

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

COURSE NOTES IN

Cryptography and Security



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

OLIVIER CLOUX

AUTUMN 2018



Contents

1 Ancient Cryptography

1.1 Introduction

Prehistory: used for confidentiality only. Enforce by different means. Nowadays, we wish for mass communication, and dedicated academic research to it. We now wish, in addition, integrity, authentication, privacy, non-repudiation, fairness, access control, timestamping and more. In ancient times, we used a lot security by obscurity. Sufficient for the time, dangerous now. First, we used *substitution* and *transpositions*. Using a simple key, we also used encryption with a configurable secret key.

But modern times changed everything. In communication with mass communication, and in computing with computer science and automata: the adversaries have more power to crack our code.

1.2 Terminology

Prof. Vaudenay is not a purist, so no problem with slightly wrong terms.

- Code: system of symbols which represent information
- Coding Theory: Science of code transformation which enables to send information through a communication channel in a reliable and efficient way. Here, the adversary is dummy, just shooting at random bits. This is basically noise.
- Cryptography: science of secret codes, enabling confidentiality of communication through insecure channel. Here, the adversary is malicious, and tries actively to break our secret.
- Cipher: Secret code,
- To cryptanalyse a cryptosystem: prove or disprove security. \neq break it.
- To break a cryptosystem: prove insecurity (disprove security)
- Cryptology: science of Cryptography and cryptanalyse (may also include steganography)
- Steganography: Science of information hiding \neq cryptography

In modern cryptography, we have several problems we have to ensure:

- confidentiality: Secure the receiving part of the communication. Only him can decrypt and read.
- Authentication: Only legitimate sender can send message.
- integrity: Received message is the same as the sent one.

1.3 Cryptography prehistory

1.3.1 Secret writings, Transpositions and Caesar's cipher

First cryptographic device was the Rosetta stone. Then, spartans had a leather belt that could be wrapped. Caesar used shifting alphabet letters by 3. In more modern times, we used ROT13 (rotate but by 13). Useful because same method encrypts and decrypts.

A more secure method: assigning to each letter another random alphabet letter. Finding one letter does not break the whole encryption. But by statistical analysis and comparison of frequencies of english alphabet and encrypted alphabet, we can easily break it.

Vigenère cipher: Use a string as a key, repeat it to length of original message, then add positions of message and key letters, modulo 26 and you have a new letter. Finding the corresponding plaintext to an encrypted letter, you don't know that other same encrypted letters are the same. This is quite strong. A good method to decypher it is using the **Kasiski test**. Look for frequent patterns, and check their positions. If their positions are often equal modulo a certain integer, this is probably the length of the key. But for a certain text, is it probable to observe a certain frequency for a certain pattern ?

Example 1. Random string of 313 characters, from alphabet of 26, we observe 5 occurrences of a trigram.

Number of k-tuples of elements in a set of size z, we can make

$$z^k$$

possible trigrams.

Number of possible subsets of t elements in a set of size n:

$$\binom{n}{t}$$

In a random sequence of 313 characters, we have 311 trigrams.

After a lot of computations, we find that the probability is less than 1 over 1 million

1.3.2 Index of Coincidence

We define the index of coincidence as

$$\begin{aligned} \text{Index}(x_1, \dots, x_n) &= Pr_{I,J}[x_I = x_J | I < J] \\ &= \frac{1}{n(n-1)} \sum_{1 \leq i \neq j \leq n} 1_{x_i = x_j} \\ &= \sum_{c \in Z} \frac{n_c(n_c-1)}{n(n-1)} \end{aligned}$$

Where $I, J \in \{1, \dots, n\}$ are iid.

1.3.3 Enigma

Used during WW2. It was patented in 1918, meaning that there was not secret behind fabrication. Specifications were secure enough. Even if a machine is stolen, you can't decrypt the messages. There was a plugboard to modify connections daily. The first rotor turned each letter, then once it did 26 turns, the next rotor turned once. And again and again.

1.3.4 Laws of modern Crypto

Law 1: the Kerchoffs Principle: Security should not rely on the secrecy of the cryptosystem itself. The adversary may get some info about the system (ex employee, corruption,...). Thus, security analysis must assume that the adversary knows the cryptosystem. But it does not imply that the cryptosystem must be public. The principles are as follow (sic):

- Le système doit être matériellement, sinon mathématiquement, indéchiffrable;
- **Il faut qu'il n'exige pas le secret, et qu'il puisse sans inconvénient tomber entre les mains de l'ennemi;**
- La clef doit pouvoir en être communiquée et retenue sans le secours de notes écrites, et être changée ou modifiée au gré des correspondants;
- Il faut qu'il soit applicable à la correspondance télégraphique;
- Il faut qu'il soit portatif et que son maniement ou son fonctionnement n'exige pas le concours de plusieurs personnes;
- Enfin, il est nécessaire, vu les circonstances qui en commandent l'application, que le système soit d'un usage facile, ne demandant ni tension d'esprit, ni la connaissance d'une longue série de règles à observer.

Law 2: the n^2 problem: In a network of n users, there is a number of potential pairs of users within the order of magnitude of n^2 . We cannot assume that every pair of users share a secret key. We must find a way for any pair of users to establish a shared secret key.

Law 3: The Moore law: the speed of CPUs doubles every 18-24 months. This allows us to estimate how long a system can remain secure, assessing security against brute force attacks. Considering current trends, the number of keys per second that can be tested at time t :

$$f_t \simeq 10^6 \times 10^{5 \frac{1}{32 \text{ years}(t-2007)}}$$

Thus, an attacker always keeping the best CPU can try between time t_0 and $t_0 + \Delta$:

$$\int_{t_0}^{t_0+\Delta} f_t dt = \int_{t_0}^{t_0+\Delta} 10^6 \times 10^{\frac{5}{32}(t-2007)} dt = \frac{10^6}{\frac{5}{32} \ln 10} \times 10^{\frac{5}{32}(t-2007)} \left(10^{\frac{5}{32}\Delta} - 1 \right)$$

\oplus	0	1
0	0	1
1	1	0

Table 1: XOR results table

Thus, to offer security between current time t_0 until time $t_0 + \Delta$ the key length must be at least

$$\log_2(\#\text{processors} \times \text{above}) \simeq \text{margin} + \frac{5}{32} \log_2 10 \times (t_0 + \Delta - 2007)$$

With Moor's law and considering a 128 bit key, we see that in 2007 we would need 770k billions CPUs that run during 14 billion years to break it. But in 2215, we *should* have computers that can solve it within 1 sec

Law 4: The Murphy Law: If there is a single security hole, the system will fall into it. Thus we should never leave a security hole, and prove security rather than bet on it.

1.4 Cryptography and Information Theory

1.4.1 XOR reminder

As seen a thousand times, we do a small reminder on bitwise exclusive OR. A XOR is basically a binary addition where carry bits are ignored. A summary of operations can be found at table ???. In addition, this operation has some very interesting properties:

- closure: XOR of two bitstrings is still a bitstring.
- associativity: $a \oplus b \oplus c = a \oplus (b \oplus c)$
- commutativity: $a \oplus b = b \oplus a$
- neutral element: $a \oplus [000 \dots 0] = a$
- (self-)invertibility: $a \oplus a = [000 \dots 0]$

1.4.2 Vernam Cipher

We can encrypt a plaintext message X using a uniformly distributed random key K (bitstring) of the same size simply with

$$X \oplus K = Y$$

. With the help of the properties of the XOR operator, we immediately see why the decryption is so simple:

$$Y \oplus K = X$$

.

There are a cases when this is insecure:

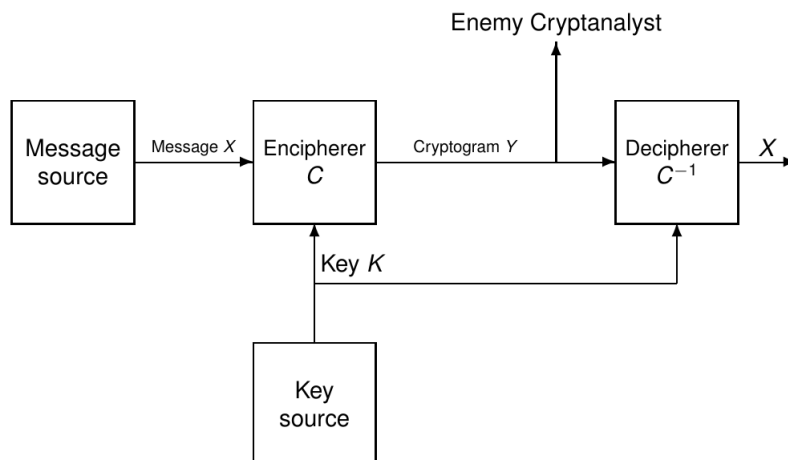


Figure 1: The Shannon Encryption Model

- Using the same key twice yields a lot of information about the original message. Applying a XOR between the two encrypted texts, we obtain a mix of both plaintext. It then becomes easy to retrieve both original messages. This is best seen using visual encryption.
- Using a key K smaller than the message (no repetition)
- Using a key not uniformly distributed (lot of same bit, or repetition, patterns).

Unfortunately, even this is perfectly secure, this makes no sense for most applications. If you have a way to transmit your key securely, you probably can transmit your message securely as well. Example of use case: transmitting keys can be slow, but you need message to be transmitted fast and you don't know the message at the moment you transmit the key. However, this algorithm is essentially

1.4.3 Generalized Vernam Cipher

G an Abelian group and let an arbitrary plaintext source producing elements in G . Key K is also uniformly distributed in G and is independent from plaintext. Then given X , we have $Y = K + X$ and $X = (-K) + Y$.

1.4.4 The Shannon Encryption Model

Reminders of the Shannon entropy are necessary, and can be found at appendix ??.

In this model, the message is a random variable with a given *a priori* distribution. The key is a random variable with specified distribution, independent from the message. We need to ensure the correctness property, that is we are absolutely certain that decryption always work:

$$Pr[C_K^{-1}(C_K(X))] = 1$$

And finally, this model assumes that an adversary can get the random variable $Y = X_K(X)$ only. This model doesn't assume any other threat.

1.4.5 Perfect secrecy

Our ultimate goal is to provide **perfect secrecy**.

Definition 1 (0.8). Perfect secrecy means that the a posteriori distribution of the plaintext X after we know the ciphertext Y is equal to the a priori distribution of the plaintext:

$$\forall x, y \ Pr[Y = y] \neq 0 \rightarrow Pr[X = x|Y = y] = Pr[X = x] \quad (1)$$

Equivalent definition: Statistic independence of X and Y

Equivalent definition: $H(X|Y) = H(X)$

For any distribution of the plaintext, the generalized Vernam cipher provides perfect secrecy.

Influence of the Plaintext Distribution:

1.5 Summary and Important points

- Milestones of Prehistory: Security by obscurity, encryption with configurable secret key (Vigenère), application of the Kerckhoffs principles.
- Milestones of modern crypto: Kerckhoffs (1883), Shannon (1949, info-theoretical approach to crypto), Diffie-Hellman (1976, public-key crypto), DES (1977, encryption standard for non-military)
- Basic security properties: confidentiality, integrity, authentication

2 Diffie-Hellman

2.1 Agreement Protocol

Assume a group generated by some public g . We suppose a communication between Alice and Bob.

1. Alice picks a random x , and compute $X = g^x$. She then sends this X to Bob.
2. Bob picks a random y , and computes $Y = g^y$ and the shared key $K = X^y$.
3. Upon receiving Y , Alice computes $K = Y^x$.

Thus, we have the “shared secret” $K = g^{xy}$. This can be defined in any group. But depending on the group, this can be easy or hard. The security requires that given (g, g^x, g^y) (that are transmitted in plaintext), computing g^{xy} must be hard.

This algorithms allows to set up a secret key over a public channel. Note that we still require authentication, to avoid a man-in-the-middle attack that transmits his own $g^{x'}$ and $g^{y'}$ to both parties.

The best attack is an exhaustive search on x . Thus, to be secured, the generated group must have a huge cardinality.

2.2 Discrete Logarithm

This problem is mentioned as such:

- Parameters: G , a group, $g \in G$ and the n the order of g
- Instance: y , power of G
- Problem: find x such that $y = g^x$

Example 2. Over \mathbf{Z}_n this is easy (with the use of the Extended Euclidean Algorithm). Over \mathbf{Z}_p^* or an elliptic curve, it **may** be hard

In a group of order n , the attack is easy on a quantum computer, easy if n only has small prime factors. But the best algorithm for a subgroup of \mathbf{Z}_p^* with n and p prime, the complexity is

$$e^{\left(\sqrt[3]{\frac{64}{9}+o(1)}\right)(\ln n)^{\frac{1}{3}}(\ln \ln n)^{\frac{2}{3}}}$$

This is mostly precomputation, so without the need of y .

2.3 The key exchange

Definition 2 (Passive adversary). A **passive adversary** just listens and tries to decrypt communications (e.g. recovering the key)

The Diffie-Hellman shall resist to passive attacks. Given only g (public), X and Y (eaves-dropped), it must be hard to compute K .

Definition 3 (Computational Diffie-Hellman (CDH) Problem). •

Parameters: G a group, $g \in G$ and n , the order of G .

- Instance: $X, Y \in \langle g \rangle$
- Problem: find $K = g^{xy}$ where $X = g^x$ and $Y = g^y$

The hardness of this problem requires the Discrete Logarithm Problem to be hard, because CDH can be reduced to DL. This means that CDH will be at most as hard as DL.

2.3.1 Correct Diffie-Hellman Key Exchange

We assume a group $\langle g \rangle$ generated by some g of prime order q . At first, Alice picks $x \in \mathbf{Z}_q^*$, and sets X

2.4 The ElGamal Public-Key Cryptosystem

2.4.1 ElGamal Cryptosystem

An important point to look at, is the **complexity** of the encryption. In subgroups of \mathbf{Z}_p^* with p of length l :

- Domain parameter selection: $O(l^4)$

- Generator: $O(l^3)$
- Encryption: $O(l^3)$
- Decryption: $O(l^3)$

3 RSA

3.1 Euler and Other Chinese

Theorem 1 (Euler Totient Function). *Given an integer n , we have the following results:*

- $\forall x \in \mathbf{Z}_n : x \in \mathbf{Z}_n^* \iff \gcd(x, n) = 1$
- \mathbf{Z}_n is a field $\iff \mathbf{Z}_n^* = \mathbf{Z}_n \setminus \{0\} \iff \varphi(n) = n - 1 \iff n$ is prime
- $\forall x \in \mathbf{Z}_n^*$ we have $x^{\varphi(n)} \equiv 1 \pmod{n}$
- If e is such that $\gcd(e, \varphi(n)) = 1$, we let $d = e^{-1} \pmod{\varphi(n)}$. For all $x \in \mathbf{Z}_n^*$, $x^d \pmod{n}$ is the only e th root of x modulo n

3.1.1 Chinese Remainder Theorem

Theorem 2 (Chinese Remainder (I)). *Let m and n be two co-primes integers. For any $a, b \in \mathbf{Z}$, there exists $x \in \mathbf{Z}$ such that*

$$\begin{aligned} x &\equiv a \pmod{m} \\ x &\equiv b \pmod{n} \end{aligned}$$

And for all such solution, $x \pmod{mn}$ is unique.

Theorem 3 (Chinese Remainder (II)). *Let m and n be two co-primes integers. We have*

- $f : \mathbf{Z}_{mn} \rightarrow \mathbf{Z}_m \times \mathbf{Z}_n$ defined by $f(x) = (x \pmod{m}, x \pmod{n})$ is a ring homomorphism.
- $f^{-1}(a, b) \equiv an(n^{-1} \pmod{m}) + bm(m^{-1} \pmod{n}) \pmod{mn}$

3.2 Primality Testing

3.3 RSA Basics

3.4 Quadratic Residuosity

3.5 The Factoring Problem

A Reminders

A.1 Theory Information

A.1.1 Shannon Entropy

Definition 4 (Entropy). We define the entropy of some string X as $H(X)$. This represents the number of bits of information to represent the value of X . It is defined as

$$H(X) = - \sum_x Pr[X = x] \log_2 Pr[X = x] \quad (2)$$

Similarly, $H(X, Y)$ is the entropy of (X, Y) , and $H(X|Y) = H(X, Y) - H(Y)$, that are defined, respectively, as

$$H(X, Y) = - \sum_{x,y} Pr[X = x, Y = y] \log_2 Pr[X = x, Y = y] \quad (3)$$

$$H(X|Y) = - \sum_{x,y} Pr[X = x, Y = y] \log_2 Pr[X = x|Y = y] \quad (4)$$

We also have the following propositions (proofs omitted):

- $H(X) \geq 0$. Equality is reached if and only if X is constant.
- $H(X, Y) \geq H(X)$. Equality is reached if and only if Y can be written as a function of X .
- $H(X, Y) \leq H(X) + H(Y)$. Equality is reached if and only if X and Y are independent.
- If $Pr[X = x] \neq 0$ for n values of x , then $H(X) \leq \log_2 n$. Equality is reached if and only if all non-zero $Pr[X = x]$ are equal to $\frac{1}{n}$.

Definition 5 (Convex). A real function f is **convex** on $[a, b]$ if and only if

$$\forall \text{ set } S \quad \forall t : S \rightarrow [a, b] \quad \forall p : S \rightarrow]0, 1[\\ \sum_{x \in S} p_x = 1 \Rightarrow \sum_{x \in S} p_x f(t_x) \geq f\left(\sum_{x \in S} p_x t_x\right)$$

A.2 Arithmetic

We remind a few definitions and theorems that will prove useful in this chapter.

Definition 6 (Prime Number). Prime number is a positive integer which has exactly two positive factors: 1 and itself

Theorem 4 (Unique factorization). *Each integer n can be uniquely written*

$$n = u \times p_1^{\alpha_1} \times \cdots \times p_r^{\alpha_r}$$

where $p_1 < \cdots < p_r$ are prime, $u = \pm 1$ and $\alpha_1, \dots, \alpha_r$ are positive integers.

Theorem 5 (Euclidean Division). *For any $a \in \mathbf{Z}$ and any $n > 0$ there exists a unique pair $(q, r) \in \mathbf{Z}^2$ such that $a = qn + r$ and $0 \leq r < n$. We denote $r = a \bmod n$ and have $q = \lfloor \frac{a}{n} \rfloor$*

About “mod” notation: $a = b \bmod n$ means “a set to (b mod n)”. on the other hand, $a \equiv b \pmod{n}$ means “a and b, once reduced modulo n, are equal”.

A.3 Group theory

Definition 7 (Group). An group is a set G together with a mapping $G \times G \rightarrow G$ which maps $(a, b) \rightarrow a + b$ with properties:

- closure $a + b \in G$
- associativity $(a + b) + c = a + (b + c)$
- neutral element $\exists 0 : a + 0 = 0 + a = a$
- invertibility $\exists(-a) : a + (-a) = (-a) + a = a - a = 0$

Definition 8 (Abelian group). An Abelian group is a group, with the additional property of commutativity

- commutativity: $a + b = b + a$

Definition 9 (Group homomorphism). Given two groups (G_1, \times_1) and (G_2, \times_2) , a mapping f from G_1 to G_2 is a group homomorphism if for any $a, b \in G_1$, we have

$$f(a \times_1 b) = f(a) \times_2 f(b) \quad (5)$$

Example 3. If $g \in G$, the mapping $\varphi : \mathbf{Z} \rightarrow G$ defined by $\varphi(a) = g^a$ is a group homomorphism, because

$$\forall a, b \in \mathbf{Z}, \varphi(a + b) = \varphi(a)\varphi(b)$$

Theorem 6 (Subgroups). *If H is a subgroup of \mathbf{Z} not reduced to $\{0\}$, then $H = n\mathbf{Z}$ where n is the smallest positive element of H*

Definition 10 (Generators). Given a group (G, \cdot) , an element g **generates/span** a subgroup

$$\langle g \rangle = \{\dots, g^{-2}, g^{-1}, g^0, g^1, g^2, \dots\}$$

If $\langle g \rangle$ is finite, of cardinality n , then $g^n = 1$ and

$$\langle g \rangle = \{g^0, g^1, \dots, g^{n-1}\}$$

If $x \in \langle g \rangle$, $\log_g x$ is uniquely determined up to some multiple of n , then $\log_g x$ is an element of \mathbf{Z} and $i \mapsto g^i$ is a group isomorphism between \mathbf{Z}_n and $\langle g \rangle$.

Theorem 7 (Lagrange). *In any finite group, the order of any element is a factor of the order of the group*

Theorem 8 (Generators in a group of prime order). *If G has a primer order, all elements (except 1) are generators*

A.3.1 Rings

Definition 11 (Rings). A ring is an Abelian group $(R, +)$ together with a mapping from $R \times R \mapsto R$ which maps $(a, b) \mapsto ab$ such that, in addition to Abelian groups properties (see definitions ?? and ??):

- Associativity: $(ab)c = a(bc)$
- neutral element: There exists 1 such that $a1 = 1a = a$
- Distributivity: $\forall a, b, c$, we have $a(b + c) = ab + ac$.

Definition 12 (Commutative Ring). A **commutative ring** is a ring R such that, in addition to Ring properties (see definition ??),

- Commutativity: $\forall a, b : ab = ba$

Definition 13 (Units). In a ring, not every element has an inverse for the multiplication. Thus, we denote R^* the set of elements having a multiplicative inverse. Those elements are called **units**. R^* with the multiplication is a group ; this is the group of units of the ring R


Definition 14 (Structures). • Sub-structures:

- Subgroup: Subset of a group stable by group law and inversion
- ideal: Subgroup of a ring stable by multiplication by any ring element.
- Spanned structure: set of all values generated by structure operations

A.3.2 Orders in a group

Given $x \in G$ then we have that:

- $\{i \in \mathbf{Z} : x^i = 1\}$ is a subgroup of \mathbf{Z}
- Thus, $\{i \in \mathbf{Z} : x^i = 1\} = n\mathbf{Z}$ for some n which is the smallest positive n such that $x^n = 1$, which is the definition of the order.



Example 4. The modulo 9 of big numbers of big numbers is quite easy. As every number can be written as a sum of powers of 10 (e.g. $2 \cdot 10^5 + 4 \cdot 10^3 + \dots$), every power of 10 can be reduced, modulo 9, to powers of 1. We quickly see that the reduction modulo 9 of any number can be done by adding the digits composing the number.

A.4 Modular theory