MÁSTER INTERUNIVERSITARIO EN CIBERSEGURIDAD

Information security

Lecture 1: Encryption

Manuel Fernández Veiga, Marcos Gestal, Fernando Pérez-González

Universidade de Vigo

Table of contents

Encryption

Outline

Encryption is simply message secrecy: we intend to transmit/store a message which is only meant for a legitimate receiver and anyone else

We shall study in this lecture

- Formal definitions of secrecy: perfect & semantic security
- Techniques for single-message encryption
- Adversarial attacks to ciphers

Shannon secrecy

Ciphering is a **mathematical mapping** of a sequence of symbols: a form of **coding**. A formal definition:

Definition: Shannon cipher

A **Shannon cipher** is a pair $\mathcal{E} = (E, D)$ of functions such that

- The encryption function $E : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C}$ takes a **key** $k \in \mathcal{K}$, a **plaintext** $m \in \mathcal{M}$ and outputs a **ciphertext** $c \in \mathcal{C}$, $c = E(k, m)$.
- The decryption function $D : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$ takes a key and a ciphertext and outputs a message, $m = D(k, c)$.
- E and D are inverses (correctness property): for all k, m

$$D\big(k, E(k, m)\big) = m.$$

Shannon ciphers

Some remarks Note the following

1. The definition of a Shannon cipher is operational: we do not specify (for the moment) the encryption and decryption functions
2. We assume that the ciphertext c is not tampered
3. We assume that k is a **secret key**
4. Intuitively, communication is secure iff it is hard to guess m only from c without knowing k: c alone gives very little or no "information" about m. Therefore, for all m, c we should see almost a random guess

$$\mathbb{P}(\mathbf{m} = m \mid \mathbf{c} = c) = \frac{1}{|\mathcal{M}|} \pm \varepsilon$$

for a **very small** ε, e.g. $\varepsilon = 2^{-128}$.

Examples of Shannon ciphers

Example 1: one-time pad

Let $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0,1\}^L$, the set of L-bit sequences. The one-time pad is

$$E(k,m) = k \oplus m \qquad D(k,c) = k \oplus c.$$

Notice that the same function $\oplus$ is used for encryption and decryption

Example 2: variable-length one-time pad

Let $\mathcal{K} = \{0,1\}^L$, $\mathcal{M} = \mathcal{C} = \{0,1\}^{\leq L}$, the sets of L-bit sequences and bit sequences up to L bits, respectively. Let ℓ be the length in bits of message m. The variable-length one-time pad is

$$E(k,m) = k_1^\ell \oplus m \qquad D(k,c) = k_1^\ell \oplus c,$$

where k_1^ℓ is the key k shortened to ℓ bits.

Examples of Shannon ciphers

Example 3: Substitution cipher

Let $\mathcal{A}$ be a finite alphabet. Put $\mathcal{M} = \mathcal{C} = \mathcal{A}^L$ and $\mathcal{K}$ the set of all permutations on $\mathcal{A}$. Then, a substitution cipher is the pair

$$E(\sigma, m) = \left(\sigma(m_1), \sigma(m_2), \ldots, \sigma(m_L)\right) \quad D(\sigma, c) = \left(\sigma^{-1}(c_1), \ldots, \sigma^{-1}(c_L)\right).$$

Many modern block ciphers (AES, DES) are in fact substitution ciphers.

Example 4: additive one-time pad

With $\mathcal{K} = \mathcal{M} = \mathcal{C} = \{0, 1, \ldots, n-1\}$, use

$$E(k, m) = k + m \mod n, \qquad D(k, c) = -k + c \mod n.$$

Again, E and D are the same function. Also, note that the additive OTP is a substitution cipher.

Perfect security

Perfect security

There are many ways to define "security" rigorously. We focus first on perfect security, which is the strongest and ideal notion of communications security.

Definition: perfect security

Let $\mathcal{E} = (E, D)$ be a Shannon cipher. $\mathcal{E}$ is perfectly secure if for all $m_0, m_1 \in \mathcal{M}$ and $c \in \mathcal{C}$ we have

$$\mathbb{P}\big(E(\mathbf{k}, m_0) = c\big) = \mathbb{P}\big(E(\mathbf{k}, m_1) = c\big)$$

where $\mathbf{k}$ is a random key uniformly distributed in $\mathcal{K}$.

In words, $E(\mathbf{k}, m_0)$ and $E(\mathbf{k}, m_1)$ are equal in distribution, so we cannot effectively distinguish between m_0 and m_1 just by looking at the ciphertexts.

Remark: note that perfect security imposes strict equality between distributions

Understanding perfect security

There are a number of equivalent ways to define perfect security

Assume that the message m is drawn uniformly from $\mathcal{M}$ and is statistically independent of the key k. Then

1. $\mathcal{E}$ is perfectly secure iff the ciphertext and the message are statistically independent, $\mathbf{c} \perp\!\!\!\perp \mathbf{m}$.
2. $\mathcal{E}$ is perfectly secure iff there does not exist a statistical test which can distinguish two messages from their ciphertexts
3. $\mathcal{E}$ is perfectly secure iff[1]

$$I(\mathbf{m}; \mathbf{c}) = 0, \qquad H(\mathbf{c} \mid \mathbf{m}, \mathbf{k}) = 0.$$

[1] $I(\cdot;\cdot)$ is the *mutual information*; $H(\cdot)$ is the Shannon entropy.

Understanding perfect security

Examples

- The one-time pad is perfectly secure.
- The substitution cipher is perfectly secure.
- The additive one-time pad is perfectly secure.
- The variable-length one-time pad **is not perfectly secure**
  Why?: because we can learn the length of the message just by looking at the ciphertext, and that length gives us information.
  We cannot use the VL-OTP for secrecy!

So, perfectly secure ciphers exist and are very simple $\Rightarrow$ problem solved?

A converse to perfect security

Actually, no!

Theorem (Shannon)

Let $\mathcal{E}$ be a Shannon cipher, and assume that $\mathcal{E}$ is perfectly secure. Then $|\mathcal{K}| \geq |\mathcal{M}|$

Hence, the key space must be at least as large as the message space for any perfectly secure cipher $\Leftrightarrow$ using a key more than once is not secure

We will see how to cipher multiple messages with the same key later

Strictly, the Shannon theorem establishes that the *entropy* of the key must be at least as large as the *entropy* of the message, $H(\mathsf{k}) \geq H(\mathsf{m})$. We shall not explore this (unless you know information theory).

Semantic security & computational ciphers

Computational ciphers

Shannon's theorem tells us that perfect security is a too strong notion of security

In practice, we only insist that there should not exist a computational device which can produce more than a negligible advantage when its input are two different ciphertexts. Formally[2]

$$|\mathbb{P}\big(\phi(E(\mathsf{k}, m_0))\big) - \mathbb{P}\big(\phi(E(\mathsf{k}, m_1))\big)| \leq \varepsilon$$

for a negligible ε, and any test ϕ.

This requirement is typically posed as an attack game between a challenger and a computational adversary

[2] Think *very carefully* about this definition, it's subtle.

Attack games & semantic security

An attack game is simply a protocol between a challenger and an adversary. For a cipher $\mathcal{E}$ and an adversary $\mathcal{A}$ define two experiments $b = 0, 1$. Under experiment b

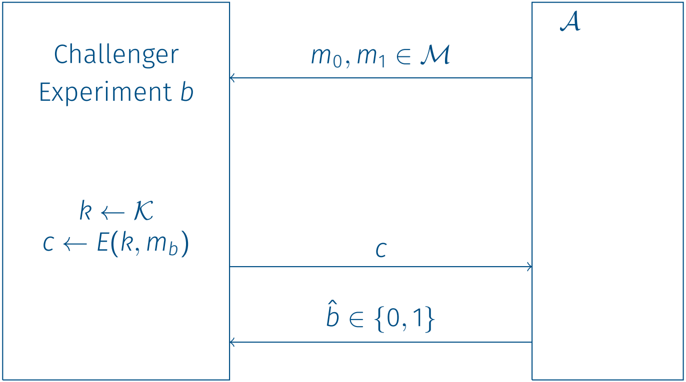
1. $\mathcal{A}$ sends two messages m_0, m_1 of his choice to the challenger
2. The challenger draws a random key k, computes $c \leftarrow E(k, m_b)$ and sends c to $\mathcal{A}$
3. The adversary outputs a bit $\hat{b}$

Definition

$\mathcal{E}$ is semantically secure against $\mathcal{A}$ if

$$|\mathbb{P}\big(\hat{b} = 1 | b = 0\big) - \mathbb{P}\big(\hat{b} = 1 | b = 1\big)| \leq \varepsilon.$$

The attack games



Understanding semantic security

Intuitively, the attack game is this: the adversary chooses two different messages. Is there any significant (i.e., computable) statistical difference between the **ciphertexts** of both messages? If not, $\mathcal{E}$ is semantically secure for any efficient adversary $\mathcal{A}$

More intuition

- $\mathcal{A}$ is allowed to use any **efficient** computational procedure
- ε is not zero, but it should be negligible, e.g., 2^{-200}, zero for all practical purposes

Theorem

A deterministic cipher which is perfectly secure is also semantically secure. The converse is not generally true.

Consequences of semantic security

Again, there are several alternative characterizations of semantic security (SS):

- For a SS cipher, it is computationally hard to predict bits of the message (bit-guessing games, e.g. predicting the parity)
- For a SS cipher, it is computationally hard for the adversary to recover the message m from the ciphertext

Attacks to a SS cipher: if the semantic security of $\mathcal{E}$ is lower than ε, then a brute-force attack on $\mathcal{E}$ —like testing all the possible keys— would take time proportional to $1/\varepsilon$

But if ε is negligible, $1/\varepsilon$ is **super-poly**! Infeasible

Application: nested encryption & onion routing

Suppose Alice wants to send a message to Bob anonymously, without disclosing her identity. She can agree with a third person, Carol, who acts as intermediate messenger

$$E\big(k_{\text{Carol}}, (\text{Bob}, m)\big) \longrightarrow E(k_{\text{Bob}}, m) \longrightarrow m$$

But

- If Carol and Bob collude, Bob can find out Alice's identity
- An eavesdropper watching the two channels can learn that Alice and Bob communicate

These two problems can be solved as follows:

- Collusion: use two or more intermediaries, so that the second one cannot reveal the identity of the source
- Mixing: the intermediaries relay messages from multiple sources **in a random order** unknown to the eavesdropper

Nested encryption & source routing

Onion routing = nested encryption + source routing

Assume a route $s \equiv h_0 \to h_1 \to \cdots \to h_{n-1} \to h_n \equiv d$

The source s sends

$$E(k_1, (h_2, E(k_2, \ldots E(h_{n-1}, E(h_n, E(k_n, m))))))$$

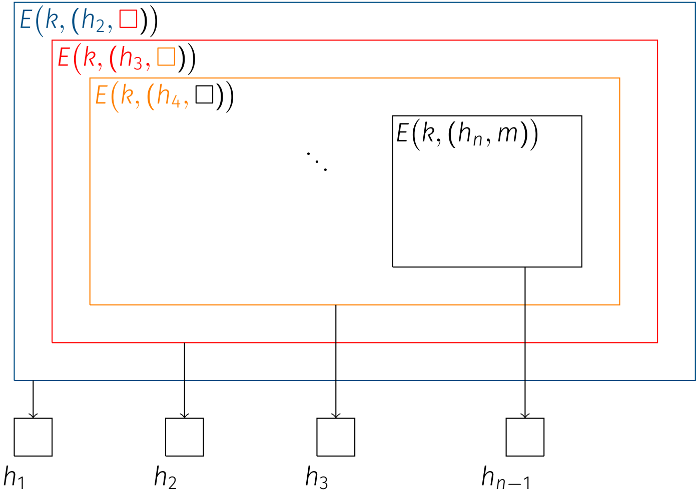
Then hop i gets the message $m_i = (h_{i+1}, m_{i+1})$ and sends m_{i+1} to h_{i+1}, for $i = 1, \ldots, n-1$, after mixing.

Remarks:

1. For $i \geq 2$, h_i cannot tell who the source is.
2. h_i cannot decrypt m_{i+1}
3. Mixing guarantees randomness in time

This is how TOR (TOR = the onion routing) routes messages in the deep web

Onion routing



Quantum key distribution (QKD)

Even with OTP for perfect security, secrecy is only possible if the two parties share a common secret (the key). How can a secret key be agreed on over an insecure channel?

Quantum key distribution (QKD) uses fundamental physical laws to solve this conundrum:

- Measurement of a quantum state inevitably disturbs the state
- No-cloning theorem: an unknown quantum state cannot be cloned (copied)

QKD

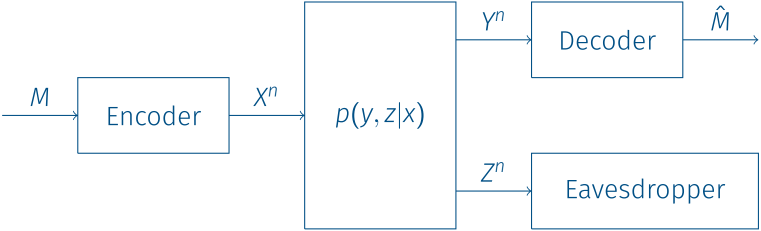
In QKD

1. A trusted source send a stream of entangled photons to Alice and Bob, randomly polarized in two different basis
2. Alice and Bob measure the received photons in a random basis, and exchange with the source the list of basis used for measurement
3. They agree on the values of a subset of the bits where the measurement basis coincide $\Rightarrow$ the key
4. Any third party that intercepts and measures the stream will change the state/basis of the entangled pair received by Alice and Bob. This can be detected

Protocols for QKD: BB84, T12 protocol,Decoy state protocol, SARG04, E91 protocol, B92 protocol, BBM92 protocol, MSZ96 protocol, COW protocol, DPS protocol,KMB09 protocol, HDQKD, ...

Information theoretic secrecy

The wiretap channel



Information leakage rate: $R_L = \frac{1}{n} I(M; Z^n)$

Error probability: $P_e^n = \mathbb{P}(M \neq \hat{M})$

Secrecy capacity: maximize the communications rate under the conditions

$$P_e^n \to 0 \qquad \text{and } R_L \to 0$$

when $n \to \infty$. This has a **known solution** in information theory.

Note that the wiretap channel is a pure *channel coding problem*, namely no secret key exists between the transmitter and the receiver

Conclusion

Summary

In this lecture

- Definition of a cipher
- Definition of perfect security
- Perfect security has limitations and it is too strong for practical use
- Semantic security: no *efficient* computational procedure exists for discovering useful information about the key or the message
- Information-theoretic aspects of secrecy: physical layer security

In the rest of the course, we will study semantically secure techniques, and we will explore other stronger forms of security: a long and winding road

Mathematical details

Negligible, super-poly, poly-bounded

A function $f(n)$ is **negligible** if for all $n \geq n_0$, $|f(n)| < 1/n^c$ for any $c > 0$. Examples: 2^{-n}, $n^{-\log n}$

$f(n)$ is **super-poly** if $1/f(n)$ is negligible, and is **poly-bounded** if $f(n) \leq n^c$ for some $c > 0$.

An algorithm parametrized by λ is **efficient** if there exist a poly-bounded function t and a negligible function ϵ such that the probability that the running time of the algorithm exceeds $t(\lambda)$ is bounded by $\epsilon(\lambda)$.

Thus, efficient algorithms are those which run in a poly-bounded time with overwhelming probability.

Some properties

If ε and ε' are negligible, and Q, Q' are poly-bounded then

1. $\varepsilon + \varepsilon'$ is negligible.
2. $Q + Q'$ and $Q \cdot Q'$ are poly-bounded.
3. $Q \cdot \varepsilon$ is negligible.