1 Last Lecture

In the last lecture, we saw two equivalent definitions of a perfectly secure encryption scheme:

- Perfect secrecy — Definition 02.03,
- Perfect indistinguishability — Definition 02.04.

We also saw a perfectly indistinguishable encryption scheme (Shannon's One Time Pad).

2 Perfect indistinguishability implies large keys

Though perfect secrecy/indistinguishability[1] should be an ideal goal, we will show in this section that for any perfectly indistinguishable and perfectly correct encryption scheme, the key space $\mathcal{K}$ must be as large as the message space $\mathcal{M}$. This makes perfect secrecy less feasible in real world.

Let us make a simple observation about encryption schemes that satisfies Definition 02.04.

Observation 2.1. Let m_0, m_1 be two distinct messages, and c be a ciphertext. Suppose there are exactly t secret keys $k_1, \ldots, k_t$ such that $\mathsf{Enc}(m_0, k_i) = c$, for $i \in [1, t]$. Then there must exist t keys $k'_1, \ldots, k'_t$ such that $\mathsf{Enc}(m_1, k'_j) = c$, for $j \in [1, t]$. Moreover, for all $i, j \in [t]$, $k_i \neq k'_j$.

Proof. Consider an encryption schemes that satisfies Definition 02.04. We have for all m_0, m_1, c,

$$\Pr_{k \in \mathcal{K}}[\mathsf{Enc}(m_0, k) = c] \;=\; \Pr_{k \in \mathcal{K}}[\mathsf{Enc}(m_1, k) = c].$$

Observe that $\Pr_k[\mathsf{Enc}(m_0, k) = c] = t/|\mathcal{K}|$. Thus, $\Pr_k[\mathsf{Enc}(m_0, k) = c]$ must also be $t/|\mathcal{K}|$, thereby proving that there are exactly t keys say $k'_1, \ldots, k'_t \in \mathcal{K}$ that maps m_1 to c.

The second part follows from perfect correctness. Suppose $k_i = k'_j$ for some $i, j \in [1, t]$. Then $\mathsf{Dec}(c, k_i) = \mathsf{Dec}(c, k'_j)$, implying m_0 must be equal to m_1. This contradicts the assumption that m_0, m_1 are distinct messages. □

Now, fix any message m^* and key k^*, and let $c^* = \mathsf{Enc}(m^*, k^*)$. There is at least one key mapping m^* to c^*. Therefore, using the observation above, for any message m, there exists a key, say $k_m \in \mathcal{K}$, such that $\mathsf{Enc}(m, k_m) = c^*$. Moreover, we showed above that for any distinct messages $m \neq m'$, the corresponding keys k_m and $k_{m'}$ are distinct. Using this, we can conclude that $|\mathcal{K}| \geq |\mathcal{M}|$.

3 Other limitations of One Time Pad

We prove in Lecture 2 that One Time Pad satisfies perfect secrecy, and therefore its key space must be as large as message space. This is a big limitation of One Time Pad. Below we list a few other limitations.

- **Two time pad attack:** Let c_1 and c_2 be two ciphertexts obtained via the one time pad encryption scheme **using the same secret key**, say k. An adversary, given two ciphertexts c_1 and c_2, can learn the XOR of the underlying messages. If $c_1 = \mathsf{Enc}(m_1, k) = m_1 \oplus k$ and $c_2 = \mathsf{Enc}(m_2, k) = m_2 \oplus k$, then $c_1 \oplus c_2 = m_1 \oplus m_2$.
- **Malleability attack:** Given an encryption of a message m using a key k, the adversary can modify the ciphertext so that the decryption using k outputs a different message. That is, suppose you encrypt your hard disk with a key k, and the resulting ciphertext is ct. The adversary does not know anything about the contents of your hard disk, but can modify the ciphertext so that (a) you will not detect any tampering (b) worse, when you decrypt the tampered ciphertext using your key k, you might receive a completely different message.

[1] We will use perfect secrecy/indistinguishability interchangeably, since they are equivalent.

Note that the adversary knows neither m nor k, but is able to modify the underlying message. For example, if the message and key are both n bits long, the adversary can flip all (or a fraction of) the message bits by XORing the ciphertext with an appropriate n bit string.

4 Overcoming the limitation of perfect secrecy

The lower bound on the size of key space is a huge limitation with regard to the practicality of encryption schemes. If we wish to have an encryption scheme where the key size is much smaller than the message size, then we cannot have perfect secrecy.

Recall that perfect indistinguishability guarantees security against ALL adversaries.[2] In reality, we don't need security against all possible adversaries – it suffices to guarantee security against *efficient* adversaries. By an efficient adversary we mean that the adversary is an arbitrary polynomial time randomized algorithm. Polynomial in what? Let us discuss that first.

4.1 Enter the security parameter

From now on, all our algorithms, and the adversary, will receive an integer input which will indicate how secure the system will be. We call this the *security parameter*.

For encryption schemes, we will assume there is a key generation algorithm that takes as input a security parameter $n \in \mathbb{N}$, runs in time polynomial in n, and outputs a key. (Note that, in general, the key space need not be $\{0,1\}^n$). Similarly, the encryption and decryption algorithms are also required to run in time polynomial in n.

Finally, we want that if the security parameter is n, then the adversary runs in time $p(n)$, where $p(\cdot)$ is some polynomial that depends on the adversary. This is what we mean by an efficient adversary.

4.2 A template for defining security against efficient adversaries

A common template for such definitions is via *security games*. We define an interactive game between a challenger and an adversary, where the game should capture the attack scenario. At the end of the game, the adversary either wins or loses, and a win corresponds to a successful attack against the encryption scheme.

We say that an encryption scheme is secure (w.r.t. the attack scenario) if, for all efficient adversaries, the probability of the adversary winning the game is *small*.

4.3 Security game for the attack scenario discussed in previous lectures

Recall, we want to prevent adversaries that, given a ciphertext, can learn some information ϕ about the underlying message. This attack is captured via the following security game. We call it the *no-query-semantic-security* game. *No-query* because the adversary is not making any encryption queries (in future lectures, we will strengthen the definition by allowing the adversary to query for encryptions of messages of its choice), and *semantic security* because the adversary only needs to learn one bit of information.

4.4 How small do we want the adversary's advantage to be?

Intuitively, we say that an encryption scheme is secure if for any efficient adversary, the probability of $\mathcal{A}$ winning the no-query-semantic-security game is bounded by some small fraction $s \in [0,1]$. Clearly, it is desirable to have s as small as possible.

[2] Note that Definition 02.04 does not explicitly talk about adversaries, but it is equivalent to the following definition (which involves adversaries):

An encryption scheme is perfectly secure if for **all** adversaries $\mathcal{A}$, $Pr[\mathcal{A}$ wins in No-Query-Semantic-Security game $] = 1/2$, where the No-Query-Semantic-Security game is defined in Figure 1.

You may want to prove that this is equivalent to Definition 02.04, but if you're too tired of perfect secrecy definitions, you can take this equivalence as a fact.

No-Query-Semantic-Security

1. Adversary sends two messages m_0, m_1 to the challenger.
2. The challenger chooses a bit $b \leftarrow \{0,1\}$, key $k \leftarrow \mathcal{K}$ and sends $\mathsf{Enc}(m_b, k)$ to the adversary.
3. The adversary sends its guess b', and wins the security game if $b = b'$.

Figure 1: No-Query Semantic Security Game

There exists a trivial adversary that, given a ciphertext, simply outputs a uniformly random bit, and will win the security game with probability $1/2$. Therefore, s must be at least $1/2$.

What if we want that for each and every efficient adversary, the probability of $\mathcal{A}$ winning the no-query-semantic-security game is at most $1/2$?

Definition Attempt 1. An encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ is secure if, for every efficient adversary $\mathcal{A}$, the probability of $\mathcal{A}$ winning the security game in Figure 1 is at most $1/2$. [3]

Unfortunately, the above definition is too strong – even the best encryption scheme will not satisfy this (if the keys are smaller than the messages).

Claim 4.1. Consider any encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ with key space $\mathcal{K} = \{0,1\}^n$ and message space $\{0,1\}^\ell$ where $\ell > n$. There exists a randomized polynomial time algorithm $\mathcal{A}$ that can win the No-Query-Semantic-Security security game with probability greater than $1/2$.

We will prove this claim in the next class.

Additional References

Relevant sections of the textbook [Boneh-Shoup]: Sections 2.1.3, 2.2.1 and 2.2.2.

[3] This probability is over the choice of bit b and key k (chosen by the challenger), as well as any randomness that the adversary uses.