ADVANCES IN CYBER-PHYSICAL SYSTEMS

Vol. 6, No. 1, 2021

USING IMPROVED HOMOPHONIC ENCRYPTION ON THE EXAMPLE OF CODE ZODIAC Z408 TO TRANSFER ENCRYPTED INFORMATION BASED ON DATA TRAFFIC

Oleksandr Mamro, Andrii Lagun

Lviv Polytechnic National University, 12, Bandery Str, Lviv, 79013, Ukraine
Authors' e-mail: *oleksandr.mamro.kn.2017@lpnu.ua, andrii.e.lahun@lpnu.ua*

https://doi.org/10.23939/acps2021.01.027

Submitted on 03.03.2021

Abstract: **This article examines the improvement of homophonic encryption using the well-known Z408 cipher as an example, and analyzes their problems to further correct the main shortcomings of this typea of encryption. In the practical part, an encryption system similar to Z408 was created and the complexity ofthe language analysis for this cipher was increased. The correlation dat of the number of alternative characters was used to improve the transmission of encrypted data depending on the load of the device. The more symbols and the higher the load, the fewer the number of alternative symbols will be used.**

Index Terms: **Zodiac, Z408, encryption, homophonic encryption, quenching, cyber-physical systems.**

I. INTRODUCTION

Advances in modern cryptography can reliably protect information from unauthorized access and distortion of information; allow you to resolve the issue of document authorship and many other useful data protection operations. However, not everything is so cloudless. Despite the fact that energy-efficient cryptographic protection microchips are built into many IoT devices, they cannot protect against hacking or data leakage if the user of the smart device does not use them for their intended purpose or the attacker is especially persistent. For example, private keys can be read from the memory of an IoT device, calculated from a dynamic change in the supply current, or even from electromagnetic radiation.

Security of smart devices can be enhanced by installing additional security electronics – miniature components that connect peripherals to host micro-controllers or microprocessors and are responsible for personalized certificates, secure placement of private keys, and management of cryptographic elements. Or the need to improve the reliability of the confidentiality of transmitted data not at the hardware level, but at the software level. This article explores homophonic encryption systems to improve data resiliency and discusses the weaknesses of homophonic cryptography, and with the help of gamma and entropy slice increase the reliability of the cipher and improve resistance to speech analysis.

II. OBJECT, PURPOSE AND TASKS OF THE RESEARCH

One of the most difficult issues in the operation of homophonic encryption is the degree of protection of homophonic encryption from speech analysis, or rather the quality of masking the frequencies of occurrence of certain letters in the text. therefore, the possibility of reducing the repetition rates of text characters, and the corresponding complication of protection against cryptocurrencies of frequency analysis should be considered.

Even though it was originally a crib which helped to crack it, Z408 has other weaknesses, most notably the way it sequentially cycles through homophones ("multiple ciphertext shapes for the same plaintext character"). For example, plaintext 't' maps to the four ciphertext homophones HI5L, and appears in the text as the sequence HI5LHI5ILHI5LHI5LHI5LHI5LI5LHL5IIHI. If you count each successful letter-to-letter transition matching the modulo-4 sequence [HI5L] as a 0.25 success event (=26) and each non-match (=8) as a 0.75 failure event, I believe you get a raw probability of less than 1 in a billion (i.e. of at least 26 successes from 34 events).

Frequency attack on simple substitution ciphers is a kind of greedy algorithm method. However, a frequency attack does not require a letter combination exceeding the 3rd graph ("trigraph".) The trigraph evaluation table consists of all combinations of any 3 consecutive English letters. For example, in English the combination "aba" has a frequency of 0.0075 %, and the combination "epa" has frequency 0.1086 %. To solve the problem of homophonic substitution, the cryptanalyst much higher tables of N-plots should be used which increase the probability to distinguish meaningful text from meaningless text.

This cipher rate approach assumes that the frequencies of all cipher symbols are equal balanced. Hence, the matches are selected based on "quantity" (i.e., numbers cipher characters.) This frequency-based encryption approach does not work on modified homophonic substitution cipher because the cipher character frequencies vary. In the case of the Z340 Zodiac, the frequency of all 62 encrypted characters is not consistent. Some

of the encrypted characters have much higher frequencies than others.

Another possibility would be the columnar double transposition method. In transpositional encryption involves reordering the cipher characters. in

In the process of decryption, the reverse operation is performed. In a double column transposition, the original message is written in matrix form, which is a series of lines with a fixed length (that is, a fixed number of columns). Row and Column Order modified based on predefined keys

The aim of this work is to try to improve and analyze the influence of the shift of block structures in homophonic ciphers, their influence on the frequency analysis of the text. Additionally, there is cut off the noise in the source text before the encryption phase, which changes the language structure and its impact on language analysis.

III. THE MAIN PART

With homophonic substitution, the frequency of each letter is balanced to prevent any statistical-based approach.

The method uses the advantages of the existing English dictionary. Typically, the dictionary used for cryptanalysis contains approximately, but not limited to, from 100.000 to 120.000 words. This approach is to use trial and error methods on some consecutive letters of the ciphertext (ie from the letter i to the letter ($i + j$), CJ .. ($J + J$). The attack method looks for similar words in a dictionary that has a similar pattern to Ci .. ($j + j$).

This method of attack assumes that the cipher is mono-alphabetical. If the cipher is a polyalphabetic substitution, this attack method cannot reuse any of its successful mappings in the attack process because the polyalphabetic substitution uses a many-to-many technique. The many-to-many display technique allows a cipher character to represent many letters of plaintext, while plaintext letters can also be represented by many ciphers characters.

Another disadvantage of a dictionary-based attack is misspelled English words. The attack itself relies heavily on the existing vocabulary. Therefore, if the cipher contains non-standard or intentionally misspelled words, the attack not provides reasonable and readable relevant plaintext – which can be used to increase protection against cryptocurrencies.

A homophonic cipher that uses a cipher alphabet with N different cipher characters has a theoretical key space of 26N compared to a theoretical key space of simple replacement being only 26! ~ $4{,}033\times10^{26}$. Due to the unmanageable size of this key space, a comprehensive key search is not possible.

For example, for a relatively short message that uses an alphabet of ciphers with N = 50 different cipher characters and a comprehensive search for keys using a personal computer (which can check approximately 106 keys/second) we have to take 2650 keys/106 keys/s = $5.6 \cdot 10^{64}$ s = $=1.8 \cdot 10^{57}$ years.

However, if the message is encrypted with a simple replacement, a comprehensive key search can only be performed in 26! keys/106 keys/s = $4.03 \cdot 10^{20}$ s = $= 1.28 \cdot 10^{13}$ years. The difference between the homophonic substitution key space and the simple substitution key space increases exponentially with the number of cipher characters in the cipher alphabet.

Of these strengths, we have many options to make them even better, and provide a high level of information protection by adding block offset after homophonic encryption and cutting off entropic noise from the source text.

The EMF quality factor, as a measure of the difference between the arbitrary probability density $p(x)$ and the probability density of the normal (Gaussian) distribution $w(x)$, is a functional that depends on the difference between the functions $p(x)$ and $w(x)$, ie

$$Y = Q\{p(x) - w(x, a_1, a_2, ..., a_n\},$$

where a_1, a_2, ..., a_n – parameters that characterize the distribution density $w(x)$. This definition involves calculating the functional by minimizing it by the parameters a_1, a_2, ..., a_n and to obtain values of the functional close to zero. Therefore, to find and select weak links and their subsequent removal, we use

$$H = -\sum_i P_i * log_2 P_i$$

where P_i is the probability of the i-th result.

For block encryption we can take the mode of tempering. This mode has no disadvantages of the simple replacement mode. Gamma mode is so called because it uses gamma – a pseudo-random sequence, which in each round is composed of module 2 with plaintext.

The gamma is formed with a synchronization S – pseudo-random sequence, which changes with each iteration and is encrypted in the mode of simple replacement, then converted into gamma and superimposed on the plaintext of Fig. 1.

To improve homophonic encryption on the example of the code Z408, we have used the practical implementation of the code Z408.

Starting with the clipping of entropic noise from the source text, as mentioned above, using the formula of the quality factor of EMF noise, we add to the source text such processing of Fig. 2.

Thus we receive the source text passed through noise and receive the text with the brought down structure of language.

There is an example of encryption process:

Source text: *I like to bill people For my investigation*

Received text: *IlietobilpeoplFormyivstgation*

Rejecting 25 % of our text, we still have a readable structure for humans, and the language structure itself is no longer 100 % correct, which further adds complexity to language analysis.

Plaintext: There are many variations of passages of Lorem Ipsum available but the majority have suffered alteration in some form by injected humor or randomized words which don't look even slightly believable If you are going to use a passage of Lorem Ipsum, you need to be sure there isn't anything embarrassing hidden in the middle of text.

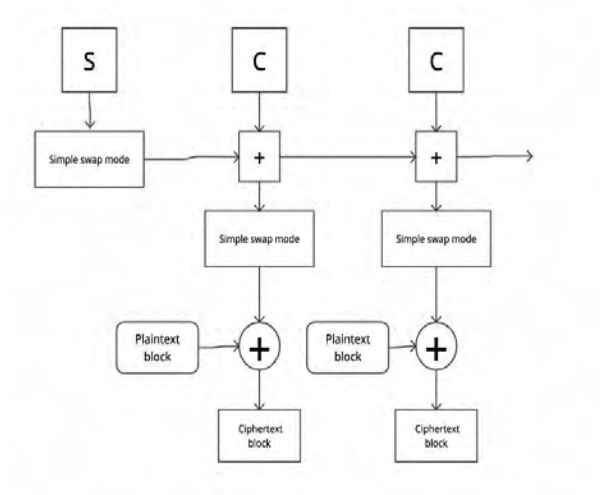

Fig. 1. The scheme of manipulation of blocks by means of hardening

All the Lorem Ipsum generators on the Internet tend to repeat predefined chunks as necessary making this the first true generator on the Internet.

It uses a dictionary of over Latin words combined with a handful of model sentence structures to generate Lorem Ipsum which looks reasonable.

The generated Lorem Ipsum is therefore always free from repetition injected humor or non-characteristic words etc.

With noise text:

ThreareanyvaraionsofpssaesoforemIpsmaailablbutthemajriyhavsuferealertioninomeormbyinjeteumourorandomisdworswichdotloovenslihtybeievabeIfyouareoingtouseaassageoforemIpsm,youneetobesurheresntanyingembarasighiddninthemiddloftxtAlltheLormpsumgneratosonthentenettedtoreeatpredenedhunksasncesaryakingtisthefirttueenerarontheIntrntItussaictonaryofoveLainworscombiewthahndfulofodelsentnctructrestogeneaeoremIsumwhicloksreasnbleThegeneaedLremIsumistherefrawaysfrefroretitionijectdhumuroron-racteristicworsetc

Encrypted text: *451 856 506 601 113 742 601 113 428 687 997 60 506 794 762 893 428 28 715 156 202 850 520 113 758 154 436 685 436 353 601 406 967 202 578 406 923 596 764 809 113 675 809 675 725 305 187 148 992 473 113 745 353 967 644 394 923 332 154 725 598 583 742 758 60 907 758 742 539 967 436 418 762 418 577 473 583 336 544 406 675 288 762 686 6 992 756 758 775 473 893 775 544 115 711 60 876 376 369 406 29 850 811 22 577 353 578 22 816 284 946 811 970 187 809 336 970 997 601 418 850 907 762 448 305 191 675 758 29 758 384 596 675 758 762 156 191 715 335 60 … .. (abbreviated)*

Frequency analysis of Fig. 3, 4 for the implemented step of changing the structure shows us that the result did not change to tangible values (we can even level by reducing repetitions referring to error), and the main purpose of this step was to minimize text coherence. This means that frequently used repetitions of English words ll, nn, or syllables are caught by language analysis much worse.

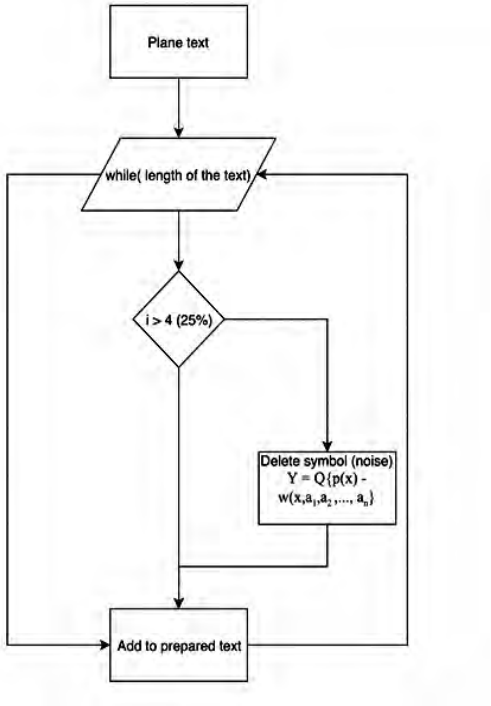

Fig. 2. Clipping of entropic noise from the source text scheme

The next step is to skip our received text, after the deformation of the language structure, through block encryption by the method of gamification, for this we need to go through several steps:

The first is to extend our alphabet to registers and special characters for further use in tempering:

Alphabetic symbols structure for now: abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ=/+-1234567890@;^<>_:\!?*

And it is necessary to add functionality for gamification with the corresponding XOR functionality as was mentioned in Fig. 1 scheme.

Example of implementation Code snippet has shown on Fig. 5.

Thus we get the following text:

GUM text: *E4GT = GT14hJ4c = <^ 2F ^: EPO4VO: WPGT1 \ aO @ P = <] = 7]> @ eH = T14 [3Ge4CTI3TN0U = 9eN4e5; X2: \ 93 ^ 17h5; [? 0U4 @ ^ IG ^ NGP21 ^ 1 <T3GUO = X? = ^ 2A ^ 3> TJ0b52YHLS99X97] 9 \ WE: d = GT; <_; A ^ IFT = EPOFP;: WP: c9EbI8h3 @ _91e37TOGTH0c9F_H4_H = X; 0S = GcO <_; = X80_ e40 \ 51] 9: WHMe] 9] H = TP: c9 \ aI8V9; TN4cO: _H = T2AT20e9; UH: c9ETHEc91X20U? @_ 7FPO; T9FPNL \ => X; AYOAY9 <cOAeI0V9; c = A ^ 3_E4G4; e408; ; PN: WJ0cP4e5: c8FR38S5; XH = P =; U: @] 33 \ 30] O0_9; RN @ RH @ c9Fe32T20c = Y ^ N88LFdK <R49 ^ 3> c94b3; P9aY92_9GPH0 ^ N0 \ LFd1 <b40c93 : \ N0T5AX3; X26e91YI8 ^ I: c3; œ? = PN4RH0XH6f3GU9AR*

Using the received text in our homophonic encryption Z408 we receive absolutely destroyed language structure which is exposed to language analysis many times more difficult, and more exact information is resulted on Fig. 6.

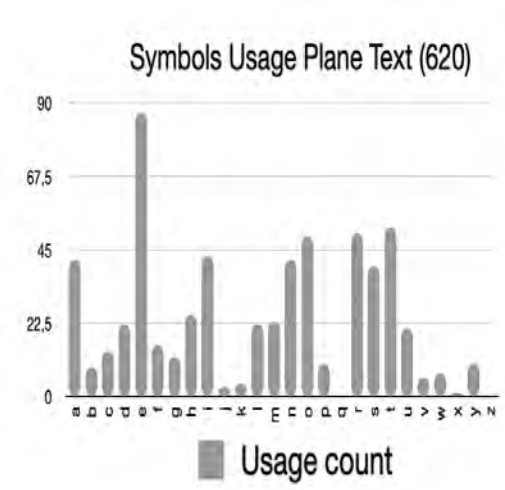

Fig. 3. Symbolic repetition of frequency analysis of the source text

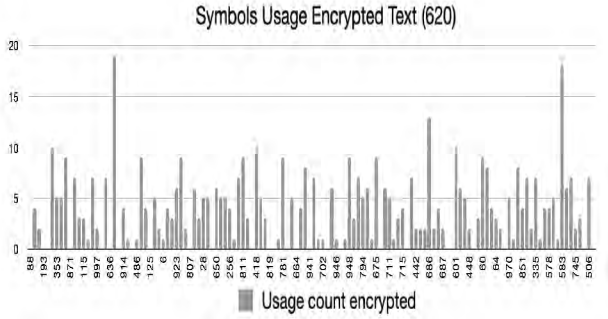

Fig. 4. Symbolic repetition of frequency analysis in ciphertext after beating the language structure of the source text

Encrypted text: *0 235 687 720 306 545 658 186 33 610 663 975 243 162 201 526 360 278 959 456 103 253 733 379 613 1 456 91 808 270 136 876 3 210 733 797 952 791 703 306 188 373 511 494 405 791 136 759 345 400 426 500 932 546 658 388 400 720 110 809 660 306 30 694 110 345 694 301 180 881 551 456 525 89 267 959 559 206 991 222 31 378 414 598 345 646 983 388 518 983 608 545 181 647 759 583 759 703 658 762 81 179 733 306 881 137 299 583 172 287 448 481 534 136 713 199 334 885 647 994 208 430 147 89 30 881 644 956 510 14 701 103 442 22 306 545 136 735 703 527 31 297 448 388 877 951 306 103 191 733 523 808 180 242 517 191 746 477 30 626 334 388 782 142 267 60 338 30 785 500 697 863 658 101 332 874 35 199….. (abbreviated)*

And finally, after ours different practical investigations, we can see that homophonic encryption with very simple additional operations can be very reliable and protected from external interference scripting. Additionally, no complex computational operations occurred, so this encryption method does not require high machine power. Analyzing the graph of symbolic repetition of frequency analysis in encrypted text (Fig. 6) we can observe a decrease in frequencies by another 23 % (compared to homophonic encryption without additional action.) But a great achievement is the lack of language structure, which makes this method of encryption analysis.

```
private static int[] encryptGum(String str, String key) {
    int[] output = new int[str.length()];
    for (int i = 0; i < str.length(); i++) {
        int o = (Integer.valueOf(str.charAt(i)) ^ Integer.valueOf(str.charAt(i %
(key.length() - 1)))) + '0';
        output[i] = o;
    }
    return output;
}

private static int[] string2Arr(String str) {
    String[] sarr = str.split(regex: ",");
    int[] out = new int[sarr.length];
    for (int i = 0; i < out.length(); i++) {
        out[i] = Integer.valueOf(sarr[i]);
    }
    return out;
}

private static String decryptGum(int[] input, String key) {
    String output = "";
    for (int i = 0; i < input.length(); i++) {
        output += (char)((input[i] - 48) ^ (int)key.charAt(i % (key.length() - 1)));
    }
    return output;
}
```

Fig. 5. Hardening functionality

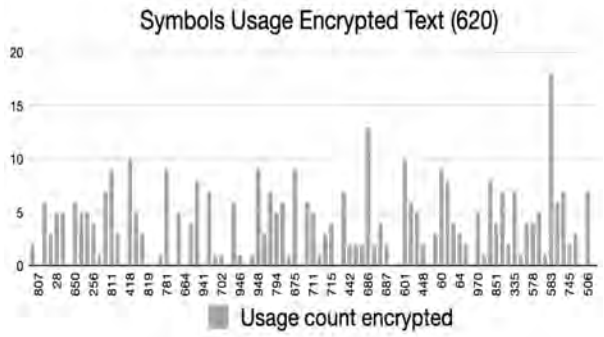

Fig. 6. Graph of language analysis after the destruction of the language structure

Usage in a cyber-physical system (transmitter encrypting incoming source information).

With high workload and high volumes of data, they are encrypted using fewer alternative characters and reduce the protection of the original text. With a low load on the transmitter, the data is encrypted using the maximum available number of alternative characters for the device, which the transmitter's RAM allows. Thus, the highest protection for the original text is preserved.

With this approach, we can reduce the load on the transmitter and encryption device (on Fig. 7) based on the amount of incoming data.

With a large amount of incoming data, the level of protection of the original information will be reduced, but due to the amount of this data, it will be very difficult to find the “necessary” information package.

With a small amount of information, the required information package will be much easier to find, but due to the high level of protection of each information package, it will be much more difficult to gain access to the package (decrypted) (correlated from alternative pairs of characters).

In this case, all data passing through all involved communication channels, including the text of the message, as well as technical information about its routing, communication protocol, etc., undergoes a cryptographic transformation.

The participants in the data transfer (for example, a switcher) will decrypt the incoming stream in order to process it, then encrypt it and transmit it to the next network node, which is an effective means of protecting information in computer networks. Since all information is encrypted, a potential attacker has no additional information about who is the source of the data, to whom it is sent, and so on.

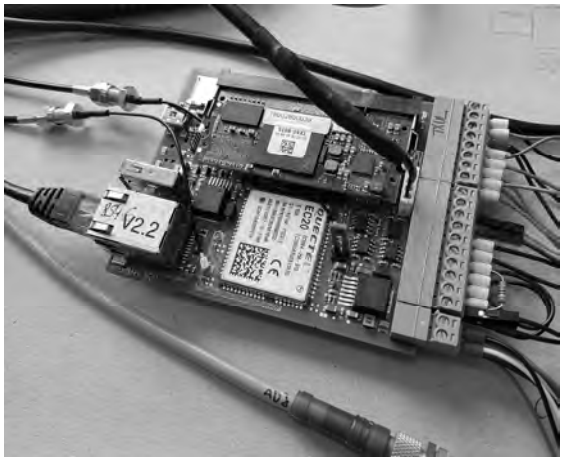

Fig. 7. Data transmitter and encoder

The simplified system (on Fig. 8) does not include methods for monitoring and collecting data for process control and processing of passing data. Also, for a simplified system, you can use a less powerful processor and less memory, because all information is processed by breaking it down into sectors, which allows you to launch this device into production with lower financial costs. If there is a need for more efficient and faster processing, it could be done by raising the amount of memory used, which will expand the capacity of the encryption blocks and reduce the number of sectors required for processing.

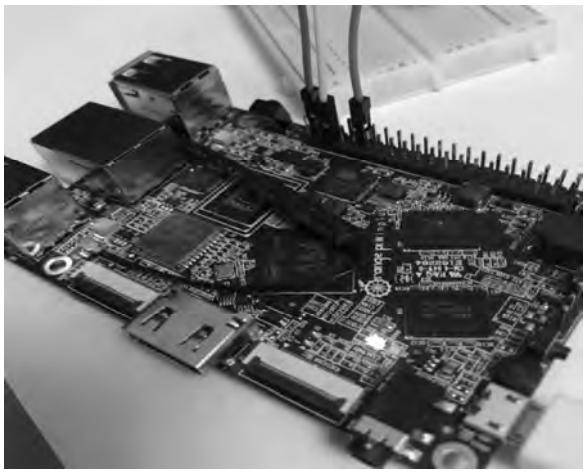

Fig. 8. Example of a simplified data transmitter and encoder system

IV. CONCLUSIONS

Today, computer security is a house of cards that can crumble at any moment. So many weak products have not yet been hacked just because they are underutilized. Once they become widespread, they will attract criminals.

And in the future, as commerce and communications become increasingly tied to computer networks, cryptography will become vital. But the cryptographic tools on the market do not provide the level of protection promised in the advertisements. In the end, the security of these products will determine the victory in the crypto product market.

In this article, homophonic encryption was considered, and encryption was improved using entropy text slicing and block encryption. The analysis of the homophonic cipher with improvements on the example of the known cipher Z408 was realized and by its analogy the automatic algorithm of ciphering of the derived text was developed.

For example, it was proved that the entropy slice of the derived text complicates language analysis, prevents unauthorized access to information and additional block operations completely close the possibility of cryptocurrencies of frequency analysis due to the complete absence of frequencies of letters in the text (due to impaired language structure).

These studies have been used to create an improved method for transmitting encrypted data, depending on the workload of the physical transmitter.

REFERENCES

[1] Polyalphabetic cipher. [Online]. Available: https://en.wikipedia.org/wiki/Polyalphabetic_cipher.

[2] Analysis of the Zodiac 340-cipher. [Online]. Available: https://scholarworks.sjsu.edu//cgi/viewcontent.cgi?referer=https://www.google.com/&httpsredir=1&article=4566&context=etd_theses.

[3] D. Welsh, “Codes and Cryptography”. Clarendon Press, Oxford, 1988.

[4] Elonka Dunin, Klaus Schmeh, “Code Breaking and Cryptograms”, A Practical Guide, Robinson, 2020, ISBN 978-1-472-14421-8.

[5] H. Beker and F. Piper, “Cipher Systems”, Northwood Books, London, 1982.

[6] Bruce Schneier, “Applied Cryptography. Protocols, Algorithms, and Source Code in C”, John Wiley & Sons, 1996.

[7] Welsh, Dominic (1988). Codes and Cryptography, Oxford University Press, A brief textbook intended for undergraduates.

[8] Smart, Nigel (2004). Cryptography: An introduction ISBN 0-07-709987-7. Similar in intent to Applied Cryptography but less comprehensive.

[9] Patterson, Wayne (1987). Mathematical Cryptology for Computer Scientists and Mathematicians, Rowman & Littlefield, ISBN 0-8476-7438-X

[10] Mao, Wenbo (2004). Modern Cryptography Theory and Practice ISBN 0-13-066943-1. An up-to-date book on cryptography.

[11] Gaines, Helen Fouché (1939). Cryptanalysis, Dover, ISBN 0-486-20097-3. Considered one of the classic books on the subject, and includes many sample ciphertext for practice.

[12] Katz, Jonathan and Lindell, Yehuda (2007 and 2014). Introduction to Modern Cryptography,[2] CRC Press.

[13] Falconer, John (1685). Cryptomenysis Patefacta, or Art of Secret Information Disclosed Without a Key. One of the earliest English texts on cryptography.

[14] Aumasson, Jean-Philippe (2017), Serious Cryptography: A Practical Introduction to Modern Encryption. No Starch Press, 2017.

[15] Anderson, Ross – Security Engineering, Wiley, ISBN 0-471-38922-6 (online version), advanced coverage of computer security issues, including cryptography. [Online]. Available: https://www.cl.cam.ac.uk/~rja14/book.html

[16] Boak, David G. A History of U.S. Communications Security (Volumes I and II); the David G. Boak Lectures, National Security Agency (NSA), 1973 [Online] Available: https://www.governmentattic.org/18docs/Hist_US_COMSEC_Boak_NSA_1973u.pdf

[17] Budiansky, Stephen, Code Warriors: NSA's Codebreakers and the Secret Intelligence War Against the Soviet Union (Knopf, 2016). (ISBN 0385352662)

[18] Marks, Leo, Between Silk and Cyanide: a Codemaker's Story, 1941–1945, (HarperCollins, 1998). (ISBN 0-684-86780-X)

[19] Levy, Steven – Crypto: How the Code Rebels Beat the Government – Saving Privacy in the Digital Age (ISBN 0-14-024432-8)

[20] Kozaczuk, Władysław, Enigma: How the German Machine Cipher Was Broken, and How It Was Read by the Allies in World War II, edited and translated by Christopher Kasparek, Frederick, MD, 1984

[21] Yardley, Herbert, The American Black Chamber (ISBN 0-345-29867-5), a classic 1931 account of American code-breaking during and after World War I; and Chinese Black Chamber: An Adventure in Espionage (ISBN 0-395-34648-7)

Alexander Mamro is a student of the Department of Information Systems and Technologies of the Institute of Enterprise and Advanced Technologies of Lviv Polytechnic National University. His research interests include software development for databases, business requirements analysis and software specification, design and research of cryptographic systems, building management of organizational processes and business tasks.

Andrii Lagun is a Head of the Department of Information Systems and Technologies of the Institute of Enterprise and Advanced Technologies of Lviv Polytechnic National University. In 2002 she obtained PhD in the field of "Elements and Devices of Computer and Control Systems" (degree of Candidate of Technical Sciences). His research interest includes Information theory, Investigation of Cryptographic and Steganographic algorithms and systems, Using Wavelet and Fourier Transforms in Cyberphysical and Cybersecurity Systems and also in the Internet of Things, Different Types of web and mobile Applications.