ECOLE
POLYTECHNIQUE
DE BRUXELLES

Introduction to Cryptography

École Polytechnique de Bruxelles

Professeur : *Gilles* VAN ASSCHE

Sami ABDUL SATER

Année académique : 2021-2022

Table des matières

1 Historical ciphers and general principles **5**
1.1 Cryptography 5
1.1.1 Confidentiality 5
1.1.2 Authentication 6
1.2 Cryptanalysis 8
1.2.1 Mathematical cryptanalysis 8
1.2.2 Key length 8
1.2.3 Time to break 8
1.3 Some ciphers 9
1.3.1 Mono-alphabetic substitution 9
1.3.2 Poly-alphabetic substitution 9
1.3.3 Vigenère cipher 9

Chapitre 1

Historical ciphers and general principles

Cryptology is a term merging two similar fields of stuy : cryptograph and cryptanalysis.

- **Cryptography** : the study of secret writing with the goal of **hiding a message**.
- **Cryptanalysis** : breaking cryptosystems.

1.1 Cryptography

In cryptography, to hide a message, there are two things that interest us : hide our content (**Confidentiality**) and authenticating a message (**Authentication**).

1.1.1 Confidentiality

For a generic cryptosystem that ensures confidentiality of a message, we talk about two operations :

- **Encryption** of a plain-text message to get a ciphertext
- **Decryption** of a ciphertext to retrieve a message

We always have to picture two persons communicating with each other, and eventually a third-party intervenant trying to have access to the conversation. Hence, encryption and decryption are meaningless without talking about the intervenants, that we choose to name Ali and Bachar [1].

Ali and Bachar's communicating schema is the following : the first encrypts a message, sends it to the seconds that knows how to decrypt it to find the original content of the message. As they must be the only ones able to encrypt and decrypt the same way, they must have some kind of **key**.

This key generation/sharing/storage is the source of the division of cryptograph in two kinds : a symmetric way or an asymmetric way.

- In **Symmetric crypto**, also called **secret-key** crypto, both parts have an encryption and a decryption method, and they **share the same key that is secret**, kepts out of the sight of any outsider. We also assume that the encryption and decryption algorithms are **publicly known**.

1. Instead of Alice and Bob, let's change continent a bit.

- In **Asymmetric crypto** (since 1976), the two possess both a private and a public key. They share their public key, but never their private key !

So in general, we will talk about encryption as a mechanism that takes a message m, encrypts it with a key k_E to get a ciphertext c, and sends it. As for decryption, it takes a ciphertext c, decyrpts it under a key k_D to obtain m.

- Symmetric : $k_E = k_D$
- Asymmetric : k_E is public, k_D is private.

1.1.2 Authentication

As for authentication, we **are not trying to hide anything**. The message is sent in full plain-text from Ali to Bachar. Our goal here is to **check** the source of our message, assure its authenticity.

Similarly to encryption/decryption, we here have two mechanisms with keys :

- Authentication : Ali generates a tag under a key k_A, and sends the couple (m, tag) to Bachar

$$m \Rightarrow (m, \text{tag})$$

- Verification : Bob receives (m, tag), and under key k_V, identifies the source.

$$(m, \text{tag}) \Rightarrow \{m, \perp\}$$

In symmetric crypto, $k_A = k_V$ and is **secret**. In this case, the tag is more commonly called **Message Authentication Code** (MAC).

In asymmetric crypto, k_A is private and k_V is public. In this case, the tag is called "**signature**". To do so, with the message, Ali sends a tag.

Confidentiality

Encryption

- plaintext $\Rightarrow$ ciphertext
- Under key $k_E \in K$

Decryption

- ciphertext $\Rightarrow$ plaintext
- Under key $k_D \in K$

Symmetric cryptography: $k_E = k_D$ is the secret key.

Asymmetric cryptography: k_E is public and k_D is private.

Authenticity

Authentication

- message $\Rightarrow$ (message, **tag**)
- Under key $k_A \in K$

Verification

- (message, **tag**) $\Rightarrow$ {message, $\perp$}
- Under key $k_V \in K$

Symmetric cryptography: $k_A = k_V$ is the secret key.
The tag is called a *message authentication code* (MAC).

Asymmetric cryptography: k_A is private and k_V is public.
The tag is called a *signature*.

1.2 Cryptanalysis

Cryptanalysis is the field that studies algorithms and ways of breaking a cryptosystem. This means, recovering the message, or recovering the key. There are several ways to do this, going from ”little average mathematician boi that exploits the inner structure of the scheme” to the ”chad asking you your password with a gun pointing to the head”. All the methods, from the first to the last, are part of **cryptanalysis**. But in this course, we focus on what we call **mathematical cryptanalysis**.

1.2.1 Mathematical cryptanalysis

Some definitions

- Key space : set of all possible keys
- Brute-force attack : attack that tries all the keys of the key space.

This branch studies brute-force attacks and analytical attacks. Analytical attacks can be of several types : exploiting some statistical patterns, length extension attack, ...

1.2.2 Key length

This is an informative section on key length, just to develop an intuition on the impact of the length of a key in a cryptographic system.

First, it is important to mention that the key length in a **symmetric** crypto system is relevant only if the brute-force attack is the best-known attack.

Excluding this case, then the guaranteed security of a cryptosystem according in function with the key length is very different depending on the kind of crypto : a 80-bit key in symmetric crypto can ensure the same security as a 1024-bits key asymmetric scheme (such as RSA).

1.2.3 Time to break

Here is an indicator of the meaning of the ”time-to-break” (TTB) of cryptosystems in function of the key length for a **symmetric scheme**.

Key length	Security estimation
56–64 bits	short term: a few hours or days
112–128 bits	long term: several decades in the absence of quantum computers
256 bits	long term: several decades, even with quantum computers that run the currently known quantum computing algorithms

1.3 Some ciphers

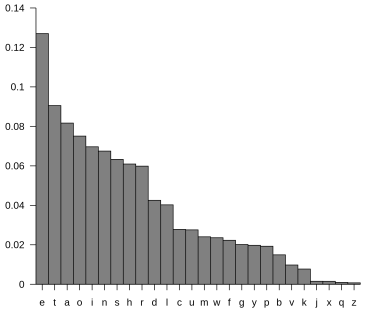
1.3.1 Mono-alphabetic substitution

The mono-alphabetic cipher consists in replacing each letter of the message by a corresponding letter in a mixed alphabet chosen randomly. So we define a substitution table, and we apply our mapping. Let's break it down a little bit.

- It is a symmetric scheme.
- The key space is $s = 26! > 4 \cdot 10^{26}$: a brute-force attack would take some time.
- It is breakable using a statistical approach.

Historical ciphers

Mono-alphabetic substitution



Frequency of invidual letters in English

9 / 57

Historical ciphers

Mono-alphabetic substitution

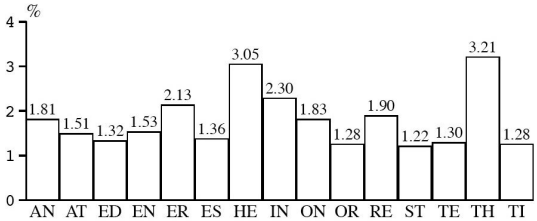


Figure 7.6: *Frequency of 15 common digrams in English text.*

10 / 57

1.3.2 Poly-alphabetic substitution

Instead of encrypting a entire message with the same mapping, we here divide the message into t blocks.

$$x = x_1 \| x_2 \| \dots \| x_t$$

Then, we define a mapping for each block. This will be encoded in the key k of the scheme. Indeed, each $k \in K$ will define a **set of permutations**

$$k \Rightarrow (p_1, p_2, \dots, p_t) \, .$$

Hence, $E_k(x)$ will be given by

$$E_k(x) = p_1(x_1) \| p_2(x_2) \| \dots \| p_t(x_t) \|$$

As for the decryption key k', it needs to define the set of the t corresponding inverse permutations :

$$k' \Rightarrow (p_1^{-1}, p_2^{-1}, \dots, p_t^{-1}) \, .$$

1.3.3 Vigenère cipher

- Message m of length $|m|$, chosen in $(\mathbb{Z}_{26})^\star$
- Key space $K \subset (\mathbb{Z}_{26})^t$
- Key k taken randomly in K, so

$$k = (k_0, k_1, \dots, k_{t-1}) \in K$$

Then, the encryption of m will result in the concatenation of the XORing of each bit m_i with a part of the key that. Remember that the key has only t parts, so we will repeat the same parts if the message is very long!

$$E_k(m) \equiv E_k(m_0 \| m_1 \| \dots \| m_{|m|-1}) = \mathop{\|}_{0 \leq i \leq |m|-1} (m_i + k_{i \mod t})$$