Votre Majesté pour le repos de la Republique en cas que les Espagnols laissassent écouler inutilement le mois de Janvier 56 306 453 120 26 44 426 302 157 321 391 42 10 34 495 10 94 334 354 26 426

Cependant, Sire, 334 44 167 10 40 42 44 16 130 135. 455 40 100 34 46 139 455 30 142 163 173 42 495 244 267 10 100 194 44 232 213 10 321 204 330 426 42 107 42 10 40 91 135 200, 94 342 40 46 505 400 104 306 194 96 93 46 30 56 194 44 401 54 10 40 50 232 213 52 500 34 157 34 377 34 46, 315 346 44 44 52 46 446 40 100 34 46 325. 10 54 495 104 30 426 157 34 367 427 351 34 334 354 26 426 447 377 372 26 44 94 10 50 295 205 33 30 505 113 426 426 40 44 167 42 194 411 321 334 44 167 10 40 42 44 16 130 321 113 426 142 177 42 377 306 194 44 232 213 142 46 416 44 157 34 46 342 321 204 330 416 427 107 42 194 44 334 354 29 429 44 30 107 426 44 346 44 10 40 305 10 126 142 470 44 306 104 14

2019 03 22

Figure 1: First Page of the Encrypted Letter (d'Avaux, 1684b)

2 Deciphering the Letter

Initially, an attempt was made to decipher the codes using computerized techniques for homophonic ciphers, described in (Lasry et al., 2020). The main challenge was that there were too many distinct codes, and the automated algorithms did not produce any valid solution.

Next, it was hypothesized that the two-digit codes might represent single letters, while the three-digit codes represent full words, names, or places. Since there are only short continuous subsequences that contain only two-digit codes, the computerized algorithm could not produce anything useful.

Next, an attempt was made to find the original key used to encipher the text, in archives. The DECODE Database contains hundreds of such original keys, but very few of them are from France, and no original key could be found that deciphers the letter (Megyesi et al., 2020).

The solution came from another direction. A year before, the author of this article had been contacted by a scholar who was studying other letters from the Comte d'Avaux, sent in 1688, also from Holland. Some of the encoded letters from 1688 had plaintext inscribed near the codes and help was requested to recover the key. The author was able to recover the key from those plaintext segments. It turned out to be the key for another French diplomatic document, recovered historically via codebreaking by John Wallis, the famous British mathematician and codebreaker (1616-1703). Wallis published his decipherment and the key in Opera Mathematica (Wallis, 1972). The author was able to recover additional codes. As shown in Figure 3, the two-digit codes mainly represent single letters. It can be seen that this is a monoalphabetic code (no homophones), with a nomenclature, and that the two-digit codes for the letters are in alphabetical order. For example, A = 20, B = 22, C = 24. Other two-digit codes represent common propositions (CE = 23, DANS = 33). The three-digits codes, shown in Figure 4, are not in a alphabetical order, but it can be seen that there is some order when looking at individual columns. For example, the syllables TA, TE, TI, TO, and TA are encoded as 322, 332, 342, 352, and 362, respectively. This pattern must have been useful to Wallis, when he broke the code.

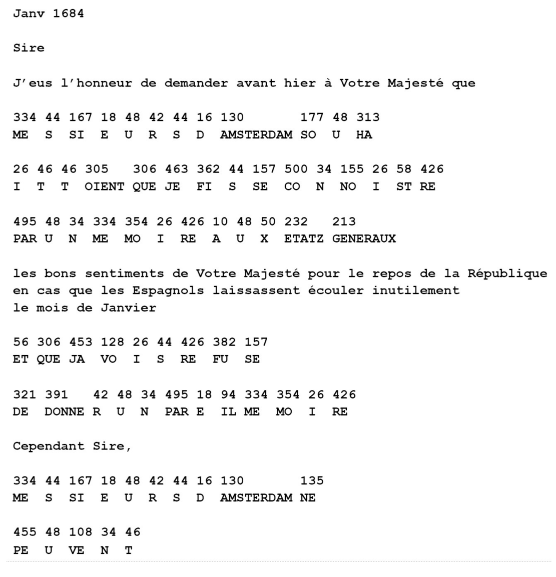
First, the author tried to decipher the letter from Comte d'Avaux's from 1684, using this code from 1688, without success.

Then, the author made a hypothesis that turned out to be correct. Since the code for the 1688 letter is quite simple, with ordered patterns, the code for the 1684 letter might also be a similar simple code, or even simpler, as it predates the 1688 code. One possibility considered was that instead of A, B, C, being 20, 22, 24 as in the 1688 code, the letters of the alphabet start from 10, that is, A = 10, B = 12, C = 14, as shown in Figure 5. Under this hypothesis, fragments of the ciphertext which consist of subsequences of two-digit codes were deciphered, and produced meaningful fragments of plaintext, validating the hypothesis.

Next, the author took advantage of those already-decrypted segments and of the fact that syllables in the 1684 code were also likely to be ordered in columns, to recover several syllables. Next, he was able to recover the codes for all the syllables, and for a number of words, places, and names, as shown in Figure 6, and as a result, to decipher almost all the original ciphertext.

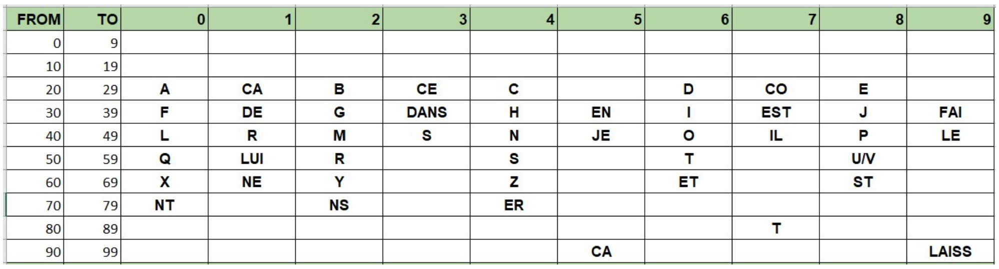
With a partial plaintext at hand, the author conducted a search for the historical context, and hopefully, the original plaintext, in other sources. With some effort the author identified the letter as a famous letter from the Comte d'Avaux, from January 9, 1684 (d'Avaux, 1684c). Based on the full plaintext, additional code entries could be recovered. The Tilda signs turned out to indicate digits to be deleted.

A sample of deciphered letter is shown in Figure 2.



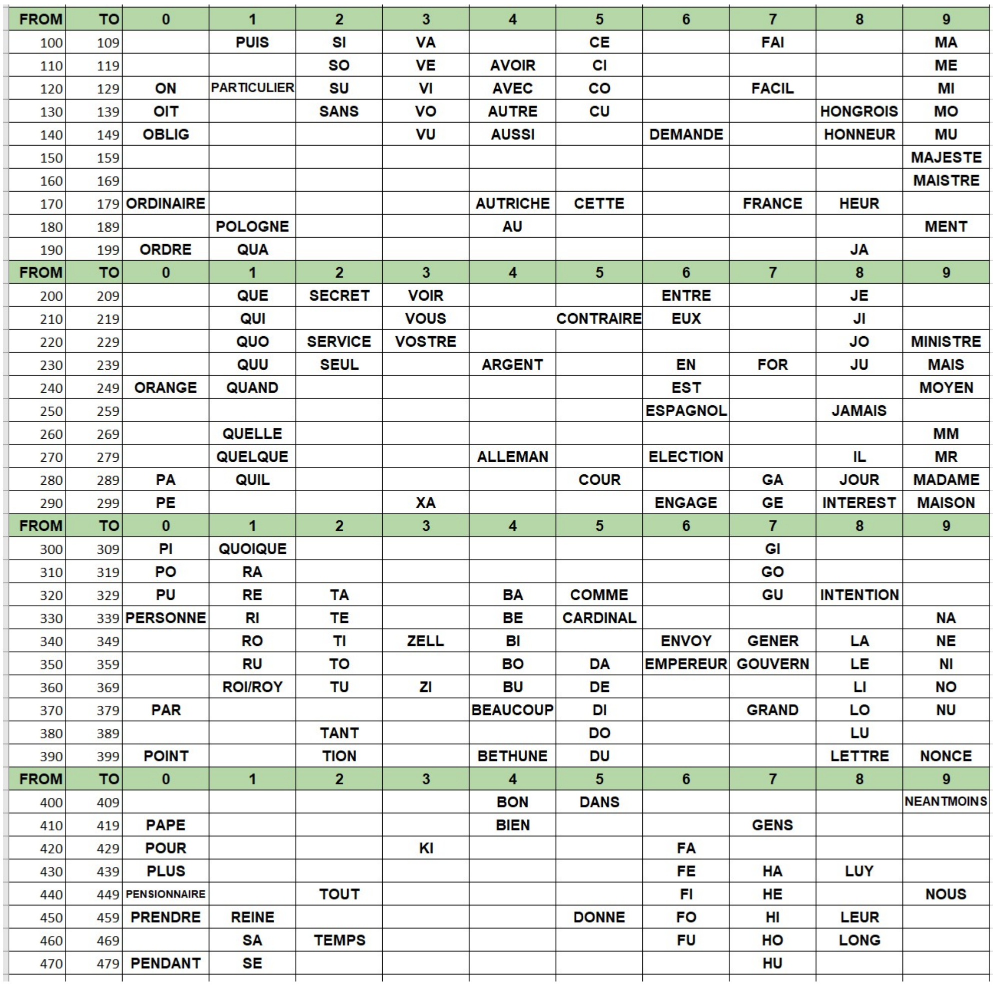
```
Janv 1684

Sire

J'eus l'honneur de demander avant hier à Votre Majesté que

334 44 167 18 48 42 44 16 130       177 48 313
ME  S  SI  E  U  R  S  D  AMSTERDAM SO  U  HA

26 46 46 305   306 463 362 44 157 500 34 155 26 58 426
I  T  T  OIENT QUE JE  FI  S  SE  CO  N  NO  I  ST RE

495 48 34 334 354 26 426 10 48 50 232   213
PAR U  N  ME  MO  I  RE  A  U  X  ETATZ GENERAUX

les bons sentiments de Votre Majesté pour le repos de la République
en cas que les Espagnols laissassent écouler inutilement
le mois de Janvier

56 306 453 128 26 44 426 382 157
ET QUE JA  VO  I  S  RE  FU  SE

321 391   42 48 34 495 18 94 334 354 26 426
DE  DONNE R  U  N  PAR E  IL ME  MO  I  RE

Cependant Sire,

334 44 167 18 48 42 44 16 130       135
ME  S  SI  E  U  R  S  D  AMSTERDAM NE

455 48 108 34 46
PE  U  VE  N  T
```

Figure 2: Sample Decryption – Codes, Decrypted Plaintext, and Cleartext



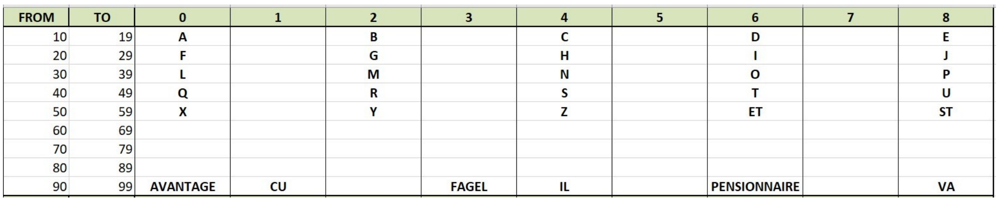
FROM	TO	0	1	2	3	4	5	6	7	8	9
0	9										
10	19										
20	29	A	CA	B	CE	C		D	CO	E	
30	39	F	DE	G	DANS	H	EN	I	EST	J	FAI
40	49	L	R	M	S	N	JE	O	IL	P	LE
50	59	Q	LUI	R		S		T		U/V	
60	69	X	NE	Y		Z		ET		ST	
70	79	NT		NS		ER					
80	89								T		
90	99						CA				LAISS

Figure 3: Similar Code Recovered by John Wallis and used by d’Avaux in 1688 – 20 to 99



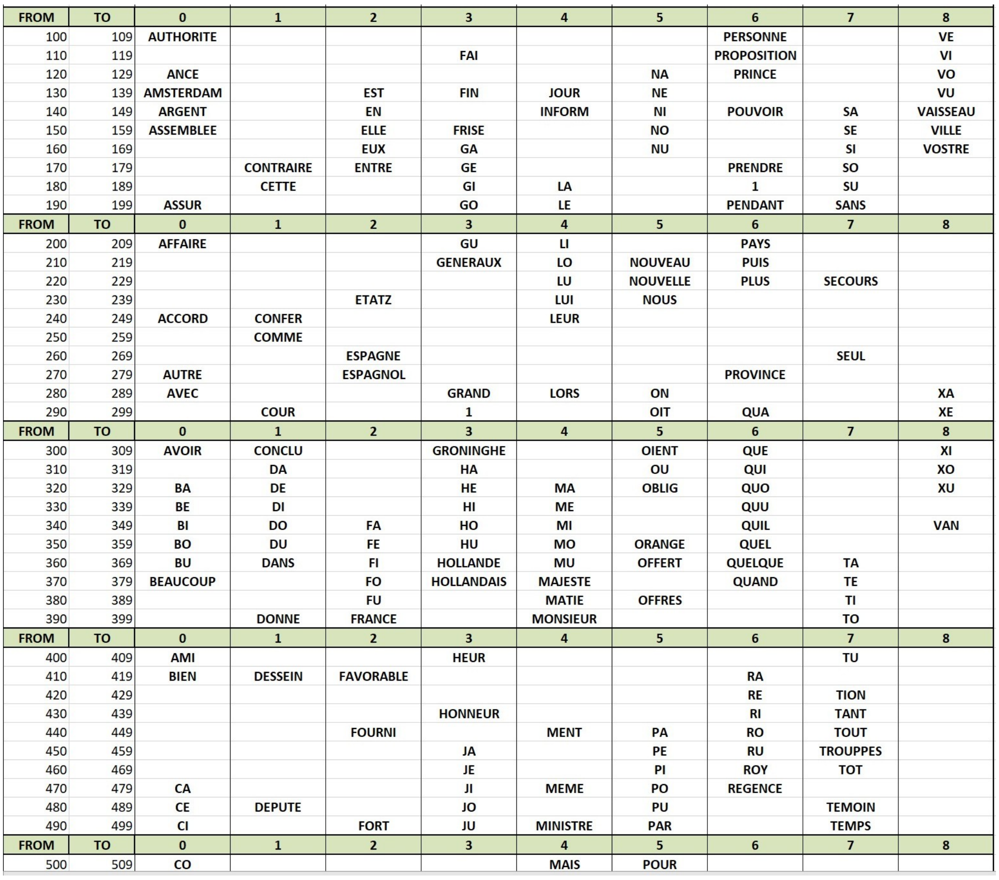
FROM	TO	0	1	2	3	4	5	6	7	8	9
100	109		PUIS	SI	VA		CE		FAI		MA
110	119			SO	VE	AVOIR	CI				ME
120	129	ON	PARTICULIER	SU	VI	AVEC	CO		FACIL		MI
130	139	OIT		SANS	VO	AUTRE	CU			HONGROIS	MO
140	149	OBLIG			VU	AUSSI		DEMANDE		HONNEUR	MU
150	159										MAJESTE
160	169										MAISTRE
170	179	ORDINAIRE				AUTRICHE	CETTE		FRANCE	HEUR	
180	189		POLOGNE			AU					MENT
190	199	ORDRE	QUA							JA	
FROM	TO	0	1	2	3	4	5	6	7	8	9
200	209		QUE	SECRET	VOIR			ENTRE		JE	
210	219		QUI		VOUS		CONTRAIRE	EUX		JI	
220	229		QUO	SERVICE	VOSTRE					JO	MINISTRE
230	239		QUU	SEUL		ARGENT		EN	FOR	JU	MAIS
240	249	ORANGE	QUAND					EST			MOYEN
250	259							ESPAGNOL		JAMAIS	
260	269		QUELLE								MM
270	279		QUELQUE			ALLEMAN		ELECTION		IL	MR
280	289	PA	QUIL				COUR		GA	JOUR	MADAME
290	299	PE			XA			ENGAGE	GE	INTEREST	MAISON
FROM	TO	0	1	2	3	4	5	6	7	8	9
300	309	PI	QUOIQUE						GI		
310	319	PO	RA						GO		
320	329	PU	RE	TA		BA	COMME		GU	INTENTION	
330	339	PERSONNE	RI	TE		BE	CARDINAL				NA
340	349		RO	TI	ZELL	BI		ENVOY	GENER	LA	NE
350	359		RU	TO		BO	DA	EMPEREUR	GOUVERN	LE	NI
360	369		ROI/ROY	TU	ZI	BU	DE			LI	NO
370	379	PAR				BEAUCOUP	DI		GRAND	LO	NU
380	389			TANT			DO			LU	
390	399	POINT		TION		BETHUNE	DU			LETTRE	NONCE
FROM	TO	0	1	2	3	4	5	6	7	8	9
400	409					BON	DANS				NEANTMOINS
410	419	PAPE				BIEN			GENS		
420	429	POUR			KI			FA			
430	439	PLUS						FE	HA	LUY	
440	449	PENSIONNAIRE		TOUT				FI	HE		NOUS
450	459	PRENDRE	REINE				DONNE	FO	HI	LEUR	
460	469		SA	TEMPS				FU	HO	LONG	
470	479	PENDANT	SE						HU		

Figure 4: Similar Code Recovered by John Wallis and used by d’Avaux in 1688 – 100 to 479



FROM	TO	0	1	2	3	4	5	6	7	8
10	19	A		B		C		D		E
20	29	F		G		H		I		J
30	39	L		M		N		O		P
40	49	Q		R		S		T		U
50	59	X		Y		Z		ET		ST
60	69									
70	79									
80	89									
90	99	AVANTAGE	CU		FAGEL	IL		PENSIONNAIRE		VA

Figure 5: Code used by d'Avaux in 1684 and Recovered by the Author – 10 to 98

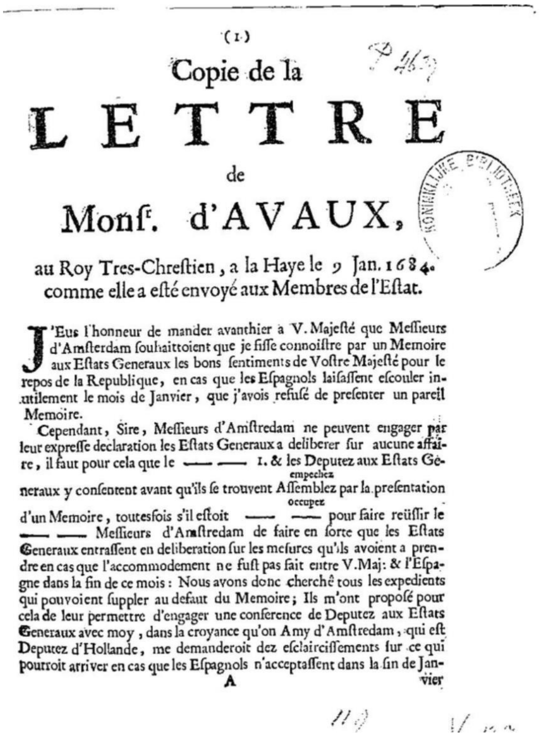


FROM	TO	0	1	2	3	4	5	6	7	8
100	109	AUTHORITE						PERSONNE		VE
110	119				FAI			PROPOSITION		VI
120	129	ANCE					NA	PRINCE		VO
130	139	AMSTERDAM		EST	FIN	JOUR	NE			VU
140	149	ARGENT		EN		INFORM	NI	POUVOIR	SA	VAISSEAU
150	159	ASSEMBLEE		ELLE	FRISE		NO		SE	VILLE
160	169			EUX	GA		NU		SI	VOSTRE
170	179		CONTRAIRE	ENTRE	GE			PRENDRE	SO	
180	189		CETTE		GI	LA		1	SU	
190	199	ASSUR			GO	LE		PENDANT	SANS	
FROM	TO	0	1	2	3	4	5	6	7	8
200	209	AFFAIRE			GU	LI		PAYS		
210	219				GENERAUX	LO	NOUVEAU	PUIS		
220	229					LU	NOUVELLE	PLUS	SECOURS	
230	239			ETATZ		LUI	NOUS			
240	249	ACCORD	CONFER			LEUR				
250	259		COMME							
260	269			ESPAGNE					SEUL	
270	279	AUTRE		ESPAGNOL				PROVINCE		
280	289	AVEC			GRAND	LORS	ON			XA
290	299		COUR		1		OIT	QUA		XE
FROM	TO	0	1	2	3	4	5	6	7	8
300	309	AVOIR	CONCLU		GRONINGHE		OIENT	QUE		XI
310	319		DA		HA		OU	QUI		XO
320	329	BA	DE		HE	MA	OBLIG	QUO		XU
330	339	BE	DI		HI	ME		QUU		
340	349	BI	DO	FA	HO	MI		QUIL		VAN
350	359	BO	DU	FE	HU	MO	ORANGE	QUEL		
360	369	BU	DANS	FI	HOLLANDE	MU	OFFERT	QUELQUE	TA	
370	379	BEAUCOUP		FO	HOLLANDAIS	MAJESTE		QUAND	TE	
380	389			FU		MATIE	OFFRES		TI	
390	399		DONNE	FRANCE		MONSIEUR			TO	
FROM	TO	0	1	2	3	4	5	6	7	8
400	409	AMI			HEUR				TU	
410	419	BIEN	DESSEIN	FAVORABLE				RA		
420	429							RE	TION	
430	439				HONNEUR			RI	TANT	
440	449			FOURNI		MENT	PA	RO	TOUT	
450	459				JA		PE	RU	TROUPPES	
460	469				JE		PI	ROY	TOT	
470	479	CA			JI	MEME	PO	REGENCE		
480	489	CE	DEPUTE		JO		PU		TEMOIN	
490	499	CI		FORT	JU	MINISTRE	PAR		TEMPS	
FROM	TO	0	1	2	3	4	5	6	7	8
500	509	CO				MAIS	POUR			

Figure 6: Code used by d'Avaux in 1684 and Recovered by the Author – 100 to 508

3 The Historical Context and the Contents of the Letter

After it was captured and deciphered in 1684, the letter was published in various pamphlets in several languages. An initial version - with the original French text - is shown in Figure 7 (d'Avaux, 1684c). It can be seen that the decryption is incomplete. An English version was also published in England (d'Avaux, 1684a).



(1)

Copie de la

LETTRE

de

Monſr. d'AVAUX,

au Roy Tres-Chreſtien, a la Haye le 9 Jan. 1684. comme elle a eſté envoyé aux Membres de l'Eſtat.

J'Eus l'honneur de mander avanthier a V. Majeſté que Meſſieurs d'Amſterdam ſouhaittoient que je fiſſe connoiſtre par un Memoire aux Eſtats Generaux les bons ſentiments de Voſtre Majeſté pour le repos de la Republique, en cas que les Eſpagnols laiſaſſent eſcouler inutilement le mois de Janvier, que j'avois refuſé de preſenter un pareil Memoire.

Cependant, Sire, Meſſieurs d'Amſtredam ne peuvent engager par leur expreſſe declaration les Eſtats Generaux a deliberer ſur aucune affaire, il faut pour cela que le —— —— 1. & les Deputez aux Eſtats Generaux y conſentent avant qu'ils ſe trouvent Aſſemblez par la preſentation d'un Memoire, toutesfois s'il eſtoit —— —— pour faire reüſſir le —— —— Meſſieurs d'Amſtredam de faire en ſorte que les Eſtats Generaux entraſſent en deliberation ſur les meſures qu'ils avoient a prendre en cas que l'accommodement ne fuſt pas fait entre V.Maj: & l'Eſpagne dans la fin de ce mois: Nous avons donc cherchê tous les expedients qui pouvoient ſuppler au defaut du Memoire; Ils m'ont propoſé pour cela de leur permettre d'engager une conference de Deputez aux Eſtats Generaux avec moy, dans la croyance qu'on Amy d'Amſtredam, qui eſt Deputez d'Hollande, me demanderoit dez eſclairciſſements ſur ce qui pourroit arriver en cas que les Eſpagnols n'acceptaſſent dans la fin de Janvier

empechez

occupez

A

Figure 7: First Version of the Deciphering of the Intercepted Letter (d'Avaux, 1684c)

Before describing the contents of the letter, an overview of the historical context is given here. The diplomatic aspects of French intervention in Holland in the 17th century are described in detail in (Jones, 1989) and in in (Kurtz, 1928). D'Avaux recorded his own version of the events in (d'Avaux, 1754). A background on Dutch cryptography is given in (De Leeuw, 1999).

The Dutch system of government at the end of the 17th century was complex, offering opportunities that the French ambassador was eager to exploit. Spain and France had just concluded a war in which Spain lost several territories in the Southern Netherlands. To counter French influence, the Prince of Orange wanted to raise new taxes, expand the army, and form new alliances. To oppose those efforts, d'Avaux established working relationships with Williams' most consistent and influential opponents, the Regents of city governments. He openly engaged delegates from the provincial States to the States-General, to block William's initiatives, that were supported by Gaspar Fagel, the Pensionary of Holland in title but Williams' loyal ally. In 1683-84, d'Avaux and his allies were ale to block William's attempt to raise any new levies of men for the army and prevented him from moving forces to the Spanish Netherlands, which were about to be invaded by a French Army.

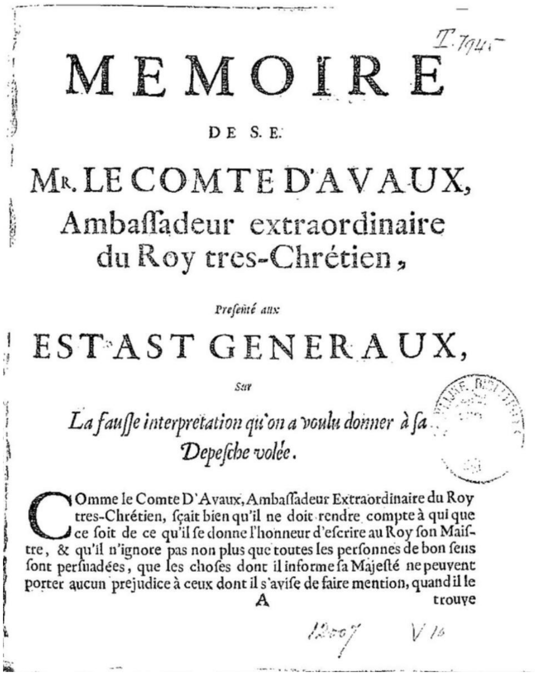
D'Avaux's most important contacts were those whom he called Messieurs d'Amsterdam, the oligarchy of burgomasters and magistrates who controlled city governments as well as banking services providing the Dutch government with loans. D'Avaux also harnessed support from members of the Frisian and Utrecht delegations to the States-General.

In September 1683, William proposed measures to the States-General to deter Louis from making an attack on Luxemburg and to bolster Spanish resistance to an attack there. William had already moved 8000 men into defensive positions in the Spanish Netherlands but he needed to increase the army to 16000 soldiers overall. Louis conducted a policy that applied military pressure on Spain, with diplomatic intervention by d'Avaux. With the delegates from Amsterdam, Friesland and Groningen, Louis XIV wanted to counter William's attempts to come to the assistance of Spain. The deputies argued that war (with France) would ruin trade and that it was not worth risking fishing trade for the sake of assisting the Spanish Netherlands. They attached impossible conditions to the voting of extra money. Even a personal appearance by William in the States failed to persuade the Amsterdam deputies to agree to extra men and money, and left him humiliated.

In this context, on January 9, 1684 d'Avaux wrote the letter to Louis XIV describing his contacts with the Messieurs d'Amsterdam in extensive detail, but without naming them. It emerges from the letter that the republican parties are trying to get concrete assurances from Louis XIV, but d'Avaux is only providing vague ones.

The courier carrying the letter was robbed of his letters just outside Maastricht, a fortified city on the border, by men wearing the uniform of the Dutch garison. The Marquis de Grana, the governor of the Spanish Netherlands, handed over a decrypted version to William, who read it to the States of Holland on February 16. Achieving a sensational effect, he denounced two delegates from Amsterdam, who claimed that they were only trying to protect the city's trading interests.

D'Avaux responded by claiming that the Spanish had deliberately distorted the text for propaganda purposes, with a letter to the States General (shown in Figure 8), with his own version of some of the key sentences in question (d'Avaux, 1684d).



T.794

MEMOIRE

DE S. E.

MR. LE COMTE D'AVAUX,

Ambaſſadeur extraordinaire du Roy tres-Chrétien,

Preſenté aux

ESTAST GENERAUX,

Sur

La fauſſe interpretation qu'on a voulu donner à ſa Depeſche volée.

COmme le Comte D'Avaux, Ambaſſadeur Extraordinaire du Roy tres-Chrétien, ſçait bien qu'il ne doit rendre compte à qui que ce ſoit de ce qu'il ſe donne l'honneur d'eſcrire au Roy ſon Maiſtre, & qu'il n'ignore pas non plus que toutes les perſonnes de bon ſens ſont perſuadées, que les choſes dont il informe ſa Majeſté ne peuvent porter aucun prejudice à ceux dont il s'aviſe de faire mention, quand il le

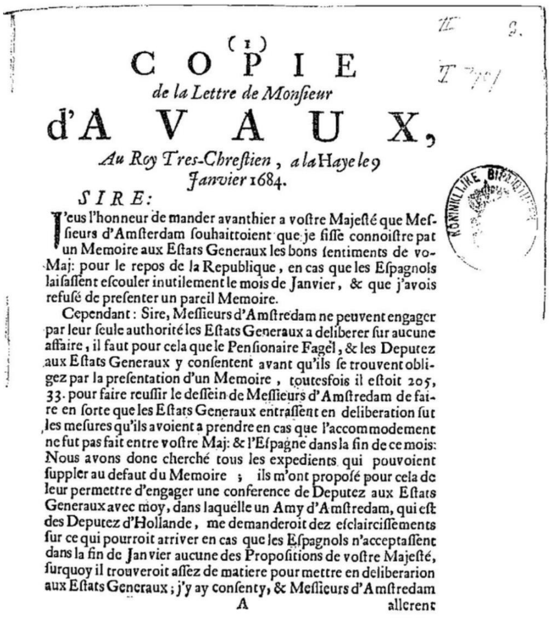
A trouve

12007 V 16

Figure 8: D'Avaux Letter to the States General in Response to the Publication of the Intercepted Letter (d'Avaux, 1684d)

An updated, more complete decryption (d'Avaux, 1684c), shown in Figure 9, further supported the claims of William. The author has compared this version to his own decipherment of the letter. They match almost entirely, refuting d'Avaux's claims that William and the Spanish had intentionally distorted his original text.

In the longer term, the incident had little effect. William was unable to support Spain, who had to accept France's terms for peace. However, attempts by d'Avaux to obtain the dismissal of the Pensionary of Holland failed, and with the expulsion of the Huguenots from France a few years laters, d'Avaux's influence had significantly diminished.



(1)

COPIE

de la Lettre de Monſieur

d'AVAUX,

Au Roy Tres-Chreſtien, a la Haye le 9 Janvier 1684.

SIRE:

J'eus l'honneur de mander avanthier a voſtre Majeſté que Meſſieurs d'Amſterdam ſouhaittoient que je fiſſe connoiſtre pat un Memoire aux Eſtats Generaux les bons ſentiments de vo- Maj: pour le repos de la Republique, en cas que les Eſpagnols laiſaſſent eſcouler inutilement le mois de Janvier, & que j'avois refuſé de preſenter un pareil Memoire.

Cependant: Sire, Meſſieurs d'Amſtredam ne peuvent engager par leur ſeule authorité les Eſtats Generaux a deliberer ſur aucune affaire, il faut pour cela que le Penſionaire Fagel, & les Deputez aux Eſtats Generaux y conſentent avant qu'ils ſe trouvent obligez par la preſentation d'un Memoire, toutesfois il eſtoit 205, 33. pour faire reuſſir le deſſein de Meſſieurs d'Amſtredam de faire en ſorte que les Eſtats Generaux entraſſent en deliberation ſut les meſures qu'ils avoient a prendre en cas que l'accommodement ne fut pas fait entre voſtre Maj: & l'Eſpagne dans la fin de ce mois: Nous avons donc cherché tous les expedients qui pouvoient ſuppler au defaut du Memoire ; ils m'ont propoſé pour cela de leur permettre d'engager une conference de Deputez aux Eſtats Generaux avec moy, dans laquelle un Amy d'Amſtredam, qui eſt des Deputez d'Hollande, me demanderoit dez eſclairciſſements ſur ce qui pourroit arriver en cas que les Eſpagnols n'acceptaſſent dans la fin de Janvier aucune des Propoſitions de voſtre Majeſté, ſurquoy il trouveroit aſſez de matiere pour mettre en deliberarion aux Eſtats Generaux; j'y ay conſenty, & Meſſieurs d'Amſtredam

A allerent

Figure 9: Second Version of the Deciphering of the Intercepted Letter (d'Avaux, 1684c)

4 Conclusion

The modern (re)decipherment of d'Avaux's letter provides some insights on the nature of the French diplomatic codes in the late 17th century. It seems that not only the same code was used in different places, but that the code used by d'Avaux in 1688 was only a minor variation of the code he had been using in 1684. Both codes have a fair amount of structure and alphabetical order, and they are highly insecure. They are even more insecure if they are used over a long period of time. It is even more striking given the fact that the 1684 code was known to be compromised, and a stronger code would have been advisable, rather than another one with minor changes. This finding is surprising, as by this time, the French cryptographers including the famous Rossignol had already introduced much more secure codes such as two-part unordered nomenclators (Kahn, 1996, p. 161).

This letter also exemplified the taking advantage of the capture and decipherment of a diplomatic coded dispatch, for propaganda purposes, despite letting the adversary know that their messages can be deciphered. Unlike another cryptographic propaganda coup, the famous Zimmermann Telegram, the capture and publication of the letter from the Comte d'Avaux did not have a major impact on diplomatic and military events.

Acknowledgments

This work has been supported by the Swedish Research Council, Grant 2018-06074, DECRYPT – Decryption of historical manuscripts.

References

Jean Antoine Comte d'Avaux. 1684a. *An exact copy of a letter from the Count d'Avaux, His Most Christian Majesties ambassador at the Hague dated the 9th of January 1684 and directed to the King his master which was intercepted by the Marquess de Grana governour of the Spanish Netherlands : as also the copies of other three letters relating to the same affair.* Hague, London: Printed by Jacobus Sikeltus, Re-printed for Randall Taylor; Retrieved: Text Creation Partnership, http://name.umdl.umich.edu/A04486.0001.001.

Jean Antoine Comte d'Avaux. 1684b. *Collection: Willem III, inv.nr. XIII-I.* Dutch Royal Archives, Koninklijk Huisarchief.

Jean Antoine Comte d'Avaux. 1684c. *Copie de la lettre de monsieur d'Avaux, au roy tres-chrestien, a la Haye le 9 janvier 1684*. Koninklijke Bibliotheek, pflt 11962, retrieved from Brill Online - https://primarysources.brillonline.com/browse/dutch-pamphlets-online/copie-de-la-lettre-de-monsieur-davaux-au-roy-treschrestien-a-la-haye-le-9-janvier-1684;dutchpamphletskb1kb14034.

Jean Antoine Comte d'Avaux. 1684d. *Memoire de S.E. Mr. le comte d'Avaux, ambassadeur extraordinaire du roy tres-chrétien, presenté aux Estast Generaux, sur la fausse interpretation qu'on a voulu donner a sa depesche volée*. Koninklijke Bibliotheek, pflt 12007, retrieved from Brill Online - https://primarysources.brillonline.com/browse/dutch-pamphlets-online/memoire-de-se-msuprsup-le-comte-davaux-ambassadeur-extraordinaire-du-roy-treschretien-presente-aux-estast-generaux-sur-la-fausse-interpretation-quon-a-voulu-donner-a-sa-depesche-volee;dutchpamphletskb1kb14080.

Jean Antoine Comte d'Avaux. 1754. *Negociations du comte d'Avaux en Hollande depuis 1679 jusqu'en 1688 (publies par Edme Mallet.)*, volume 2. Durand.

Karl De Leeuw. 1999. The Black Chamber in the Dutch Republic during the War of the Spanish Succession and its aftermath, 1707-1715. *Historical Journal*, pages 133–156.

James Jones. 1989. *'French intervention in English and Dutch politics, 1677–88', in Black Jeremy (ed.), Knights Errant and True Englishmen: British Foreign Policy, 1660–1800*. Edinburgh, John Donald Publishers.

David Kahn. 1996. *The Codebreakers: The comprehensive history of secret communication from ancient times to the internet*. Simon and Schuster.

Gerdina Hendrika Kurtz. 1928. *Willem III en Amsterdam, 1683-1685*. Utrecht, Kemink en zoon.

George Lasry, Beáta Megyesi, and Nils Kopal. 2020. Deciphering papal ciphers from the 16th to the 18th Century. *Cryptologia*, pages 1–62.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. 2020. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 44(6):545–559.

John Wallis. 1972. *Johannis Wallis Opera Mathematica*, volume 3. E Theatro Sheldoniani.

Modern Cryptanalysis of Schlüsselgerät 41

George Lasry
The CrypTool Team
george.lasry@cryptool.org

Abstract

The Schlüsselgerät 41 was an highly secure encryption machine developed by Fritz Menzer and used from 1944 by the Abwehr. Bletchley Park could not decipher its traffic. In this article, we provide a functional description of the SG-41 and present a novel cryptanalytic method to recover the key settings from ciphertext and known-plaintext. This attack requires extensive computing power, a testimony to the resilience of the SG-41 even against modern cryptanalysis. We also present an alternative method, based on acoustic cryptanalysis, which allows for the recovery of the key settings in minutes.

1 Overview

This article is structured as follows: In Section 2, a brief overview of the history of the SG-41 is given and a functional description. Section 3 describes cryptanalysis attempts by Bletchley Park and the US against the SG-41. In Section 4, we describe a novel known-plaintext attack that is feasible but requires extensive computing power, and in Section 5, a highly-efficient side-channel attack that relies on acoustic analysis of the device. Finally, in Section 6, we assess the security of the SG-41 compared to other encryption machines of the 1940s.

2 The SG-41 – Introduction

The SG-41 was an encryption machine introduced by Fritz Menzer, Regierungsoberinspektor of OKW/Chi, the cryptographic branch of the Wehrmacht. While inspired by the Hagelin pin-and-lug devices, the design of the SG-41 incorporated several novel features that significantly enhanced its security. Logistical reasons prevented its production in large volumes, and it was only deployed in 1944 on a few Abwehr networks. Bletchley Park could not decipher its traffic unless multiple messages were sent in-depth. Until recently, little was known about the inner functioning of the SG-41. Several historical documents have been declassified that provide extensive details about its functioning. A small number of SG-41 have survived, and some have been restored.

In this section, we provide an overview of the history of the SG-41, as well as a functional description, and an analysis of its keyspace size.

2.1 Fritz Menzer and the SG-41

Fritz Menzer (1908–2005) was the Government Inspector (Regierungsoberinspektor) of OKW/Chi, the cryptographic arm of the Wehrmacht, and later, Admiral Canaris, the head of the Abwehr charged him with ensuring the security of the organization's communications. Menzer designed and led the development of several cipher devices, methods, and procedures, some of which created some difficulties for British and U.S. codebreakers. In a post-war NSA publication, Menzer is described as "Cryptologic Inventor Extraordinaire", and the peak of his achievements, however, is most probably the invention of the SG-41, shown in Figure 1 (Mowry, 1983).

Having previously worked on the cryptanalysis of the Enigma and the Hagelin C-36, Menzer understood their weaknesses. While much of the SG-41 borrows from the Hagelin pin-and-lug design, Menzer introduced some features that provided enhanced security (Mowry, 1983). Boris Hagelin later complained to William Friedman that the Germans had stolen his design. He had obtained one of the SG-41 machines, wrongly calling it C-41 (Friedman, 1955).[1]

The SG-41 was designed with a keyboard and a strip printer to speed up the process of enciphering

[1] It is also called C-41 in some U.S. documents (Agency, 1947).

and deciphering (unlike the Enigma that required at least two operators, one to type into the keyboard, and another one to write down the lamps activated).

The army ordered 11,000 units in 1942, and a prototype was presented in 1943, but by the end of the war, only 1000–1500 had been produced by the firm Wanderer-Werke in Chemnitz. The challenges of wartime production and the lack of material may have prevented its production in higher volumes. In addition, the device was considered too heavy - over 13 kg - to be used at the frontlines (Dahlke, 2018; Mowry, 1983). Near the end of 1944, it was deployed on at least three Abwehr links, between Berlin, Bordeaux in Southern France, Northern Italy, and Vienna, replacing the Enigma G machines (Batey et al., 1945).

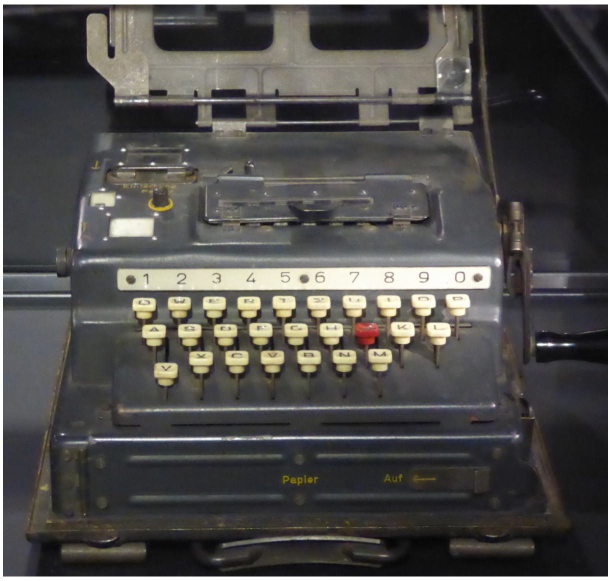


Figure 1: The SG-41

2.2 Functional Description of the SG-41

Until recently, little was known about the internal mechanism of the SG-41 (Dahlke, 2018; Museum, 2020b; Schmeh, 2004). Recently declassified U.S. and British documents, and in particular, a wartime report from Bletchley Park, provide enough details to fully reconstruct the functioning of the SG-41 (Batey et al., 1945; Mowry, 1983; Mowry, 1989; Mowry, 2003). While very few devices have survived, most in bad condition, some units are in the hands of museum curators and crypto collectors, who were able to analyze the physical/mechanical design of the SG-41. At least one machine has been restored so that it is fully functional (Historica, 2019; Dahlke, 2018).

While the SG-41 is described in several documents (Mowry, 1989; Mowry, 2003; Mowry, 1983; WDGAS-14, 1946), those descriptions are incomplete, and sometimes conflicting. The most reliable historical source describing the SG-41 is a G.C. & C.S. report titled *Secret Service SIGINT Volume II - Cryptographic Systems and Their Solutions - Machine Cyphers* written by Keith Batey, Mavis Batey, Margaret Rock, and Peter Twinn in 1945. The authors were part of ISK - Intelligence Services Knox (headed by Dilly Knox before his death in 1943) and were responsible for analyzing Abwehr traffic with its agents and offices worldwide. While most of the report is about the cryptanalysis of the Abwehr Enigmas, against which ISK had considerable success, the last seven pages of the report are dedicated to a detailed functional description of the SG-41 and to the mostly unsuccessful attempts by ISK to decipher its traffic (Batey et al., 1945).

The focus in this section is on the logical and functional aspects of the SG-41, rather than on its physical design and implementation. Figure 2 shows a functional diagram of the SG-41. The SG-41 enciphers symbols of the A-Z alphabet into symbols of the same alphabet. To encipher, the operator presses a plaintext symbol on the keyboard (spaces are represented by the symbol J). The plaintext symbol is encrypted, and the resulting ciphertext symbol is printed on a paper strip (together with the plaintext symbol). The decryption process is similar: The operator presses a ciphertext symbol on the keyboard. The encryption process, which is reciprocal, converts back the ciphertext symbol into a plaintext symbol printed on the paper strip (together with the ciphertext symbol).

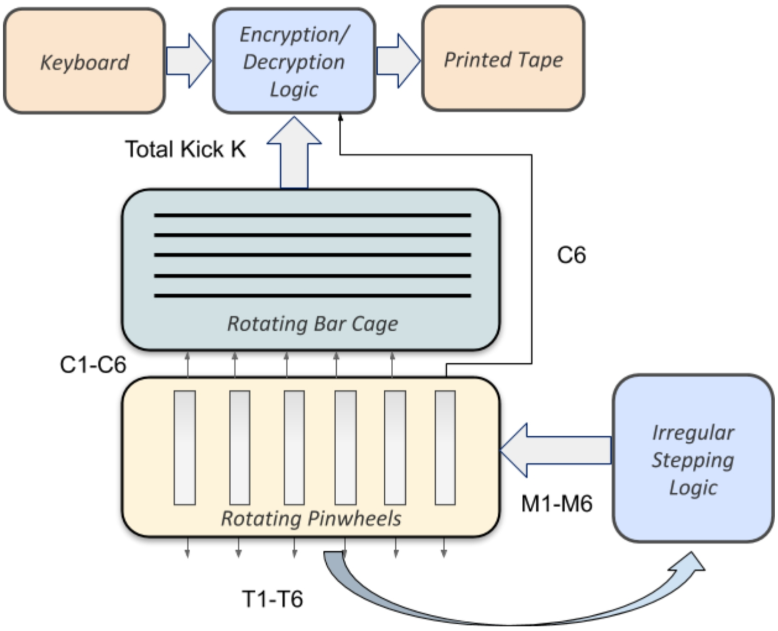


Figure 2: The SG-41 - Functional Diagram

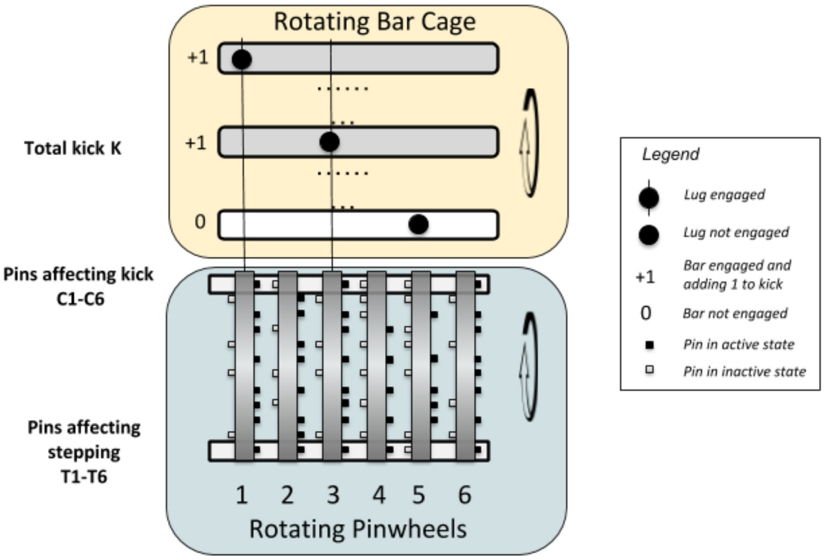


Figure 3: The SG-41 - Functional Diagram - Pinwheels and Bar Cage

The encryption logic is governed by the outputs of a *rotating bar cage* with 25 bars, as well as of a set of six rotating *pinwheels*, as shown in Figure 3. The pinwheels are numbered 1 to 6, from left to right, and have 25, 25, 23, 23, 24, and 24 pins each, respectively. Each pin can be set to an *active* or an *inactive* state. Each wheel from 1 to 5 affects one or more bars. The pin currently in front of the cage determines whether the bar is engaged or not. Those pins are denoted as C1 to C5, for wheels 1 to 5, respectively. C6 does not affect the cage bars, but it affects encryption as described in Section 2.4.

Each bar has a fixed lug positioned in front of one of the wheels 1 to 5. In front of wheels 1, 2, 3, 4, and 5, there are 1, 2, 4, 8, and 10 bars with such a fixed lug, respectively. The bar cage performs a full rotation during the encryption of a single symbol. When a bar has its lug against an active pin, it is engaged and it adds one to a total additive kick, denoted as K (its function is described in Section 2.4). If a bar is not engaged, it does not add to K. Therefore, wheels 1 to 5 may add 1, 2, 4, 8, and 10, to K, respectively. When all the bars are against active pins, and thus, they all are engaged, the total kick is $1+2+4+8+10=25$.

So far, the mechanism of the first five wheels and the bar cage is very similar to the Hagelin C-35, which also had bars with fixed lugs, and five wheels that affect kick (Museum, 2020a). But the SG-41 introduces two features which greatly enhance its cryptographic security: irregular stepping (see Section 2.3), and complementary kick (see Section 2.4). It also features a fixed substitution (see Section 2.5), which has no significant effect on cryptographic security.

2.3 Irregular Stepping

The SG-41 features an irregular wheel stepping mechanism. The stepping of each wheel is governed by the state of pins in other wheels (see Figure 2). The pins of wheels 1 to 6 that affect other wheels' stepping are denoted as T1 to T6. Those pins are at a certain distance on the wheel from the pins that control the bars (C1-C6). With wheel 1, if pin 1 currently affects the bar cage (it generates C1), at the same time, pin 14 affects the stepping of another wheel (generating T1). Similarly, if the wheel has advanced one step, pin 2 generates C1 and pin 15 generates T1, and so forth. The same applies to C2 and T2 for wheel 2, and similarly to the other wheels.

A full *encryption cycle*, in which a plaintext symbol is encrypted (or a ciphertext symbol is decrypted), and the wheels advance, consists of three stages, one of which is optional (Batey et al., 1945):

1. An optional *pre-encryption stepping stage* that occurs before encryption, only if T6 was active (at the beginning of the cycle).
2. The *encryption stage*.
3. A *post-encryption stepping stage* that always occurs after encryption.

The two stepping stages are identical. Each stepping stage consists of two phases, as follows:

- Wheels 2 to 6 step if the pin affecting stepping on the wheel on its left was active (at the beginning of the stage). For example, if T3 was active, wheel 4 steps.
- All the wheels (1 to 6) step.

This mechanism creates a circular interdependence between the wheels, as illustrated in Figure 4. This circular interdependence means that any wheel may affect the stepping of any other wheel, directly or indirectly, and that there is no way to know how the wheels step without first determining the pin settings of all wheels.

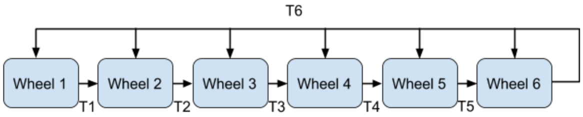


Figure 4: Wheel Stepping – Circular Interdependency

This two-stage mechanism ensures that every wheel will step at least once, and no more than four times, per encryption cycle. Depending on T6 at the beginning of the encryption cycle, either:

- Wheels 2 to 6 step two to four times, and wheel 1 steps twice, or
- Wheels 2 to 6 step once or twice, and wheel 1 steps once.

Wheel 2 has 25 pins and completes a full rotation after 7 to 13 cycles. Wheels 3, 4, 5, and 6 (with 24 or 23 pins) complete a full rotation after 6 to 12 cycles.

2.4 Complementary Kick

The second security enhancement has to do with how K affects encryption. In the regular Hagelin C machines, encryption is according to the Beaufort reciprocal formula (p is the plaintext symbol, c the resulting ciphertext symbol, and K the total kick) (Lasry et al., 2016):

$$c = (25 - p + K) \mod 26 \qquad (1)$$

Decryption works similarly (modulo 26):

$$p = 25 - c + K = 25 - (25 - p + K) + K = p \quad (2)$$

The SG-41 introduces a *complementary feature*, that works as follows (modulo 26):

- If C6 is inactive: $c = 25 - p + K$
- If C6 is active: $c = 25 - p + (25 - K)$, effectively complementing the kick.

The complementary feature complicates the relationship between the effective kick K_e, computed as $K_e = c + p - 25$, and the state of C1-C5, the pins that affect encryption. Without the complementary feature, if $K_e = 1$, for example, we could clearly establish that C1 is active, and C2-C6 are inactive. But with the complementary feature, $K_e = 1$ could also be obtained if C1 is inactive, C6 is active, and C2-C5 are active. In general, there are two possible C1-C6 options for any K_e between 0 and 9, and from 16 to 25 (instead of one without the complementary feature). Similarly, there are four possible C1-C6 options for any K_e between 10 and 15 (instead of two without the complementary feature), as illustrated in Table 1.

2.5 Fixed Substitution

It should be noted that SG-41 first applies a substitution alphabet (denoted as S) to the input symbol, and its inverse to the output symbol, after encryption. The substitution alphabet is as follows (the letter on the top row maps to the letter on the bottom row, e.g., A maps to P, B maps to T, etc...):

```
ABCDEFGHIJKLMNOPQRSTUVWXYZ
PTOIUHVRFWACXQSEZKGMYJBNLD
```

So therefore the full encryption formula (modulo 26) is as follows:

- If C6 is inactive: $c = S^{-1}(25 - S(p) + K)$
- If C6 is active: $c = S^{-1}(25 - S(p) + (25 - K))$

Effective Kick K_e	Active C1-C6
0	none active C1+C2+C3+C4+C5+C6
1	C1 C2+C3+C4+C5+C6
2	C2 C1+C3+C4+C5+C6
3	C1+C2 C3+C4+C5+C6
4	C3 C1+C2+C4+C5+C6
5	C1+C3 C2+C4+C5+C6
6	C2+C3 C1+C4+C5+C6
7	C1+C2+C3 C4+C5+C6
8	C4 C1+C2+C3+C5+C6
9	C1+C4 C2+C3+C5+C6
10	C5 C2+C4 C1+C3+C5+C6 C1+C2+C3+C4+C6
11	C1+C5 C1+C2+C4 C3+C5+C6 C2+C3+C4+C6
12	C3+C4 C2+C5 C1+C3+C4+C6 C1+C2+C5+C6
13	C1+C3+C4 C1+C2+C5 C3+C4+C6 C2+C5+C6
14	C3+C5 C2+C3+C4 C1+C5+C6 C1+C2+C4+C6
15	C1+C3+C5 C1+C2+C3+C4 C5+C6 C2+C4+C6
16	C2+C3+C5 C1+C4+C6
17	C1+C2+C3+C5 C4+C6
18	C4+C5 C1+C2+C3+C6
19	C1+C4+C5 C2+C3+C6
20	C2+C4+C5 C1+C3+C6
21	C1+C2+C4+C5 C3+C6
22	C3+C4+C5 C1+C2+C6
23	C1+C3+C4+C5 C2+C6
24	C2+C3+C4+C5 C1+C6
25	C1+C2+C3+C4+C5 C6

Table 1: Options for Effective Kick K_e

In a still-classified TICOM report, Menzer claims that this alphabet was designed to flatten the frequency counts in the ciphertext (Mowry, 1983; I-72, 1945). While it is true that the effective kick stream is not randomly distributed (the values 10 to 15 are more likely to appear), it is not clear to what extent this additional substitution enhances the cryptographic security of the SG-41.

2.6 Analysis of the Keyspace

Any pin on a pinwheel may be set to be either active or inactive. There are $25+25+23+23+24+24=144$ pins, therefore the size of the theoretical keyspace is 2^{144}. In practice, operational procedures on how to set the pins would probably have reduced this number. Unfortunately, no documents have survived that describe the operational procedures of the SG-41.

3 Historical Cryptanalysis of the SG-41

In this section, we present historical attempts at the Cryptanalysis of the SG-41.

3.1 Attacks on Depths

(Batey et al., 1945) describes how the mechanism of the SG-41 was reconstructed from depths by Bletchley Park, but could only be fully understood after a unit was captured in 1945. The SG-41, similarly to other Hagelin cipher machines, is still susceptible to attacks on messages in-depth, that is, encrypted with the same key settings. If we have two ciphertexts originally enciphered with the same key settings, and we look at the ciphertext symbols c_1 and c_2 at the same position in the message, then, assuming that C6 is inactive at that encryption cycle, we obtain (modulo 26):

$$c_1 = S^{-1}(25 - S(p_1) + K) \tag{3}$$

$$c_2 = S^{-1}(25 - S(p_2) + K) \tag{4}$$

where p_1 and p_2 are the corresponding unknown plaintext symbols.

After applying S (the known fixed substitution) on both sides, we obtain:

$$S(c_1) = 25 - S(p_1) + K \tag{5}$$

$$S(c_2) = 25 - S(p_2) + K \tag{6}$$

and therefore:

$$S(p_1) + S(c_1) = S(p_2) + S(c_2) \tag{7}$$

It can easily be seen that Equation 7 also applies if C6 is active and $c = S^{-1}(25 - S(p) + (25 - K))$.

Since S, c_1, and c_2 are known, if we can guess p_1, we obtain p_2.

The same techniques historically used for recovering depths (e.g., from Hagelin ciphertexts) can be applied here (Lasry et al., 2018).

Depths are available if operational discipline is poor, and the same key and starting positions are reused for different messages. From the historical reports, it can be understood that the same key settings (the active and inactive pins on the wheels) were used for a certain period of time. To avoid sending messages in-depth, the operator would first change the starting positions of the wheels for each message and securely communicate to the other party those starting positions, using concealed indicators (Batey et al., 1945).

3.2 Conditions for a Long Period

Another way depths may occur is if the machine repeats the keystream (the series of K_e) after a relatively short period, which we denote as *motion period*. In theory, because of irregular stepping, this should happen only after 25 * 25 * 23 * 23 * 24 * 24 = 190,440,000 stepping stages. In practice, the longest achievable period will be shorter than that, as there might be one or two stepping stages per encryption cycle.

An historical report by the predecessor to the NSA, the Army Security Agency, analyses the preconditions for a full motion period. According to the report, the Germans came up with a list of necessary and sufficient conditions to ensure a maximum period. The keys were selected to always comply with those conditions (Agency, 1947; I-72, 1945). We denote the number of active pins on the wheels as N_1 to N_6 and list the conditions:

1. N_1 is not divisible by 5
2. $N_2 \neq 21$
3. $N_3 \neq 0$ and $N_3 \neq 23$
4. $N_4 \neq 1 \mod 2$ and $N_4 \neq 1 \mod 3$
5. N_5 is neither divisible by 2 nor by 3

In (Agency, 1947), an example is given demonstrating that by violating only one of the five conditions, it is possible to obtain a motion period with only 70 stepping stages. Generally, if the conditions are not systematically followed, the vast majority of (randomly-selected) settings would result in periods shorter than the maximum period. It can be seen that condition 1 leaves only 0.8 of the possible wheel settings, condition 4 one-third of those, and condition 5 leaves one-third of the latter, so that they may generate a complete period. With just those three conditions, we are left with 0.8/9, which is less than one-tenth. Therefore, more than 90% of randomly selected settings would be sub-optimal, and even if the motion period is longer than one million, on a day with heavy traffic, with tens of thousands of symbols intercepted, overlaps (partial depths) are likely to occur.

The Bletchley Park report, and a report written by Walter Fried, the U.S. liaison officer in Bletchley Park, states that no generic solution could be devised to read SG-41 traffic, for messages not in-depth. Furthermore, even the availability of a crib, or plaintext recovered from depths, did not allow for the reconstruction of the key settings (Batey et al., 1945; Fried, 1944).

4 A Novel Known-Plaintext Attack

Because of the complex stepping mechanism, there are no periodic patterns that would allow statistical attacks to be effective, such as the ciphertext-only and known-plaintext attacks that were developed against Hagelin systems with regular stepping (Lasry et al., 2016; Lasry et al., 2018). Furthermore, the circular dependencies of the wheels with regards to their stepping (as illustrated in Figure 4) make the problem even more challenging. Basically, to know how the wheels will step, one needs to know all the pin settings. But to recover the pin settings, one needs to know how the wheels step.

To break that circular dependency, one approach is to exhaustively test some of the elements of the circular logic chain and to validate the elements under test and/or reconstruct additional elements further in the logical chain. There is a trade-off between the number of options to test and the number of elements under test. On the one hand, the richer the information in the elements under test, the easier it is to rule out wrong options in an efficient manner. On the other hand, more elements under test means that more options need to be tested.

Finding the right balance between the number of options to test and the amount of information that allows for a definitive evaluation requires extensive trial-and-error with various testing scenarios. In this section, we present a recursive, backtracking algorithm to validate candidate settings of wheels 1 and 6, based on a sequence of effective kick K_e (obtained from the ciphertext and known-plaintext). The technique is illustrated in Figure 5. It not only tests the settings of wheels 1 and 6 but also reconstructs the settings of wheel 2. The algorithm starts with unknown states for all the pins of wheels 2. It recursively processes the sequence of K_e, testing all possible T5 options at each encryption cycle, advancing wheels 1, 2, and 6 accordingly. It then validates candidate C1 and C6 against K_e and Table 1 (C1 can always be determined unambiguously from C6 and K_e), and tries to deduce C2 from K_e. If C2 can be deduced unambiguously, the pin at the current C2 position is updated accordingly. If C2 at that position has already been updated (this is possible if the wheel has already rotated once or more), then the algorithm validates that there is no conflict. If there is a conflict, the algorithm discards the option for T5, and if all T5 options have been discarded, it backtracks. If there is no conflict (neither with C1 nor with C2), the algorithm recursively processes the next encryption cycle and its associated K_e. If this is the last encryption cycle for which there is known-plaintext, and there are no more K_e elements to process, the (tested) settings of wheels 1 and 6 and the reconstructed settings of wheel 2, constitute a solution candidate.

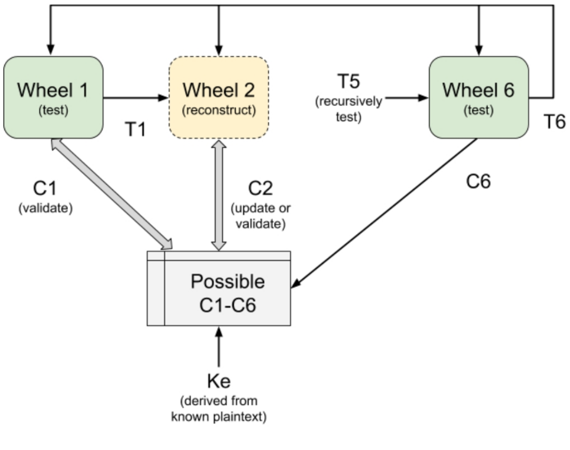


Figure 5: Known-Plaintext Attack – Recovering Wheel 2 Settings

We present here the algorithm to recover the settings of wheel 2 from the settings of wheels 1 and 6 and from known-plaintext. Its complexity reflects the complexity of the stepping logic.

Recursive procedure:

1. Repeat (2) to (5) for all possible pin settings of wheels 6 and wheels 1. There are $2^{25+24} = 2^{49}$ such settings.

2. Initially mark the state of all the pins of wheel 2 as *unknown*. We will mark them as *active* or *inactive* as we gather unambiguous evidence in the procedure described here.

3. Assume that the starting position of all wheels is pin 1 (another position may be assumed, and the algorithm would produce equivalent, shifted, pin settings).

4. Start by processing the first encryption cycle (the first ciphertext and known-plaintext symbols).

5. Determine T6 (from the wheel 6 settings under test, at the pin currently driving T6):
   (a) If T6 is active, the optional pre-encryption stepping stage is applied:
      i. Advance wheels 1, 2, and 6, and advance again wheel 2 if T1 was active.
      ii. For each possible state of pre-encryption T5 (active or inactive):
         A. Advance wheel 6 if T5 is active.
         B. Validate/update C1 and C2 (see below). If a conflict is detected, discard this option for T5, or backtrack if both T5 options result in a conflict.
         C. Advance wheels 1, 2, and 6, and advance again wheel 2 if T1 was active.
         D. For each possible state of (post-encryption) T5 - active or inactive, advance wheel 6 if T5 is active, and recursively perform (5) for the next encryption cycle (the next ciphertext and plaintext symbols). If this is the last encryption cycle (for which there is known-plaintext), store the settings of wheel 2 as a candidate solution.
   (b) If T6 is inactive, only the post-encryption stepping stage is relevant:
      i. Validate/update C1 and C2 (see below). If a conflict is detected, discard this option for T5, or backtrack if both T5 options result in a conflict.
      ii. Advance wheels 1, 2, and 6, and advance again wheel 2 if T1 was active.
      iii. For each possible state of (post-encryption) T5 - active or inactive, advance wheel 6 if T5 is active, and recursively perform (5) for the next encryption cycle (the next ciphertext and plaintext symbols). If this is the last encryption cycle (for which there is known-plaintext), store the settings of wheel 2 as a candidate solution.

Procedure to validate/update C1 and C2:

1. Compute K_e, the effective kick for the current ciphertext and known-plaintext symbol.

2. Determine the expected state of C1 from K_e and the current C6 (C1 can be determined unambiguously - see Table 1). If the expected state of C1 is different from the state of C1 at the current position (based on the wheel 1 settings being tested), the procedure fails.

3. Update or validate C2, as follows:
   (a) If K_e is between 0 and 9, or 16 and 25, it is possible to determine the state of C2 unambiguously from K_e and the current C6 (see Table 1).
      i. If the state of C2 at the current position was previously marked as active or inactive, verify that it does not conflict with the C2 derived from K_e and C6. If there is a conflict, the procedure fails.
      ii. If the state of C2 at the current position was previously marked as unknown, update it with C2 derived from K_e.
   (b) If K_e is between 10 and 15, it is not possible to determine C2 unambiguously from K_e and the current C6. No update or validation for C2 can be done in this encryption cycle.

When processing the initial known-plaintext symbols, the state of C2 at the current position can only be updated and not validated, as there is no prior knowledge. As wheel 2 completes a full rotation, previously updated C2 states can be compared with C2 states newly derived from K_e, checking for contradictions and pruning wrong T5 assumptions, or wrong options under test (settings of wheels 1 and 6). If SG-41 were designed so that wheels advance only once or twice (versus up to four times) per encryption cycle, this attack would have been less effective.

A similar technique is employed to recover the pin settings of wheel 3 from the pin settings of wheels 6, 1, and 2. Similarly, the pin settings of wheels 4 and 5 can be recovered. The candidate pin settings that survive all the algorithm stages are finally verified by decrypting the ciphertext and ensuring that the resulting decryption indeed matches the known-plaintext.

To rule out all wrong settings of wheel 2, a crib of about 150 symbols is required. However, a crib of 80 symbols is enough to rule out most of the wrong wheel 2 settings, while the additional phases (recovering wheel 3, wheel 4, and wheel 5) can discard the remaining wrong ones.

The first phase of the algorithm needs to test 2^{49} options, for all possible settings of wheels 6 and 1. Subsequent phases - for recovering the pins of rotors 3 to 5 - require only 2^{23} to 2^{25} runs. Based on preliminary benchmarks, it is estimated that a few thousands of PCs would complete the process in a month. Further research is needed to evaluate whether additional optimizations or a GPU implementation could further speed up the process.

5 Possible Side-Channel Acoustic Attack on SG-41

The SG-41 is a purely mechanical machine. As wheels step up to four times, it is also a noisy machine, as can be heard in a video of a machine recently restored (Historica, 2019). The sound emitted by the machine is likely to leak extensive information about its internal functioning, and wheel stepping in particular. In this attack, we assume that it is possible to determine at each encryption cycle, based on the sound generated by the SG-41, whether there was one or two stepping stages (that is, whether the optional pre-encryption stepping stage took place). In other words, it is possible to extract a sequence of T6 states from an acoustic recording. This assumption has not been checked yet against a real machine, but it is highly plausible.

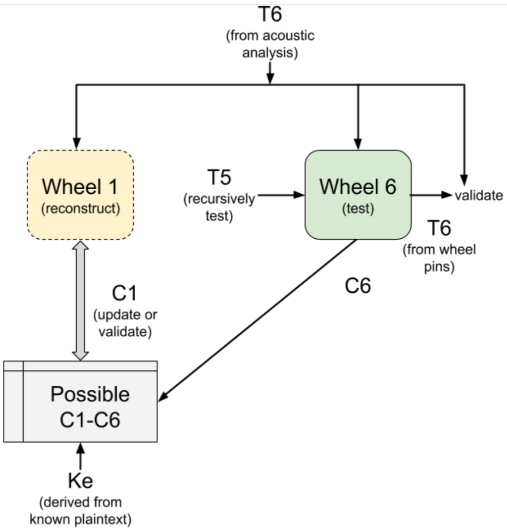


Figure 6: Acoustic Attack – Recovering Wheel 1 Settings

The process for recovering the wheel settings is similar to the process described in Section 4, and only its outline is presented here. The algorithm is applied to all possible settings of wheel 6 (there are 2^{24} such settings). The recursive backtracking algorithm is illustrated in Figure 6. It recursively tests all T5 options at each encryption cycle and it recovers the states of wheel 1 pins, based on C1 that can be derived from K_e and C6 (after wheel 1 completes a full rotation, contradictions can also be detected). It also checks whether T6 derived from wheel 6 pins matches the T6 pattern predicted via acoustic analysis. If there is a conflict (in either C1 or T6), it backtracks. If there is no conflict, the next encryption cycle (the next known-plaintext and ciphertext symbols) is recursively processed.

After candidate settings of wheel 1 have been recovered, the settings of wheels 2 are similarly recovered, and so forth for the remaining wheels. With about 100 known-plaintext symbols (and the relevant T6 sequence), only a handful of candidate solutions survive the last stage of the algorithms, and the wrong ones can be eliminated with a simple decryption test. The algorithm takes a few minutes to test all possible wheel 6 settings.

6 Conclusion

The functional description of the SG-41 in this article is based on historical British and U.S. reports(Batey et al., 1945; Mowry, 1989; Mowry, 2003; Mowry, 1983), and on information that has been made available recently, following the work of curators and collectors who own a SG-41 (Museum, 2020b; Historica, 2019; Dahlke, 2018; I-72, 1945). The acoustic attack described in Section 5 might need to be refined, based on further analysis of the precise information that may leak acoustically, but based on our work, we can provide an revised assessment of the security of the SG-41.

The attack described in Section 4, when nothing is known except for a segment of plaintext, requires 2^{49} runs of the core algorithm. Taking into account the complexity of the core algorithm, the author estimates that the security of the SG-41 is comparable to a modern cipher with 60-bit key (DES as a 56-bit key). The fact that a significant amount of processing power is required for its cryptanalysis with modern techniques is a testimony to the high level of security of the device, compared to other WWII German and Allied cipher machines. It is much more secure than Enigma, and probably provides the same level of security as SIGABA and T52e, the most sophisticated cipher machines of the time (Lasry, 2019). An historical report by the Army Security Agency even suggested designing a new device based on the same principles as the SG-41, to be used by the U.S. (WDGAS-14, 1946; Mowry, 1983).

Some of the features of the SG-41 such as irregular stepping with circular dependencies, and the complementary feature, are nowhere to be seen in other devices, until the 1950s, with some advanced models of the Hagelin CX-52 (Museum, 2020b; Friedman, 1955).

Acknowledgments

This work has been supported by the Swedish Research Council, Grant 2018-06074, DECRYPT – Decryption of historical manuscripts. In addition, the author would like to thank Ralph Erskine and Frode Weierud for providing access to key historical documents and for reviewing an early version of this article, and Klaus Kopacz for clarifying some of the details of the SG-41 stepping mechanism.

References

Army Security Agency. 1947. *Observations on C-41 Cycles. Coll. 354: STINFO, Box 278, Folder16380.* NARA, College Park, MD.

Keith Batey, Mavis Batey, Margaret Rock, and Peter Twinn. 1945. *Secret Service SIGINT Volume II – Cryptographic Systems and Their Solutions – Machine Cyphers.* G.C. & C.S., TNA, Kew, HW 43/7.

Carola Dahlke. 2018. What We Know About Cipher Device “Schlüsselgerät SG-41” so Far. In *Proceedings of the 1st International Conference on Historical Cryptology HistoCrypt 2018*, number 149, pages 109–111. Linköping University Electronic Press.

Walter J. Fried. 1944. *ISK Report, F-119, IR 4065.* War Department.

William F. Friedman. 1955. *Report of Visit to CRYPTO A.G. (Hagelin).* National Security Agency.

Hermann Historica. 2019. *YouTube Video: Schlüsselgerät 41 - Cipher Machine 41 (SG-41).*

TICOM I-72. 1945. *First part of the report by Wm. Buggisch on S.G.41 (still classified).*

George Lasry, Nils Kopal, and Arno Wacker. 2016. Automated Known-Plaintext Cryptanalysis of Short Hagelin M-209 Messages. *Cryptologia*, 40(1):49–69.

George Lasry, Nils Kopal, and Arno Wacker. 2018. Ciphertext-Only Cryptanalysis of Short Hagelin M-209 Ciphertexts. *Cryptologia*, 42(6):485–513.

George Lasry. 2019. A Practical Meet-in-the-Middle Attack on SIGABA. In *2nd International Conference on Historical Cryptology HistoCrypt 2019*, page 41.

David P. Mowry. 1983. Regierungs-Oberinspektor Fritz Menzer: Cryptographic Inventor Extraordinaire. *Cryptologic Quarterly*, Volume 2, Nos. 3-4.

David P. Mowry. 1989. *The Cryptology of the German Intelligence Services.* United States.

David P. Mowry. 2003. *German Cipher Machines of World War II: Description Based on Print Version Record.* National Security Agency, Center for Cryptologic History.

Crypto Museum. 2020a. *Hagelin C-35.* www.cryptomuseum.com/crypto/hagelin/c35/index.htm.

Crypto Museum. 2020b. *Schlüsselgerät 41.* www.cryptomuseum.com/crypto/sg41/index.htm.

Klaus Schmeh. 2004. *Hitlers letzte Maschinen.* Telepolis. Heise.

WDGAS-14. 1946. *European Axis Signal Intelligence in World War II – Volume 2: Notes on German High Level Cryptography and Cryptanalysis.* Army Security Agency.

A Massive Machine-Learning Approach For Classical Cipher Type Detection Using Feature Engineering

Ernst Leierzopf[1]*, Nils Kopal[2], Bernhard Esslinger[2], Harald Lampesberger[1], Eckehard Hermann[1]
[1]University of Applied Sciences Upper Austria, Hagenberg, Austria
[2]University of Siegen, Germany
*e.leierzopf@gmail.com

Abstract

Cryptanalysis of enciphered documents typically starts with identifying the cipher type. A large number of encrypted historical documents exists, whose decryption can potentially increase the knowledge of historical events. This paper investigates whether machine learning can support the cipher type classification task when only ciphertexts are given. A selection of engineered features for historical ciphertexts and various machine-learning classifiers have been applied for 56 different cipher types specified by the American Cryptogram Association. Different neuronal network models were empirically evaluated. Our best-performing model achieved an accuracy of 80.24% which improves the current state of the art by 37%. Accuracy is calculated by dividing the total number of samples by the number of true positive predictions. The software-suite is published under the name ”Neural Cipher Identifier (NCID)”.

1 Introduction

Historical records show that encryption is about as old as scripture itself. The earliest documented use of cryptography can be traced back to the Old Egyptian Empire in the third millennium BC (Lieven, 2007). In ancient history, cryptography was mainly used by the aristocracy and the military. In principle, classical ciphers can be divided into substitution ciphers and transposition ciphers. With simple substitution, each letter is substituted with a different one from the alphabet. Homophonic substitution replaces letters by several different substitutes, so the ciphertext alphabet is bigger than the plaintext alphabet. Transposition ciphers mix (permute) the letters of the plaintext into a quasi-random order. There are also ciphers combining substitution and transposition like ADFG(V)X (Friedman, 1941).

A typical cryptanalysis method for classical substitution ciphers is frequency analysis. Here, the frequencies of single or groups of multiple ciphertext symbols are counted and then compared to the frequencies of the assumed plaintext language. Then, based on the different frequencies in the plaintext language, assumptions of which letter was replaced by which symbol, can be made. Knowledge of the used cipher type allows the application of cipher-specific and heuristic algorithms to find the plaintexts more precisely. For example the Kasiski examination (1863) of the Vigenère cipher takes advantage of the fact that, by chance, repeated words are sometimes encrypted using the same key letters and therefore give indication for the possible key lengths.

The goal of this research is to determine how cipher type detection can be improved with machine learning approaches like feedforward neural networks (FFNN), decision trees (DT), random forests (RF) and naïve Bayes networks (NBN) using newly calculated statistics in a massive feature engineering approach (see Section 3.7). A systematic, exhaustive evaluation over all American Cryptogram Association (ACA) ciphers (2005) has been performed. To achieve the best results, multiple optimizers, activation functions and features were implemented and evaluated with several relatively unparameterized neural networks. The identified features are mainly based on previous work starting with Kopal's prototype (2020) for the MysteryTwister[1] challenge ”Cipher ID”, and on the implementations from Bion (Mason, 2021).

[1] MysteryTwister: https://www.mysterytwisterc3.org/

The result of this work is the software suite ”Neural Cipher Identifier (NCID)” (Leierzopf, 2021), which also will be available online.[2] It can be used for training and evaluation of neural networks with different classifiers like FFNN, DT and NBN. A potential use case for the software suite could be the DECRYPT project, with the aim to offer a working infrastructure for researchers enabling the collection, automated digitization, analysis, and decryption of encrypted historical documents (Megyesi et al., 2020).

This paper is structured as follows. The next chapter discusses all important related work for cipher type detection with neural networks. The third chapter describes all implemented cipher types, data generation procedures, feature selection and different classifier architectures. The results of the empirical evaluation are summarized in Chapter 4. Chapter 5 concludes this paper.

2 Related Work

The search for related work included classification of classical ciphers as well as modern ciphers. The idea was that recognition methods, which work on modern ciphers, most certainly also work on classical ciphers. It is state-of-the-art to use the same datasets for training to achieve better comparability to other work. Unfortunately no such standard dataset exists for the field of cipher type detection.

Nuhn and Knight achieved remarkable results in the area of the classification of classical ciphers with neural networks and were the benchmark (2014). The researchers trained a neural network with a linear classifier and a Stochastic Gradient Descendent (SGD) optimizer with default parameters for 50 ACA ciphers. An accuracy of 58.5% was achieved by using a quadratic loss function and adaptive learning rates with 1 million ciphertexts and 20 epochs. According to Nuhn and Knight, squared features have not improved accuracy. They implemented 55 features from Bion and developed three features themselves. The random text lengths without defined ranges of lengths are a major drawback in the comparability of their work. The reason for this assumption is that most features rely on statistical calculations, which are more precise for longer texts.

Results from the work of Sivagurunathan et. al (2010), where the three classical ciphers Playfair, Hill and Vigenère were analyzed with a simple neural network, coincide with the results of Kopal (2020). Both discovered the difficulty of classifying (distinguishing) the Hill and Vigenère ciphers, because of their similar statistical values.

A multi-layer classifier has been introduced by Abd and Al-Janabi (2019) to classify plaintexts and ten different cipher types. The impressive results of over 99% accuracy are lessened by the enormous ciphertext length of about one million characters, which is the equivalent of an average book with 500 pages. Ciphertexts with these lengths are seldom. The greatest part of original historic encrypted manuscripts are only between a few lines and some pages of ciphertext long.

Krishna (2019) developed approaches that have not yet been used by other authors for the four ciphers Simple Substitution, Vigenère, Transposition and Playfair. An important point for comparison is that the Hill cipher was not used here. The first approach, a support vector machine (SVM), uses the ciphertexts of length 10 to 10,000, which are mapped in a number range, as training data. The SVM uses the implementation of the Sklearn Library[3] with a 10-Fold StratifiedKFold Cross-Validation and a One-vs-Rest Classifier for the calculation of the confusion matrix. This means that 9 out of 10 datasets were used for training and 1 dataset for testing in order to find the most suitable class. In the second and third approach, a Hidden Markov Model (HMM) was trained for 1000 ciphertexts per class and used by means of conversion as input for a convolutional neural network (CNN) in the second approach and an SVM in the third approach. The first approach achieves an accuracy of 100% with a text length of 200, the second 71% with a text length of 155 and the third 100% with a text length of 155.

Zhao et al. (2018) extracted 54 features from 15 different NIST 800-22 (Rukhin et al., 2010) randomness tests. The efficiency was tested in several 10-fold-cross-validation SVM one-to-one classifiers for six modern block ciphers (AES, Blowfish, Camellia, DES, 3DES and IDEA). The result was that 42 features gave better results than random guessing, i.e. 50%. 12 of these features even provide a recognition rate of over 60%.

Tan and Ji (2016) developed a very similar model with the five modern ciphers AES, Blowfish, DES, 3DES and RC5. The experiments are

[2] https://www.cryptool.org/ncid

[3] Sklearn Library: https://scikit-learn.org/stable/

Author	# Features	Accuracy in %	Text Length	Dataset Size	Epochs	# Cipher Types	Cipher Category	Technology
Nuhn	58	58.50	random	1,000,000	20	50	classical	Vowpal Wabbit
Nils Kopal	4	90	100	4,500	20	5	classical	FFNN
Sivagur.	12	84.75	1,000	900	N/A	3	classical	FFNN
Abd	7	99.60	1,000,000	N/A	500	11	classical	3-Level-Classifier
Krishna	N/A	100	155	4,000	N/A	4	classical	SVM, Hid. Markov M.
Zhao	54	47.8-89.5	512,000	6,000	N/A	6	modern	One-vs-One SVM
Tan	N/A	39	100,000	1,100	N/A	5	modern	SVM
Manjula	10	72.20	1-2,000	1500	N/A	11	modern	DT
Chandra	46	80	12,800	1,000	N/A	3	modern	One-vs-One FFNN

Table 1: Summarized results and attributes of related work

carried out in this work with two scenarios: Once the same key material for training and test data and the other time with different key material. For the same key, the result is 85% from 20 kB of data and 96% from 100 kB of data. For different key lengths with the same amount of data it is 35% or 39%. The parameters used by the SVM were not explained in more detail.

Manjula and Anitha (2011) designed a C4.5 classifier for eleven modern ciphers and achieved a recognition rate of over 70% for ciphertexts with a variable length of 1-2000 bytes. The C4.5 algorithm creates a decision tree based on the information gain ratio. A total of ten features were designed, seven of which are based on the maximum entropy of different characters. Further features are the entropy of all characters, the correlation coefficient of capital letters and the length of the ciphertext, since the expected entropies depend on this.

Different algorithms for the one-to-one classification, i.e. a comparison of individual modern stream and block ciphers, were presented by Chandra et al. (2007). The tested neural network architectures were back propagation, back propagation with momentum, resilient propagation, scaled conjugent gradient, conjugent gradient with Powell-Beale restarts and conjugent gradient with Polak-Ribiere update. On average, all algorithms achieved an accuracy of over 80%, but resilient propagation was able to achieve over 6% better results, especially comparing one stream cipher with another stream cipher. The training was carried out with texts with a length of 12.8 kB and 46 features that are not described in detail.

Table 1 summarizes the state of the art with respect to the number of features, the self-reported accuracy, the utilized text lengths for evaluation and the training dataset size in the respective paper.

3 Neural Cipher Identifier

In this paper general classifier architectures are referred to as classifiers. Trained instances of these classifiers are called models. The selected architectures and algorithms of the models were, more or less, biased by the knowledge of the authors, and the hyperparameters were set to default values without optimization.

A common theme in the related work is that features are selected from one or a few sources and not further questioned or tested. These features are often incomplete and correlated to other features, which can even make the models worse. By selecting features based on individually tested results and implementing and optimizing multiple feature engineering machine learning approaches, better results for more cipher types can be expected. This approach has been implemented in this paper. Every test used newly generated data to prevent specialization on a specific dataset.

3.1 Implemented Cipher Types

This work is based on Kopal (2020), who analyzed the five ciphers, simple monoalphabetic substitution, columnar transposition, Vigenère, Hill and Playfair with an FFNN. As a result from previous work, an FFNN with five hidden layers and a hidden layer size of

$$2 \cdot \frac{input\_layer\_size}{3} + output\_layer\_size$$

is used as the starting point of research (further on called **"baseline reference model"**). In the first step, the solution was expanded by adding interfaces, cipher implementations, a custom data loader and a testsuite for all classes. Training and test data is generated on-the-fly.

For the test setup, all Bion features plus the already existing features from previous work together with a selection of 55 of the 60 ACA ciphers and plaintext were used. The ciphers Twin Bifid and Twin Trifid were excluded, because they combine two ciphertexts, Incomplete Columnar Transposition and Interrupted Key were also excluded, because they are indistinguishable ciphers. Syllabary (Friedman, 2012) was not implemented, because it was invented 2012 and does not fit into the classical cipher period. Table 2 shows the ciphers used during this evaluation.

amsco	grandpre	per. gromark	ragbaby
autokey	grille	phillips	railfence
baconian	gromark	phillips rc	redefence
bazeries	gronsfeld	plaintext	route transp.
beaufort	headlines	playfair	running key
bifid	homophonic	pollux	seriatedpfair
cadenus	key phrase	porta	slidefair
checkerboard	mnmedinome	portax	swagman
col. transp.	morbit	progkey	tridigital
condi	myskowski	quagmire1	trifid
cmbifid	nicodemus	quagmire2	trisquare
digrafid	nihilist subst.	quagmire3	two square
foursquare	nihilist transp.	quagmire4	variant
fract. morse	null	numbered key	vigenère

Table 2: All 56 implemented ACA ciphers

To get comparable results with different architectures, the training and validation text length is fixed to 100 characters, after all non-alphabet characters are filtered. According to the American Cryptogram Association (2005), all ACA ciphers need 40 to 220 characters to be broken.

3.2 Keywords

For historical reasons, all ACA ciphers, whose keys do not consist of digits, do not choose the key words and alphabets at random, but rather use English words. So called key alphabets use one keyword and fill the rest of the alphabet in the alphabetical order. This allows the following three training scenarios to be defined and sorted by their classification difficulty:

1. Keywords are chosen from a dictionary. Key alphabets use key words from a dictionary and the rest of the alphabet is arranged in the correct order.
2. All characters of keywords are chosen at random. Key alphabets use keywords with all characters being chosen randomly and fill the rest of the alphabet in the correct order.
3. All characters of keywords are chosen at random. Key alphabets are arranged randomly.

By default, all tests were run in the second scenario which use keywords with all characters chosen randomly and key alphabets are arranged in the correct order after the keyword. Filling the rest of the alphabet in the correct order is a major weakness of each cipher. However, historically ciphers were used in the first scenario with a word chosen from a dictionary. At the time of invention this procedure offered enough security. Compared to scenario 1, the advantage of the second scenario is that the model is less likely to be overfitted due to the lack of different keywords for specific lengths and it should be more general and more secure than with predefined keywords.

3.3 Optimizer Selection

Before testing new architectures some tests were made beforehand. The algorithm used to determine the weights after every training cycle is referred to as optimizer. The optimizer was selected by the best result of empirical test runs with the default parameters. SGD with Momentum = 0.9, RMSprop, Adam, Adadelta, Adagrad, Adamax and Nadam were tested. To find out which of the seven optimizers is the best for our scenario, each model was trained with a different optimizer with 100 million data records, i.e. 1.8 million per cipher. Ciphers which need keywords were trained with the key lengths 5 to 8 as these lengths were typical at the time of invention. The rest was trained with no keywords and the same amount of data. A baseline reference model with plaintexts of the exact size of 100 characters and the second training scenario were used for the comparison.

Optimizer	Accuracy in %	Top 3 Accuracy in %	Training Time	Converge?
SGD with Momentum	64.78	81.50	4h 21m	Yes
RMSprop	69.96	84.60	4h 27m	Yes
Adam	72.97	87.18	4h 21m	No
Adadelta	48.04	68.82	4h 23m	No
Adagrad	56.31	74.74	4h 23m	No
Adamax	73.71	87.80	4h 25m	No
Nadam	72.42	86.91	4h 33m	Yes

Table 3: Results of the comparison of 7 optimizers

From the results in Table 3 it can be seen that Adam and Adamax deliver the best results in terms of accuracy with default parameters. The training time of the model is very close for all optimizers which can be attributed to the preprocessing time on the CPUs being greater than the training time on the GPUs. For better comparability, further tests and the search for the best hyperparameters are carried out with the Adam algorithm.

3.4 Activation Functions

Activation functions are mathematical functions which are used to adapt the weights in a neural network. In order to be able to determine how the training corresponds to different activation functions, the baseline reference model was trained up to convergence with 10 different activation functions. The exponential function is an exception in which only one hidden layer was used, as otherwise the loss cannot be calculated. These results were calculated after the selection of the best features, which is described later on. Using the activation functions in Keras (2021), the results of the activation function comparison can be seen in Table 4.

Function	Accuracy in %	Top 3 Accuracy in %	Training Time	Converges after Mio. Iterations
ReLU	74.70	88.94	16h 50m	146
Leaky ReLU	72.64	87.45	12h 30m	99
Parametric ReLU	75.18	89.02	19h 29m	152
Sigmoid	72.32	87.01	1d 22h	385
tanh	65.48	81.87	9h 33m	68
ELU	68.51	84.18	10h 24m	76
SELU	67.54	83.58	13h 48m	103
Exponential	70.10	85.62	17h 54m	140
Swish	70.32	86.07	20h 33m	150
RBF	1.59	4.92	3h 30m	16

Table 4: Results of the activation function comparison

With the exception of the Parametric ReLU function, the ReLU function delivered the best results in terms of accuracy and training time. Due to the more reliable results of the ReLU function it was preferred over the more complex Parametric ReLU function in further tests. The exponential function delivered impressive results with only one hidden layer.

3.5 Data Generation

Figure 1 shows the training process of the cipher classification model. 14 GB of English texts of the Gutenberg Project[4], which is free to use for research purposes, were used as dataset for training and validating of the models. Loading and preprocessing of the features is done by an own data loader and is described in more detail in the next paragraph. After the training process is completed, the model is saved and evaluated.

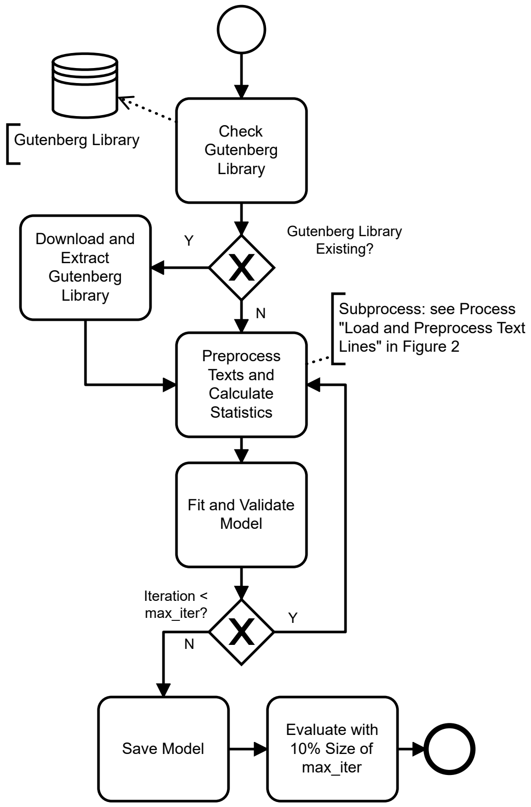

Gutenberg Library
Check Gutenberg Library
Download and Extract Gutenberg Library
Y
Gutenberg Library Existing?
N
Subprocess: see Process "Load and Preprocess Text Lines" in Figure 2
Preprocess Texts and Calculate Statistics
Fit and Validate Model
Iteration < max_iter?
Y
N
Save Model
Evaluate with 10% Size of max_iter

Figure 1: Training process

Figure 2 shows the data loader process described in the last section. This process loads one or multiple lines of text, depending on the defined ranges, from the given dataset, and adapts them with the appropriate filter function for the specific cipher. As the final length after a text is read from file can not be determined due to the filtering of non-alphabetic characters, lines are read in loops. For example, the filter of the Playfair cipher replaces all J characters with I characters. The length of the entire text can be set using command line arguments. It can be assumed that longer messages are easier to classify because the calculated features are more meaningful. The process described must be carried out until the number of plaintexts generated equals the size of the required dataset, which is defined as a parameter in the program itself, divided by the sum of the ciphers and

[4] https://www.gutenberg.org/

their configured key lengths. This means that each plaintext can be used once for each cipher with each of the configured key lengths for training the model. After enough lines of text are available, several processes, so-called workers, are started in figure 2 to calculate the features in parallel.

3.6 Features

The selected features can be divided into the following groups:

- frequency statistics (e.g. unigrams, bigrams)
- distribution statistics (e.g. IoC)
- binary features (e.g. HAS_J, HAS_X)
- cipher-specific features (e.g. A_LDI)

Abbr	Term	Description
SDD	Average Single Letter – Digraph Discrepancy Score	This feature uses a table of the differences between unigrams and bigrams. The score is calculated by adding each value at the position of the first letter in the alphabet times 26 plus the position of the second letter in the alphabet from the SDD table. The score is then divided by the length of the text minus 1. For normalization the scores are divided by 10.
CHI^2	Chi Square	With the Chi^2 function, the deviation from the distribution of English letters, which is known, can be calculated. This value is divided by 100 to be normalized.
DIC	Digraphic Index of Coincidence	Sum of all probabilities of the occurrence of two identical pairs of characters in a text times 1000.
DBL	Double Letter	Binary value about the occurrence of a double character in an even place and that the total length is even.
AUTO	Estimated Auto Correlation	Autocorrelation is useful in identifying repeating patterns in a sequence. Due to the different lengths of the ciphertexts (the Null cipher has ciphertexts a maximum of 10 times as long as plain texts), the remaining data points must be filled with 0.
FREQ	Frequencies	Recursive calculation of the probability of occurrence up to and including bigrams.
HAS_0	Has Digit 0	Binary value based on the occurrence of the digit 0.
HAS_H	Has Hash	Binary value based on the occurrence of the # sign.
HAS_J	Has Letter J	Binary value for the occurrence of the letter J.
HAS_X	Has Letter X	Binary value for the occurrence of the letter X.
HAS_SP	Has Space	Binary value based on the occurrence of the space character.
IoC	Index of Coincidence	Sum of all probabilities of the occurrence of two identical characters in a text.
LDI	Log Digraph Score	Bigrams in a text are searched for in a list of precalculated English letter frequencies and added up. The average of this sum is the score. At Bion, the real numbers are used instead, but these are too large values, which is why the probability of occurrence divided by 10 is more suitable.
A_LDI, B_LDI, P_LDI, S_LDI, V_LDI, PTX	Log Digraph Score for Autokey, Beaufort, Porta, Slidefair, Vigenère, and Portax	The LDI calculates this set of Vigenère statistics for different ciphers by converting the ciphertexts with the respective shift functions. The score is divided by 1000. For ciphertexts that contain characters other than letters, the PTX feature is 0.
LR	Long Repeat	Percentage of characters that are repeated exactly three times. For this purpose, all the same characters are counted for each character from position +1. The root of this result is divided by the length of the text.
BDI	Max Bifid DIC for Periods 3-15	As in the Bifid cipher, texts are read in periods of 3-15 and the DIC is calculated from this. The highest score is divided by 1000 and returned. For ciphertexts that contain characters other than letters, this feature is 0.
CDD	Max Columnar SDD Score for Periods 4-15	As in the columnar transposition cipher, texts are read in periods and the SDD score is calculated for them. The result of this feature is the maximum SDD score divided by 1000. This feature is 0 for ciphertexts that contain characters other than letters.
MKA	Max Kappa	Texts are shifted by p to the right for Periods 1-15. The remaining p characters are padded with values that are not contained in the text (e.g. -1). The result of this statistic is the maximum percentage of match between the moved text and the original text.
NIC	Max Nicodemus IC	Texts are divided into periods 3-15. The highest NIC is calculated by dividing and reading the text as with the Nicodemus cipher. The highest value is returned.
SSTD	Max STD Score for Swagman Periods 4-8	As in the Swagman cipher, texts are read in periods and the STD score is calculated. The result of this feature is the maximum STD score divided by 100.
MIC	Maximum Index of Coincidence	Texts are divided into periods 1-15. The highest IoC of all subgroups is calculated by dividing the text into p groups. Each group consists of all characters spaced p. If p = 3 there are 3 groups, whereby the first group contains every third character starting with 0; the second group every third character starting with 1 and the third group every third character starting with 2. The highest value is returned.
NOMOR	Normal Order	The frequency of each character is calculated and sorted by size. The normal order is the sum of the distances of all characters from their normal position divided by 1000.
PHIC	Phillips IC	Calculates the IC using a fixed column size = 5 and a fixed period = 8. The result is multiplied by 10. For ciphertexts that contain characters other than letters, this feature is 0.
REP	Repetition Feature	This feature is adopted from Nuhn and Knight (2014). It consists of the normalized number of exactly n times occurring identical characters for $2 \leq n \leq 5$. The normalization is calculated by dividing through the total number of repetitions.
ROD	Repetition Odd	Percentage of odd-spaced repeating characters to the sum of repeating characters. For this purpose, all the same characters are counted for each character from position +1. The result is sum_odd / sum_all
RDI	Reverse Log Digraph	Bigrams in a text are searched for in a list of precalculated English letter frequencies and added up, but the order of the letters is reversed, e.g. AB -¿ BA. The average of this sum is the score. With Bion, the real numbers are used instead, but these are too large values, which is why the probability of occurrence divided by 10 is more suitable.
SHAN	Shannon's Entropy Equation	Entropy is a measure for determining the information content of a text. Basically, a higher entropy indicates that data is encrypted. This value is divided by 10.

Table 5: Feature definitions

The value ranges of the features are normalized to [0..1] so that small changes in a feature with a higher value range do not have disproportionate

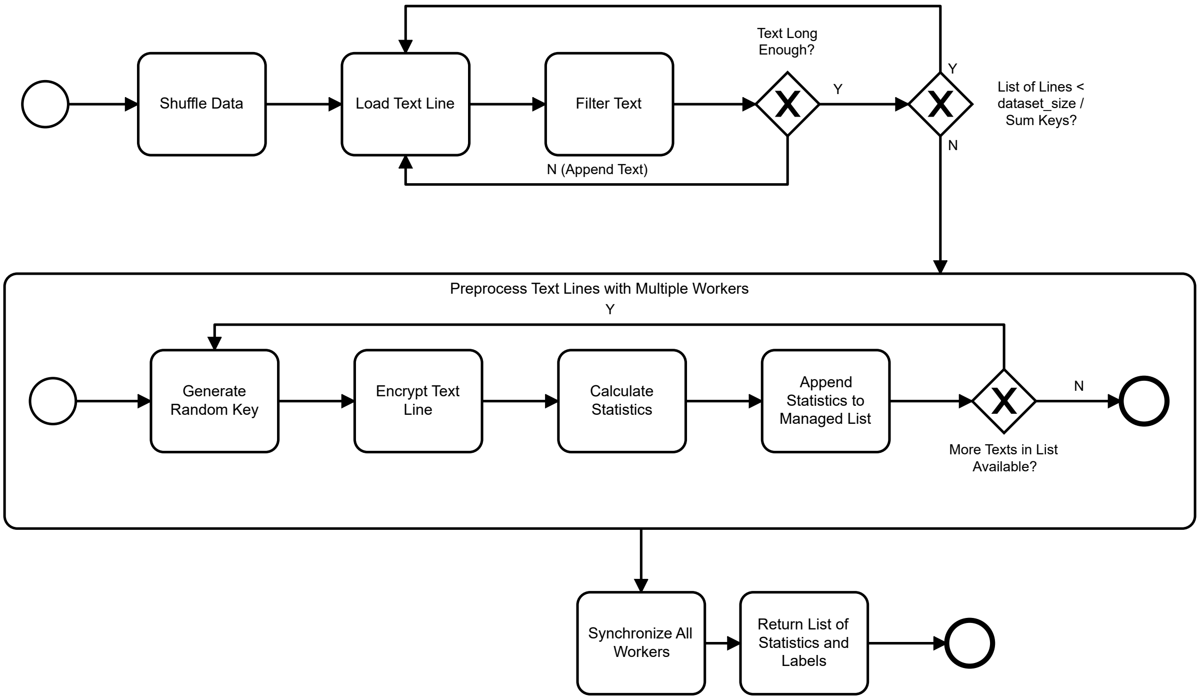

Shuffle Data
Load Text Line
Filter Text
Text Long Enough?
Y
N (Append Text)
Y
List of Lines < dataset_size / Sum Keys?
N
Preprocess Text Lines with Multiple Workers
Y
Generate Random Key
Encrypt Text Line
Calculate Statistics
Append Statistics to Managed List
N
More Texts in List Available?
Synchronize All Workers
Return List of Statistics and Labels

Figure 2: Load and preprocess text lines

effects on the decision and on the evaluation of other features in the learning process. Every feature was calculated 1000 times. The average calculation time contributed to the decision whether a feature was included or not. Totally 28 features were implemented and out of these 20 were selected after extensive testing (see Chapter 4). The tests were run evolutionary. This method does not need as much testing as a grid search. Every feature which achieves better accuracy than the last configuration is included in the following test. Table 5 describes all implemented and tested features. The selection is based on tests with adding one feature at the time and checking the difference in the results.

3.7 Classifier Architectures

Feature engineering classifiers have the property that selected features are provided as input for the model through expert knowledge. With regard to text classification, features can be properties like HAS_J, which checks if a J can be found in the text, and statistics like the index of coincidence. An essential advantage of the feature-engineering method is that known weaknesses of ciphers can be modeled as features and trained in the neural network. The greatest disadvantage of this method is the initial effort required to design and implement the features mentioned.

Feedforward neural networks (FFNN) are based on differentiable activation functions and the finding of the local minimum for the gradient descent error function.[5] The structure of an FFNN consists of one or more layers. The layers between the input and output layers are called hidden layers. A neuron never has connections to other neurons in the same layer. The output of one layer is used as the input of the next layer. The goal of the training phase is to calculate the optimal multiplier (weight) for every connection to minimize the error (loss) by using a small factor (learning rate). The weight is adjusted with a small part of the calculated loss after each update iteration. The complexity of a model is determined by the number and width of the hidden layers and must not be too simple or too complicated. The statistics bias and variance are mostly used to evaluate models. Bias refers to errors due to relationships that have not been learned. The variance is the sensitivity to training data. A model that is too simple can be recognized by a high bias and a low variance (underfitting). A model that is too complicated gets a low bias and a high variance (overfitting) due to the irrelevant features of the data. (Tino et al., 2019)

[5] Gradient descent error function: https://towardsdatascience.com/an-overview-of-the-gradient-descent-algorithm-8645c9e4de1e

Decision tree (DT) algorithms construct binary trees from data for decision making. Each DT has a root node, internal nodes and leaf nodes. Ultimately, the decision always takes place in a leaf node. DT are prone to overfitting and therefore misclassification, which is why the structure of the DT is built up in two phases: training and pruning. During training, the tree is built with all the nodes. The task of the pruning phase is to remove rarely used nodes in order to improve the accuracy and runtime of the DT. (Anyanwu and Shiva, 2009)

DT can be used in serial algorithms (e.g. C4.5 or CART) and in parallel algorithms (e.g. RF). RF consist of multiple incomplete DT with randomly selected features from the entire feature map. These DT are called estimators.

Naïve Bayes networks (NBN) are based on the assumption that all attributes or features of data are completely independent of one another. NBN classifiers make decisions by using the maximum a posteriori estimation with the individual attributes. (Huang and Li, 2011)

Depending on whether the classification problem requires one or more classes, a decision function must be implemented. In the case of clear decision-making problems, in most cases the class with the greatest probability is chosen. Classification problems with multiple outcomes can be classified using a threshold method. Basically, a distinction can be made between Bernoulli and multinomial NBN. Bernoulli NBN can only use binary features. In contrast, multinomial NBN are able to use discrete data for classification.

4 Empirical Evaluation

The best feature map combination from Table 5, which consists of the 20 features SDD, DIC, FREQ, HAS_0, HAS_H, HAS_J, HAS_X, HAS_SP, IoC, LDI, LDI_STATS, LR, BDI, PTX, MKA, NIC, MIC, NOMOR, PHIC and ROD, led to 80.24% accurracy with the FFNN classifier.

Simple decision trees (DT) achieved an accuracy of 61.68%. Random forest classifiers (RF) achieved 71.15% accuracy with 1000 estimators and a maximal depth of 30 without using the LDI_STATS feature. RF achieved good results with a fraction of the training time and data. An essential drawback of RF are the enormous memory requirements, which peaked at about 350 GB, and a very large model to be saved. Therefore, a small RF model with only 100 estimators and setting the parameters minimal samples leaf and split to 10 achieved 74.35% with only 6.4 GB of space, using the LDI_STATS feature. Naïve Bayes networks did not perform well for this specific problem with the provided features. They only achieved 54.17% accuracy. Overall, FFNN achieve the best results for feature engineering classifiers.

Table 6 shows a comparison between all four tested models and Nuhn's work concerning accuracy and memory requirements. All of these models, excluding Nuhn's, used 20 features and a plaintext length of 100 for 56 ciphers. Nuhn's work has been selected to compare with, because it is the most comparable work from Table 1 to this one. The other authors from Table 1 used a much smaller set of different cipher types.

Technology	Accuracy in %	Memory Usage in MB
Nuhn's Vowpal Wabbit	58.50	N/A
FFNN	80.24	45
DT	61.68	300
RF	74.35	6,400
NBN	54.17	2

Table 6: Summarized results compared to Nuhn's work

5 Conclusion

Random English plaintexts were encrypted with 56 different cipher types specified by the American Cryptogram Association. The task was to train models which can be used to determine the cipher type of given ciphertexts. In the feature testing and hyperparameter optimization phases more than 100 models were systematically trained, each one having a computing time of about one day on a Nvidia DGX-1 V100 deep learning machine. As a result, the best configurations for different types of machine learning models were found. In summary, feedforward neural networks (FFNN) provide the best models in terms of accuracy. Random forest classifiers (RF) on the other side only need small amounts of data with about 3 million records to deliver good results in comparison to 200-250 million records with the FFNN.

Further work in this field could include training models with texts from different languages or with texts including errors, as these likely happened in historical documents. Another related question is, whether different features can help in finding the key of a ciphertext and if feature engineering is the best approach for this problem. More modern ciphers used in World War II can also be implemented and tested with the existing classifiers. This work can be further extended by testing if feature-extracting neural networks can achieve similar or even better results without engineering and testing features. Another extension would be to train and apply these classifiers for modern ciphers.

Acknowledgements

This work has been supported by the Swedish Research Council (grant 2018-06074, DECRYPT – Decryption of historical manuscripts) and the University of Applied Sciences Upper Austria for providing access to the Nvidia DGX-1 deep learning machine.

References

A. Abd and S. Al-Janabi. Classification and Identification of Classical Cipher Type using Artificial Neural Networks. *Journal of Engineering and Applied Sciences*, volume 14, 2019.

American Cryptogram Association. Cryptogram. 2005. https://www.cryptogram.org/.

M. Anyanwu and S. Shiva. Comparative Analysis of Serial Decision Tree Classification Algorithms. *International Journal of Computer Science and Security*, volume 3, June, 2009.

B. Chandra, P. Pallath, P. Saxena, and S. Kant. 3rd Indian International Conference of Artificial Intelligence (IICAI-07). *Neural Networks for Identification of Crypto Systems*, pages 402–411, New Delhi, India, January, 2007.

W. F. Friedman. Military cryptanalysis. volume 61, 1941.

Robert J. Friedman. The syllabory cipher. *Cryptogram*, May-June, 2012.

Y. Huang and L. Li. Naïve Bayes classification algorithm based on small sample set. *IEEE International Conference on Cloud Computing and Intelligence Systems*, pages 34–39, Beijing, China, September, 2011.

Friedrich Kasiski. *Secret writing and the Art of Deciphering*. E. S. Mittler und Sohn, Kingdom of Prussia, 1863.

Keras. Keras. 2021. https://keras.io/api/layers/activations/.

Nils Kopal. Of ciphers and neurons–detecting the type of ciphers using artificial neural networks. *Proceedings of the 3rd International Conference on Historical Cryptology HistoCrypt 2020*, number 171, pages 77–86. Linköping University Electronic Press, 2020.

N. Krishna. Classifying Classic Ciphers using Machine Learning. Master's thesis, San Jose State University, California, USA, May, 2019.

E. Leierzopf. NCID - Neural Cipher Identifier. 2021. https://github.com/dITySoftware/ncid.

A. Lieven. *Grundriss des Laufes der Sterne. Das sogenannte Nutbuch*. The Carsten Niebuhr Institute of Ancient Eastern Studies, Denmark, 2007. ISBN 978-3-15-96137-8.

R. Manjula and R. Anitha. Identification of Encryption Algorithm Using Decision Tree. *CCSIT 2011, Part III, CCIS 133*, pages 237–246, New Delhi, India, 2011.

W. Mason. Bionsgadgets. January, 2021. https://bionsgadgets.appspot.com.

Beáta Megyesi, Bernhard Esslinger, Alicia Fornés, Nils Kopal, Benedek Láng, George Lasry, Karl de Leeuw, Eva Pettersson, Arno Wacker, and Michelle Waldispühl. Decryption of historical manuscripts: the DECRYPT project. *Cryptologia*, 2020. DOI:10.1080/01611194.2020.1716410.

M. Nuhn and K. Knight. Cipher Type Detection. *Conference on Empirical Methods In Natural Language Processing*, pages 1769–1773, Doha, Quatar, October, 2014.

A. Rukhin, J. Soto, J. Nechvatal, M. Smid, E. Barkerand S. Leigh, M. Levenson, M. Vangel, D. Banks, A. Heckert, J. Dray, and S. Vo. A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications. *Computer Security*. NIST - National Institute of Standards and Technology, April, 2010.

G. Sivagurunathan, V. Rajendran, and T. Purusothaman. Classification of Substitution Ciphers using Neural Networks. *IJCSNS International Journal of Computer Science and Network Security*, volume 10, pages 274–279, March, 2010.

C. Tan and Q. Ji. An Approach to Identifying Cryptographic Algorithm from Ciphertext. *8th IEEE International Conference on Communication Software and Networks*, Chengdu, Sichuan Province, China, 2016.

P. Tino, L. Benusova, and A. Sperduti. Natural-logarithm-rectified activation function in convolutional neural networks. China, December, 2019. IEEE 5th International Conference on Computer and Communications.

Z. Zhao, Y. Zhao, and F. Liu. The Research of Cryptosystem Recognition Based on Randomness Test's Return Value. *Cloud Computing and Security - 4th International Conference*, pages 1–15, Haikou, China, June, 2018.

Key Design in the Early Modern Era in Europe

Beáta Megyesi and Crina Tudor
Dept. of Linguistics and Philology
Uppsala University
Sweden
`first.last@lingfil.uu.se`

Benedek Láng and Anna Lehofer
Budapest University of Technology
and Economics, and ELTE
Hungary
`(lang|lehofer.anna)@filozofia.bme.hu`

Abstract

We present an empirical study on historical keys in their original form from Early Modern Times (1400-1800) in Europe. We describe the internal structure of keys, and specify what was encoded and how. We present some trends of the construction of historical keys over time. Some of these trends have been sensed but never systematically documented by crypto historians, some other trends however are revealed here for the first time.

1 Introduction

Many studies in historical cryptology have been published on the cryptanalysis of single ciphers but systematic studies on the development of ciphers and the way encryption was carried out are rather few. Studying a large number of keys from various time periods and geographic areas gives insights into the evolution of encryption. To study original keys over time in a systematic way requires a significantly large sampled set of original keys, collected from archives and libraries. Large-scale studies have not been possible due to the lack of infrastructural resources and tools for historical cryptology.

The DECODE database (Megyesi et al., 2019) developed recently for the collection of historical ciphertexts and keys contains over 1 000 keys, of which ca 41% have been transcribed with publicly available transcriptions at the time of writing. The transcribed keys allow us to carry out large, quantitative studies to investigate and compare the internal structure of keys.

Relying on materials published by other scholars and on the basis of the DECODE collection containing many different types of keys, in many languages and from various European territories, we provide some insight into the evolution of encryption, describe some trends, along with a structural description of keys to present their typology.

The study described in this article seeks to get insights into answers to the following research questions:

- What types of keys were used in Europe between the 15th and 18th centuries? What were their specific characteristics?
- What was encoded and how?
- How did encryption evolve over time?
- Can we apply simple statistical methods to large-scale analysis of transcribed historical keys?

We focus on original keys from the Early Modern times, ca 1400-1800 found in European archives and libraries.

We start with an overview of previous studies on encryption methods with the main focus on key structure and an overview of the morphology of keys. We continue with a description of the data collection used in our study and the automatic structural description of keys. Then, in Section 5, we present results about what is encoded in keys and how, and describe some trends in key design over the centuries. Lastly, we discuss some issues and conclude our findings.

2 Historical Cipher Keys

In classic cryptography, a key defines the transformation of the plaintext units (characters, words, phrases, etc) into ciphertext to encrypt the plaintext message, and vice versa, to decrypt ciphertext. The plaintext units are replaced with a code as specified by the key. The code can be represented by symbols from alphabetic characters and digits to many kinds of graphic signs.

While large-scale systematic studies on historical keys are missing, we can find a few late

19th and mid 20th century text editions of cipher keys that did not go beyond simply publishing the tables, see e.g. Rockinger (1892) and Devos (1950). The most well-known studies on keys were performed by Aloys Meister in the beginning of the twentieth century, who first offered systematic analyses of this kind of source. In two volumes, he focused on the cipher system of the Vatican (Meister, 1906) [p. 69], and other Italian city-states (Meister, 1902)), not only publishing, but also classifying the keys. Meister collected keys from the 14th to the 17th centuries from various archives in the Vatican and identified 12 types of keys using digits, and described an advanced system of cryptography carried out by professionals involving training in both the creation and the cryptanalysis of ciphers. Meister focused on keys and did not publish ciphertexts so we cannot draw any conclusion from the actual usage of keys.

The Vatican ciphers were revisited in a recent study (Lasry et al., 2020) aiming at the decryption of ciphertexts and the recovering of keys, originated from the papal correspondence in European countries between the 16th and the 18th century. The study gave unique insights into papal cryptographic practices and showed that in the 16th century, and in accordance with Meister's study, there is strong evidence for diversity, innovation, and sophistication in the development and use of (papal) cipher methods and keys. The cipher types from that period include simple (one plaintext entity – one code), homophonic (one plaintext entity – several codes), and polyphonic (several plaintext entity – one code) substitutions with or without nomenclature elements, i.e. codewords, the cipher equivalents of proper names, geographical entities, common words, etc. Most of the homophonic ciphers use variable length codes for various plaintext entities, making codebreaking much harder. In the 17th and 18th centuries, on the other hand, shorter or longer nomenclatures were standard and the ciphers were homophonic with codes of fixed-length, thus easier to use, but also easier to break, allowing deterministic parsing and decoding.

To our knowledge, the only study that systematically described early modern code keys was carried out by David Kahn published in his famous *Codebreakers* (Kahn, 1996).

Not to mention here a great number of useful case studies published in the following half century (more often than not in the journal *Cryptologia*) that did not exceed local relevance, in 2018 Benedek Láng (Láng, 2018) chose a fairly large, but still limited territorial scope, that of East-Central Europe. On the source material of this territory, he carried out a systematic analysis. He mapped the many small steps stages through which monoalphabetic ciphers evolved first into large homophonic systems, which finally gave the floor to code-booklets. On this rich, but geographically well defined area, he managed to match cipher keys with the corresponding encrypted documents. In this matching process, such structural features as we present in this article, were of great help.

To quote David Kahn again, he emphasised first that a systematic research is to be done in the historical evolution of nomenclators. Note that Kahn uses the word nomenclator in a more general sense than we defined nomanclatures above: he refers to the whole cipher key. Kahn writes:

> "At first, the substitution symbols were neither letters or numbers but fanciful signs like % or . But nobody has looked into when, in the later evolution, as nomenclators ran out of easily distinguishable symbols and began using numbers, the cipher secretaries began forming two-part nomenclators. This research requires merely examining the many nomenclators in the archives of Italy and France and timing and quantifying the change. I suppose it will be tough, living in Europe for a year and having an aperitif after a day examining antique manuscripts. But somebody should do it!" (Kahn, 2008) [p.58].

And this is exactly what the authors of this paper are up to.

3 The Morphology of Keys

A key defines how each entity in the original plaintext shall be encrypted. Keys contain a mapping between the plaintext entities and their corresponding codes used for encryption. There are some basic elements in historical keys that can be structurally described. We introduce the term "morphology" to describe the form and structure of keys with respect to codes and their corresponding plaintext entities.

Entities that can be encrypted range from characters in the plaintext alphabet and space to hide word boundaries, to nomenclature elements that are plaintext entities with two or several characters, such as syllables, morphemes, common words, and/or named entities, typically referring to persons, geographic areas, or dates. Punctuation marks or capital letters might also occur in keys while diacritics are often not encoded. A key might also contain nulls, i.e. symbols without any corresponding plaintext characters to confuse the cryptanalyst and make decryption even harder.

Each type of entity to be encrypted might be encoded by one symbol only, two symbols, three symbols, and so on. The codes in a key might be of fixed or of variable length. For example, one key might contain only two-digit codes while another key might contain two-digit numbers for the encryption of the characters in the plaintext alphabet, three-digit numbers used for the nomenclature elements, one-digit numbers for space, and four-digit numbers for the nulls. To make decryption difficult, the most frequently occurring plaintext characters in a language might have several corresponding codes.

Figure 1 illustrates a key based on homophonic substitution with nomenclature from the second half of the 17th century. Each letter in the alphabet has at least one ciphertext symbol represented as a two-digit number or a symbol, and the vowels and double consonants have one additional graphical sign (e.g. A – 18, m; B – 20; C – 19). The key also contains encoded syllables with two-digit numbers or bigram characters (e.g. BA – 65; BE – 66), followed by a nomenclature in the form of a list of Spanish words encoded with three-digit numbers or symbols (e.g. ajustiamento – 106).

Given a transcribed key, we can automatically derive the key's morphological structure. Next, we describe our method for the empirical study on historical keys using computational methods.

4 Analysing Keys

4.1 Key Collection

Finding original keys in archives and libraries is a time-consuming and frustrating endeavor as these manuscripts are rarely indexed as keys. The DECODE database (Megyesi et al., 2019) provides a collection of encryption keys with information about their origin and other relevant documents. At the time of writing, the database contains over 1 116 original cipher keys originating from the 15th to the 18th centuries. They have been collected in libraries from European countries, mainly from Austria, Belgium, Germany, Hungary, Italy, the Netherlands, the UK, and the Vatican. 41% of the keys have been manually transcribed, following the transcription guidelines developed for historical ciphers (Megyesi, 2020). The distribution of keys throughout the centuries in this study is shown in Figure 2.

The digitized and transcribed cipher keys allow us to make large-scale studies of the morphology of ciphers, and make comparisons across time periods, geographic areas, and other information of interest. In order for our analysis to be as accurate as possible, we must first establish a transcription standard. This way, we ensure a stable and uniform basis to provide a reliable comparison across keys.

Our method makes use of plain text files (".txt") containing the transcription of the original key document. The transcription replicates the original document as closely as possible, both in terms of its structure as well as its content. In large terms, we follow the same guidelines (Megyesi, 2020) as those used in the DECODE database (Megyesi et al., 2019), and expand on them in order to adapt to the specific key structure.

Next, we describe the automatic process of the structural description of keys.

4.2 Automatic Structural Description of Keys

We provide automatic description of keys based on their transcription and extract statistical information from the transcription file b y u tilising a Python script that analyses the text file and returns a detailed analysis of its content, as described in Tudor (2019) and Tudor et al. (2020).

The first major section of our output focuses on the analysis of ciphertext symbols, beginning with the type of symbols used for encryption. Here we differentiate between 3 major types, namely Latin alphabet, digits, and graphic signs.

The next section of the output looks more in-depth into the internal structure of the ciphertext symbols, which we will refer to as unigraphs, bigraphs, trigraphs, and 4+graphs. What counts as unigraphs are usually digits, isolated letters or graphic signs.

We then move on to investigate plaintext units. Similarly to ciphertext, these are separated in unigrams, bigrams, trigrams, and 4+grams. We do

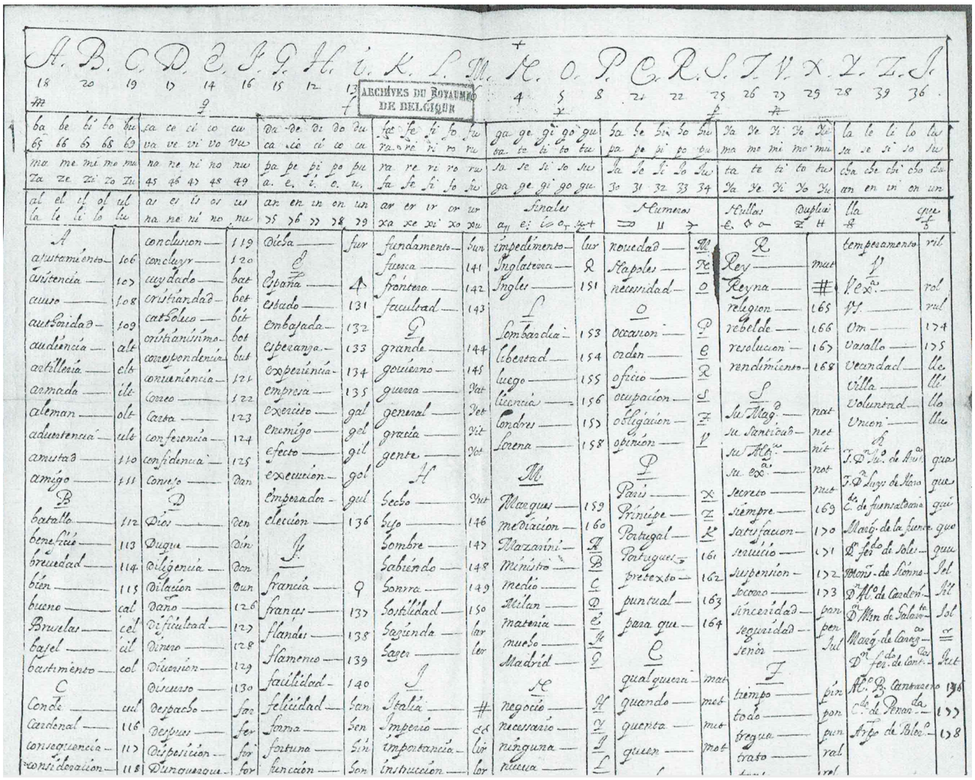


Figure 1: Example of homophonic key with variable length code (ARA Brus SEG inr.2chiffres1647-98 key3, 2018).

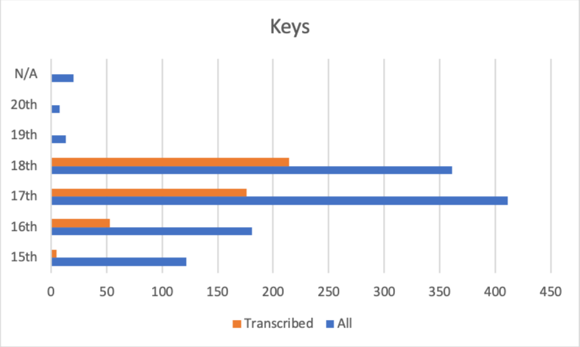


Figure 2: Key distribution throughout centuries in the DECODE database

add 3 additional ones, namely nulls, empty and cancellation signs.

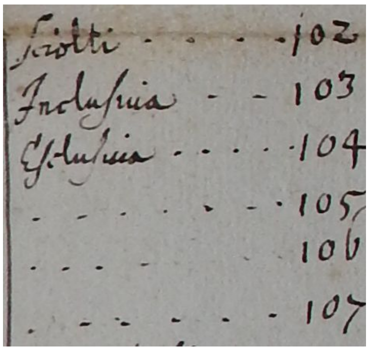
For the most part, the type of plaintext unigrams that we find in keys are either letters or digits, even punctuation in some cases. Bigrams and trigrams are commonly either non-lexical units (e.g. double letters that occur frequently in the language of encryption, such as “ll” or “ee” in English, syllables, morphemes etc.), or short function words (“at”, “for”, “to”, “and” etc.). Under 4+grams we include those units that consist of 4 or more elements, such as longer function words or nomenclature entries, which can consist of names, places, common words. Nomenclatures can also include words that are specific to the lingo used in the topic the key was designed for, such as army terms in military correspondence.

Even though nulls and empty elements might sound the same in theory, we differentiate between them in terms of their purpose; we look at nulls as entities that have been purposefully inserted by the author of the key to hinder the decryption process, while ”empty” entities are unintentional. The latter usually occurs in preset tables of codes that are later filled in with plaintext unit, but some codes are not assigned semantic significance, as shown in Figure 3.

The last category, cancellation signs, refers to those codes that not only do not carry significance, but also negate a certain number of codes in their vicinity, rendering them null as well, which we exemplify in Figure 4.

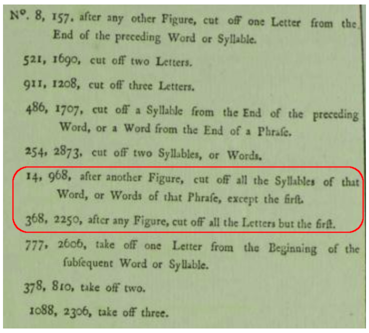
Once we described the code and plaintext structure, we can analyze the distribution of ciphertext symbols to plaintext elements from several different perspectives.

First, we establish the cipher type, such as simple, homophonic or polyphonic substitution, or a



ſciolti	102
Incluſiua	103
Eſcluſiua	104
	105
	106
	107

Figure 3: Excerpt from key containing empty entities.



N°. 8, 157. after any other Figure, cut off one Letter from the End of the preceding Word or Syllable.

521, 1690, cut off two Letters.

911, 1208, cut off three Letters.

486, 1707, cut off a Syllable from the End of the preceding Word, or a Word from the End of a Phraſe.

254, 2873, cut off two Syllables, or Words.

14, 968, after another Figure, cut off all the Syllables of that Word, or Words of that Phraſe, except the firſt.

368, 2250, after any Figure, cut off all the Letters but the firſt.

777, 2606, take off one Letter from the Beginning of the ſubſequent Word or Syllable.

378, 810, take off two.

1088, 2306, take off three.

Figure 4: Excerpt from key containing cancelling signs.

mix of these. Then, we also look into the length of the codes used for encryption, be it fixed or variable. We also indicate how many of the codes are used for encrypting each level of n-grams and similar plaintext representations.

The last significant portion of the automatic analysis looks into the specific distribution of ciphertext to plaintext units for each section of the key, separated into alphabet, nomenclature, nulls, empty, and cancellation signs.

The final step is to output all of the specifications for each key into a global csv file.

5 Results

In the transcriptions and their structural descriptions, we study the entities that were chosen to be encoded, the codes themselves, and the relation between the codes and the plaintext entities.

5.1 What is encoded

Given the plaintext entities, we analyze them with respect to the number of characters and their types as well as the language(s) they represent.

5.1.1 Plaintext

Plaintext entities, such as characters, syllables, words, or sentences that are described to be coded in the keys, can be rather short, like a size of the alphabet of ca 20-30 entities, to several hundred like a long list of a nomenclature. 72% of the keys contain over 100 different plaintext entities, of which all contained the plaintext alphabet and an additional list of word-like elements, such as syllables, function words, frequent content words, and named entities. We present the distribution over the keys on the basis of the length of plaintext divided into unigrams of length 1, bigrams of length 2, trigrams of length 3, and 4+ grams of length 4 or more in Figure 5.

5.1.2 Languages

The involved languages that we find among the plaintext elements in the transcribed keys are: German (DE), English (EN), Spanish (ES), French (FR), Hungarian (HU), Italian (IT), and Latin (LA). See Figure 6 for an overview. Keys may encode entities not only in one but also in several languages. The involved languages depend on the time period, the geographic area of the corresponding people, and the lingua franca of that time.

Almost 30% of the keys contain several languages, which is hardly surprising due to the well-known property of code-switching in historical texts. Latin occurs in almost half of the keys, followed by English, French, and Italian.

Figure 6 illustrates the distribution of the languages, occurring as the only language, or as one of several languages.

5.1.3 Nulls

Keys might also contain nulls, elements that are fake codes without any underlying plaintext. Ca 32% of the keys contains one or several nulls. How many nulls are used vary across keys, as illustrated in Figure 7. Nulls can be listed as a finite set of numbers, or defined in cleartext corresponding to several hundred codes.

5.1.4 Empty plaintext

Keys are not necessarily complete, sometimes we find a list of codes in some structural manner without any corresponding plaintext. In fact, 19% of the keys contained some empty plaintext elements ranging from 1 up to 2500 empty places.

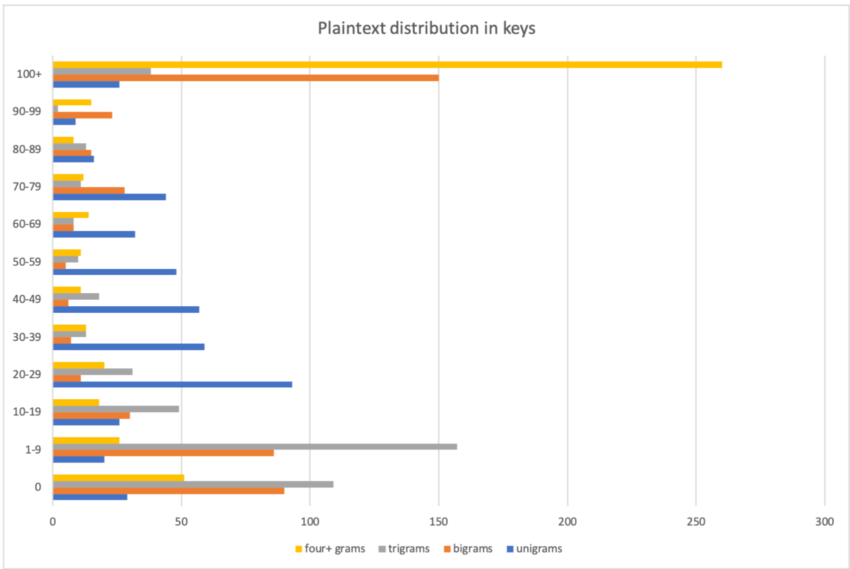


Figure 5: The distribution of plaintext entities of variable length: unigrams, bigrams, trigrams and four+grams.

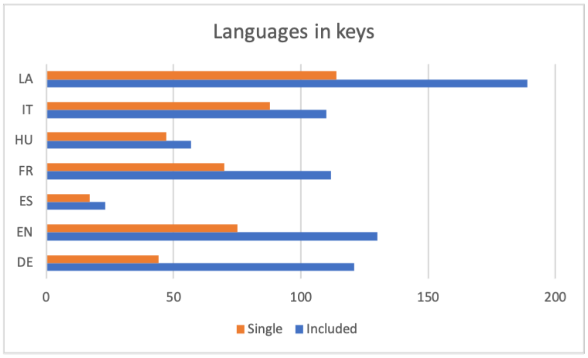


Figure 6: The distribution of languages in keys: blue marks the number of keys the language occurs, and orange marks the number of keys where the language is the only one used.

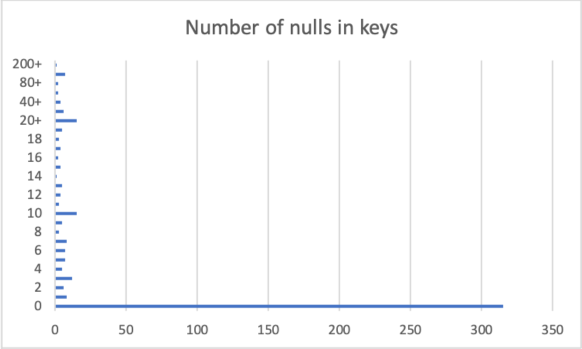


Figure 7: The number of nulls in keys.

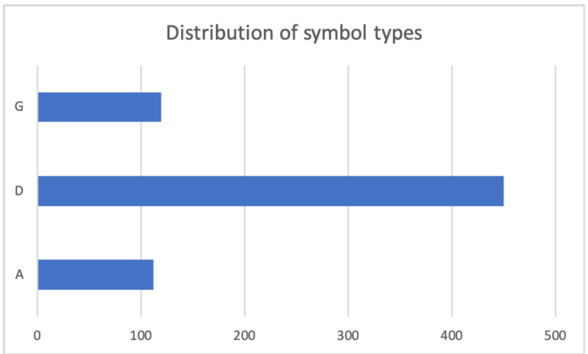


Figure 8: Symbols in keys: A=alphabet, D=digit, G=graphic sign.

5.2 How it is encoded

Encoding systems have been varying over time, and here we try to summarize the encoding systems in terms of symbols and code types.

5.2.1 Symbol systems

We distinguish between alphabets such as Latin and Greek, digits, and graphic signs such as alchemical symbols or Zodiac signs. The great majority, 98% of the keys contain digits (0-9) and only 25% use codes expressed as alphabetical characters or graphic signs, as show in Figure 8. In 72% of the keys, the only symbols that are used are digits. The remaining ones combine digits with alphabets, oftentimes Latin letters. Graphic signs occur only in few keys. The distribution of symbol sets across keys is illustrated in Figure 9.

5.2.2 Code types

85% of the keys contain codes of variable length, and only 15% have a fixed length code, mostly 2-

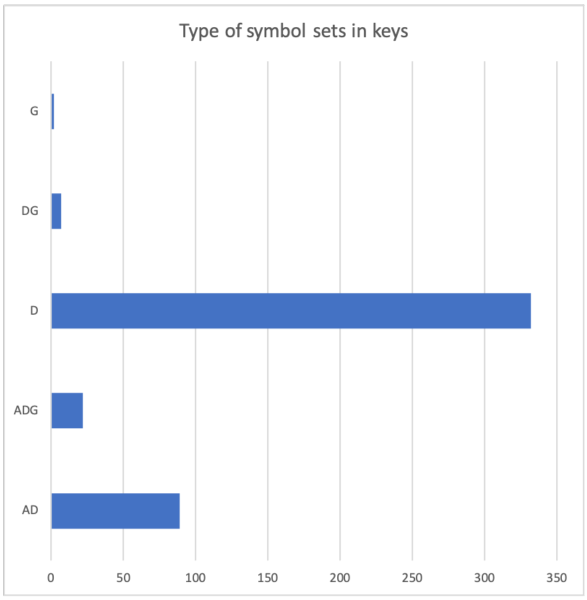


Figure 9: The combination of symbols in keys: A=alphabet (Greek or Latin), D= digits, G=Graphic signs.

digit codes.

Code types vary across the plaintext entity types not only in length but also in type. For example, it is common that the alphabet is encoded as 2-digit homophonic codes while nomenclatures have 3-digit simple substitution code system. Thus, the distribution of code types vary not only across but within a single key. In Figure 10, we show the code types for alphabets, nomenclatures as well as for nulls. Typically, while several characters in alphabets are often encoded with two or more codes resulting in a homophonic substitution, elements in nomenclatures tend to have one code only.

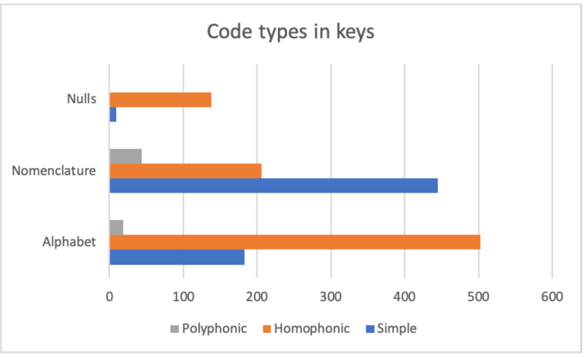


Figure 10: The number of nulls in keys.

Given the various code types in a key, we analyze the type given their components, see Figure 11. Homophonic substitution is far most popular either on its own or combined with simple substitution. Purely polyphonic or simple substitution occur seldom, and if they do they are often combined with homophonic codes. In a partly homophonic, partly polyphonic cipher key, for example, some elements of the plaintext alphabet are substituted by several cipher text characters (that is the homophonic component), while some elements of the nomenclature are substituted by the same code (that is the polyphonic part).

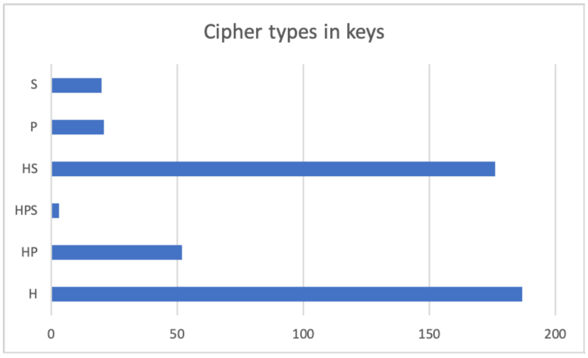


Figure 11: The distribution of cipher types in keys.

5.2.3 Cancellation

Cancellation, i.e. codes that define elements that should be removed in the plaintext, are not very common but appear in ca 4% of the keys, and not until the 18th century. Cancellation can be defined in many different ways, not only as codes but as in cleartext describing how cancellation is performed, which can be seen in Figure 4.

5.3 Trends

Given the keys' structural description, we can investigate the trends throughout the centuries concerning what has been chosen to be encoded and how. Since the set of structurally described keys that have been automatically extracted from transcriptions originate from the 17th to 18th centuries, (see the orange bars in Figure 2), we manually extracted structural information from 251 keys without any transcriptions originating from the 15th and 16th centuries. In total, we investigate 700 keys. In the subsequent paragraphs, we report some of our findings about the main trends of key structure over the centuries.

The usage of the types of symbols that have been chosen for encoding varied over the centuries, as illustrated in Figure 12. While alphabetical characters, digits, and graphic signs were evenly distributed in the 15th century, we can see a clear increase in tendency to use digits as the main encoding at the expense of Latin letters or graphic signs, which we can hardly find in keys from the 18th century.

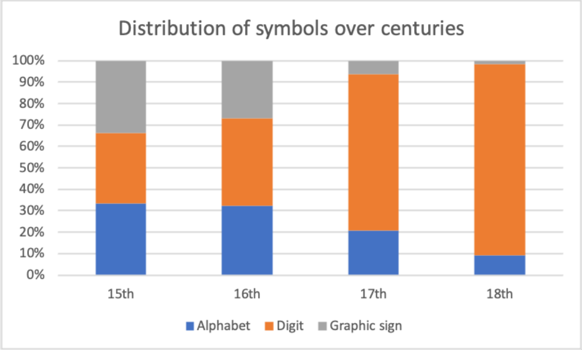


Figure 12: The distribution of symbols over time.

The symbol systems used in keys often contain a combination of digits, letters, and graphic signs. In the 15th century, all three types of symbols were combined in almost all keys, but this eclectic symbol set have been reduced in the 16th and 17th centuries in favor of digits in combination with Latin letters. The distribution of various symbols sets over centuries is shown in Figure 13.

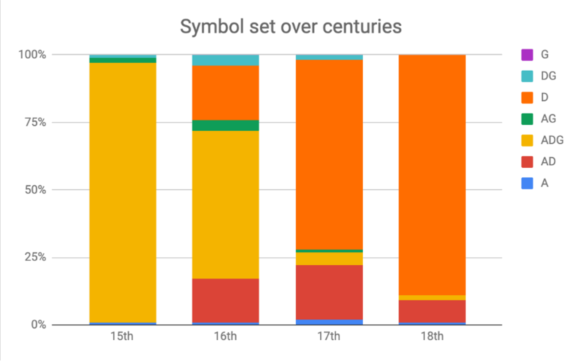


Figure 13: The distribution of symbols set containing (a combination of) Latin alphabet (A) digits (D) and/or graphic signs (G) over time.

The usage of the length of the codes also varies over time, as illustrated in Figure 14. The great majority of keys contain codes of variable length and the length typically differ between alphabetical elements, nomenclatures, as well as nulls.

To investigate the type of codes in more detail, we analyzed the type of codes used for alphabets and nomenclatures separately, distinguishing between simple, homophonic, and polyphonic distributions.

Encodings of alphabetical signs were mostly homophonic, as shown in Figure 15. Quite surprisingly, however, we can see a decrease in favor of simple substitution which became more frequent in the 17th and 18th centuries. This might be due to the increase in the size of the nomenclatures over time.

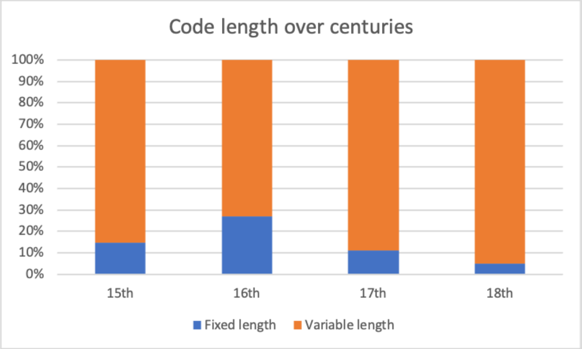


Figure 14: The distribution of fixed vs variable length codes over time.

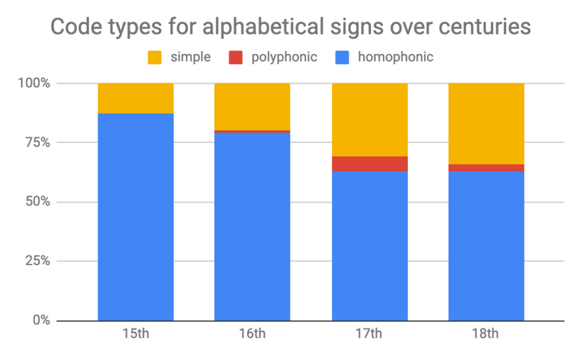


Figure 15: The distribution of code types for alphabetical signs over time.

Encodings of nomenclatures, on the other hand, are mostly simple substitution, but homophonic and even polyphonic encodings become standard in the 17th and 18th centuries, see Figure16.

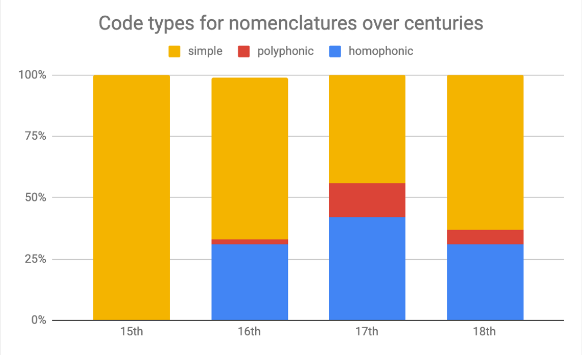


Figure 16: The distribution of code types for nomenclatures over time.

The usage of nulls in keys also varied over time, as illustrated in Figure 17. While nulls have been frequently occurring in keys, i.e. 96% of keys included nulls in the 15th century, we find nulls in 27% of the keys in the 18th century. The nulls were in the great majority of the cases (94%) encoded with at least two possible codes.

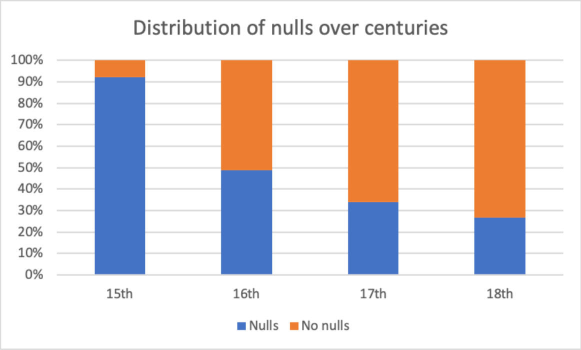


Figure 17: The distribution of nulls over time.

Clearly the usage of nulls decreased over time, and codes for cancellations have not been used until the 18th century.

6 Discussion

One surprising result that emerged after looking more in-depth into the structure of keys was a rather large amount of nomenclatures that use homophonic substitution. This was particularly interesting to investigate as the phenomenon was mostly visible in the keys that were automatically analysed by our script, and not nearly as much in those that passed through a manual analysis. Upon further inspection, we were able to isolate two main factors that cause this phenomenon.

- Frequent bigrams that can occur in a language, such as ”ae”, ”oe”, ”au” in Latin (NAH G15 CAPS C FASC 43 18, 2018), or ”gy”, ”cz, ”sz” in Hungarian (NAH G15 CAPS C FASC 43 40, 2018), can often be encoded by means of homophonic substitution. In our analysis, we consider bigrams to be part of the nomenclature, whereas some keys include them on the same level as the alphabet. For example, if at first sight it seems like a key is using homophonic substitution at alphabet level and simple substitution at nomenclature level, we may discover that the author included some bigrams which are encoded by 2 or more codes at alphabet level, which in turn makes the nomenclature homophonic as well.
- Some very large tables (100+ ngrams) can use homophonic substitution only for a few entities in the nomenclature table, oftentimes those that are used most frequently in the language (e.g. ”aller” - *to go*, ”peu” - *few* in French (KHA_ A29_ PWIV_ inr301_ B, 2019)) or for the purpose of the correspondence (e.g. titles, such as ”The King”, ”His Majesty” (ÖStA HHStA Stk Int Chiffrenschlüssel fasc 20 kt14 152, 2020)). These tend to be rather hard to spot with the naked eye among the multitude of plaintext entries.

This only goes to show that automatic methods are a lot more reliable when it comes to picking up subtle elements of key structure.

All in all, given the results presented above, we cannot draw certain conclusion about how the keys have been used — we can only see what the intentions of the key creators have been. More ciphertexts and systematic studies would be needed about the actual usage of the keys.

7 Conclusion

In this paper, we investigated 700 cipher keys from the 15th to the 18th centuries, all originating from European archives and libraries. We described the keys’ internal structure and their morphology looking at what has been chosen to be encoded and how over four centuries. In particular, we described the type of the symbol set and the code structures used, and the changes and trends of each century.

Not surprisingly, we found that keys evolved over time, and their structure changed in various ways. While codes with various symbols including alphabets, digits, and graphic signs were dominating in the 15th century, using digits only became more frequent to became the standard in the 18th century. The codes varied in length for alphabetical signs and nomenclatures throughout all centuries while codes with fixed length seemed to be most popular in the 16th century. Coding alphabetical signs were mostly homophonic, but simple substitution of letters became more frequent as the length of the nomenclatures increased over time. Nomenclatures, however, were mostly encoded as simple substitution. Nulls have been frequently used in the 15th century and decreased significantly over time. Cancellation as phenomenon became popular in the 18th century.

Our results presented in this paper are based on 700 original keys from four centuries, but the dataset is rather opportunistic — we took what was available to us — the data is not evenly distributed across geographic areas, countries, or senders/receivers. In the future, we intend to extend our collection with more keys from a large number of places, and make in-depth analyses of the nomenclatures and the involved plaintext languages.

Acknowledgments

We would like to thank Anne-Simone Rous and Karl de Leeuw for generously sharing their keys with the public through the DECODE database.

This work has been supported by the Swedish Research Council, grant 2018-06074, DECRYPT – Decryption of Historical Manuscripts.

References

ARA Brus SEG inr.2chiffres1647-98 key3. 2018. Reproduced image from Algemeen Rijksarchief, Secretairerie d'Etat et de Guerre, inv.nr. 2, DECODE link: https://cl.lingfil.uu.se/decode/database/record/960 .

J. P. Devos. 1950. *Les chiffres de Philippe II (1555-1598) et du Despacho Universal durant le XVIIe siècle.* Brussels: Académie Royale de Belgique.

David Kahn. 1996. *The Codebreakers: The Comprehensive History of Secret Communication from Ancient Times to the Internet.* Scribner, New York, NY.

David Kahn. 2008. The future of the past—questions in cryptologic history. *Cryptologia*, 32:56–61.

KHA_A29_PWIV_inr301_B. 2019. Reproduced image from Koninklijk Huisarchief (KHA), Prins Willem IV, inv.nr. 301 B, DECODE link: https://cl.lingfil.uu.se/decode/database/record/1024.

Benedek Láng. 2018. *Real Life Cryptology: Ciphers and Secrets in early modern Hungary*. Atlantis Press, Amsterdam University Press.

George Lasry, Beáta Megyesi, and Nils Kopal. 2020. Deciphering Papal Ciphers from the 16th to the 18th Century. *Cryptologia.*

Beáta Megyesi, Nils Blomqvist, and Eva Pettersson. 2019. The DECODE Database: Collection of Ciphers and Keys. In *Proceedings of the 2nd International Conference on Historical Cryptology, HistoCrypt19*, Mons, Belgium, June.

Beáta Megyesi. 2020. Transcription of Historical Ciphers and Keys. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt20*, Budapest, Hungary, June.

Aloys Meister. 1902. *Die Anfänge der modernen diplomatischen Geheimschrift.* Paderborn: Ferdinand Schöningh.

Aloys Meister. 1906. *Die Geheimschrift im Dienste der Päpstlichen Kurie von Ihren Anfängen bis zum Ende des XVI. Jahrhunderts*, volume 11. F. Schöningh.

NAH G15 CAPS C FASC 43 18. 2018. Reproduced image from National Archives of Hungary, G15 Caps. C. Fasc. 43. 18., DECODE link: https://cl.lingfil.uu.se/decode/database/record/600 .

NAH G15 CAPS C FASC 43 40. 2018. Reproduced image from National Archives of Hungary, G15 Caps. C. Fasc. 43. 40., DECODE link: https://cl.lingfil.uu.se/decode/database/record/581 .

Ludwig von Rockinger. 1892. Über eine bayerische Sammlung von Schlüsseln zu Geheimschriften des sechzehnten Jahrhunderts. *Archivalische Zeitschrift*, pages 18–92.

Crina Tudor, Beáta Megyesi, and Benedek Láng. 2020. Automatic Key Structure Extraction. In *Proceedings of the 3rd International Conference on Historical Cryptology, HistoCrypt20*, Budapest, Hungary, June.

Crina Tudor. 2019. Studies of Cipher Keys from the 16th Century: Transcription, Systematisation and Analysis. Master thesis in Language Technology, Uppsala University, Sweden.

ÖStA HHStA Stk Int Chiffrenschlüssel fasc 20 kt14 152. 2020. Reproduced image from Österreichisches Staatsarchiv, Haus-, Hof- und Staatsarchiv, Staatskanzlei Interiora, Chiffrenschlüssel, Kt. 14. Fasc. 20. f 152., DECODE link: https://cl.lingfil.uu.se/decode/database/record/1397.

Cryptographic postcards

Tobias Schrödel
IT Security & Awareness
Munich, Germany
tobias@schroedel.email

Abstract

This document is about postcards written in code or cipher, which is a field in historic cryptography with only few information available. While military and business related cryptography is examined in depth, these cards give insight in the civil use of pen & paper ciphers. The author has scanned and evaluated his private collection of more than 400 encrypted postcards. Although statistics and data are not representative, this paper allows a first classification and should encourage other collectors to contribute data as well.

1 Introduction

According to a base of more than 400 encrypted postcards, private correspondence in code was mainly performed in the early 20th century. Nevertheless, encrypted postcards of the 19th century are rare but also exist. While historical cryptographic books, letters, and documents have been evaluated and inventoried since decades, this has not been the case for encrypted postcards. About these, the reader can only find minimal information.

2 Finding encrypted postcards

Historic postcards can be found in different places. But besides special fairs, flea markets, and online auctions, there is no typical marketplace for historic postcards. Especially not for encrypted postcards.

2.1 Sources

Bookstores selling antiquarian books sometimes offer historical postcards as well. The author notes that private stamp collectors quite often also collect postcards. They sometimes offer them on flea markets. Finding encrypted postcards in packs of hundreds of cards is like searching the needle in a haystack. The author notes, that most of the (private) collectors are of higher age and are not aware what an encrypted postcard is. Their reason for collecting is either the stamp or the motive of the postcard (city, area, military, or cards with applications). Therefore, many postcards written in code or cipher may rest unnoticed in such big packs of cards. Some collectors have mentioned that the interest of younger people in ancient postcards is minimal and that the cards have no high value in general. It is, therefore, possible, that the cards will be disposed once the collector dies.

Professional stamp and postcard dealers can be found on the Internet.[1] They offer thousands of catalogued cards often including a search function. The author unregularly performs checks on search terms like "Geheimschrift" (German for "secret writing"), "written in code", or "cipher". The hit rate over the last approximately ten years was minimal (below 5 pieces). According to akpool, a German vendor with over 1 million cards and a sales quantity of 400.000 cards per year, they have no interest in marking cryptographic cards, as this is not asked for by buyers.

A very promising place for finding encrypted postcards is eBay. The reason why this platform is popular for the sale of cryptographic postcards is unclear. It is possible, that once a layman identifies a postcard written in cipher, due to "strange" symbols, it is acknowledged as something special and put up for sale (maybe in expectation of a high price). This assumption is

[1] https://www.ansichtskarten-center.de

https://www.delcampe.net

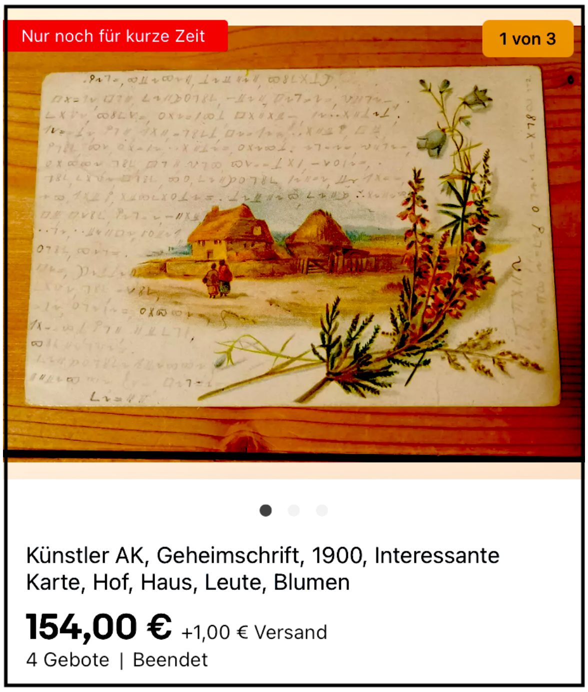
https://www.akpool.de

hardened due to the many false-positives on eBay. Private sellers very often sell postcards written in shorthand as “a very rare card written in code”. However, shorthand was very common in the early decades of the past century and is no cipher or code.

2.2 Prices

The prices for encrypted postcards vary from 1€ to 250€ and sometimes even more. These cards are collector items and as such the price depends extremely on the number of collectors, interested in it. For a regular and “common” encrypted card, the author normally payed up to a maximum of 30€. For special cards with beautiful symbols or cards belonging to an existing series, the author is willing to pay more.

Average prices for encrypted postcards have increased massively in the past three to five years. Even unattractive cards nowadays regularly achieve prices of over 50€. The reason for the increase is unknown and may be a sign of more collectors. Proper statistics on payed prices for the cards described in this paper are not available.



Künstler AK, Geheimschrift, 1900, Interessante Karte, Hof, Haus, Leute, Blumen

154,00 € +1,00 € Versand

4 Gebote | Beendet

Figure 1: An encrypted postcard from 1900 sold on eBay Germany for 154€ (June 20th, 2021). There were four bidders. At least two were willing to pay more than150€ for this postcard.

2.3 Other resources

The author knows only of one other collector (in Thuringia) specialized in encrypted postcards.

Another source for encrypted postcards is Klaus Schmeh’s blog “Cipherbrain” aka “Klausis Krypto Kolumne” where more than one hundred postcards have been described over the past years.[2]

3 The collection

The following paragraph will provide numbers and statistics about the evaluated card collection. Although the collection has no focus and spans over all periods and locations, the numbers are not representative. The author has mainly searched and bought cards in Germany and mainly from German websites and sellers. In addition, not all cards ever on sale are in the evaluation, as there are other collectors, and the author has lost several eBay auctions to other people.

3.1 Quantity

The author’s collection contains 428 encrypted postcards. This includes 38 cards most probably written in shorthand, which will be left out or marked in the statistics. Shorthand is no encryption method. But there are many abbreviations in shorthand that are often personalized and therefore difficult for others to identify. So a few cards cannot clearly be identified as code or shorthand by the author. Some of these cards also combine shorthand with number codes. Anyway, there is a certain uncertainty in the numbers.

3.2 Period of time

The date of an encrypted postcard can be evaluated in two different ways. Sometimes, there is a written date on the card. However, it is more accurate to read the postmark on a devalued stamp.
Sometimes a handwritten date differs from the postmark date. This happens when the card has been sent days after writing. For the following statistics, the author used whatever date was the

[2] https://scienceblogs.de/klausis-krypto-kolumne/

most clearly readable one. If none was readable, they do not appear in the following table.

Written dates: Handwritten dates appear in cleartext just like on unencrypted postcards. The author knows of no postcard, where the date was also part of the encryption. (This applies only to cards, that are already deciphered).

Postmark dates: Some postcards are postmarked twice. The first punch was made, when the card was delivered to the sender's post office and a second time, when it arrived at the recipient's post office.

Period	Count
1880-1889	6
1890-1899	21
1900-1909	148
1910-1919	161
1920-1929	4
1930-1939	1
1940-1949	3
1950-1959	0
1960-1969	2

Table **1**: Overview by decade

The date of the postmarks on the cards do not follow a standard format. Therefore, some dates cannot be identified with certainty, especially when only two digits and not four represent the year. For example, 01-03-14 can be read either as 14th of March 1901 but also as 1st of March 1914. As philatelist catalogues (and websites) state in which period a specific stamp was in use, the stamp (if present) can help to clarify or at least to narrow the date of the card.

The number of encrypted cards during WW1 and WW2 is lower than in the preceding years. The author assumes, that this is (at least partially) a consequence of the ban of the use of codes during the war in many countries. However, for the period of WW1 the collection contains seven encrypted cards (shipped within Germany) and a large set (same sender and recipient) of 140 cards shipped within Hungary. For the period of WW2 there are only two encrypted cards available in the collection (plus one in shorthand). It is not known, why these two cards (1940 within Nazi Germany and 1942 within France) were not removed by censorship.

The above table clearly indicates a significantly higher number of encrypted postcards in the time between 1900 and 1920. This may correlate to the availability of books about simple cryptography for the public (and not military nor government). These books were mostly addressed to lovers, who wanted to correspond secretly. Many are listed on the cryptobooks-website, and some are available on Google-books.[3] The method(s) described in these books are normally easy-to-use pen & paper ciphers, mainly monoalphabetic substitutions (simple MASC). Although the authors of these books claim "absolute security" for the messages, a MASC was no problem for an experienced cryptanalyst at that time. However, the method probably fulfilled its task to hide the message on a postcard against curious family members and the postman.

Geheimschrift für Liebende
Leichtfaßliche und doch ganz sichere Anleitungen von
Deutobold Symbolowitsch Mystifizinsky
Dresden 1923
Rudolph'sche Verlagsbuchhandlung
Sicherster Schutz des Briefgeheimnisses
Eine unbedingt sichere Geheimschrift für Liebende und Freunde, Geschäftsleute und Politiker etc.
Von unbefugten nicht zu entziffern.
von Emil Katz.

Figure **2**: Books aimed to lovers about encrypting private correspondence. Left: Geheimschrift für Liebende (Erwin Le Mang, 1923). Right: Sicherster Schutz des Briefgeheimnisses (Emil Katz, 1901).

[3] https://cryptobooks.org

https://books.google.com

3.3 Sender & Recipient

The following paragraph gives an overview of the origin and destination of all evaluated encrypted postcards.

Some cards can be assigned to be part of a set. A set is defined as postcards from one sender to the same recipient. They normally used the identical cipher over time. In the examined collection of 428 postcards, 347 cards are part of 24 sets. While two large sets of 45 and 120 pieces are outstanding, the average number of cards in a set is 6.6.

Sender: The country and city of the sender can only be obtained, when the punched stamp on the card is readable, as it shows the name of the city (post office). If the recipient's address does not contain a country name, it is assumed, that the card was shipped within one country, as only international cards require the destination country to be named. Sometimes, the originating city can be derived from the card itself. E.g., when the printed picture on the card states "Greetings from …". However, this is an assumption.

The author found some ambiguities, especially from cards around the time of WW2. There were cases, where cards were sent within Germany with German stamps. However, the origin and/or destination city is today in Poland. In the following table, these cards are counted for the country of the time, they were sent.

Recipient: The recipients address is mostly easy to find out. It has to be written on the card, if the card was shipped. Anyhow, a small number of cards has not been shipped. It is mostly unclear, whether they were dropped personally, sent in an envelope, or just used as a note.

Country	From	To
HU Hungary	171	171
DE Germany	159	165
US United States	27	25
UK United Kingdom	17	17
AT Austria	12	9
FR France	6	7
BE Belgium	4	3
CH Switzerland	2	5
CZ Czech Republic	1	1

Table **2**: Country of origin and destination

165 of the Hungarian cards are part of only two sets while all other countries represent a variety of senders and recipients including smaller sets.

3.4 Stamps

Collectors removed the stamps on some cards. In most cases, this led to unreadable parts or missing postmarks made by the post office. In these cases, it was much harder to find out the shipping date as well as the origin city – if even possible.

However, many stamps remain on the cards, and it is obvious, that they were quite often not placed straight in the right corner of the postcards. Some of the stamps were put on in a 45° angle or even upside down. The author wants to note, that the alignment of the stamps was often used for a short note, such as "I miss you", "Write back soon" or "Forever yours". The interested reader can find different instructions and meanings for the stamp placement using Google or other Internet search engines with the keywords "language of stamps".

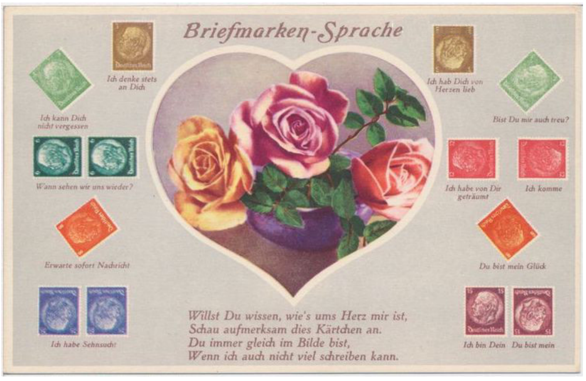

Briefmarken-Sprache
Ich denke stets an Dich
Ich hab Dich von Herzen lieb
Ich kann Dich nicht vergessen
Bist Du mir auch treu?
Wann sehen wir uns wieder?
Ich habe von Dir geträumt
Ich komme
Erwarte sofort Nachricht
Du bist mein Glück
Ich habe Sehnsucht
Willst Du wissen, wie's ums Herz mir ist,
Schau aufmerksam dies Kärtchen an.
Du immer gleich im Bilde bist,
Wenn ich auch nicht viel schreiben kann.
Ich bin Dein
Du bist mein

Figure **3**: A postcard explaining the "language of stamps". Date unknown, according to the shown stamps probably around 1935

4 The Ciphers

The used cipher type is only known for sure, when the postcard has been decrypted or if the used cryptographic method is obvious (e.g.

pigpen). In most cases, a simple monoalphabetic substitution cipher (simple MASC) was used.

4.1 Cipher type

This tables shows, what kind of encryption method is used on the cards – if known or obvious.

Cipher type	Count
Simple MASC	71
(Shorthand)	38
Pigpen	20
Morse code	8
Anamorph writing	2
Square writing	2
Mirror writing	1
Wigwag	1
Caesar shift	1

Table **3**: Cipher types

4.2 Used symbols and characters

In most cases, numerical substitutions are used. However, some cards use symbols or standard Latin characters. The following table shows substitution with symbols as the leading type. The reason is that two sets from Hungary contain 165 cards with symbols. To get a more meaningful statistic, the reader might want to subtract them.

Special encryption methods such as square writing (writing horizontal and vertical) or hidden messages (under stamp) are not counted.

Symbols used	Count
Symbols	199
Numbers	132
Characters	3

Table **4**: Used symbols for the cipher text

4.3 Plaintext

For 76 cards, the plaintext is known. The length of these messages ranges from 27 to 1.955 characters. The average length of these messages is 297 characters.

Length of message	Count
1-99	24
100-199	17
200-299	11
300-399	9
400-499	3
500-599	5
600-999	2
> 1.000	5

Table **5**: Message length

4.4 Language

The language of the encrypted message is a very interesting and an important part regarding cryptography. However, there were no surprises. For the evaluated collection of postcards, the language of the known plaintext matches the language spoken in the sender's and/or recipient's country. All evaluated cards within Germany as well as from Germany to German speaking countries (like Austria) were written in German language. Cards within Hungary were written in Hungarian language, cards within Czech Republic were written in Czech. A card from Paris (France) to San Francisco (USA) was written in French. Only one card differs. It was sent from Finland to Russia in 1906 and is written in German language.

5 Conclusion

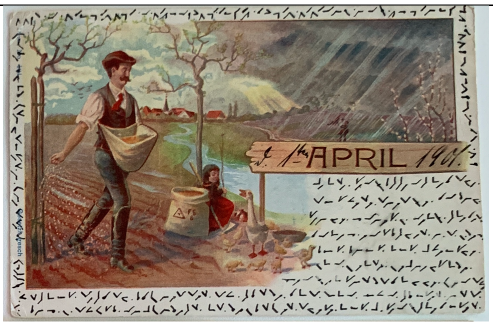
This collection of cards soon is subject to a more detailed analysis within the DECRYPT[4] project. Therefore, all postcards were scanned (both sides) and uploaded to the DECODE[5] database. Other collectors are invited to do so as well.

A complete publication of the scans is planned shortly to allow students and any other interested people to participate in the exciting world of "postcard cryptology".

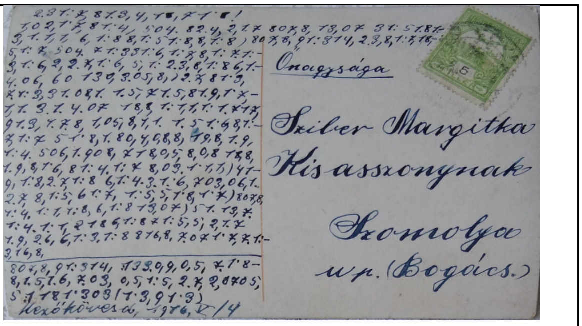
[4] https://www.de-crypt.org/

[5] https://de-crypt.org/decrypt-web/

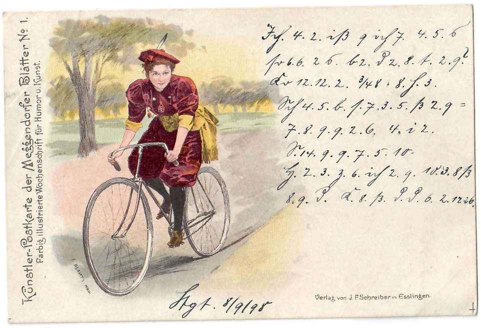
6 Addendum



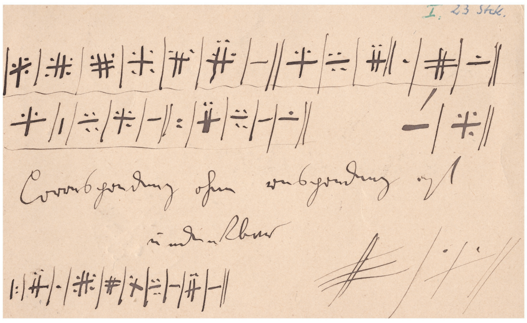
Card with a MASC from a set of twelve cards in German language



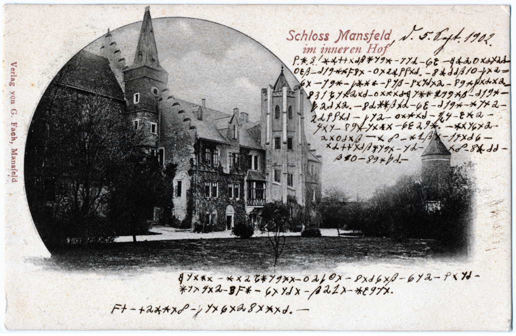
A numerical code and a stamp in 45° angle dated May 4th, 1916


Künstler-Postkarte der Meggendorfer Blätter No 1.
Farbig illustrierte Wochenschrift für Humor u. Kunst.
Verlag von J. F. Schreiber – Esslingen

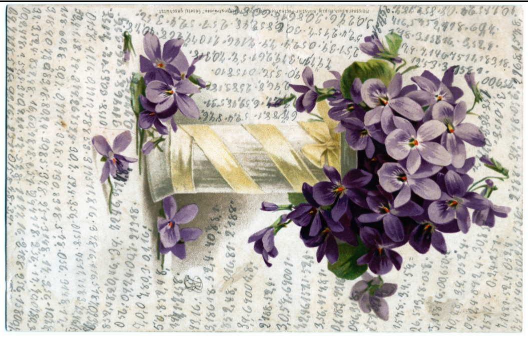
Mix of a numerical code with shorthand from 1898



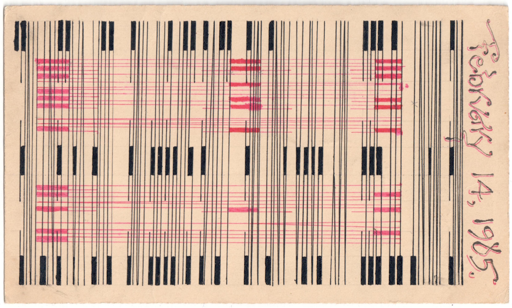
Postcard to a Bavarian princess from her brother (1890)



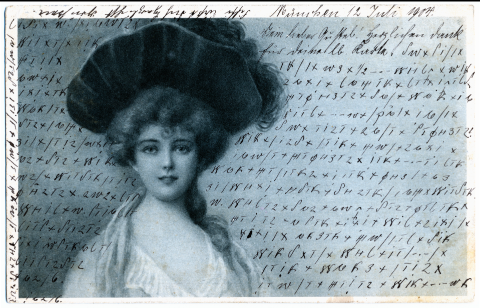
The daugther of an Earl has a message for the shoemaker’s son. She wants to meet him on Sunday and hopes for good weather (1902)



MASC using numbers



Postcard with anamorph writing and Valentine’s day greetings



Beautiful, encrypted card from July 1907

The American Army Bombe

Dermot Turing
Visiting Fellow, Kellogg College
60-62 Banbury Road
Oxford, OX2 6PN
dermotturing@btinternet.com

Abstract

This paper presents the U.S. Army's version of the anti-Enigma cryptanalytical bombe machine, which has not previously received attention in the literature on Enigma. Its unique features and applications are discussed, and the paper describes the sensitive context of the machine's development and deployment.

1 Introduction

In 1941, many months before the attack on Pearl Harbor, a courageous act took place in which the United States and the United Kingdom agreed to share their achievements in the sphere of cryptanalysis. Two years later this tentative, awkward and unstable agreement was nearly rescinded. The cause of the near-rift was the desire of the British to inspect certain cryptanalytical and cryptographic devices being developed for the U.S. Army at Bell Labs. The cryptanalytical devices in question included the U.S. Army bombe.

While the literature covers each of the Polish bomba (Link, 2009; McCarthy, 2019), the British 'Turing-Welchman' bombe (Davies, 1999; Carter, 2010) and the U.S. Navy's 'Desch' bombe (Erskine et al., 2002; DeBrosse and Burke, 2004), the U.S. Army bombe has largely been ignored. The fact that this branch of the bombe family has been overlooked is perhaps remarkable, given its innovative features: its significance may go far beyond a mid-war spat between intelligence services about who could see what. The purpose of this paper is to begin to fill the gap with a description of the U.S. Army bombe and to open a discussion on the role of this interesting piece of equipment.

The reader is assumed to be familiar with the standard Wehrmacht version of the Enigma cipher machine. As to bombes, their object was to identify the secret 'key' or set-up of the Enigma machine. In very brief summary, the British bombe tested all 26×26×26 possible positions of three chosen coding rotors to determine if a single starting-position of the rotors could consistently transform a segment of guessed-at plaintext (called a 'crib') into an observed, intercepted message. Additionally, the machine identified one possible pairing of letters effected on the Enigma machine's plugboard. When a logically consistent rotor orientation arose, the bombe machine would stop, allowing the operator to identify that orientation and the single plugboard pairing.

By the time of the historic visit of four Americans to Bletchley Park in January 1941, the bombe was already making a contribution to the solution of Enigma messages and thereby to the wartime intelligence picture. One outcome of the American visit was that the British would – albeit with some reluctance and delays – share the particulars of their bombe-based attack with the Americans. (Sherman, 2016)

2 The American Army Solution

Much has been written about the development by the US Navy of a four-rotor bombe at the National Cash Register Corporation in Dayton, Ohio under Joe Desch. However, that was not the only American response to the challenge of Enigma. Within two weeks of the launch of the Desch project, William F. Friedman, then the U.S. Army's principal cryptanalyst, put forward his own argument for autonomous American cryptanalytic machinery for deployment against Enigma. Relying on the British could be unwise: the three-rotor bombes would be of no use if the German forces rolled out four-rotor Enigma modifications to their land and air forces; and 'should a few well-placed bombs destroy the present three buildings in which the Enigma-solving machinery is housed, all Enigma solution will stop…. Consequently, it appears vital that we take immediate steps to establish an Enigma solution unit of our own.'[1]

To implement the new plan, the U.S. Army turned to Bell Telephone Laboratories.[2] Bell Labs was the research offshoot of the American Telephone and Telegraph corporation, which contributed many technological breakthroughs in the mid-twentieth century (Gertner, 2012). Among the galaxy of intellectual stars in the Bell Labs sky were George Stibitz and Claude Shannon. In 1937, Stibitz had created a digital adding machine, stimulating the development of digital computing at Bell Labs. In the same year, Shannon had discovered that Boolean algebra and electrical circuitry shared features which enabled mathematical and logical functions to be represented in physical form through switching. It seems, though, that the idea of using electrical switching for the U.S. Army bombe originated with Lt Leo Rosen of Friedman's team, which led to Bell Labs being chosen for the Army's project.

2.1 The technology

The U.S. Army concept for a bombe was to omit the rotating parts of the British and Desch bombes, which wore out, needed specialist engineering, and were limiting components in that physical movement takes time and therefore slows the operation of the machine. Instead, the army bombe would rely on relays and switching. Relays are simple electromechanical instruments, which rely on electric current to generate a magnetic field which pulls into place an electrical contact, thus switching the path of a current in a new direction.

The U.S. Army bombe used relay technology to replace rotating drums by switching. 'M' units, also called 'Multiple Paths', to direct electricity into fixed-wire circuitry imitating the internal wiring of Enigma rotors in a progressive fashion, so that each entry-connection on a 'rotor' would be connected in succession, with suitable switching to copy the stepping pattern of the 'middle' and 'slow' rotors of an Enigma machine.

To bring about this progression, the continuous supply of voltage of traditional bombes was replaced by pulses of electricity. Each pulse not only coursed through the circuitry to carry out the logic test designed by Alan Turing for the British bombes, but operated on the relays in the M units so as to change the electrical path to be followed by the succeeding pulse.

Replacing the rotating drums of the British bombe with circuitry required a new method for set-up of the cryptanalytic machinery. Running a 'menu' – the logic diagram resulting from comparing crib and intercept – requires a number of three-rotor devices each imitating the behaviour of the moving parts of an Enigma machine, each of which compares an actual transformation of a letter from plaintext to ciphertext as observed in the intercepted message. (A plaintext-ciphertext letter pair is referred to as a 'constatation'.) As different constatations came from different parts of the intercept, the Enigma analogues needed to be moved on an appropriate number of steps to reflect the progression of Enigma rotors as the message was enciphered. This would be done on a traditional bombe by moving the drums round; on the U.S. Army bombe, by advancing the progression of switching on the M units.

As explained by Alan Turing in his technical report, written after an inspection of a single M unit and Enigma emulator on 5 February 1943, the progression was essentially a 'Vigenère slide' achieved by electrical arithmetic in base 3. Three pairs of relays were connected in series, and the connection point to the Enigma emulator achieved by the additive effect of the relays, as illustrated in Box 1. 'If any particular total slide is required it is possible to choose certain of the six relays to energise so that this total will be obtained.'[3] This was done in a 'control turret'

[1] Friedman to Bullock, 14 September 1942. NARA RG 457, HMS Entry A1-9032, Box 1283, Nr 3815.

[2] Special Research History No 361 'History of the Signal Security Agency, Volume Two, The General Cryptanalytic Problems', page 257. NARA RG 457, HMS Entry A1-9002, Box 96.

[3] Turing report, 11 February 1943. TNA HW 62/5.

from which other aspects of menu set-up were done, such as the patching-together of the Enigma analogues testing the different constatations. Choice of rotors was also made from a control panel, rather than physically selecting drums.

Relay	Neither closed	One closed	Both closed
A , A′	0	1	2
B , B′	0	3	6
C , C′	0	9	18

Box 1: Relays which are in the 'on' position contribute units, threes, or nines in base-3 arithmetic. Combining the results identifies the input contact to an Enigma emulator. With appropriate choice of closures, each value from 0 to 25 can be obtained.

Another innovation was to do with 'stops'. British and Desch bombes were designed to stop when the machine detected a rotor start-position and plugboard cross-wiring consistent with the plaintext having been transformed into the observed intercept. A typical bombe-run would yield several 'stops', each of which had to be checked. The U.S. Army machine dealt with stops by not stopping, but recording the result.[4]

All of this required a vast amount of switching equipment and plenty of space. A demonstration version consisting of a single M unit was 3m high, 2m wide and 50cm deep; the finished machine had 72 of these, together with all the associated rotor-emulator racks, patch panels and so forth. The capacity of the U.S. Army bombe was equivalent to four British bombes, but it occupied four times the space (see Figure 1).

There were compensating advantages. The machine was fast (7 minutes for a run, compared with around 12 for a British bombe); the components were nothing more than standard telephone equipment, which aided both maintenance and secrecy in manufacture; omitting heavy moving parts eliminated mechanical stress and saved on wear and tear; fewer operators were needed; rotor changeover took 30 seconds as compared to 10 minutes for a rotary bombe, and the U.S. Army machine, being digital, was more accurate.[5]

2.2 Flexibility and future-proofing

The relay-based approach was highly flexible and future-proof. Given that the German navy had already devised a way to squeeze a fourth rotor into its Enigma machines, it was likely that further modifications would arise if the German forces continued to rely on Enigma. Indeed, towards the end of the war, new components such as a settable reflector (*Umkehrwalze D*), a hand-turned attachment to the plugboard to rotate its cross-wirings (the *Uhr*) and rotors with adjustable stepping notches (*Lückenfüllerwalzen*) were all proposed or rolled out at some stage. Even abandonment of Enigma might be possible, in which case some new encryption device might come into being. The British or Desch bombes would be more-or-less useless against such developments.

By using readily available components and relying on circuit design rather than hardware for its problem-solving logic, the U.S. Army bombe was highly adaptable. Over the course of 1943-44 a range of peripherals were developed to tackle specific Enigma problems:[6]

- Machine-gun (October 1943). This attachment automated the checking process for stops. 'Checking' meant testing the cross-plugging implied for each constatation in the menu to identify inconsistencies: if letter P was supposed to be cross-plugged to T it could not also be cross-plugged to K, so if checking led to that result, it would be a 'false stop'.
- Double-input (October 1943). This adaptation allowed the machine to test two menus simultaneously.
- Dud-buster (October 1944). A 'dud' was a message where the message setting (orientation of rotors at the start of encipherment) was not known, but all other aspects of the Enigma set-up (rotor choice and order, plugboard and ring-settings) were. The dud-buster found the

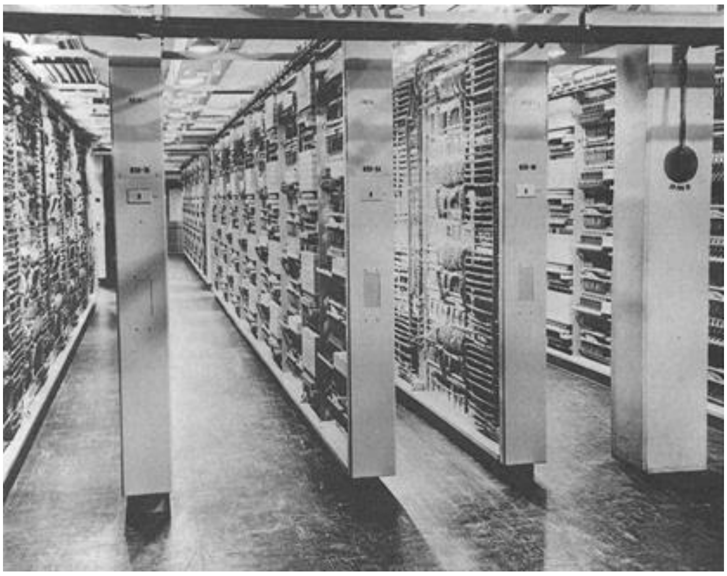
[4] Stevens report of Bell Labs visit, 3 February 1942. NARA RG 457, HMS Entry A1-9032, Box 1283, Nr 3815.

[5] Stevens report; Friedman to Corderman, 29 March 1944. NARA RG 457, HMS Entry A1-9032, Box 950, Nr 2809.

[6] SRH 361, pages 265-267.

missing setting. Many of the most valuable applications of dud-busting were naval, but it does not appear that the U.S. Navy had a machine solution to duds; the record is obscure as to whether naval problems were among those solved on the Army's equipment.

- Autoscritcher (by December 1944). This device was invented to counter the settable reflector, by identifying its wiring pattern in force for the time being. A functionally equivalent device built in Britain, called the Giant, linked four rotary bombes together. The U.S. Navy also built a machine called the Duenna for the same purpose.



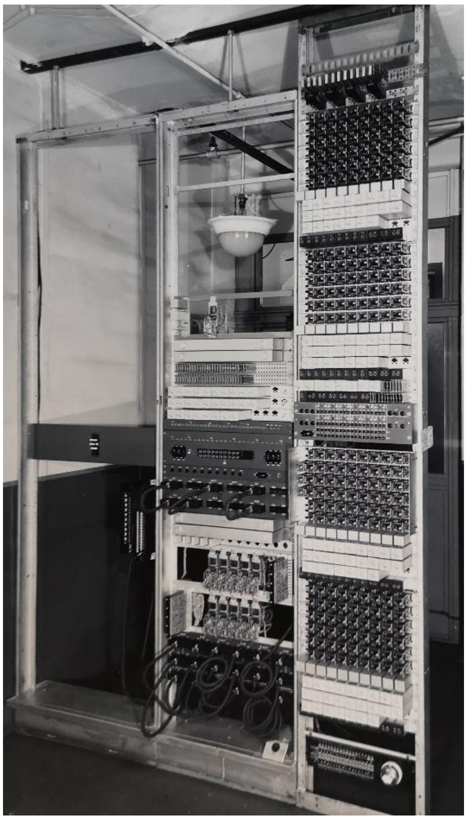


Figure 1: Two photographs of the US Army bombe: above, the full installation; right, an 'M' switching unit. (NCM online collection; NARA RG 457 HMS Entry A1-9032, Box 939. Declassification Authority for both images NND 963016.)

All these developments showed the versatility of a machine to which plug-on additions could be attached simply without the need for re-engineering. But, perhaps more importantly, the machine showed the way forward for future cryptanalytical problems as yet unforeseen. As William F. Friedman mentioned to Alan Turing on the occasion of the latter's visit to Bell Labs, 'the machine is intended to be "a general cryptographic machine".'[7] Indeed so: Friedman noted just after the end of the war that it would be useful if 56 of the 144 frames of the bombe machinery were redeployed to test the security of the US Army's own encryption systems, 'since it will facilitate certain investigations of a general nature in connection with rotor cryptographic machines.'[8]

3 The Secrecy issue in 1943

Despite the cooperation between Britain and the United States on cryptographic matters during World War II, in its early stages there was official resistance at the highest level in the U.S. Army to the British being allowed to see what they were building. Given that the British had invented the concept of the rotary bombe, it was going to be hard for the British to understand

[7] Turing report.

[8] Friedman to Hayes, 6 July 1945. NARA RG 457, HMS Entry A1-9032, Box 1283, Nr 3815.

why they (and the rotary bombe's chief logical designer, Alan Turing) should not take a look at the U.S. Army's bombe project.

It was not easy for the British authorities to work out what the Americans were concerned about in 1942. The British thought they had given full details of their Bombe technology, but because the details did not include 'blueprints', in July 1942 the Americans accused the British of holding back on them, notwithstanding 'assurances that it was not intended to build bombes'. (A separate agreement covered the American plan to build four-rotor naval Enigma bombes.) It was against this backdrop that Alan Turing was sent to America to work with Desch and to 'advise on the security of a U.S. speech scrambling device made by Bell Laboratories.' Friedman thought he had obtained approval on behalf of the U.S. Army's Signals Security Service for Turing to have access to Bell Labs.

However, U.S. Army Staff demurred. Various problems were mentioned to the UK's own chief cryptographer, John Tiltman, but these were unconvincing, and raised British suspicions. 'This thoroughly bad impression was reinforced a hundredfold by Colonel Tiltman's report that the War Department had without our knowledge or consent begun… building a bombe machine at the Bell Laboratories.' [9]

Matters did not end there. Turing's clearance to visit Bell Labs apparently did not extend to the speech encryption device, now known by the name SIGSALY and then under code reference X61753. The British were informed that the device was 'considered too secret to allow Dr. Turing to look in on it'. [10] Friedman was too junior – despite being the top military code-breaker – and Turing's visit should have been cleared at a much higher level. The British were told that the objection came from the very top, namely General George C. Marshall, the US Chief of Staff.

To deny Turing access to the speech encipherment machine did not appear logical. After all, the speech machine was in part a response to insecurity of the transatlantic radio-telephone link, which was used not just to keep the Chiefs of Staff connected to their commanders in the European Theatre of Operations but to allow political liaison between President Franklin D. Roosevelt and Prime Minister Winston Churchill. If Churchill was going to use it then the British were going to see it sooner or later. Perhaps the secrecy of X61753 was a specious reason for excluding Turing from Bell Labs. In any case, Bell Labs was a huge building, and to keep him away from one project while he looked at another would have been perfectly feasible. Perhaps something more was afoot, perhaps something reflecting embarrassment about the American change of policy on building their own non-naval bombes.

Now that the U.S. Army bombe documentation has been largely declassified, it is possible to put forward a more convincing reason for the desire to keep the British away in 1943. The possibilities suggested by digitising the logic of the bombe – and in particular the power and versatility of the new approach, and how they might be exploited and even turned against the United States itself in the wrong hands – may have been a secret far more important than X61753/SIGSALY or any short-term operational considerations relating to Enigma intelligence.

In the early months of 1943 the British were still, just, the dominant partner in the trans-Atlantic intelligence relationship. A single hint that the British would simply cut out the Americans if Turing's access was not granted was followed within two days by a removal of the obstacles. On 4 January, formal permission to inspect project X68003 – the Army bombe – was granted to Tiltman and Turing once again.[11] Turing was admitted to Bell Labs two weeks later to see the speech machinery, and at last, on 5 February 1943, to see the Army bombe. Once the ruffled feathers between the two allies had been smoothed over, a cooperative arrangement was worked out between Bletchley and Arlington Hall (where two finished Army bombes were installed) whereby specific problems, well-suited to the versatility of the X68003 equipment, were agreed for the U.S. Army's machine cryptanalysis team. Indeed, eventually the team's tasks seem to have been largely directed from Bletchley Park.[12]

[9] Briefing for Travis (undated, April 1943). TNA HW 50/13

[10] Dill to Marshall, 2.12.42. TNA HW CAB 122/14.

[11] Memorandum by Bullock, 4 January 1943. NARA RG 457, HMS Entry A1-9032, Box 1283, Nr 3815.

[12] SRH 361, page 269.

4 Digital cryptanalysis

The American army bombe represents a step forward in the mechanisation of cryptanalysis. Its development marks a change in thinking, from seeing large key-space cryptanalytical problems thrown up by the invention of cipher machines as case-by-case challenges, each demanding a bespoke mechanical response, towards a more universal, digital, computerised approach. Cryptanalysis was part of the business case for the United Kingdom's post-war computer project called ACE, which mentioned cryptically that 'the promised support of Commander Sir Edward Travis [by then the head of GCHQ], of the Foreign Office, will be invaluable.'[13]

In retrospect, it seems likely that the American fears about the innovative aspects of their army bombe becoming shared intellectual property were well-founded. The evolutionary pathway from wartime cryptanalytical devices to postwar programmable computing machines is well known (Corera, 2015). Electronics added speed, but the real breakthrough in this era was the ability to conceptualise machine-solvable problems in digital terms. While one can argue that the British bombe was digital – in the sense that its output was a binary presence or absence of voltage in a single wire of a 26-wire cable, the precondition for a 'stop' – it is probably more accurate to see the British bombe as a pre-computing-era hybrid between single-purpose analogue devices and digital data-processing machinery such as Hollerith punched-card sorters. The Desch bombe did not break from that tradition, whereas the U.S. Army bombe depended on binary processing of electrical pulses for its entire logical operation. Furthermore, the army bombe was to a degree programmable for new tasks, albeit not a 'stored-program computer' of the post-war era.

The use of electrical pulses and logical path moderation through relay switching shifted the focus of thought towards logic and programming and away from engineering: the design features of the U.S. Army bombe implied a new direction for computing. These lessons were not lost on Alan Turing, who appears to have spent the years after his Bell Labs visit in developing his own thoughts about computing machinery, culminating in his 1945 design proposal for the ACE.

Acknowledgments

The author gratefully acknowledges assistance and materials from John Harper, Michael Barbakoff, Philip Marks and Rob Simpson in with the research for this article.

References

Frank Carter. 2008. *The Turing Bombe*. Report No.4, Bletchley Park Trust, Milton Keynes, UK.

Gordon Corera. 2015. *Intercept*. Weidenfeld and Nicolson, London, UK.

Donald Davies. 1999. *The Bombe – a Remarkable Logic Machine*. Cryptologia, 23(2):108-138.

Jim DeBrosse and Colin Burke. 2004. *The Secret in Building 26*. Random House, New York, NY, USA.

Ralph Erskine, Philip Marks and Frode Weierud. 2002. *Review of US Bombes*. IEEE Annals of the History of Computing, 24(3):85-87.

Jon Gertner. 2012. *The Idea Factory*. Penguin Books, New York, NY, USA.

David Link. 2009. *Resurrecting Bomba Kryptologiczna*. Cryptologia, 33(2):166-182.

Jeremy McCarthy. 2019. *The Enigma of the Polish Bomba*. ITNOW, 61(3):26–27.

David J. Sherman. 2016. *The First Americans. The 1941 US Codebreaking Mission to Bletchley Park*. United States Cryptologic History, vol.12. National Security Agency/Center for Cryptologic History, Fort George G. Meade, MD, USA.

[13] Womersley to Darwin, undated memo entitled 'ACE Machine Project'. alanturing.net/turing_archive/archive/index/aceindex.html documents, accessed January 2021.

The Upplandic Non-Lexical Rune Stones: Ciphers or Nonsense?

Viktor Wase
Stockholm, Sweden
viktorwase@gmail.com

Abstract

The so-called non-lexical rune stones use ordinary runes but contain nothing but nonsensical "words". It is not entirely uncommon for rune stones to contain hidden and enciphered messages, which is why this study investigates the possibility of the Upplandic non-lexical stones being ciphers. This is done using a graph clustering algorithm that sorts the stones into groups based on how similar their texts are.

The algorithm labeled all non-lexical stones as outliers (belonging to no group), with the exception of U1126 and U1128 that form a group on their own. As such it is deemed unlikely that any of the non-lexical stones (perhaps excluding U1126 and U1128) are ciphers.

1 Background

The occurrence of ciphers in and among runes are not at all uncommon. Even rune stones, placed out in the public for everyone to see, contain messages in the form of ciphers. It is therefore a natural conclusion that the purpose of the ciphers was not to convey a hidden meaning, but something else.

Take the Kareby baptismal font (signum Bo NIYR5;221B), for example. Its transcription, excluding a complicated bind rune, reads raþesaerkannamnorklaski (Bæksted, 1949). This gives: *raðe sa er kan namn orklaski*, which roughly translates to *Read those who can the name orklaski*. Orklaski is not a known name. However, if one replaces each rune with the one that precedes it in the younger futhark, then orklaski becomes þorbiarn, which is still a common name in Norway. This is a Caesar cipher and is simple to solve once you know how (Suetonius, 1914).

In a similar vein, the stone U 1165 ends with a series of long and short lines: ||′′′′||′′′|||′′′′′||′′′|||′′′′′′|||′′′′′ hiuk (Nordby, 2018, p. 392). This is a binary rune cipher, where each rune can be reduced to a pair of numbers. By pairing up the long lines with the short ones following to the right one gets 2/4, 2/3, 3/5, 2/3, 3/6, 3/5. The younger futhark is commonly divided into three parts (ætt): fuþork hnias tbmlR. The first number indicates which ætt, and the second number the index in the ætt. It should be noted that the ættir are numbered backwards as 3, 2, and 1. This gives airikr hiuk, which translates to Erik carved. Once again the name was the only part that was encrypted.

A third type of cipher can be found in DR 239, which contains the following inscription *þmk iii sss ttt iii lll*. This is called an istil-formula, since the runes can be shuffled into three words that end with istil: *þistil* (thistle), *mistil* (mistletoe) and *kistil* (box) (Nordby, 2018, p. 104).

1.1 Non-Lexical Stones

There are some rune stones, mostly in Uppland and Södermanland, Sweden, that have no apparent meaning. They look like regular rune stones and the runes are of standard runic form, but they do not form words and sentences. They are commonly referred to as non-lexical stones. The common belief is that they are produced by illiterate carvers (Bianchi, 2010, p. 165), but there are fringe theories about their actual message.

For example Stig Eliasson argues that these stones show some patterns that would not show up if it were pure and random gibberish (Eliasson, 2014). This could indicate, he argues, that they might be written in an unexpected language. This is concretized by suggesting that the Danish Sørup stone might be written in Basque (Eliasson, 2010).

Perhaps not surprisingly, people have considered the possibility of the non-lexical stones being ciphers. In 1923 Erik Brate wrote this about the stone U 466: "...designed with the intention to test the wit of the reader, which supersedes the abilities of our time" (from Swedish: utförd i avsikt att sätta läsarens skarpsinne på prov, som överstiga vår tids förmåga) (Wessén and Jansson, 1946, p. 279-281). Regarding U 298 he wrote that he believed it to be "hidden writing" (from Swedish: lönnskrift) (Wessén and Jansson, 1946, p. 6-7). Rikard Dybeck, the creator of the de-facto Swedish national anthem, wrote about U 427 in 1877: "the inscription, as of yet uninterpreted, will probably remain so for a long time to come." (from Swedish: inskriften, hittills otydd, lärer länge nog förblifva det.) (Wessén and Jansson, 1946, p. 214-216).

More recently Craig P. Bauer argued similarly in his book Unsolved!. He concludes with the following remark: “A statistical study needs to be conducted on groups of related stones, such as those from Uppland, Sweden, with currently unreadable runic inscriptions to see if they might have been enciphered in the same manner.” (Bauer, 2017, p. 115-126).

1.2 Classification of Runic Cryptology

K. Jonas Nordby created a classification of runic cryptology (Nordby, 2018, p. 76). The two top classes are permutation and substitution ciphers. Permutation means that the runes are sorted in some unusual order, and substitution means that a specific symbol represents a specific rune. Permutation ciphers are simple to detect, since the frequencies of the runes are the same as in non-encrypted texts. Substitution ciphers are a bit trickier. One has to differentiate between mono-alphabetic substitutions ciphers (commonly abbreviated MASC) and homophonic substitution ciphers. The former being a cipher in which one symbol represents one rune, and the latter several symbols can represent one rune (Dooley, 2018, p. 9). The homophonic substitution ciphers can be excluded from this study since they require more symbols than the used alphabet, and the non-lexical stones only use the symbols from the futharks.

However, one of the sub-classes of substitution in Norby’s classification is neither mono-alphabetic nor poly-alphabetic. It is called jǫtunvillur, and in it each rune is replaced by the last rune in its name (Nordby, 2018, p. 135). In English this would entail that B is enciphered to E, since the letter is pronounced bee. Likewise F would be enciphered to F, since it is pronounced eff. The problem is that C would also be enciphered to E. This makes it a very inpractical cipher that is very hard to read. Nordby argues that it might have been a tool for learning the names of the runes (Nordby, 2018, p. 149).

There exists ciphers that are dependent on the position of the letter as well. For example, A might be encoded as B if it is the first letter of a text but encoded as C if it is the second letter. These ciphers tend to be highly complex and nothing of the sort has been found in the Viking era Scandinavia. The earliest examples found are from the 16:th century (Bonavoglia, 2020, p. 46). These are therefore excluded from the search, and the algorithm is not expected to be able to find any such ciphers.

1.3 Aim

The aim of this study is to develop an algorithm that takes a collection of short texts, from the stones, and divides them into groups. Each group will contain stones that are similar, in the sense that the frequencies of the runes are similar. If a stone is dissimilar to all the other stones then it will be classified as a singleton. This algorithm will then be applied to a collection of stones with both ordinary texts and non-lexical texts.

There are two foreseeable outcomes. Either, only one large group is formed with most of the regular stones and all of the non-lexical stones are filtered away as singletons. Or, a large group is formed with most of the regular stones, and a second group is formed with a portion of the non-lexical stones. Note that all non-lexical stones do not need to be in this second group, since it is possible that some of them are ciphers while others are not. The second outcome would indicate that the ciphers are distinct from the regular stones, but similar to each other. This means that there is some underlying pattern that could indicate the existence of a cipher. If neither of these are the true outcome, then the algorithm will not have been successful in separating the regular stones from the non-lexical stones, and a new algorithm will have to be developed.

The goal of the algorithm should be to be able to find stones that use mono-alphabetic or jǫtunvillur-like substitution ciphers, without being

Baseline	Non-Lexical
U 32, U 46, U 56, U 69, U 91	U 298
U 96, U 99, U 109, U 124	U 370
U 132, U 135, U 144, U 147	U 427
U 151, U 155, U 164, U 165	U 466
U 166, U 175, U 184, U 186	U 468
U 189, U 192, U 193, U 217	U 483
U 224, U 227, U 240, U 244	U 522
U 257, U 259, U 261, U 276	U 811
U 292, U 305, U 327, U 328	U 902
U 342, U 345, U 365, U 368	U 983
U 372, U 373, U 390, U 397	U 1126
U 423, U 431, U 435, U 441	U 1128
U 442, U 486, U 494, U 495	
U 528, U 530, U 574, U 577	
U 580, U 582, U 585, U 594	
U 606, U 620, U 660, U 662	
U 683, U 732, U 750, U 768	
U 814, U 826, U 856, U 866	
U 875, U 903, U 911, U 941	
U 943, U 949, U 960, U 961	
U 967, U 969, U 972, U 978	
U 994, U 1003, U 1028, U 1037	
U 1045, U 1060, U 1070, U 1127	
U 1129, U 1131, U 1146, U 1148	
U 1151, U 1157, U 1172	

Table 1: The stones in the baseline group and the non-lexical group.

confused by permutations.

2 Dataset

The dataset used is the offline version of the Scandinavian Runic-text Database (samnordisk runtextdatabas). The scope of the study will be limited to the Upplandic stones. There are over 1100 such stones. We form two groups based on these stones. The first is the baseline group and the second is the non-lexical group. See table 1.

The baseline stones are 100 stones that are longer than 10 total runes and contain only sixteen-rune younger futhark without extensions. They also had to have a translation in the Scandinavian Runic-text Database, to ensure that they do indeed have a lexical meaning. The stones are chosen randomly. To reduce the scope of the study we will only focus on stones with young futhark. The non-lexical group is based on the separation made by Marco Bianchi in his doctoral thesis (Bianchi, 2010, p.170-199) with some removals. The four stones U 523, U 835, U 1170 and U 1175 were removed since they contain rune-like signs, but no actual runes. U 888 and U 1179 were removed since they contain very little of the original message. U 493 and U 1180 were removed because they contained the letter e, which is not part of the younger futhark. U 529 was removed since the runes are very shallow and hard to read, to the point that Scandinavian Runic-text Database did not have any runes in its entry. U 1061 and U 596 did not have entry either, so it was removed. Finally, U 1078 was removed since it only had four symbols on it. This leaves 12 stones.

Figure 1: U 99 from the baseline group. Picture taken 1931.

All uncertain runes, guesses and non-futhark signs were removed from the dataset. Old sources (indicated by [ ] in the Scandinavian Runic-text Database) were used. All stones with ciphers (indicated by < >) and variants of words (indicated by /) in the transcription, were excluded from the baseline group. This gives 499 runes in the non-lexical group and 4998 in the baseline group.

3 Algorithm

3.1 Similarity Measure

The goal of the algorithm is to cluster the stones based on how similar they are. But similarity has yet to be defined in this context. Each stone is converted to a list of 16 numbers, one for each rune in the younger futhark. This number represents the frequency of the rune in the stone. For example if a carving were to have 60 % i-runes and 40 % l-runes, then its list would contain zeros except for the numbers representing i and l which would be 0.6 and 0.4. The distance metric chosen is the common Pythagorean metric, but in 16 dimensions.

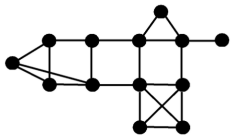
A clustering algorithm based on this similarity measure won't be confused by permutations since the frequencies of the runes remain unchanged if the order of the runes is changed. As a matter of fact, this means that the algorithm will not be able to differentiate pure permutations from non-enciphered stones. Substitution ciphers should be detectable since their distribution of frequencies will change.

3.2 Clustering

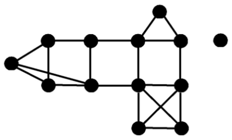
Before the data is passed to the algorithm it has to be converted into a graph of points connected by edges. Each point represents a stone in the dataset, and it is connected to all other stones that are similar to it.

First the Pythagorean distance of all pairs of stones are calculated. The median of these numbers is set as a threshold; if the distance is lower than the median then the pair is connected by an edge (similar) otherwise they are not. This choice of threshold is arbitrary and two other threshold values are used as well to ensure robust results. These values are the 40:th and 30:th percentile of the distance of all pairs of stones.

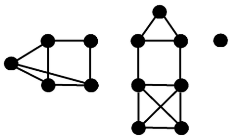
The clustering part of the algorithm is based on an algorithm called Highly Connected Subgraphs (HCS). It is a rather simple algorithm that takes a graph and looks for the smallest set of edges without which the graph will become disconnected - a so-called minimum cut. It then repeats this procedure on the two new separated graphs. It stops dividing a graph when its minimum cut contains n / 2 or more edges, where n is the number of points in the graph (Erez Hartuv, 2000). Consider the graph below, for example.



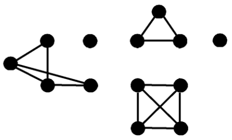
The minimum cut contains only one edge, the rightmost one. If that edge is removed, then the graph is divided into two disconnected sub-graphs, thus creating the graph below.



The minimum cut of this graph contains two edges, since one cannot divide the graph into two disconnected sub-graphs by removing only one edge. It is worth noting that the algorithm has two choices here: either it removes the triangle at the top (creating a singleton) or the horizontal edges of the square in the middle. In these cases the outcome is random. Let us say that it chooses the square.



With two sub-graphs (ignoring the singleton) the algorithm will process them separately. In both cases the minimum cut is two, and removing the edges gives the following graphs.

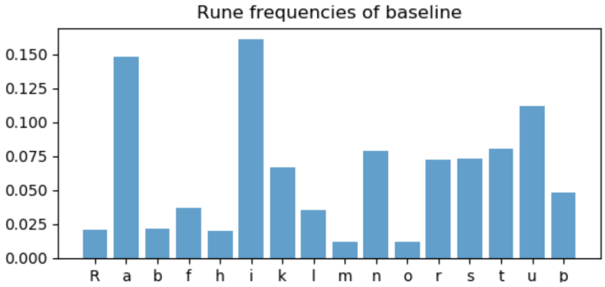


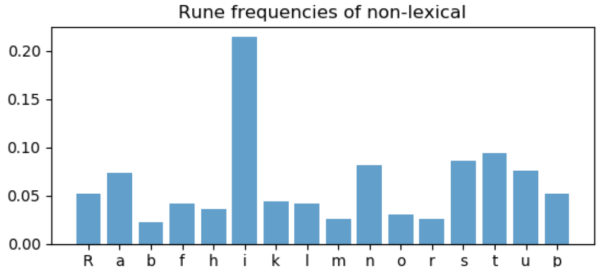
At this point the minimum cut of each sub-graph is larger than or equal to half of the number of points in each sub-graph, which means that the algorithm stops.

This paper used the Python implementation found att *github.com/53RT/Highly-Connected-Subgraphs-Clustering-HCS*.

4 Results

Before we get to the result of the clustering algorithm, let's quickly examine the rune frequencies of the baseline group and the non-lexical group, as seen in the figures below.





The frequency distributions are clearly different, as seen by the huge spike in *i* in the non-lexical group and the lack of such a spike in *a*. However, it is clear that the non-lexical distribution is not simply a reordered version of the baseline. This indicates that there is no widespread use of a substitution cipher in the non-lexical stones.

4.1 Graph Algorithm Result

The results from the algorithm are quite interesting. The overall behaviour was the same no matter if the threshold value was the median, or the 40:th or 30:th percentile. The result was: one large sub-graph containing the majority of the stones, one tiny sub-graph containing the pair U 1126 and U 1128, and then a lot of singletons. The remarkable part is that the non-lexical stones were almost always filtered out from the large sub-graph - the only exception being U 298 when the threshold was the median. See table 2 for the full results for the 40:th percentile. The non-lexical group is marked in bold font.

The algorithm can clearly filter out the non-lexical stones from the bulk of the regular stones. The fact that it did not group the non-lexical stones together means that it seems unlikely that there is any widespread use of substitution ciphers, permutation ciphers or a combination of the two. Roughly a third of the baseline was excluded from the large group. It should be noted that the outlier group contains stones with common forms. For example

U 135 (translated): *Ingifastr and Eysteinn and Sveinn had these stones raised in memory of Eysteinn, their father, and made this bridge and*

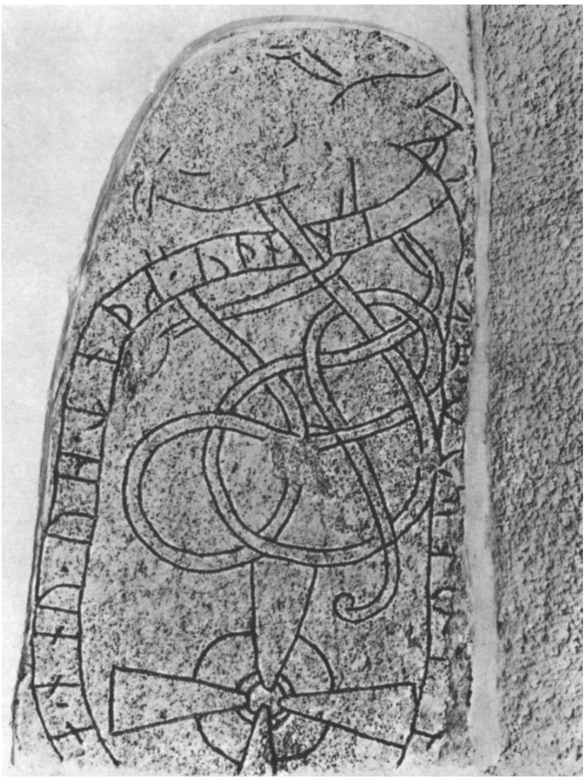


Figure 2: The non-lexical stone U 1126. From *Upplands runinskrifter*, part 2 (1946).

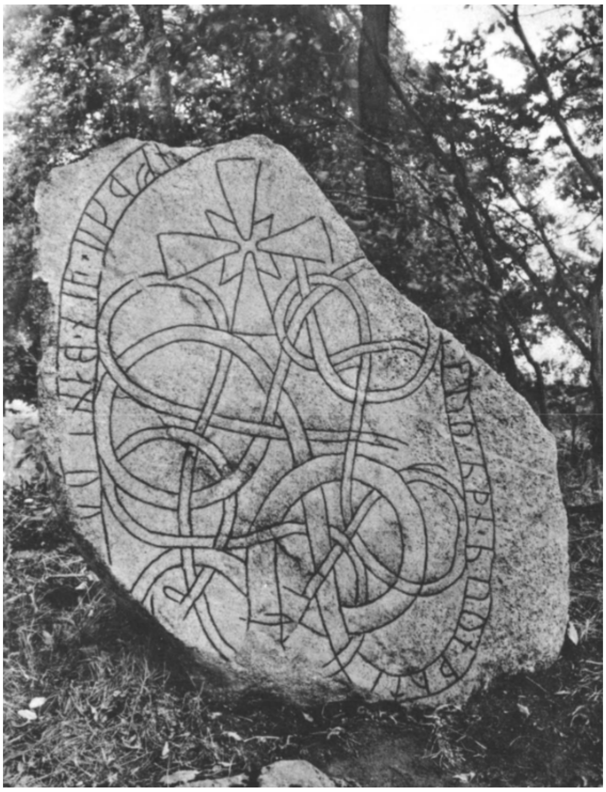


Figure 3: The non-lexical stone U 1128. From *Upplands runinskrifter*, part 2 (1946).

Large Group	Small Group	Singletons
U 32, U 56, U 69	**U 1126**	U 46
U 91, U 96, U 99	**U 1128**	U 132
U 109, U 124, U 144		U 135, U 155
U 147, U 151, U 164		U 184, U 192
U 165, U 166, U 175		U 217, U 244
U 186, U 189, U 193		U 257, U 292
U 224, U 227, U 240		**U 298**, U 305
U 259, U 261, U 276		U 345, U 365
U 327, U 328, U 342		U 368, **U 370**
U 372, U 390, U 423		U 373, U 397
U 431, U 435, U 486		**U 427**, U 441
U 494, U 530, U 580		U 442, **U 466**
U 582, U 585, U 606		**U 468**, **U 483**
U 660, U 662, U 732		U 495, **U 522**
U 750, U 768, U 814		U 528, U 574
U 826, U 866, U 875		U 577, U 594
U 903, U 911, U 941		U 620, U 683
U 949, U 960, U 961		**U 811**, U 856
U 969, U 972, U 978		**U 902**, U 943
U 1003, U 1028		U 967, **U 983**
U 1060, U 1070		U 994
U 1127, U 1129		U 1037
U 1146, U 1151		U 1045
U 1157, U 1172		U 1131
		U 1148

Table 2: The results from the algorithm. Most stones belong to one large group, many do not belong to any group (singletons) and two form a small group. The non-lexical stones are in boldface.

this mound.

U 244 (translated): *Fasti had the stone cut in memory of Fastulfr, his son.*

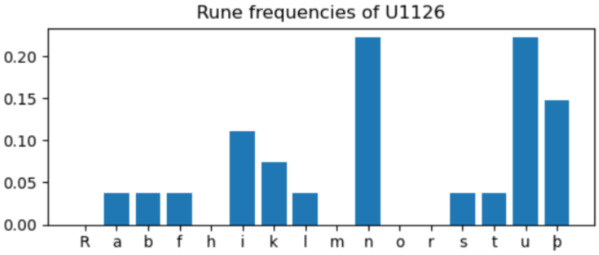
Both of these have common forms and it would be expected that a similar ones would exist in the baseline group, but even so they have been marked as singletons. This indicates that the methodology is not perfect at the stone-level, even if the algorithm manages to catch the overall larger trends.

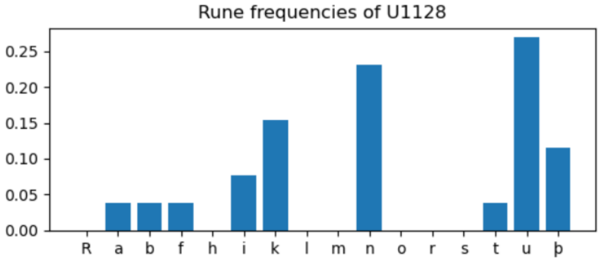
This brings us to U 1126 and U 1128. The fact that these stones are grouped together is rather intriguing. These stones are currently placed next to each other at the Alunda church This was not known to the algorithm, and yet it managed to pair them together. This does, of course, not mean that they are ciphers. It only means that they are similar. The inscriptions of the two stones are:

U 1126 *uluiuþnis-... ...-þnf]a · nnu · ub ’ tnþk · uþnki*

U 1128 *...nfþku* × *–in · ban-iuu ...-nuu ’ kþn ’ kuunþkt-*

See the figures below for the runic frequencies of U 1126 and 1128.





Both U1126 and U 1128 differ greatly from the baseline frequencies. They might be monoalphabetic substitution ciphers. Or perhaps they are encoded with a cipher in which multiple runes are enciphered to the same symbol? They are unlikely to be jotunvillur since that only has 6 unique runes (Nordby, 2018, p. 137) and U 1126 and U 1128 has 11 unique runes together.

It seems Rikard Dybeck was right. The inscriptions will remain uninterpreted for at least a while more.

5 Acknowledgements

A big thank you goes out to Dr. James Knirk for vast and valid feedback. Any and all shortcomings and faults are of course my own.

References

Craig P. Bauer. 2017. *Unsolved!: The History and Mystery of the World’s Greatest Ciphers from Ancient Egypt to Online Secret Societies*. Princeton University Press.

Marco Bianchi. 2010. *Viking age written culture in Uppland and Södermanland*. Department of Nordic Languages. Uppsala University.

Paolo Bonavoglia. 2020. Trithemius, Bellaso, Vigenère - origins of the polyalphabetic ciphers. *Proceedings* *of the 3rd International Conference on Historical Cryptology*, pages 46–51.

Anders Bæksted. 1949. Kareby-fontens runeindskrift. *Fornvännen - Journal of Swedish Antiquarian Research*, pages 49–53.

John F. Dooley. 2018. *History of Cryptography and Cryptanalysis - Codes, Ciphers and Their Algorithms*. Springer International Publishing.

Stig Eliasson. 2010. Change resemblances or true correspondences? on identifying the language of an 'unintelligible' scandinavian runic inscription. In *Transeurasian verbal morphology in a comparative perspective: Genealogy, contact, chance*, pages 43–79. Otto Harrassowitz Verlag.

Stig Eliasson. 2014. Runic inscriptions in an unrecognized foreign tongue? methodological preliminaries to language identification. *8:th International Symposium on runes and runic inscriptions*.

Ron Shamir Erez Hartuv. 2000. A clustering algorithm based on graph connectivity. *Information Processing Letters*, 76:175–181.

K. Jonas Nordby. 2018. Lønnruner - kryptografi i runeinnskrifter fra vikingtid og middelalder.

Suetonius. 1914. *Lives of the Caesars, Volume I: Julius. Augustus. Tiberius. Gaius. Caligula.* Harvard University Press.

Elias Wessén and Sven B. F. Jansson. 1946. *Upplands runinskrifter, part 2*. Kungliga Vitterhets Historie Och Antikvitets Akademin.

Published by:

NEALT Proceedings series 48
Linköping University Electronic Press, Sweden
Linköping Electronic Conference Proceedings, No. 183
ISSN: 1650-3686
eISSN: 1650-3740
ISBN: 978-91-7929-026-9
URL: https://doi.org/10.3384/ecp183
Cover photo: Ecolex I, Crypto Museum Eindhoven