Solving Classical Ciphers with CrypTool 2

Nils Kopal
Applied Information Security – University of Kassel
Pfannkuchstr. 1, 34121 Kassel, Germany
nils.kopal@uni-kassel.de

Abstract

The difficulty of solving classical ciphers varies between very easy and very hard. For example, monoalphabetic substitution ciphers can be solved easily by hand. More complex ciphers like the polyalphabetic Vigenère cipher, are harder to solve and the solution by hand takes much more time. Machine ciphers like the Enigma rotor machine, are nearly impossible to be solved only by hand. To support researchers, cryptanalysts, and historians analyzing ciphers, the open-source software CrypTool 2 (CT2) was implemented. It contains a broad set of tools and methods to automate the cryptanalysis of different (classical and modern) ciphers. In this paper, we present a step-by-step approach for analyzing classical ciphers and breaking these with the help of the tools in CT2. The primary goals of this paper are: (1) Introduce historians and non-computer scientists to classical encryption, (2) give an introduction to CT2, enabling them to break ciphers by their own, and (3) present our future plans for CT2 with respect to (automatic) cryptanalysis of classical ciphers. This paper does not describe the used analysis methods in detail, but gives the according references.

1 Motivation

There are several historical documents containing text enciphered with different encryption algorithms. Such books can be found for instance in the secret archives of the Vatican. Often, historians who find such encrypted books during their research are not able to decipher and reveal the plaintext. Nevertheless, these books can contain secret information being of high interest for historians. In such cases, cryptanalysts and image-processing experts are needed to support deciphering the books, thus, enabling the historians to continue their research.

The ciphers used in historical books (from the early antiquity over the Middle Ages to the early modern times) include simple monoalphabetic substitution and transposition ciphers, codebooks and homophone ciphers.

With the open-source tool CrypTool 2 (CT2) (Kopal et al., 2014) historians and cryptanalysts have a powerful tool for the analysis as well as for the (automatic) decryption of encrypted texts. Throughout this paper, we present how CT2 can be used to actually break real world ciphertexts.

The following parts of this paper are structured as follows: The next section gives a short introduction to classical ciphers, as well as an overview of cryptanalysis. In Section 3, we briefly introduce the CT2. In Section 4, we present a general step-by-step approach for the analysis of ciphers with CT2. Section 5 shows real example analyses done with the help of CT2. Section 6 gives an overview of cryptanalysis components (for classical ciphers) already implemented in CT2 as well as components planned for the future. Finally, Section 7 summarizes the paper.

2 Foundations of Classical Ciphers and Cryptanalysis

After a brief introduction to classical ciphers we discuss the cryptanalysis of classical ciphers.

2.1 Classical Ciphers

Ciphers encrypt plaintext into ciphertext based on a set of rules, i.e. the encryption algorithm, and a secret key only known to the sender and intended receiver of a message.

Classical ciphers, as well as ciphers in general, can be divided into two different main classes: substitution ciphers and transposition ciphers. A

Proceedings of the 1st Conference on Historical Cryptology, pages 29– 38,
Uppsala, Sweden, 18-20 June, 2018

substitution cipher replaces letters or groups of letters of the plaintext alphabet with letters based on a ciphertext alphabet. Transposition ciphers do not change the letters themselves but their position in the text, i.e. plaintext alphabet and ciphertext alphabet are equal. There also exist ciphers that combine both, substitution and transposition, to create a composed cipher, e.g. the ADFGVX cipher (Lasry et al., 2017).

Substitution ciphers can be furthermore divided into monoalphabetic and polyalphabetic ciphers (Forsyth and Safavi-Naini, 1993). With monoalphabetic ciphers, only one ciphertext alphabet exists. Thus, every plaintext letter is always replaced with the same letter of the ciphertext alphabet. If there are more possibilities to choose from the ciphertext alphabet the substitution cipher is a homophone substitution cipher (Dhavare et al., 2013). If there are more than one ciphertext alphabet which are exchanged after each encrypted letter, the substitution is a polyalphabetic substitution, e.g. the Vigenère cipher (Schrödel, 2008). Substitution may also not only be based on single letters but on multiple letters, e.g. the Playfair cipher (Cowan, 2008). In history, for military and diplomatic communication, codebooks and nomenclatures were used. With a nomenclature, not only letters were substituted, but additionally, complete words were substituted. Codebooks contained substitutions for nearly all words of a language.

Transposition ciphers change the positions of each letter in the plaintext based on a pattern that is based on a key. The most used transposition cipher is the columnar transposition cipher (Lasry et al., 2016c). Here, a plaintext is written in a grid of columns. Then, the columns are reordered based on the lexicographical order of a keyword written above the columns. Finally, the ciphertext is read out of the transposed text column-wise. Decryption is done the same way but in the reverse order.

Composed ciphers execute different cipher types in a consecutive order to strengthen the encryption. One famous composed cipher is ADFGVX. Here in the first step, each plaintext character is substituted by a bigram only consisting of the 6 letters A,D,F,G,V, and X. After that, the intermediate ciphertext is encrypted with a columnar transposition cipher. ADFGVX was used by the Germans during World War I. It introduced a new concept, called fractionation. With fractionation, a plaintext symbol (here a bigram) is afterwards fractionated into two different symbols, making cryptanalysis even harder.

Ciphers based on codebooks were often super-enciphered, thus, first the words were substituted, for instance with numbers. Then, the resulted ciphertexts were additionally super-encrypted by changing them according to special rules.

Many encrypted historical books that survived history are available. Most of them are encrypted either with simple monoalphabetic substitutions or with homophone substitutions. For some books the type of cipher is unknown.

Many encrypted historical messages are encrypted with simple substitution ciphers, homophone substitution ciphers, polyalphabetic substitution ciphers, nomenclatures, or codebooks. Transposition ciphers were also used, but not as much as substitution ciphers since transpositions are more complex with respect to the encryption procedures. Additionally, performing a transposition cipher is more prone to errors. In modern times, transposition was used by the IRA (Mahon and Gillogly, 2008) and during World War II by the Germans and the British.

In World War II, rotor cipher machines like the German Enigma (Gillogly, 1995) performing polyalphabetic encryptions were introduced and widely used.

2.2 Cryptanalysis

Cryptanalysis is the science and art of breaking ciphers without the knowledge of the used key. Today, cryptanalysis is used to evaluate the security of modern encryption algorithms and protocols.

We divide the cryptanalysis of classical ciphers into two different approaches: the classical paper-based cryptanalysis and the modern computer-based cryptanalysis. In his paper we focus on modern computer-based cryptanalysis which can be done with CT2.

Substitution ciphers can be broken with the help of language and text statistics. Since every letter in a language as well as in the plaintext alphabet of a cipher has its unique frequency it can be used to guess and identify putative plaintext letters. With monoalphabetic substitutions, plaintext and ciphertext frequencies are identical, but the letters differ. For example an 'E' is substituted by an 'X' – 'X' has then the same frequency in ciphertext as 'E' has in plaintext. Thus, an algorithm

to break a substitution ciphers aims at recovering the original letter distribution.

Homophone substitutions as well as polyalphabetic substitutions flatten the distribution of letters, hence, aiming to destroy the possibility to break the cipher with statistics. Nevertheless, having enough ciphertext and using sophisticated algorithms, e.g. hill climbing and simulated annealing, it is still possible to break them.

Transposition ciphers can also be attacked with the help of statistics. Since transposition ciphers do not change the letters, the frequency of the unigrams in plaintext and ciphertext are exactly the same. Thus, to break transposition ciphers, text statistics of higher orders (bigrams, trigrams, tetragrams, or n-grams in general) are used to break them. Besides that, similar sophisticated algorithms, e.g. hill climbing and simulated annealing, are used to break transposition ciphers.

For breaking a classical cipher, it is useful to know the language of the plaintext. It is possible to break a cipher using a "wrong" language, but the correct one yields a higher chance of success. For cryptanalysis most of the algorithms implemented in CT2 contain a set of multiple languages, e.g. English, German, French, Spanish, Italian, Latin, and Greek. In many cases, the language of an encrypted book is known to the cryptanalyst or can be guessed by its (historical) context.

To identify the type of the cipher, whether it is a substitution cipher or a transposition cipher, cryptanalysts use the Index of Coincidence (IC) (Friedman, 1987). The IC, invented by William Friedman, is the probability of two randomly drawn letters out of a text to be identical. For English texts the IC is about 6.6% and for German texts about 7.8%. Simple monoalphabetic encryption, where a single letter is replaced by another letter, does not change the IC of the text. Same applies to all transposition ciphers, since these do not change the text frequencies. Polyalphabetic substitution aims at changing the letter distribution of a text to become the uniform distribution. Thus, the IC is about $\frac{1}{26} \approx 3.8\%$ (where 26 is the length of the ciphertext alphabet and all letters are used equally distributed). Homophone substitution also aims at changing the letter distribution of a text to become the uniform distribution, but here the IC is about $\frac{1}{n}$, where n is the amount of different symbols in the text.

Thus, having an IC close to 6.6% indicates that we have either a plaintext, a monoalphabetic substituted text, or a transposed text. And it is probably German. On the other hand, having an IC close to 3.8% indicates that we have a polyalphabetic encrypted text. Clearly, the IC is more accurate having long ciphertexts. Identification of homophone ciphers can be done by counting the number of different used letters or symbols. If the number is above the expected alphabet size, it is probably a homophone substitution.

State-of-the-art for breaking classical ciphers are search metaheuristics (Lasry, 2018). Because with classical ciphers, a "better guessed key" often yields a "better decryption" of a ciphertext, such algorithms are able to "improve" a key to come close to the correct key and often finally reveal the correct key. "Better" in this context means, that the putative plaintext that is obtained by decrypting a given ciphertext is rated higher by a so-called cost or fitness function. An example for such a function is the aforementioned IoC, which comes close to a value indicating natural language when the key comes closer to the original one. A common and very successfully used search metaheuristic is hill climbing. A hill climbing algorithm first randomly guesses a putative "start key". Then, it rates its cost value using a cost function. After that, it tries to "improve" the key by randomly changing elements of the key. With the Vigenère cipher for example, it would change the first letter of the keyword. After changing the letter, it again computes the cost function. If the result is higher than for the previous key, the new key is accepted. Otherwise, the new key is discarded and another modified one is tested. The algorithm performs these steps until no new modified key can be found that yields a higher cost value, i.e. the hill (= local maximum) of the fitness score is reached. Most of our classical cryptanalytic implementations in CT2 are based on such a hill climbing approach.

3 An Introduction to CrypTool 2

CrypTool 2 (CT2) is an open-source tool for e-learning cryptology. The CrypTool community aims to integrate into CT2 the best known and most powerful algorithms to automatically break (classical and modern) ciphers. Additionally, our goal is to make CT2 a tool that can be used by everyone who needs to break a classical cipher. Another well-known Windows analyzer for classical

ciphers is CryptoCrack (Pilcrow, 2018).

CT2 consists of a set of six main components: the Startcenter, the Wizard, the WorkspaceManager, the Online Help, the templates, and the CrypCloud, which we present in detail in the following.

The **Startcenter** is the first screen appearing when CT2 starts. From here, a user can come to every other component by just clicking an icon.

The **Wizard** is intended for CT2 users that are not yet very familiar with the topics cryptography or cryptanalysis. The user just selects step by step what he wants to do. The wizard displays at each step a small set of choices for the user.

The **WorkspaceManager** is the heart of CT2 since it enables the user to create arbitrary cascades of ciphers and cryptanalysis methods using graphical icons (components) that can be connected. To create a cascade, the user may drag&drop components (ciphers, analysis methods, and tools) onto the so-called workspace. After that, he has to connect the components using the connectors of each component. This can be done by dragging connection lines between the inputs (small triangles) and outputs (also small triangles) using the mouse. Data in CT2 can be of different types, e.g. text, numbers, binary data. The type of data is indicated by a unique color. A simple rule is, that connections between the same colors are always possible. Connections between different colors (data types) may also be possible, but then data has to be converted. CT2 can do this automatically in many cases, but sometimes special data converters are needed.

Figure 1 shows a sample workspace containing a so-called *Caesar cipher* (very simple monoalphabetic substitution) component, a *TextInput* component enabling the user to enter text, and a *TextOutput* component displaying the final encrypted text. The connectors are the small colored triangles. The connections are the lines between the triangles. The color of the connectors and connections indicate the data types (here text). When the user wants to execute the flow, he has to start it by hitting the *Play* button in the top menu of CT2. Currently, CT2 contains more than 160 different components for encryption, decryption, cryptanalysis, etc. Many components that can be put onto the workspace have a special visualization that can be viewed when opening the component by double clicking on it. Figure 2 shows such a maximized visualization of a standard component.

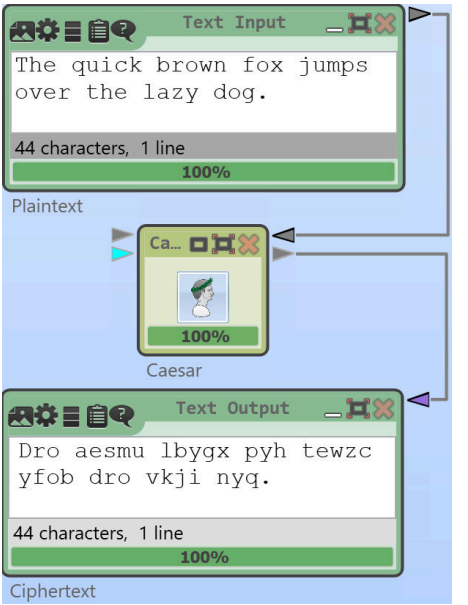

Figure 1: CT2 Workspace with Caesar Cipher

CT2 contains a huge **Online Help** describing each component. By pressing F1 on a selected component of the WorkspaceManager, CT2 automatically opens the online help of the corresponding component.

CT2 also contains a huge set of more than 200 so-called **Templates**. A template shows how to create a specific cipher or a cryptanalytic scenario using the graphical programming language and is ready to use. The Startcenter contains a search field that enables the user to search for specific templates using keywords.

Finally, the **CrypCloud** (Kopal, 2018) is a cloud framework built in CT2. We developed it as a real-world prototype for evaluating distribution algorithms for distributed cryptanalysis using a multitude of computers.

4 A Step-by-Step Approach for Analyzing Classical Ciphers in CrypTool 2

In this section, we show a step-by-step approach for analyzing classical ciphers in CT2. The first step is to make the cipher processable for CT2, so we create a digital transcription of the ciphertext. Then, we identify the type of the cipher. The third

step then finally breaks the cipher with CT2.

4.1 Create a Transcription

There are two ways to create a transcription of a ciphertext for CT2. The first method is to manually assign to each ciphertext symbol a letter by hand outside of CT2, e.g. with Windows Notepad. The transcription is saved as a simple text file. This file can be loaded into CT2 by using the *FileInput* component and then be processed further.

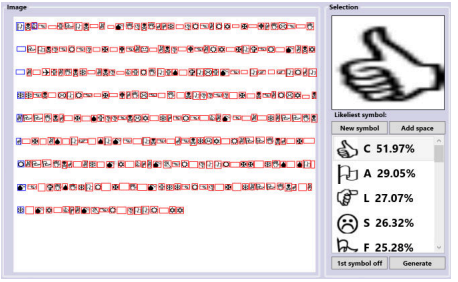

Figure 2: CT2 Transcriptor Component with Marked Symbols for Transcription

The second method to create a transcription uses the CT2 component *Transcriptor* (Figure 2). With the transcriptor, a user can load a picture, e.g. a scan of a document. Then, he can assign letters to the scanned symbols by marking them. Finally, the transcriptor is able to output the complete transcription. It supports the user in two different ways: (1) It automatically guesses, which symbol the user just had marked by showing the most likely symbols and (2) it can be set to semi-automatic mode. In semi-automatic mode, it automatically marks all other symbols that are similar to the one just marked by the user.

The DECODE project (Megyesi et al., 2017) already hosts a huge set of transcriptions of encrypted historical books done by experts. Within 2018 there will be an interface to call either CT2 from the DECODE website or to download DECODE records from within CT2.

4.2 Identify the Cipher

After creating the transcription of the cipher it is now possible to analyze its characteristics. A first analysis would be to create a text frequency analysis. For that, CT2 contains a *Frequency Test* component. It can be configured to show unigram distribution, bigram distribution, etc.

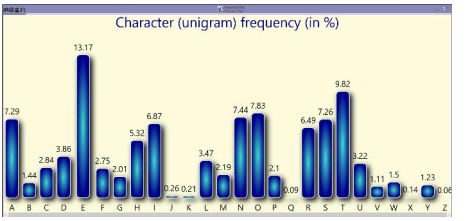

Figure 3: Frequency Test Component Showing Distribution of Plaintext

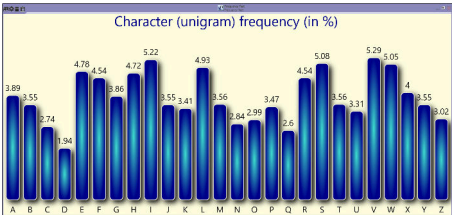

Figure 4: Frequency Test Component Showing Distribution of Ciphertext

In Figure 3 we show the distribution of a plaintext ("The Declaration of Independence" of the US). It can easily be seen, that the text follows the letter distribution of the English language, i.e. the 'E' is the most frequent letter, the letters 'X', 'Q', and 'Z' are very rare. In Figure 4 we show the distribution of a ciphertext ("The Declaration of Independence" of the US, encrypted with a Vigenère cipher). Here, all letters are more or less equally distributed, showing the cryptanalyst that it is possibly a polyalphabetic substitution cipher.

Another component that helps to analyze and identify a cipher is the *Friedman Test*, invented by William Friedman. With this test the key length (number of letters of a key word or phrase) of a polyalphabetic cipher can be calculated.

In Figure 5 we show the result of the Friedman test performed on plaintext ("The Declaration of Independence" of the US). It shows that the given text is possibly plaintext or a monoalphabetic substitution. Furthermore, the ciphertext could be transposed since the transposition does not change the letter distribution. In Figure 6 we show the result of the Friedman test performed on ciphertext ("The Declaration of Independence" of the US, encrypted with a Vigenère cipher). It shows that the given text is possibly ciphertext and polyalphabetic. Additionally, it shows that the estimated

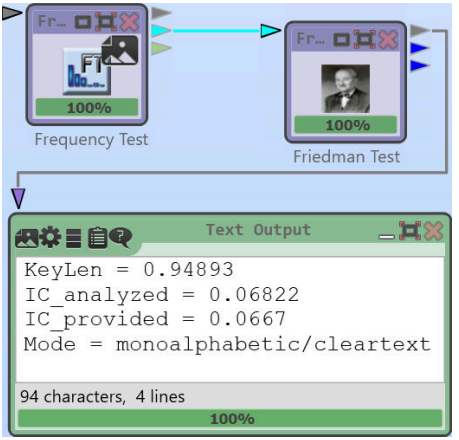

Figure 5: Friedman Test Component Showing Result of English Plaintext

key length is about 9. The component needs a provided IC (*IC_provided*) which is used as a reference value for the analyzed IC (*IC_analyzed*).

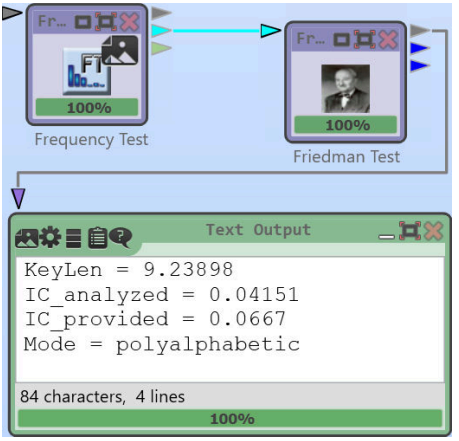

Figure 6: Friedman Test Component Showing Result of Ciphertext

4.3 Break the Cipher

After identifying the cipher type it can now be broken with the help of different cryptanalysis components. CT2 contains components for the automatic breaking of the monoalphabetic substitution cipher, the Vigenère cipher, and the columnar transposition cipher.

In Figure 7 we show the *Vigenère Analyzer* component which automatically solved a Vigenère cipher ("The Declaration of Independence" of the US, encrypted with a Vigenère cipher. The solver automatically tested every keylength between 5 and 20 using hill climbing. Only about ten seconds are needed for the component to automatically break the cipher. The decrypted text is automatically outputted by the component and can be displayed by an *TextOutput* component.

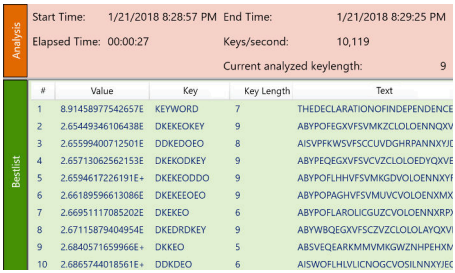

Figure 7: Vigenère Analyzer Solving a Cipher

All automatic cryptanalysis components have the same style of user interface. Besides start and end-time, the elapsed time for the analysis is shown. Furthermore, some components estimate the time for the remaining automatic analysis.

5 Example Cryptanalysis of Original Classical Ciphers

In this section we present two different real-world classical ciphers that can be broken with CT2.

5.1 Message in a Bottle Sent to General Pemberton in the US Civil War

The following message was sent in a bottle by a Confederate commander at the 4th of July 1863 in Vicksburg to General Pemberton. It was broken by the retired CIA codebreaker David Gaddy in 2010 (Daily Mail Reporter, 2010). We here use this message (221 letters) as our first real-world example for breaking classical ciphers with CT2.

In the first step, to automatically analyze the ciphertext, we had to create a transcription as shown in Section 4.1. We could have used the *Transcriptor* component or do it manually. Since the letters are written differently, the scanned image has only a low resolution, and the message contains ink spots, we did it manually. We show the result of the transcription of the ciphertext in Figure 9.

Now, we could analyze the text to identify the

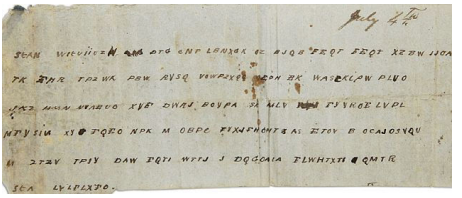

Figure 8: Encrypted Message in a Bottle Sent by General Johnston

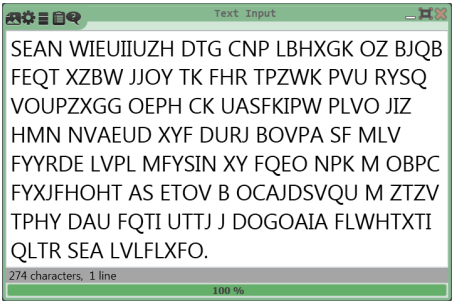

Figure 9: Transcription of Encrypted Message in a Bottle

type of the cipher. First, we created a letter frequency analysis (see Figure 10).

The distribution of letters indicated that the message is not encrypted with monoalphabetic substitution and possibly not transposed. Based on the more or less equal distribution of the letters we assume that the message is encrypted with a polyalphabetic cipher. To further strengthen our assumption, we applied the Friedman test and calculated the IC. In Figure 11 we show the results of the computation of the IC and the Friedman test.

The IC equal to 0.03834 indicated that the message is possibly encrypted with a polyalphabetic cipher. The estimated length of the key by the Friedman analysis is ≈ 5730, which is impossible for a text of only 221 letters. Thus, the message is either encrypted with a running key cipher, meaning the key length is infinity, or the Friedman test just fails because of the short length of the message. Since we know that in the Civil War the Vigenère cipher was often used, we assumed it could be encrypted with the Vigenère cipher. Other possibilities would be a codebook or a homophone cipher.

In the last step, we try to break the cipher. Since we assume it to be a Vigenère cipher, we used the *Vigenère Analyzer* component to break it.

We automatically test all key lengths between 1 and 20. Figure 12 shows the final result of the Vigenère Analyzer component. The component displays a toplist of "best" decryptions based on a cost function that rates the quality of the decrypted texts. The higher the cost value (sum of n-gram probabilities of English language) the higher the place in the toplist. Furthermore, the component shows the used keyword or pass phrase. With "MANCHESTERBLUFF" (15 letters), the message can be broken. The analysis run took 5 seconds on a standard desktop computer with 2.4 GHz. We present the final plaintext in Figure 13.

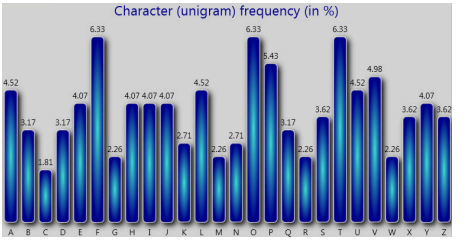

Figure 10: Letter Frequency Analysis of Encrypted Message in a Bottle

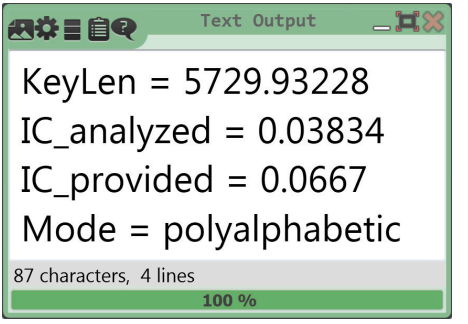

Figure 11: Friedman Test and IC of Encrypted Message in a Bottle

5.2 Borg Cipher – Encrypted Book from the 17th Century

The Borg cipher is a 408 pages manuscript, probably from the 17th century. The manuscript is located at the *Biblioteca Apostolica Vaticana* (Aldarrab et al., 2018). It is written using special ciphertext symbols. Figure 14 shows a small part of the Borg cipher. We here use the book as our

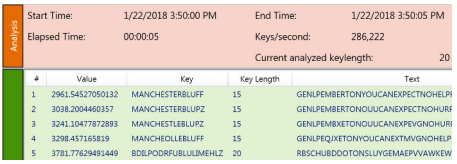

Figure 12: Breaking the Encrypted Message in a Bottle with the Vigenère Analyzer

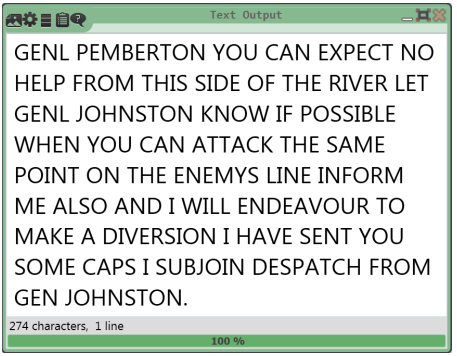

Figure 13: Message in a Bottle – Revealed Plaintext by Vigenère Analyzer

second real-world example for breaking classical ciphers with CT2. The cipher was already broken by (Aldarrab et al., 2018).

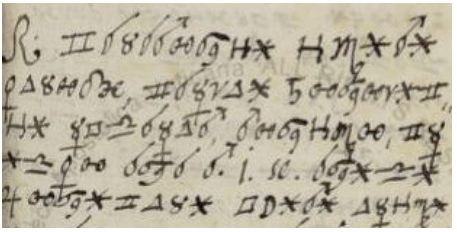

Figure 14: Picture Taken from the Borg Cipher

We took the complete transcription of the book from (Aldarrab et al., 2018).

First, we performed a frequency analysis of the text shown in Figure 15.

Then, we applied the Friedman test on the ciphertext and computed the IC (see the result in Figure 16). Both indicated, that the Borg cipher is encrypted using the monoalphabetic substitution.

Thus, we finally used the *Monoalphabetic Substitution Analyzer* component of CT2 to break the cipher, see Figure 17. We tested different languages to be used by the analyzer. Latin produced the best results, since the original text is Latin. The analysis run took 8 seconds.

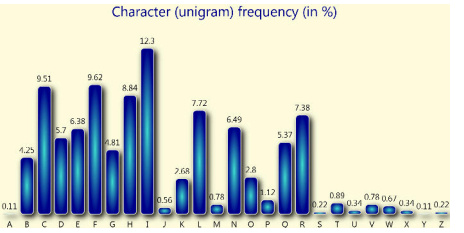

Figure 15: Letter Frequency Analysis of the Borg Cipher

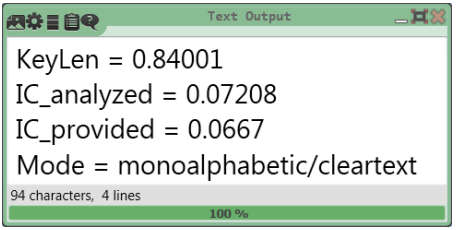

Figure 16: Friedman Test of the Borg Cipher

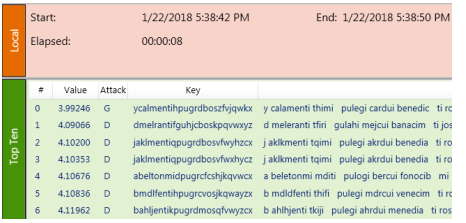

Figure 17: Breaking the Borg Cipher with the Monoalphabetic Substitution Analyzer

We present the first part of the finally decrypted Borg cipher in Figure 18.

6 Current Cryptanalysis Components in CrypTool 2 and Open Tasks

CT2 contains a set of different components for the automated cryptanalysis of classical ciphers. In Table 1 we show an overview of already implemented components for the cryptanalysis of classical ciphers. Green marked entries refer to components which we already implemented. Yellow marked entries refer to components that are not implemented yet. The monoalphabetic substitu-

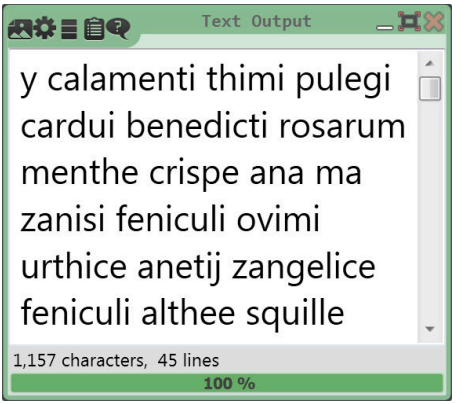

Figure 18: Borg Cipher – Revealed Plaintext by Monoalphabetic Substitution Analyzer

tion, the columnar transposition, the Vigenère cipher, and the Enigma machine are already breakable using CT2. We plan to implement homophone cipher analysis, codebook cipher analysis, grille analysis (a special transposition cipher), Playfair (a special substitution), strip and cylinder cipher analyzers, ADFGVX analyzer, analyzers for the Hagelin Machine (Lasry et al., 2016a) (Lasry et al., 2016b) (e.g. the M209 cipher machine), and a Hill cipher analyzer.

The currently implemented analysis tools, like frequency analysis, transcriptor, or Friedman test, were shown and used in Section 5. For the future, we also plan to implement a "Cipher Detector" component which is able to automatically detect the used type of cipher (with a high probability).

7 Conclusion

In this paper, we gave a brief introduction in the e-learning program CrypTool 2 (CT2) and how it can be used to automatically cryptanalyze classical ciphers. First, we gave an introduction to classical ciphers as well as to their cryptanalysis. Then, we shortly presented CT2 and its usage. After that, we showed an approach consisting of three steps (transcription, identification, and analyzing) for breaking classical ciphers using CT2. We showed two classical real-world ciphers ("Message in a Bottle Sent to General Pemberton in the US Civil War" and "Borg Cipher – Encrypted Book from the 17th Century") and described step-by-step how we broke them with components already implemented in CT2. Then,

Cipher	Component
Mono. Substitution	Mono. Subst. Analyzer
Homophone Subst.	Homophone Analyzer
Colum. Transposition	Transp. Analyzer
Codebook	Codebook Analyzer
Vigenère	Vigenère Analyzer
Grille	Grille Analyzer
Playfair	Playfair Analyzer
Enigma Machine	Enigma Analyzer
Strip/Cylinder Ciphers	Strip/Cylinder Analyzer
ADFGVX	ADFGVX Analyzer
Hagelin Machines	Hagelin Analyzer
Hill Cipher	Hill Cipher Analyzer
Analysis Tools	**Component**
Transcription	Transcriptor
Friedman Test	Friedman Test
Kasiski Test	Kasiski Test
Text Freq. Analysis	Text Freq. Analysis
Index of Coincedence	Cost Function
Autocorrelation	Autocorrelation
Cipher Detector	Cipher Detector

Table 1: Cryptanalysis Components for Classical Ciphers in CT2 – Overview

we gave an overview of methods for the automatic cryptanalysis already implemented in CT2 as well as an overview of cryptanalytic components that we plan to implement.

CT2 is a project that now runs for nearly 10 years. Within this time, we extended CT2 with state-of-the-art methods for the cryptanalysis for classical as well as for modern ciphers. CT2 contains possibilities to cryptanalyze ciphers by connecting different CT2 instances over the Internet (*CrypCloud*). In the future, we plan to extend CT2 in such a way that it becomes easier and more user-friendly, thus, non-computer scientists can more easily use it for breaking their classical ciphers. There are still a lot of open tasks besides the implementation of cryptanalytic methods. We plan to extend the existing components by a huge set of different languages (e.g. Latin, Greek, Hebrew, etc). Since many of the historical encrypted books are written in these languages, historians and cryptanalysts will benefit by the newly added languages. Furthermore, we will extend existing cryptanalytic components to be more robust and more general with respect to the used alpha-

bets. Currently, the monoalphabetic substitution analyzer needs (for the transcription) a specific input alphabet consisting of Latin letters. Till end of 2018 all kind of symbols a computer can process will be possible (e.g. a support of UTF-8 characters). Furthermore, new kinds of classical ciphers and cryptanalytic methods will be added. Examples are grilles and codebooks, which were extensively used in history.

The CT2 team highly welcomes suggestions, wishes, and ideas of historians, cryptanalysts, and everybody else for additional ciphers and automated cryptanalysis methods which should be included in CT2 in the future. The list shown in Table 1 is open for new entries proposed by everyone. Since CT2 is open-source software, we welcome everyone in contributing to the CT2 project (programmers, testers, etc). Finally, everyone interested in CT2 may download the software for free from https://www.cryptool.org/.

References

Nada Aldarrab, Kevin Knight, and Beata Megyesi. 2018. The Borg.lat.898 Cipher. `http://stp.lingfil.uu.se/~bea/borg/`.

Michael J Cowan. 2008. Breaking short playfair ciphers with the simulated annealing algorithm. *Cryptologia*, 32(1):71–83.

Daily Mail Reporter. 2010. CIA codebreaker reveals 147-year-old Civil War message about the Confederate army's desperation. `https://dailym.ai/2JkVFCu`.

Amrapali Dhavare, Richard M Low, and Mark Stamp. 2013. Efficient cryptanalysis of homophonic substitution ciphers. *Cryptologia*, 37(3):250–281.

William S Forsyth and Reihaneh Safavi-Naini. 1993. Automated cryptanalysis of substitution ciphers. *Cryptologia*, 17(4):407–418.

William Frederick Friedman. 1987. *The index of coincidence and its applications in cryptanalysis*. Aegean Park Press California.

James J Gillogly. 1995. Ciphertext-Only Cryptanalysis of Enigma. *Cryptologia*, 19(4):405–413.

Nils Kopal, Olga Kieselmann, Arno Wacker, and Bernhard Esslinger. 2014. CrypTool 2.0. *Datenschutz und Datensicherheit-DuD*, 38(10):701–708.

Nils Kopal. 2018. Secure Volunteer Computing for Distributed Cryptanalysis. `http://www.upress.uni-kassel.de/katalog/abstract.php?978-3-7376-0426-0`.

George Lasry, Nils Kopal, and Arno Wacker. 2016a. Automated Known-Plaintext Cryptanalysis of Short Hagelin M-209 Messages. *Cryptologia*, 40(1):49–69.

George Lasry, Nils Kopal, and Arno Wacker. 2016b. Ciphertext-only cryptanalysis of Hagelin M-209 pins and lugs. *Cryptologia*, 40(2):141–176.

George Lasry, Nils Kopal, and Arno Wacker. 2016c. Cryptanalysis of columnar transposition cipher with long keys. *Cryptologia*, 40(4):374–398.

George Lasry, Ingo Niebel, Nils Kopal, and Arno Wacker. 2017. Deciphering ADFGVX messages from the Eastern Front of World War I. *Cryptologia*, 41(2):101–136.

George Lasry. 2018. *A Methodology for the Cryptanalysis of Classical Ciphers with Search Metaheuristics*. kassel university press GmbH.

Thomas G Mahon and James Gillogly. 2008. *Decoding the IRA*. Mercier Press Ltd.

Beata Megyesi, Kevin Knight, and Nada Aldarrab. 2017. DECODE – Automatic Decryption of Historical Manuscripts. `http://stp.lingfil.uu.se/~bea/decode/`.

Phil Pilcrow. 2018. CryptoCrack. `http://www.cryptoprograms.com/`.

Tobias Schrödel. 2008. Breaking Short Vigenere Ciphers. *Cryptologia*, 32(4):334–347.