Encryption Algorithms for Secure Communication over the Internet

Prof. Dr. Thomas ENGEL*, PhD Student Asya MITSEVA†, BICS student: Desislava MARINOVA ‡
Faculty of Science, Technology and Communication, University of Luxembourg, Luxembourg

*Email: *thomas.engel@uni.lu, †asya.mitseva@uni.lu, ‡desislava.marinova.002@student.uni.lu*

***Abstract*—This document is a template for the scientific and technical report that is to be delivered by any BiCS student at the end of each Bachelor Semester Project (BSP). The Latex source files are available at: https://github.com/nicolasguelfi/lu.uni.course.bics.global**

This template is to be used using the Latex document preparation system or using any document preparation system. The whole document should be in between 6000 to 8000 words and the proportions must be preserved. The other documents to be delivered (summaries, . . .) should have their format adapted from this template.

1. Introduction (± 5% total words)

We are living in the information age enabled by computing and communication technologies whose rapid evolution is almost taken for granted today. This is an age dense with electronic connectivity, eavesdropping and fraud. We send messages around the world instantaneously and we do not want them to be intercepted or stolen. Thus, information is an asset that has precious value like any other. Societys reliance on a changing panoply of information technologies and technology-enabled services, the increasing global nature of commerce and business, and the ongoing desire to protect freedoms all suggest that future needs for data security is vital. Network based attacks spring up every day, distance does not matter anymore since data has been digitalized and there is the issue of privacy being collected, processed and misrepresented. For this reason, the disciplines of cryptography and network security have matured. Thus, internet security is complex yet fascinating. To develop a security mechanism (algorithm or protocol) one must always consider potential attacks, take countermeasures, act like an intruder and go against ones own intuition and exploit eventual weaknesses in the algorithm.

We dive into the terminology that governs our topic in the next paragraph. The word cryptography comes from two Greek words meaning secret writing [1]. Cryptography is a vast, rich subject. It is the art and science of concealing meaning, of keeping the enciphered information secret and of designing and analyzing encryption schemes. It involves mathematics as a supporting tool. Cryptanalysis is the breaking of codes and uses statistical and mathematical approaches. The areas of cryptography and cryptanalysis together are called cryptology. The basic component of cryptography is a cryptosystem which is based on substitution (it changes characters in the plaintext to produce the ciphertext) and permutation (an ordered sequence of elements in a finite set with each character appearing only once). If someone wants to break the ciphertext, knowing the algorithm but not the specific cryptographic key, there are 3 main types of attacks [Ref: Stallings, 7th ed, p.90]: - ciphertext only attack ciphertext known, the goal is to find the corresponding plaintext. The key may be found as well if possible. (A Caesar cipher is susceptible to a statistical ciphertext-only attack) - known plaintext attack both ciphertext and plaintext and known; the goal is to find the key - chosen plaintext attack specific plaintexts are enciphered in order to find out the corresponding ciphertexts and the goal is to find the key. a good cryptosystem protects against all three types of attacks.

We often stumble upon impending security threats such as information leakage; integrity violation; denial of service; illegitimate usage; trojan horses; insider attacks; problems related to access or authentication control. Thus, there are instilled security requirements of utmost importance. We present them as follows: confidentiality; integrity; authentication; availability; access control; non-repudiation, which are often intermingled. Hence, one of the mechanisms promising to ensure security is encryption, along with digital signatures and hashing.

In order to make information secret, we use a cipher an algorithm that converts plain text into ciphertext, which is in unreadable form, unless we have a key that lets us convert back the cipher. The process of making text secret is called encryption, and the reverse process is called decryption.

There are two types of encryption: symmetric and asymmetric. For this project, we shall focus mainly on the symmetric, also called conventional. It comprises a single key for both encryption and decryption and it is the preferred choice when we need to encrypt large amounts of data. On the other hand, asymmetric cryptography, also called public key and two-key, consists of two keys: public and private. The public key is used to encrypt the message, whereas the private key is used to reverse the encryption. This method cannot deal with large quantities of data due

to the runtime of supporting processes employed. However, it is more secure when we want to transfer over an insecure channel. Nevertheless, the choice to be made on which system to use for encryption and decryption depends on the purposes and on each algorithms functions.

2. Project description (± 10% total words)

2.1. Domain

The report draws on a variety of disciplines. However, it is impossible to appreciate the vastness and significance of the topic in a limited amount of content pages. Nevertheless, we attempt at making the report self-contained by providing the reader with an intuitive understanding of our survey and application results. The domains that are associated with our project are:

- Internet security: "measures to protect data during its transmission over a collection of interconnected networks" [Stallings, ed.3,]
- Computer Security: "generic name for the collection of tools designed to protect data and to thwart hackers and malicious software (viruses) " [Stallings, ed.3,]; at its heart there are three key objectives: confidentiality, integrity and availability
- Cryptographic primitives
- Symmetric-key cryptography
- Cryptographic algorithms: "study of techniques for ensuring the secrecy and/or authenticity of information =¿ 3 main studies of this category (1) symmetric encryption, (2) asymmetric encryption and (3) cryptographic hash functions" [Stallings, 5th ed, p. 3]

2.2. Objectives

We aim for a comprehensive survey of both principles and practice of cryptography and network security. The objectives of this project are three-fold.

1) As a preliminary task, we address a highly needed background material. We give a basic introduction to the concepts and primitives of cryptography, i.e. transposition/permutation, encryption/decryption, symmetry/asymmetry, block ciphers/stream ciphers and how they obtain their security in constantly emerging attacks
2) Our main focus falls on symmetric encryption, taking into account both classical and modern algorithms. Therefore, we explore the following encryption algorithms: Caesar cipher, ROT13, One-time pad, DES, 3DES, AES in order to yield two types of approaches: information theoretic and computational
3) Lastly, we implement an encrypted chat system that could be of use to provide network security.

2.3. Constraints

Our project takes into account substitution/transposition; block/stream ciphers but we do not consider:

- Asymmetric ciphers (public-key algorithms, including RSA and elliptic curve);
- Data integrity algorithms (cryptographic hash functions, message authentication codes (MACs); digital signatures));
- Authentication techniques: key management and key distribution topics (protocols for key exchange);

3. Background (± 15% total words)

3.1. Scientific

Historical ciphers

We would like to highlight that ciphers have been used long before apparition of computers. In 1900BC, there were nonstandard hieroglyphics. In 600BC, there were Atbash ciphers used to encrypt the Hebrew alphabet.

We go to ancient Rome times with Julius Caesar's wish to encrypt his private correspondance and use what now is called the Caesar cipher. The cipher replaces a letter from the alphabet with another that is 3 places further down. To decrypt the message, the other party needs to know both the algorithm and the shifting number.

ROT13 is a simple monoalphabetic substitution cipher, special class of Caesar cipher, that encodes a certain letter with another letter that is 13 positions after it. Only those letters which occur in the English alphabet are affected; numbers, symbols, whitespace, and all other characters are left unchanged. It is an example of a cipher providing weak encryption since both operations encryption and decryption are identical, hence, this cipher is its own inverse. Because we know there are 26 letters in the English alphabet, if we wish to apply twice 13 it would give us one shift of 26, thus restoring the original text. Moreover, the direction of the shift is of no importance since it will always give the same output. [REF: p. 5 https://books.google.lu/books?id=oLoaWgdmFJ8Cpg=PA5redir$_esc = yv = onepageqf = false].ROT13isusedinonlineforumsasameansofhidingspoilers, puWikipedia].

Substitution ciphers

In order to produce the ciphertext, a substitution cipher modifies characters in the plaintext. There are two types of substitution ciphers: simple and polyalphabetic. We can break the former by using letter frequency analysis since letter frequencies are preserved (e.g. E is the most common letter in English). The latter can be broken by decomposing into individual alphabets and as a consequence to treat is as a simple substitution cipher. A historical example of breaking the substitution cipher is the execution of the Queen of Scots, Mary, in 1587 for plotting to kill Queen Elizabeth.

One - time pad (1917)

It is considered unbreakable and for this is employed by Russian spies, the CIA covert operations and the ”hotline” Washington-Moscow. A pad (secret key) is never reused, what is more, its ”random key is as long as the message, utilized to both encrypt and decrypt a single message and afterwards it is discarded”(Stallings, ed.5). To generate a new message, a new key is required, thus, this scheme leads to an output unrelated to the plaintext. If we try to decipher a ciphertext using this method, it would be difficult to decide upon the correct decryption if we somehow managed to find the two keys. Stallings states there are ”two fundamental difficulties” related to the one-time pad: (1) making large quantities of random keys; (2) key distribution and protection (since a key of equal length is needed by both sender and receiver). Hence, the author argues that the one-time pad is ”useful primarily for low-bandwidth channels requiring very high security”. In addition, Stallings explores the concept of the one-time pad being ”the only cryptosystem that exhibits (...) perfect secrecy”. Cipher machines attempted unsuccessfully, first mechanically, then electronically, to create approximations to OTPs. Many snake oil algorithms claim unbreakability by claiming to be OTPs, hence, giving a rise to pseudo-OTPs providing pseudo-security [REF: http://www.interhack.net/people/cmcurtin/snake-oil-faq.html].

Up until now, we have examined techniques involving substitution of a ciphertext symbol for a plaintext symbol. A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher. For example, a simple cipher of this sort is the rail fence technique, in which the plaintext is written as a sequence of diagonals and read off as a sequence of rows.

In the following paragraph, we introduce the class of systems known as rotor machines, an important application of the principle of multiple stages of encryption. We cite famous rotor machines: Converter M-209 (USA); Typex (UK); Red, Purple (Japan); Enigma (Germany). However, these codes were broken in the wars outcome.

Cipher machines (1920s)

Cryptography was mechanized by the 1900s in the form of encryption machines. The basic component of a cipher machine is the wired rotor. When we get prior to the Second World War, the Germans realized that thanks to radios messages can be sent across the battlefield in an instant but that always meant that the other side could also tapped into those radio channels. Therefore, pens and paper would not be good enough and in order to have high-tech radio communication, we need to have high-tech encryption. The Germans invented Enigma,whose complexity guaranteed their privacy. The Enigma was a substitution cipher, but a more sophisticated one because it used three rotors in a row, each feeding into the next. The Germans typed their messages on a keyboard that came out as gibberish on a lampboard and sent it over the radio to the other side which also has the same Enigma machine to help them decrypt the unintelligible message. The Germans distributed cookbooks with daily settings for the machines. Furthermore, to decrypt the message, it is assumed that the other side must know the algorithm and must have configured the machine the same way as the Enigma machine encrypting the message. The Enigma relies on a random letter generator, hence, its encryption does not seem to follow any kind of pattern. There are 26 wires coming out of the keyboard, running through three rotors with 6 permutations, eventually going into the lamps. Once the first of the three rotors hits a full evolution, it kicks the next rotor to start moving, yet again, when it completes it cycle, it transmits the process to the last rotor. The inside of the rotors resembles scrambled wiring. A rotor’s side has 26 junctions/contacts, accepting the incoming wires and outputting them to the other side of the rotor. The rotor moves each time a letter is typed in. Even if the same letter is typed sequentially, the rotor would move and a lamp representing a different output letter would light up. Thus, the dynamism of the rotors accounts for the complicated encryption. Finally, there was a plugboard at the front of the machine that allowed letters to be optionally swapped so that the machine it is interacting with is configured the same way. It is argued that rotor machines ”would have been secure if they had been used properly”. They used predictable opening sentences: Mein Fuehrer!, Nothing to report, etc. as well as the same key over an extended period of time and messages with old compromised and new keys were encrypted. The circuits configuration showed that it was impossible for a letter to be encrypted as itself, which turned out to be a cryptographic weakness. Alan Turing and his colleagues at Bletchley Park were able to break the Enigma codes and automate the process.

Stream ciphers

A stream cipher is one that encrypts a digital data stream one bit or one byte at a time. Examples of classical stream ciphers are Vigenre cipher and the Vernam cipher. Stream ciphers are vulnerable to attacks, mostly to bit-flipping. We should never reuse a key with a stream cipher. These ciphers have a binary pad keystream and use XOR-operation instead of addition. What is worth noting is that using the keystream and the ciphertext (encrypted data), we can recover the plaintext (original unencrypted data) as well as using the plaintext and the ciphertext, the keystream can be recovered. Furthermore, using two ciphertexts from the same keystream, we can recover the XOR-encryption of the plaintexts.

Block ciphers

Block ciphers date back to late 1960s when IBM attempted to develop banking security systems [Ref http://www-03.ibm.com/ibm/history/ibm100/us/en/icons/cryptography/]. The result was Lucifer, an encryption method, with 128-bit key and block size, aimed at protecting data for cash-dispensing system in the UK. However, it was not secure in any of its version implementations. When we use block ciphers, each block is encrypted independently producing a ciphertext block of equal length. Block ciphers are also called product and have a cipher structure named ”Feistel”, as first described by Horst Feistel of IBM

in 1973. The Feistel structure consists of a number of identical rounds of processing. In each round,a substitution is performed on one half of the processed data, followed by a permutation that interchanges the two halves. The original key is expanded so that a different key is used for each round. Symmetric block encryption algorithms are based on this structure. In general, it is accepted that block ciphers are applicable to a broader range of applications than stream ciphers.

SHOULD I DESCRIBE FEISTEL CIPHER STRUCTURE ANY FURTHER ???

DES

Cryptography gradually moved from hardware to software with the advent of computers. Most famous example of the block cipher design and the classic Feistel structure is the Data Encryption Standard (DES), designed by IBM under the advisement of NASA in 1977. It was meant to encipher sensitive but non classified data. DES processes the plaintext in 64-bit blocks (64-bit input block and 64-bit output block), in 16 rounds (during each round a different set of keys is extracted), with a key of effective 56 bits (out of 64 since the parity bits are dropped) which means there are 2 to the power of 56 different keys. Bishop states that if the order in which the round keys is used is reversed, then the input is deciphered. He adds that "the rounds are executed sequentially, the input of one round being the output of the previous round"[Ref: Introduction to Computer Security, p.108, 2005]. DES is bit oriented which implies that it uses both transposition and substitution. The larger the block size, the key size and number of rounds means greater security. However, when there are more than 16 rounds or more than 56 keys, the security will neither be increased nor the encryption will be made any stronger. The possible attacks that often occur on product ciphers are differential cryptanalysis and linear cryptanalysis, however, Stallings maintains that DES "has been shown to be highly resistant to these two types of attacks". By 1999, a computer could try every possible key in a couple of days rendering the cipher insecure.

TRIPLE DES/3DES has replaced the older version of DES and is a more secure method of encryption, as it encrypts data three times (which makes it much slower than other methods of encryption), i.e. the 56-bit key becomes a 168-bit key and uses a different key for at least one of the versions. Furthermore, it is a symmetric-key encryption using 3 individual 56-bit keys.. In addition, 3DES uses shorter block lengths implying that it is easier to decrypt and lead to leakage of data.

AES

The Advanced Encryption Standard (AES) replaced DES as the US Government encryption technique to protect classified information in 2000. It has 128 bit block size and uses much bigger keys 128/192/256-bits in size making brute force attacks (cryptanalytic attacks attempting all possible key variants to decrypt any enciphered data) much harder. AES chops data up into 16-byte blocks and then applies a series of substitutions and permutations, based on the key value, plus some other operations to obscure the message, and this process is repeated ten or more times for each block. Today, AES is used everywhere, from encrypting files and sensitive data, transmitting data over WiFi with WPA2, to accessing websites using HTTPS.

3.2. Technical

4. BSPro - A First Bachelor Semester Project in BiCS-land

4.1. Requirements ($\pm$ 15% total words)

Describe here all the properties that characterize the deliverables you produce. It should describe what are those deliverables, who are the actors exploiting the deliverables, what are the expected functional and non functional qualities of the deliverables.

4.2. Design ($\pm$ 20% total words)

Provide the necessary and most useful explanations on how those deliverables have been produced.

4.3. Production ($\pm$ 20% total words)

Provide descriptions of the deliverables concrete production. It must present part of the deliverable to illustrate and explain its actual production.

4.4. Assessment ($\pm$ 15% total words)

Provide any objective elements to assess that your deliverables reached or not the requirements described above.

Acknowledgment

The authors would like to thank the BiCS management and education team for the amazing work done.

5. Conclusion

The conclusion goes here.

References

[1] BiCS Bachelor Semester Project Report Template. https://github.com/nicolasguelfi/lu.uni.course.bics.global. University of Luxembourg, BiCS - Academic Bachelor in Computer Science (2017).
thebibliography1

[2] Wikipedia. Cryptography. https://en.wikipedia.org/wiki/Cryptography

6. Appendix

All images and additional material go there.