Perfect Security

Somitra Sanadhya

11 January 2024

1 Introduction

In this short note, I describe what I covered in the class on Monday and today.

Firstly, we have the following sets:

1. **K**: Random variable corresponding to the secret key.
2. **M**: Random variable corresponding to the plaintext.
3. **C**: Random variable corresponding to the ciphertext.

We have the three sets, $\mathcal{K}, \mathcal{M}$, and $\mathcal{C}$ corresponding to the three sets associated with keys, plaintexts, and ciphertexts, respectively.

1.1 Encryption scheme

An encryption scheme consists of the following 3 algorithms:

1. KeyGen(1^n): It takes the security parameter n as the input and produces a secret key $k \in \mathcal{K}$. Usually, the key is produced uniformly at random (but it is not necessary). This algorithm is necessarily randomized.
2. Enc(k, m): Encryption function which takes a key and a plainext message and produces a ciphertext c. This algorithm may be randomized. (Later, we will see that it *must be* randomized.
3. Dec(k, c): Decryption function which takes a secret ket and the ciphertext and produces a plaintext. This algorithm must be deterministic.

Correctness requirement :

$\forall k \in \mathcal{K}$ and $\forall m \in \mathcal{M}$, we should have that Dec$(k,$ Enc$(k, m)) = m$.

Security notion : We discussed why it is not very easy to come up with a formal definition of security which can capture our intuitive notion of security.

2 Different flavors of perfect security

1. **Shannon Security**: This is the notion defined by Shannon in his groundbreaking paper "A mathematical theory of communication systems". It captures the notion that the ciphertext does not permit extraction of any *non-trivial* information about the plaintext.

   $$\Pr_{k,m}(M = m | C = c) = \Pr_m(M = m)$$

   In the above, we have used the notation $\Pr_a(\cdot)$ to denote that the probability is over randomness of the variable a, and any randomness of the algorithms used in the process itself.

2. **Independence of M and C**: This notion says that the random variables corresponding to the plaintext and the ciphertext are independent.

   Recall that X and Y are independent random variables if $\Pr[X \cap Y] = \Pr[X] \cdot \Pr[Y]$.

3. **Perfect Security**: This notion states that the probability of any two plaintexts have equal probability to result in some ciphertext c, where the probability is over randomness of the key, and possibly of the encryption algorithm itself (remember that the encryption function itself may be randomized).

   $$\forall m, m' \in \mathcal{M}, \forall c \in \mathcal{C} : \Pr[\mathrm{Enc}_K(m) = c] = \Pr[\mathrm{Enc}_K(m') = c].$$

Interestingly, all three notions defined above are identical. That is,

$$(1) \implies (2) \implies (3) \implies (1).$$

3 Proof of equivalence of the different security notions

3.1 (1) $\implies$ (2)

We are given that (1) holds. That is, $\Pr_{k,m}(M = m|C = c) = \Pr_m(M = m)$. Our goal is to prove that M and C are independent.

$$\begin{aligned} \Pr_{k,m}(M = m|C = c) &= \Pr_m(M = m) \\ \frac{\Pr_{k,m}((M = m) \cap (C = c))}{Pr_{k,m}(C = c)} &= \Pr_m(M = m) \\ \implies \Pr_{k,m}((M = m) \cap (C = c)) &= \Pr_{k,m}(C = c) \cdot \Pr_m(M = m). \end{aligned}$$

This is precisely what we wanted to prove.

3.2 (2) $\implies$ (3)

Let us pick an $m \in \mathcal{M}$, and a ciphertext $c \in \mathcal{C}$. As a first step, we simply play with the way we defined the random variables M and C, and can write the following:

$$\begin{aligned} \Pr_{k,m}\left[\text{Enc}[k,m]=c\right] &= \Pr_{k,m}\left[\text{Enc}[k,M]=c \,\middle|\, M=m\right] \\ &= \Pr_{k,m}\left[C=c \,\middle|\, M=m\right] \end{aligned}$$

Now, pick any two messages m_1 and m_2, and any ciphertext c. *If* we can prove that $\Pr[C=c \mid M=m_1] = \Pr[C=c \mid M=m_2]$ (where we permit any random m_1 and m_2), then we will be done.

We will prove this by showing that

$$\begin{aligned} \Pr[C=c \mid M=m_1] = &\quad \Pr[C=c] \quad \text{and} \\ \Pr[C=c \mid M=m_2] = &\qquad\quad '' \quad . \end{aligned}$$

You may already notice that the above statement is true. It is a direct consequence of (1).

3.3 (3) $\implies$ (1)

Left as an exercise.

4 Adversarial perfect security or Indistinguishability of ciphertext

This notion is based on a game being played between a challenger and an adversary. The adversary algorithm is denoted by $\mathcal{A}$. It follows the following steps:

1. $\mathcal{A}$ outputs a pair of messages m_0, m_1.
2. The challenger generates a random key as $k \leftarrow \text{KeyGen}(1^n)$. She also generates a random bit $b \overset{\$}{\leftarrow} \{0,1\}$.
3. Challenger computes $c = \text{Enc}(k, m_b)$, and sends c to the adversary.
4. Adversary produces a bit b' which is his guess for the message encrypted.
5. A wins the games if $b' = b$, else he loses.

The encryption scheme is perfectly secure if $\Pr(b' = b) = \frac{1}{2}$.

It is easy to see that this notion is also equivalent to the previous definitions. How would you prove this statement? If the adversary had any advantage in distinguishing between the two messages by observing the ciphertext, then you can show that one of the previous definitions will not be satisfied. This proof by contradiction is rather easy and is left as an exercise.

Notes

1. There is a subtle difference between $\Pr_{k,m^*}(\mathrm{Enc}(k, m^*) = c^*)$ and $\Pr_{k,m}(C = c^*)$. I use m^* and c^* to emphasise the point I am trying to explain. In the first term, we talk about the probability of the ciphertext being equal to c^* when the message is m^*. The second term is the unconditional probability of getting the ciphertext to be c^*. The second term can be computed as follows:

$$\begin{aligned}
\Pr_{k,m}(C = c) = & \quad \Sigma_i \Pr_{m_i}(M = m_i) \times \Pr(C = c^* \mid M = m_i) \\
= & \quad \Sigma_i \Pr_{m_i}(M = m_i) \times \alpha \\
= & \quad \alpha \times \Sigma_i \Pr_{m_i}(M = m_i) \\
= & \quad \alpha \\
= & \quad \Pr(C = c^* \mid M = m_i).
\end{aligned}$$

   Notice how I use the fact that probability of obtaining a ciphertext c^* is a constant α irrespective of which message is selected.