**Cryptography**

# Notes: On the notion of security

*Written by: Marius Zimand*

What does it mean that a crypto system is secure?

Of course, if the adversary finds the entire plaintext or the entire secret key, that would be a severe failure. But even if the adversary finds a small part of the plaintext or the key, or even if the adversary determines that, say, the first letter of the plaintext is more likely to be an $A$ than the usual frequency of an $A$ at the beginning of a word in a typical English text, that would also be a weakness.

We want to give a formal definition of what we mean when we say that a cryptosystem is secure.

**Idea:** A cryptosystem is secure to an attack if the adversary does not learn anything after the attack compared to what he knew before the attack.

We will formalize this idea.

In these notes we consider the case of the *ciphertext-only attack*. The other types of attacks can be formalized similarly.

We define two types of security:

1. **perfect security**: the adversary does not learn anything, no matter her computational power and how much time the attack takes. This is the ideal, but cannot be realized by practical cryptosystems.

2. **computational security**: the adversary does not learn anything unless she is performing more than $N$ operations, where $N$ is some huge number (so that the attack takes thousands of years). This is good enough and may be achieved by practical cryptosystems.

To formally define the notion of security we need some **probability notions**. First, we introduce some notation:

- $M$ is a random variable that denotes a message chosen from the set of messages $\mathcal{M}$;
- $K$ is a random variable that denotes the e-key chosen from the set of keys $\mathcal{K}$;
- $C$ is the encryption of $M$, i.e., $C = E_K(M)$.
- $M$ is characterized by its distribution (see example below);
- the key $K$ is chosen uniformly at random (i.e., all the keys are equally likely).

Simple example: Suppose the message comes from a military base. To keep things simple, let us assume that the base sends only three messages: "nothing to report," "attack with 5 planes" and "attack with 10 planes.". Then

$\mathcal{M} =$ {"nothing to report," "attack with 5 planes," "attack with 10 planes"}.

This is called the *set of messages*. We can endow a set of messages with a probability distribution (in short, just *distribution*), indicating how likely each message is.

For example, one possible distribution can be

$$M = \begin{pmatrix} \text{nothing to report} & \text{attack with 5 planes} & \text{attack with 10 planes} \\ 0.6 & 0.3 & 0.1 \end{pmatrix}$$

We should assume that the attacker knows the distribution $M$ (similar to knowing the frequency of letters in English).

Let us introduce the notion of *conditional probability.*

**Conditional Probability:** $P(A \mid B)$ = prob. of $A$ conditioned by $B$, where $A$ and $B$ are two random events.

Intutively: The prob. of event $A$ if we know that event $B$ has happened.

Example: Throw two fair dice.

$A$ = the event "sum is 6" (6=1+5, 2+4, 3+3, 4+2, 5+1).
$P(A) = \frac{5}{36}$. (Because $A$ consists of 5 elements of the probability space each having probability 1/36).

$B$ = the event "both dice are even".
$P(A \mid B) = \frac{2}{9}$. (This time the probabilistic space is $B$ with 9 elements; $A$ consists of 2 elements each with probability 1/9).

**Theorem 1** *(sometimes this is taken as the def. of conditional probability)*
*For any events $A$ and $B$, $P(A \mid B) = \frac{P(A \cap B)}{P(B)}$.*

**Definition 2 Independent events:** *$A$ and $B$ are independent if $P(A \cap B) = P(A) \times P(B)$.*

**Observation:** Events $A$ and $B$ are indeoendent if and only if $P(A \mid B) = P(A)$. Example:
$A$ = first die is 5.
$B$ = 2nd die is 2 or 4.

$P(A \cap B) = \frac{2}{36} = \frac{1}{18}$ (because there are 2 good pairs, namely $(5, 2)$ and $(5, 4)$.
$P(A) \times P(B) = \frac{1}{6} \times \frac{1}{3} = \frac{1}{18}$.

An encryption scheme is a mapping that takes a message $M$ from the space of messages and a key $K$ from the space of keys and calculates the ciphertext $C$. We denote this by $E_K(M) = C$.

**Definition 3** *An encryption scheme over message space $\mathcal{M}$ is* perfectly secure - version 1 *if for all distributions $M$ over $\mathcal{M}$, for any fixed message $m$ and for any fixed ciphertext $c$, we have*

$$P(M = m \mid C = c) = P(M = m),$$

*where* $C = E_K(M)$.

*Here the probabilities are taken over the distribution* $M$ *and over choosing the key* $K$ *uniformly at random in the space of all keys.*

Notes

- This is equivalent to saying that $M$ and $C$ are independent.
- What the definition is saying:

  The distribution on $\mathcal{M}$ is supposed to be known by the adversary. This is called the *prior distribution*.

  The distribution that the attacker knows (for the space of messages) after seeing $c$ is $P(M = m \mid C = c)$. This is the *posterior distribution*.

  We just want that the cryptosystem does not leak any additional information. This is captured in the definition by saying that knowing the ciphertext $c$ does not change the distribution $M$, because the prior distribution and the posterior distribution are identical.

- We have intuitively argued that the one-time pad has the above property. Now we can prove this assertion rigorously.

**Theorem 4** *One-time pad is perfectly secure.*

**Proof** of a special case (the general case is similar).

The special case is that we take $\mathcal{M} = \{0, 1\}$ - just two messages.

Let us denote $C = E_K(M)$.

We need to show that $P(M = m \mid C = c) = P(M = m)$, for all possible messages $m$ and ciphertexts $c$. Recall that the probability is taken over the choice of $M$ according to some arbitrary distribution, and the random choice of the key using the uniform distribution, i.e., all keys are equally likely; also, recall that in one-time pad, $C = M + K$, where $+$ is addition modulo 2, or equivalently, bitwise XOR.

For example, let's show that $P(M = 0 \mid C = 0) = P(M = 0)$. (The other 3 cases are similar).

$$\begin{aligned} P(M = 0 \mid C = 0) &= \frac{P(M = 0 \cap C = 0)}{P(C = 0)} = \frac{P(M = 0 \cap (M + K) = 0)}{P(C = 0)} \\ &= \frac{P(M = 0 \cap K = 0)}{P(C = 0)} = \frac{P(M = 0)P(K = 0)}{P(C = 0)}. \end{aligned}$$

Now we show that

$$P(K = 0) = 1/2 \text{ and } P(C = 0) = 1/2,$$

from which the conclusion follows.

$P(K = 0) = 1/2$ is immediate, because there are 2 equally likely keys (namely 0 and 1).

$$\begin{aligned} P(C=0) &= P(M=0 \cap K=0) + P(M=1 \cap K=1) \\ &= P(M=0)P(K=0) + P(M=1)P(K=1) \\ &= P(M=0)\cdot(1/2) + P(M=1)\cdot(1/2) \\ &= (1/2)(P(M=0)+P(M=1)) \\ &= 1/2. \end{aligned}$$

In the case of the one-time pad cryptosystem, the key is as long as the message, which means that the space of keys is as large as the space of messages. The next theorem, proved by Claude Shannon, shows that this is the case for any encryption scheme that is perfectly secure-version 1. In other words, any encryption scheme that is perfectly secure-version 1 suffers from the same impracticality issue as the one-time pad.

Notation: $||A||$ denotes the number of elements of the finite set $A$.

**Theorem 5 (Shannon's Theorem)** *If an encryption scheme is perfectly secure-version 1 over message space $\mathcal{M}$, then the set of keys $\mathcal{K}$ must satisfy $||\mathcal{K}|| \geq ||\mathcal{M}||$.*

**Proof.** Let $c$ be a ciphertext. Suppose $||\mathcal{K}|| < ||\mathcal{M}||$. Then when we decrypt $c$ with all possible keys, we obtain at most $||\mathcal{K}||$ possible plaintexts. So there is a message $m$ that is not obtained. Then $P(M = m \mid C = c) = 0$. But clearly we can make a distribution with $P(M = m) > 0$.

Thus if for example we look at messages that are say 1000 bits long, there are $2^{1000}$ possible messages, and we need at least $2^{1000}$ keys, so a key on average must be at least 1000 bits long. So, **perfectly secure version 1** is too much to ask, because it can be achieved only by very impractical encryption schemes (such as one-time pad).

The definition of an encryption that is *perfectly secure - version 1* may seem to be too abstract and not be very convincing. Let us try another attempt for defining security. This definition has the merit that it models the fact that the adversary does not get anything if she is doing a ciphertext-only attack.

**Definition 6** *An encryption scheme over message set $\mathcal{M}$ is* perfectly secure- version 2 *if for any two messages $m_1$ and $m_2$ in $\mathcal{M}$ and for any algorithm $A$, we have*

$$P(A(C) = m_1 \mid C = E_K(m_1)) = P(A(C) = m_1 \mid C = E_K(m_2)).$$

Notes:

- Think that $A$ is an attacker that wants to guess whether $C$ is the encryption of $m_1$ or of $m_2$.
- The definition assumes that the enemy does a *ciphertext-only attack*, because $A$ has as input only $C$. Security against the other kind of attacks can be defined (more or less) similarly.
- The probabilities are taken over the random choice of the key $K$ (and the random decisions of $A$ if $A$ is a probabilistic algorithm).

- A succesfull atacker would have the LHS big (ideally 1) and the RHS small (ideally 0).
- The definition says that $A$ is not doing any better at guessing the message when it is given an encryption of $m_1$ than when it is given an encryption of $m_2$.

**Theorem 7** *perfectly secure - version 2 = perfectly secure - version 1.*
*(This means that an encryption scheme is secure according to version 1 if and only if it is secure according to version 2).*
*(Proof omitted, not hard but long).*

Thus perfectly secure - version 2 cannot be achieved by practical encryption schemes either.

So we adopt a more relaxed definition – this is "computational security."

**Definition 8** *Let $\epsilon$ be a small parameter (for ex. $\epsilon = 0.0001$) and $N$ be a large parameter (for ex. $N = 10^{80}$). An encryption scheme over message space $\mathcal{M}$ is* computational secure *(with parameters $\epsilon$ and $N$) if for any two messages $m_1$ and $m_2$ in $\mathcal{M}$ and for any algorithm $A$ that performs $N$ operations, we have*

$$|P(A(C) = m_1 \mid C = E_K(m_1)) - P(A(C) = m_1 \mid C = E_K(m_2))| < \epsilon.$$

Notes:

1. There are two relaxations compared with "perfectly secure - version 2."
   - We don't require equality between the two probabilities, just closeness within $\epsilon$.
   - And it is ok if the attacker can break the system by doing a huge number of operations: if an attacker must spend billions of year to break the cryptosystem, then the cryptosystem is considered secure.
2. The above definition only defines security against ciphertext-only attacks. In the same spirit, we can define computational security against stronger types of attacks, such as chosen plaintext attack, or chosen ciphertext attack.
3. What should be the concrete values for $N$ (the number of operation we allow the adversary to do) and $\epsilon$ (the bias we allow the adversary to achieve)? The current recommendations state that it is ok if no adversary running for at most $N = 2^{80}$ CPU cycles can break the system with probability greater than $2^{-64}$. Some people recommend these days to have $N = 2^{128}$.

   Let's get a feel for these values. A modern 3 GHz computer with eight "cores" does approx 24 billions operations per second. Suppose we have an extraordinary computer which does one trillion $\approx 2^{40}$ operations per second. Then to perform $2^{80}$ operations would require $2^{80-40} = 2^{40}$ seconds, which is 34900 years. Executing $2^{128}$ operations requires $9 \cdot 10^{18}$ years.

   The age of universe is estimated to be $13.8 \cdot 10^9$ years.

An event that occurs once every hundred years can be roughly estimated to occur with probability $2^{-30}$ in any given second. Something that occurs with probability $2^{-60}$ in any given second is $2^{30}$ times less likely and might be expected to occur roughly once every 100 billion years.