SAPIENZA
UNIVERSITÀ DI ROMA

Sapienza University of Rome

Cryptography Notes

Alessandro Dori

Instructor:

Prof. Daniele Venturi

Academic Year 2025/2026

Cryptography

Alessandro Dori

September 27, 2025

Contents

Chapter 1

Introduction

Classical cryptography was concerned with designing and using codes (also called ciphers) that enable two parties to communicate secretly in the presence of an eavesdropper who can monitor all communication between them. In modern parlance, codes are called encryption schemes and that is the terminology we will use here. Security of all classical encryption schemes relied on a secret—a key—shared by the communicating parties in advance and unknown to the eavesdropper. This scenario is known as the *private-key* setting. In the setting of private-key encryption, two parties share a key and use this key when they want to communicate secretly. One party can send a message, or plaintext, to the other by using the shared key to encrypt (or "scramble") the message and thus obtain a ciphertext that is transmitted to the receiver. The receiver uses the same key to decrypt (or "unscramble") the ciphertext and recover the original message. Note the same key is used to convert the plaintext into a ciphertext and back; that is why this is also known as the symmetric-key setting, where the symmetry lies in the fact that both parties hold the same key that is used for encryption and decryption. This is in contrast to asymmetric, or public-key, encryption, where encryption and decryption use different keys.

The goal of encryption is to keep the plaintext hidden from an eavesdropper who can monitor the communication channel and observe the ciphertext.

The syntax of encryption. Formally, a private-key encryption scheme is defined by specifying a message space $\mathcal{M}$ with three algorithms: a procedure for generating keys (Gen), a procedure for encrypting (Enc), and a procedure for decrypting (Dec). The message space $\mathcal{M}$ defines the set of "legal" messages. The algorithms have the following functionality:

1. The *key-generation* algorithm **Gen** is a probabilistic algorithm that outputs a key k chosen according to some distribution.
2. The *encryption algorithm* **Enc** takes as input a key k and a message m and outputs a ciphertext c. We denote by $Enc_k(m)$ the encryption of the plaintext m using the key k.
3. The *decryption algorithm* **Dec** takes as input a key k and a ciphertext c and outputs a plaintext m. We denote the decryption of the ciphertext c using the key k by $Dec_k(c)$.

An encryption scheme must satisfy the following correctness requirement: for every key k output by **Gen** and every message $m \in M$, it holds that

$$Dec_k(Enc_k(m)) = m.$$

The set of all possible keys output by the key-generation algorithm is called the key space and is denoted by $\mathcal{K}$. Almost always, Gen simply chooses a uniform key from the key space.

An encryption scheme can be used by two parties who wish to communicate as follows. First, Gen is run to obtain a key k that the parties share. Later, when one party wants to send a plaintext m to the

other, she computes $c := Enc_k(m)$ and sends the resulting ciphertext c over the public channel to the other party.[1] Upon receiving c, the other party computes $m := Dec_k(c)$ to recover the original plaintext.

Symmetric Encryption (SKE): $\Pi = (Enc, Dec)$ such that:

- Enc: $\mathcal{M} \times \mathcal{K} \to \mathcal{C}$
- Dec: $\mathcal{C} \times \mathcal{K} \to \mathcal{M}$
- k is uniform over $\mathcal{K}$

Correctness:

$\forall k \in \mathcal{K}, \forall m \in \mathcal{M} \qquad Dec(k, Enc(k, m)) = m$

Keys and Kerckhoffs' principle.

Security only should depend on security of the key, not of algorithms.

As is clear from the above, if an eavesdropping adversary knows the algorithm Dec as well as the key k shared by the two communicating parties, then that adversary will be able to decrypt any ciphertexts transmitted by those parties.

An encryption scheme should be designed to be secure even if an eavesdropper knows all the details of the scheme, so long as the attacker doesn't know the key being used. Stated differently, security should not rely on the encryption scheme being secret; instead, Kerckhoffs' principle demands that security rely solely on secrecy of the key.

[1] We use ":=" to denote deterministic assignment, and assume for now that Enc is deterministic.

Chapter 2

Perfectly Secret Encryption

In this chapter, we look at the other extreme and study encryption schemes that are *provably* secure even against an adversary with unbounded computational power. Such schemes are called *perfectly secret*. (Beginning in this chapter, we assume familiarity with basic probability theory. The relevant notions are reviewed in Appendix.)

2.1 Definitions

We begin by recalling and expanding upon the syntax that was introduced in the previous chapter. An encryption scheme is defined by three algorithms **Gen**, **Enc**, and **Dec** as well as a specification of a (finite) *message space* $\mathcal{M}$ with $|\mathcal{M}| > 1$. The key-generation algorithm **Gen** is a probabilistic algorithm that outputs a key k chosen according to some distribution. We denote by $\mathcal{K}$ the (finite) *key space*, i.e., the set of all possible keys that can be output by **Gen**. The encryption algorithm **Enc** takes as input a key $k \in \mathcal{K}$ and a message $m \in \mathcal{M}$ and outputs a ciphertext c. We now allow the encryption algorithm to be probabilistic (so $Enc_k(m)$ might output a different ciphertext when run multiple times), and we write $c \leftarrow Enc_k(m)$ to denote the possibly probabilistic process by which message m is encrypted using key k to give ciphertext c. (In case **Enc** is deterministic, we may emphasize this by writing $c := Enc_k(m)$. Looking ahead, we also sometimes use the notation $x \leftarrow S$ to denote uniform selection of x from a set S.) We let $\mathcal{C}$ denote the set of all possible ciphertexts that can be output by $Enc_k(m)$, for all possible choices of $k \in \mathcal{K}$ and $m \in \mathcal{M}$. The decryption algorithm **Dec** takes as input a key $k \in \mathcal{K}$ and a ciphertext $c \in \mathcal{C}$ and outputs a message $m \in \mathcal{M}$. We assume *perfect correctness*, meaning that $\forall k \in \mathcal{K}, \forall m \in \mathcal{M}$, if $c \leftarrow Enc_k(m)$, then $Dec_k(c) = m$ with probability 1. We will thus write $m := Dec_k(c)$, since **Dec** is deterministic.

In the definitions and theorems below, we refer to probability distributions over $\mathcal{M}$, $\mathcal{K}$, and $\mathcal{C}$. The distribution over $\mathcal{K}$ is the one defined by running **Gen** and taking the output. We let K be a random variable denoting the value of the key output by **Gen**; thus, for any $k \in \mathcal{K}$, $Pr[K = k]$ denotes the probability that the key output by **Gen** is equal to k. Similarly, we let M be a random variable denoting the message being encrypted, so $Pr[M = m]$ denotes the probability that the message takes on the value $m \in \mathcal{M}$.

K and M are assumed to be independent, i.e., what is being communicated by the parties is independent of the key they happen to share. This make sense, among other reasons, because the distribution over $\mathcal{K}$ is determined by the encryption scheme itself (since it is defined by Gen), while the distribution over $\mathcal{M}$ depend on the context in which the encryption scheme is being used.

Fixing an encryption scheme and a distribution over $\mathcal{M}$ determines a distribution over the space of ciphertexts $\mathcal{C}$ given by choosing a key $k \in \mathcal{K}$ (according to Gen) and a message $m \in \mathcal{M}$ (according to the given distribution), and then computing the ciphertext $c \leftarrow Enc_k(m)$. We let C be the random variable denoting the resulting ciphertext and so, for $c \in \mathcal{C}$, write $Pr[C = c]$ to denote the probability that the ciphertext is equal to the fixed value c.

Example

We work throgh a simple example for the shift cipher. Her, by definition, we have $\mathcal{K} = \{0, 1, \ldots, 25\}$ with $Pr[K = k] = 1/26$ for each $k \in \mathcal{K}$.

Say we are given the following distribution over $\mathcal{M}$:

$$Pr[M = a] = 0.7 \text{ and } Pr[M = z] = 0.3.$$

What is the probability that the ciphertext is B? There are only two ways this can occur: either $M = a$ and $K = 1$, or $M = z$ and $K = 2$. By independence of M and K, we have

$$Pr[M = a \wedge K = 1] = Pr[M = a] \cdot Pr[K = 1] = 0.7 \cdot \frac{1}{26}$$

Similarly, $Pr[M = z \wedge K = 2] = Pr[M = z] \cdot Pr[K = 2] = 0.3 \cdot \frac{1}{26}$. Therefore,

$$Pr[C = B] = Pr[M = a \wedge K = 1] + Pr[M = z \wedge K = 2] = 0.7 \cdot \frac{1}{26} + 0.3 \cdot \frac{1}{26} = \frac{1}{26}.$$

We can calculate conditional probabilities as well. For example, what is the porbability that the message a was encrypted, given that we observe ciphertext B? Using Bayes' theorem, we have

$$Pr[M = a \mid C = B] = \frac{Pr[C = B \mid M = a] \cdot Pr[M = a]}{Pr[C = B]} = \frac{0.7 \cdot Pr[C = B \mid M = a]}{1/26}.$$

Note that $Pr[C = B \mid M = a] = 1/26$, since if $M = a$ then the only way $C = B$ can occur is if $K = 1$ (which occurs with probability $1/26$). We conclude that $Pr[M = a \mid C = B] = 0.7$.

Perfect Secrecy:

An encryption scheme with message space $\mathcal{M}$ is *perfectly secret* if for every probability distribution over $\mathcal{M}$, every message $m \in \mathcal{M}$, and every ciphertext $c \in \mathcal{C}$ for which $Pr[C = c] > 0$:

$$Pr[M = m \mid C = c] = Pr[M = m].$$

(The requirement that $Pr[C = c] > 0$ is a technical one needed to prevent conditioning on a zero-probability event.)

We now give an equivalent formulation of perfect secrecy. Informally, this formulation requires that the probability distribution of the ciphertext does not depend on the plaintext.

Formally, for every $m, m' \in \mathcal{M}$ and every $c \in \mathcal{C}$,

$$Pr[Enc_K(m) = c] = Pr[Enc_K(m') = c]$$

(where the probabilities are over choice of K and any randomness of **Enc**). This implies that the ciphertext contains non information about the plaintext, and that is impossible to distinguish an encryption of m from an encryption of m', since the distributions over the ciphertext are te same in each case.

Lemma

The following definitions of perfect secrecy are equivalent:

1. **Perfect secrecy**: $Pr[M = m] = Pr[M = m \mid C = c]$
2. M and C are independent (i.e. $I(M; C) = 0$)
3. $\forall m, m' \in \mathcal{M}, \forall c \in \mathcal{C}$: $Pr[Enc_K(m) = c] = Pr[Enc_K(m') = c]$

Proof. We need to show that (1) $\implies$ (2) $\implies$ (3) $\implies$ (1).

- (1) $\implies$ (2): By the definition of perfect secrecy, we have:

$$Pr[M = m] = Pr[M = m \mid C = c].$$

Using the definition of conditional probability:

$$Pr[M = m \mid C = c] = \frac{Pr[M = m \wedge C = c]}{Pr[C = c]}.$$

Substituting this back, we get:

$$Pr[M = m] = \frac{Pr[M = m \wedge C = c]}{Pr[C = c]}.$$

Rearranging terms, we obtain:

$$Pr[M = m] \cdot Pr[C = c] = Pr[M = m \wedge C = c].$$

This implies that M and C are independent, and therefore $I(M;C) = 0$.

- (2) $\implies$ (3): Fix m from M, c from C:

$$Pr[Enc_K(m) = c] = Pr[Enc_K(m) = c \mid M = m]$$

But now "$Enc_K(m) = c$" is C, so:

$$Pr[C = c \mid M = m] = Pr[C = c]$$

By same argument:

$$Pr[Enc_K(m') = c] = Pr[Enc_K(m') = c \mid M = m']$$
$$Pr[C = c \mid M = m'] = Pr[C = c]$$

- (3) $\implies$ (1): Consider any $c \in \mathcal{C}$:

$$Pr[C = c] = Pr[C = c \mid M = m].$$

By assumption (3), this holds for all $m \in \mathcal{M}$. Using the definition of conditional probability:

$$Pr[M = m \mid C = c] = \frac{Pr[M = m \wedge C = c]}{Pr[C = c]}.$$

Substituting $Pr[M = m \wedge C = c] = Pr[C = c \mid M = m] \cdot Pr[M = m]$, we get:

$$Pr[M = m \mid C = c] = \frac{Pr[C = c \mid M = m] \cdot Pr[M = m]}{Pr[C = c]}.$$

By assumption (3), $Pr[C = c \mid M = m] = Pr[C = c]$, so:

$$Pr[M = m \mid C = c] = \frac{Pr[C = c] \cdot Pr[M = m]}{Pr[C = c]} = Pr[M = m].$$

Thus, we have shown that:

$$Pr[M = m] = Pr[M = m \mid C = c].$$

Claim: $Pr[C = c] = Pr[C = c \mid M = m]$.

Proof of claim: Using the law of total probability:

$$Pr[C = c] = \sum_{m' \in \mathcal{M}} Pr[C = c \wedge M = m'] = \sum_{m' \in \mathcal{M}} Pr[C = c \mid M = m'] \cdot Pr[M = m'].$$

By assumption (3), $Pr[C = c \mid M = m'] = Pr[C = c \mid M = m]$ for all $m' \in \mathcal{M}$. Therefore:

$$Pr[C = c] = Pr[C = c \mid M = m] \cdot \sum_{m' \in \mathcal{M}} Pr[M = m'].$$

Since $\sum_{m' \in \mathcal{M}} Pr[M = m'] = 1$, we conclude:

$$Pr[C = c] = Pr[C = c \mid M = m].$$

□

2.2 The One-Time Pad

In 1917, Vernam patented a perfectly secret encryption scheme noe called the *one-time pad*. Approximately 25 years later, Shannon introduced the definition of perfect secrecy and demonstrated that the on-time pad achieves that level of security.

The One-Time Pad:

Fix an integer $n > 0$. The message space $\mathcal{M}$, key space $\mathcal{K}$, and ciphertext space $\mathcal{C}$ are all equal to $\{0,1\}^n$.

- **Gen**: the key-generation algorithm chooses a key from $\mathcal{K} = \{0,1\}^n$ according to the uniform distribution (i.e., each of the 2^n strings in the space is chosen as the key with probability exactly 2^{-n}).
- **Enc**: given a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, the encryption algorithm computes the ciphertext $c := k \oplus m$, where $\oplus$ denotes bitwise XOR.
- **Dec**: given a key $k \in \{0,1\}^n$ and a ciphertext $c \in \{0,1\}^n$, the decryption algorithm computes the message $m := k \oplus c$.

In the one-time pad encryption scheme the key is a uniform string of the same length as the message. Before discussing security, we first verify correctness: for every key k and every message m it holds that $Deck(Enc_k(m)) = k \oplus (k \oplus m) = m$, and so the one-time pad constitutes a valid encryption scheme.

One can easily prove perfect secrecy of the one-time pad using Lemma 2.1 and the fact that the ciphertext is uniformly distributed regardless of what message is encrypted.

Theorem:

The one-time pad encryption scheme is perfectly secret.

Proof. We use (3) from Lemma 2.1. Fix any two messages $m, m' \in \mathcal{M}$ and any ciphertext $c \in \mathcal{C}$.

$$Pr_K[Enc_K(m) = c] = Pr_K[K \oplus m = c] = Pr_K[K = c \oplus m] = 2^{-n}$$

By the same argument, $Pr_K[Enc_K(m') = c] = 2^{-n}$. Therefore, $Pr_K[Enc_K(m) = c] = Pr_K[Enc_K(m') = c]$, and the one-time pad is perfectly secret. □

The one-time pad has two major drawbacks:

- The key must be as long as the message.
- The key can be used only once.

In fact assume we encrypt two messages m and m' using the same key k. The adversary can compute $c \oplus c' = (k \oplus m) \oplus (k \oplus m') = m \oplus m'$, and this can leak information about m and m'.

If I know a single pair (m, c), I can decrypt all the other messages encrypted with the same key.

Theorem:

In any perfectly secret encryption scheme, the key space must be at least as large as the message space. Formally, if $\Pi = (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ is a perfectly secret encryption scheme with message space $\mathcal{M}$ and key space $\mathcal{K}$, then $|\mathcal{K}| \geq |\mathcal{M}|$.

Proof. We show that if $|\mathcal{K}| < |\mathcal{M}|$, then Π cannot be perfectly secret. Assume $|\mathcal{K}| < |\mathcal{M}|$. Consider the uniform distribution over $\mathcal{M}$ and let $c \in \mathcal{C}$ be a ciphertext such that $Pr[C = c] > 0$. Let $\mathcal{M}'$ be the set of all possible messages that are possible decryptions of c; that is,

$$\mathcal{M}' = \{m \mid m = Dec_k(c) \text{ for some } k \in \mathcal{K}\}.$$

Clearly $|\mathcal{M}'| \leq |\mathcal{K}| < |\mathcal{M}|$, and so there must exist a message $m \in \mathcal{M} \setminus \mathcal{M}'$. But then

$$Pr[M = m \mid C = c] = 0 \neq Pr[M = m] = \frac{1}{|\mathcal{M}|},$$

and so Π is not perfectly secret. □

2.3 Shannon's Theorem

Theorem (Shannon's Theorem):

Let $\Pi = (\mathbf{Gen}, \mathbf{Enc}, \mathbf{Dec})$ be an encryption scheme with message space $\mathcal{M}$, key space $\mathcal{K}$, and ciphertext space $\mathcal{C}$. Suppose that $|\mathcal{K}| = |\mathcal{C}| = |\mathcal{M}|$ and that the key-generation algorithm chooses a key from $\mathcal{K}$ according to the uniform distribution. Then Π is perfectly secret if and only if the following two conditions hold:

1. For every $m \in \mathcal{M}$ and every $c \in \mathcal{C}$, there exists a unique key $k \in \mathcal{K}$ such that $c = Enc_k(m)$.
2. The key-generation algorithm chooses a key from $\mathcal{K}$ according to the uniform distribution (i.e., each of the $|\mathcal{K}|$ keys is chosen with probability exactly $1/|\mathcal{K}|$).

Chapter 3

Message Authentication Codes

Also called MACs, are a way to ensure integrity and authenticity of a message.

Alice generates a tag τ for a message m using a secret key k shared with Bob. She sends (m, τ) to Bob, who verifies the tag using the same key k. An adversary who intercepts (m, τ) should not be able to produce a valid tag for a different message m'.

Correctness: By definition if tag is deterministic.

Unforgeability: An adversary who sees tags for messages of his choice should not be able to produce a valid tag for a new message. Is hard to produce a valid pair (m', τ') as long as $m' \neq m$.

Definition: Statistical Secure MAC

We say that Π =Tag has ε-statistical security (Unforgeability) if $\forall m \neq m' \in \mathcal{M}, \forall \tau, \tau' \in \mathcal{T}$:

$$Pr_K[Tag_K(m') = \tau' \mid Tag_K(m) = \tau] \leq \varepsilon.$$

Here ε is a small parameter, e.g., 2^{-80}.

Is impossible to get $\varepsilon = 0$ because a random $\tau' \leftarrow Tag_K(m')$ is valid with probability at least $1/|\mathcal{T}|$.
Observation: note that the definition is one time!

We will show:

- The notion is achievable.
- It's inefficient.

In fact:

Theorem:

Any t-time $2^{-\lambda}$-statistical secure Tag has a key of size at least $(t+1) \cdot \lambda$.

We now show that any family of hash function with a particular property satisfies the definition.

Definition: Pairwise Independence

A family of hash functions $\mathcal{H} = \{h_k : \mathcal{M} \to \mathcal{T}\}_{k \in \mathcal{K}}$ is called *universal* if $\forall m \neq m' \in \mathcal{M}$ then:

$$(h_k(m), h_k(m'))$$

is uniform over $\mathcal{T}^2 = \mathcal{T} \times \mathcal{T}$ when k is uniform over $\mathcal{K}$.

Appendix A

Appendix

A.1 Identities and Inequalities

We list some standard identities and inequalities that are used at various points throughout the text.

THEOREM A.1 (Binomial expansion theorem) *Let x, y be real numbers, and let n be a positive integer. Then*

$$(x+y)^n = \sum_{i=0}^{n} \binom{n}{i} x^i \, y^{n-i}.$$

PROPOSITION A.2 *For all $x \geq 1$ it holds that $(1-1/x)^x \leq e^{-1}$.*

PROPOSITION A.3 *For all x it holds that $1 - x \leq e^{-x}$.*

PROPOSITION A.4 *For all x with $0 \leq x \leq 1$ it holds that*

$$e^{-x} \leq 1 - \left(1 - \frac{1}{e}\right) \cdot x \leq 1 - \frac{x}{2}\,.$$

The *conditional probability of E_1 given E_2*, denoted $\Pr[E_1 \mid E_2]$, is defined as

$$\Pr[E_1 \mid E_2] \stackrel{\text{def}}{=} \frac{\Pr[E_1 \wedge E_2]}{\Pr[E_2]}$$

as long as $\Pr[E_2] \neq 0$. (If $\Pr[E_2] = 0$ then $\Pr[E_1 \mid E_2]$ is undefined.) This represents the probability that event E_1 occurs, given that event E_2 has occurred. It follows immediately from the definition that

$$\Pr[E_1 \wedge E_2] = \Pr[E_1 \mid E_2] \cdot \Pr[E_2]\,;$$

equality holds even if $\Pr[E_2] = 0$ as long as we interpret multiplication by zero on the right-hand side in the obvious way.

We can now easily derive Bayes' theorem.

(Bayes' Theorem) *If $\Pr[E_2] \neq 0$ then*

$$\Pr[E_1 \mid E_2] = \frac{\Pr[E_2 \mid E_1] \cdot \Pr[E_1]}{\Pr[E_2]}.$$

PROPOSITION A.10 (Markov's inequality) *Let X be a non-negative random variable and $v > 0$. Then $\Pr[X \geq v] \leq \mathsf{Exp}[X]/v$.*

PROPOSITION A.11 (Chebyshev's inequality) *Let X be a random variable and $\delta > 0$. Then:*

$$\Pr[|X - \mathsf{Exp}[X]| \geq \delta] \leq \frac{\mathsf{Var}[X]}{\delta^2}.$$