1 Last Lecture

In the last lecture, we discussed that any encryption scheme $(\mathsf{Enc}, \mathsf{Dec})$ with key space $\mathcal{K}$ and message space $\mathcal{M}$ that satisfies perfect correctness and perfect secrecy (as per Definition 02.04 in Lecture 02) must have $|\mathcal{K}| \geq |\mathcal{M}|$.

Next, we discussed that a reasonable relaxation of the security definition is to require security against efficient adversaries, instead of requiring security against ALL adversaries. To define security against efficient adversaries, we introduce a *security parameter* — an integer n that indicates how secure the system will be. We introduced a special *key generation algorithm* KeyGen that takes the security parameter as input, and outputs the secret key. All algorithms (KeyGen, Enc and Dec) as well as the adversary, must have running time bounded by $\mathsf{poly}(n)$.

We then formulated security in terms of a *security game* between a challenger and an adversary. Security games in general, capture two things:

- what information is the adversary receiving?
- what is the adversary trying to achieve using this information?

Below, we define a security game for the attack scenario where the adversary receives a single ciphertext, and wants to learn some information about the underlying message.

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: The No-Query Semantic Security Game

Intuitively, if an encryption scheme is secure, then for any probabilistic[1] polynomial time algorithm $\mathcal{A}$, the probability of $\mathcal{A}$ winning the above security game should be small. How small do we want this probability to be? Clearly, if we require that this probability is too small (say close to 0), then even the best encryption scheme will not satisfy this definition. Consider an adversary that outputs a random bit b' in the final step of the security game. This adversary wins with probability $1/2$ (and the adversary is an efficient algorithm). Our security definition will along these lines, but we need to figure out the correct threshold.

2 An efficient adversary that wins with probability $p > 1/2$

Let $\mathcal{K} = \{0,1\}^n$ and $\mathcal{M} = \{0,1\}^\ell$, where $\ell > n$. We will now present an adversary that can win the security game with probability $p > 1/2$. The adversary sets message $m_0 = 0^\ell$, chooses m_1 uniformly at random, and sends them to the challenger. On receiving a ciphertext ct, the adversary picks a uniformly random key $k' \leftarrow \mathcal{K}$, and checks if $\mathsf{Dec}(\mathsf{ct}, k') = m_0$ or m_1. If the decryption is equal to either m_0 or m_1, then the adversary outputs 0 or 1 accordingly. If the decryption outputs neither m_0 nor m_1, then the adversary simply outputs a uniformly random bit b'.

[1] A probabilistic algorithm $\mathcal{A}$ takes the input x, samples a random string r and performs some computation that depends on x and r. In this course, all security games will involve prob. algorithms. Therefore, in the analysis, we should also take the algorithm's randomness into consideration.

In this security game (and most security games that we'll see from now on) the adversary can 'guess' the key used by the challenger, and this guess is correct with some non-zero probability.

The crucial issue here is that the adversary must somehow 'check' whether its guess is correct or not (in order to decide whether to go with the decryption, or to output a random bit). This checking makes the analysis a bit complicated. In our case, the adversary, in order to check if the guessed key is the same as the one chosen by the challenger, tries to decrypt the challenge ciphertext. If the decryption outputs neither m_0 nor m_1, then the adversary concludes that its guessed key is not same as the challenger's key, and it outputs a random bit. If the decryption was either m_0 or m_1, the adversary concludes that its guess is correct, and outputs 0 or 1 accordingly. Note that this conclusion may be wrong – maybe the challenge ciphertext was an encryption of m_0 using key k_0 (chosen by the challenger), but the adversary chose a key k_1 such that the challenge ciphertext is also an encryption of m_1 under k_1. In this case, the adversary will (wrongly) conclude that its guessed key is the correct key. However, we will see below that the adversary still wins the game with probability strictly greater than $1/2$ (albeit only slightly greater than $1/2$).

Claim 04.01. The success probability of the above described adversary is at least $\frac{1}{2} + \Omega(\frac{1}{2^n})$.

Proof. We present our analysis only for the case $\ell > 3n$. The analysis for the general case is left as an exercise.

Recall $m_0 = 0^\ell$. Fix the challenger's key k. We will prove that for any choice of key k,

$$\Pr\left[\mathcal{A} \text{ wins the } \mathsf{No\text{-}Query\text{-}Semantic\text{-}Security} \text{ game}\right] \;\geq\; 1/2 + \Omega(1/2^n)$$

where, the probability is taken over the random choices of: message m_1 (chosen by $\mathcal{A}$), bit b (chosen by the challenger), key k' (chosen by $\mathcal{A}$), and bit b' (chosen by $\mathcal{A}$). If the above inequality holds for all keys k, it will hold even if k was chosen at random.

For this proof, we consider the following bad event:

$\mathsf{BadEvent}_k$:= the randomly chosen message m_1 is such that there exists a key k' satisfying $\mathsf{Enc}(m_0, k) = \mathsf{Enc}(m_1, k')$ or $\mathsf{Enc}(m_0, k') = \mathsf{Enc}(m_1, k)$.

This event captures the 'bad event' where decrypting $\mathsf{Enc}(m_0, k)$ (using a different key k') can output message m_1, or when decrypting $\mathsf{Enc}(m_1, k)$ (using a different key k') can output message m_0. The probability of this bad event (where the probability is over the choice of m_1) is very small since $|\mathcal{K}| \ll |\mathcal{M}|$. In particular,

$$\Pr[\mathsf{BadEvent}_k] \;\leq\; 2|\mathcal{K}|/|\mathcal{M}| \;\leq\; 2^{-2n+1} \quad (why?)$$

Let us now compute the success probability of $\mathcal{A}$. We have,

$$\Pr\left[\mathcal{A} \text{ wins } \wedge \neg\mathsf{BadEvent}_k\right] = \Pr\left[\mathcal{A} \text{ wins } | \neg\mathsf{BadEvent}_k\right] \cdot \Pr\left[\neg\mathsf{BadEvent}_k\right] \tag{1}$$

Let us focus on $\Pr\left[\mathcal{A} \text{ wins } | \neg\mathsf{BadEvent}_k\right]$, since $\Pr\left[\neg\mathsf{BadEvent}_k\right]$ is close to 1. If the bad event does not happen, and the adversary's key k' is not equal to k, then the decryption cannot give a wrong output. This observation is used to simplify the probability of $\mathcal{A}$ winning, conditioned on $\neg\mathsf{BadEvent}_k$.

$$\Pr\left[\mathcal{A} \text{ wins } \mid \neg\mathsf{BadEvent}_k\right] = \Pr\left[\mathcal{A} \text{ wins } \mid k = k' \ \wedge \ \neg\mathsf{BadEvent}_k\right] \cdot \Pr\left[k = k' \mid \neg\mathsf{BadEvent}_k\right] + \Pr\left[\mathcal{A} \text{ wins } \mid k \neq k' \ \wedge \ \neg\mathsf{BadEvent}_k\right] \cdot \Pr\left[k \neq k' \mid \neg\mathsf{BadEvent}_k\right]$$

Here, note the following:

- $(*)$ $\Pr\left[\mathcal{A} \text{ wins } \mid k = k' \ \wedge \ \neg\mathsf{BadEvent}_k\right] = 1$ ($\mathcal{A}$ definitely wins if $k = k'$)
- $(*)$ $\Pr\left[\mathcal{A} \text{ wins } \mid k \neq k' \ \wedge \ \neg\mathsf{BadEvent}_k\right] \geq 1/2$ ($\mathcal{A}$ sends random guess b')
- $(*)$ $\Pr\left[k = k' \mid \neg\mathsf{BadEvent}_k\right] = 1/2^n$ (k' and m_1 are independently chosen)

Therefore, $\Pr\left[\mathcal{A} \text{ wins } \mid \neg\mathsf{BadEvent}_k\right] \geq 1/2^n + 1/2(1 - 1/2^n) = 1/2 + 1/2^{n+1}$.

Finally, we get that

$$\Pr\left[\mathcal{A} \text{ wins}\right] \ \geq \ \Pr\left[\mathcal{A} \text{ wins } \wedge \neg\mathsf{BadEvent}_k\right] \ \geq \ (1/2 + 1/2^{n+1})(1 - 1/2^{2n-1}) \ \geq \ 1/2 + \Omega(1/2^n).$$

□

3 Defining security against efficient adversaries

In the previous section we saw an efficient adversary that wins the game with probability slightly greater than $1/2$. However, note that $1/2^n$ (and more generally, $\mathsf{poly}(n)/2^n$) is a very small quantity, and therefore, the adversary's guess is almost like a uniformly random guess. For all practical purposes, we do not need to worry about such rare attacks. Consider, on the other hand, an adversary that can win the game with probability $1/2 + 1/\mathsf{poly}(n)$. Such adversaries can be dangerous in real world scenarios.

Therefore, for an encryption scheme to be secure, for any probabilistic polynomial time adversary, the probability of winning the security game must be at most $1/2 + f(n)$, where $f(n)$ is eventually smaller than any inverse polynomial function. We call such functions *negligible functions*.

Definition 04.01. A function $\mu : \mathbb{N} \to [0, 1]$ is said to be negligible if, for any polynomial $p(\cdot)$, there exists $n_0 \in \mathbb{N}$ such that for all $n > n_0$, $\mu(n) < 1/p(n)$.

Examples of negligible functions:
$\mu(n) = 1/2^{n/2}$, $\mu(n) = 1/n^{\log n}$, $\mu(n) = 1/n!$

An example of non-negligible function:

$$\mu(n) = \begin{cases} 1/n \text{ if n is prime} \\ 1/2^n \text{ otherwise} \end{cases}$$

4 A formal security definition for security against efficient adversaries

Given the notion of negligible functions, we are now ready to present our first security definition for efficient adversaries. [2]

Definition 04.02. An encryption scheme $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to satisfy no-query-semantic-security if, for any probabilistic polynomial time adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

$$\Pr\left[\mathcal{A} \text{ wins the } \mathsf{No\text{-}Query\text{-}Semantic\text{-}Security} \text{ game}\right] \leq 1/2 + \mu(n)$$

where the $\mathsf{No\text{-}Query\text{-}Semantic\text{-}Security}$ game is defined in Figure 1.

5 Security proofs

Armed with the above security definition, we are ready to see our first **security proof**. Suppose you are given an encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with $\mathcal{M} = \mathcal{C} = \{0,1\}^\ell$ that satisfies Definition 04.02. Consider the following modified encryption scheme $\mathcal{E}' = (\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$:

- $\mathsf{KeyGen}' = \mathsf{KeyGen}$
- $\mathsf{Enc}'(m,k) = \mathsf{reverse}\Big(\ \mathsf{Enc}(\mathsf{reverse}(m),k)\ \Big)$
- $\mathsf{Dec}'(\mathsf{ct},k) = \mathsf{reverse}\Big(\ \mathsf{Dec}(\mathsf{reverse}(\mathsf{ct}),k)\ \Big)$

We will show that $\mathcal{E}'$ also satisfies Definition 04.02, assuming $\mathcal{E}$ does. We will prove this statement by taking the contrapositive. Suppose, there exists a prob. poly. time adversary $\mathcal{A}$ that breaks security of $\mathcal{E}'$. Then we will show a prob. poly. time adversary $\mathcal{B}$ (that using $\mathcal{A}$) breaks the security of $\mathcal{E}$.

More formally, suppose there exists a non-negligible function $\epsilon_{\mathcal{A}}(\cdot)$ such that for infinitely many n,

$$\Pr\left[\mathcal{A} \text{ wins the } \mathsf{No\text{-}Query\text{-}Semantic\text{-}Security} \text{ game against } \mathcal{E}'\right] \geq 1/2 + \epsilon_{\mathcal{A}}(n).$$

Note: the probability is over the choice of bit b and key k chosen by the challenger, as well as any randomness that the adversary might use. Also, in the probability event, we specify the security game ($\mathsf{No\text{-}Query\text{-}Semantic\text{-}Security}$), as well as the encryption scheme $\mathcal{E}'$.

We will use $\mathcal{A}$ and the non-negl. function $\epsilon_{\mathcal{A}}$ to break the security of $\mathcal{E}$ as follows:

- build a prob. poly. time adversary $\mathcal{B}$ that uses $\mathcal{A}$ as a *black-box*.
- propose a non-negligible function $\epsilon_{\mathcal{B}}$ that depends on $\epsilon_{\mathcal{A}}$
- show that $\Pr\left[\mathcal{B} \text{ wins the } \mathsf{No\text{-}Query\text{-}Semantic\text{-}Security} \text{ game against } \mathcal{E}\right] \geq 1/2 + \epsilon_{\mathcal{B}}(n)$

[2] A common template for defining security via security games (in this course) is to first define a security game, and then identify the trivial winning strategy in this game. A construction is secure (w.r.t. this security game) if no prob. poly. time adversary can win the game with probability significantly larger than the trivial winning probability.

Reductions will be the bread-and-butter of this course, so please make sure you understand all the details here.

Description of Reduction Algorithm
The reduction algorithm $\mathcal{B}$ interacts with the adversary $\mathcal{A}$ and the challenger for $\mathcal{E}$.

1. First, $\mathcal{B}$ uses the adversary $\mathcal{A}$ to choose two messages m'_0, m'_1.
2. $\mathcal{B}$ sets $m_0 = \mathsf{reverse}(m'_0)$, $m_1 = \mathsf{reverse}(m'_1)$ and sends m_0, m_1 to the challenger for $\mathcal{E}$.
3. The challenger sends a ciphertext ct (which is either an encryption of m_0 or m_1).
4. $\mathcal{B}$ computes $\mathsf{ct}' = \mathsf{reverse}(\mathsf{ct})$ and sends it to $\mathcal{A}$.
5. $\mathcal{A}$ finally sends its guess b' to $\mathcal{B}$, and $\mathcal{B}$ forwards it to the challenger for $\mathcal{E}$.

Note that $\mathcal{B}$, together with the adversary $\mathcal{A}$, act as an adversary that wants to break the security of $\mathcal{E}$. From the viewpoint of the challenger for $\mathcal{E}$, it is interacting with an adversary that wants to win the game against $\mathcal{E}$.

Before proceeding with the formal proof, think about the following questions:

- what happens if $\mathcal{B}$ sets $m_0 = m'_0$, $m_1 = m'_1$ (instead of reversing the messages m'_0, m'_1)?
- what happens if $\mathcal{B}$ sets $\mathsf{ct}' = \mathsf{ct}$ (instead of reversing the ciphertext ct)?

Claim 04.02.

$$\Pr\Big[\mathcal{B} \text{ wins the No-Query-Semantic-Security game against } \mathcal{E}\Big] = \Pr\Big[\mathcal{A} \text{ wins the No-Query-Semantic-Security game against } \mathcal{E}'\Big].$$

Proof. Let us consider the LHS probability.

$$\Pr\Big[\mathcal{B} \text{ wins the No-Query-Semantic-Security game against } \mathcal{E}\Big]$$

This probability is over the choice of bit b, key k by the challenger, and the randomness of $\mathcal{A}$. Expanding the above probability, we get the following:

$$\Pr_{\substack{b\leftarrow\{0,1\},k\leftarrow\mathcal{K}\\ \text{rand. of } \mathcal{A}}}\left[\begin{array}{c}\mathcal{A} \text{ chooses } m'_0, m'_1\\ \mathsf{ct}' = \mathsf{reverse}\Big(\mathsf{Enc}(\mathsf{reverse}(m_b), k)\Big)\\ \mathcal{A}, \text{ on receiving } \mathsf{ct}' \text{ outputs } b\end{array}\right]$$

Now, note that this is identical to

$$\Pr_{\substack{b\leftarrow\{0,1\},k\leftarrow\mathcal{K}\\ \text{rand. of } \mathcal{A}}}\left[\begin{array}{c}\mathcal{A} \text{ chooses } m'_0, m'_1\\ \mathsf{ct}' = \mathsf{Enc}'(m_b, k)\\ \mathcal{A}, \text{ on receiving } \mathsf{ct}' \text{ outputs } b\end{array}\right]$$

and this is the probability that $\mathcal{A}$ wins the No-Query-Semantic-Security security game against $\mathcal{E}'$. □

6 Lecture summary and additional References

This lecture concludes our discussion related to perfect security, and provides an introduction to computational security. Key points to remember from Lectures 02-04:

- There are multiple ways of defining perfect secrecy. The simplest definition (Definition 02.04) is equivalent to the other definitions.
- If an encryption scheme satisfies perfect secrecy, then the key space must be as large as the message space (Theorem 03.01).
- Therefore, we relax the security requirements. Instead of seeking perfect secrecy, we aim for computational security: security against efficient adversaries. These security definitions are formalized using security games. (Definition 04.02)
- Security proofs via reductions are useful for proving security of a cryptographic construction, assuming the security of the underlying building blocks. We show that if there exists an adversary that breaks the construction, then there exists a reduction algorithm that breaks one of the underlying building blocks.

  Security reductions can also be used to compare security definitions (that is, whether one definition is stronger/weaker than the other).

Relevant sections from textbook [Boneh-Shoup]: 2.3.1, 2.3.2, 2.3.3.1 (this subsection discusses Question 04.03), 2.4.

7 Questions

Question 04.01. Consider the following function:

$$\mu(n) = 1/n^i \text{ for all } n \in [2^i, 2^{i+1}]$$

Is this function negligible?

Hint: The function is closely related to $1/n^{\log n}$, which is a negligible function.

Question 04.02. Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme that satisfies Definition 04.02. Consider the following modified encryption scheme $(\mathsf{KeyGen}', \mathsf{Enc}', \mathsf{Dec}')$:

- KeyGen' = Run KeyGen twice, and let k_1, k_2 be the two output keys. Output $k = (k_1, k_2)$ as the key.
- $\mathsf{Enc}'(m, k = (k_1, k_2)) = (\mathsf{Enc}(m, k_1), \mathsf{Enc}(m, k_2))$
- $\mathsf{Dec}'(\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2), k = (k_1, k_2)) = \mathsf{Dec}(\mathsf{ct}_1, k_1)$

Show that $\mathcal{E}'$ satisfies Definition 04.02, assuming $\mathcal{E}$ does.

Hint: This question is discussed in one of the following lectures.

Question 04.03. Security proofs can also be used for studying the relationship between various security definitions. Let $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ be an encryption scheme with message space $\mathcal{M}$.

1. Propose a security game (and a security definition) to capture the following 'intuitive definition': an encryption scheme is secure against message recovery attacks if no adversary can recover the message, given the encryption of a uniformly random message using a uniformly random key.

   Hint: there exist prob. poly. time adveraries that can win the above game with probability $1/|\mathcal{M}|$.

2. Show that if an encryption scheme $\mathcal{E}$ satisfies No-Query-Semantic-Security, then it also satisfies security against message recovery attacks.

   Hint: The reduction algorithm interacts with the No-Query-Semantic-Security challenger. It picks two uniformly random messages m_0, m_1 and sends them to the challenger. On receiving the challenge ciphertext ct, it sends ct to the adversary. The adversary sends its 'guess message' m. If $m = m_0$, the reduction guesses 0, else it guesses 1.

Question 04.04 (*). Can you modify the above definition to allow the adversary to choose a message distribution that it wants to attack? Would the same reduction work?

Hint: No, this will not work if the adversary is allowed to choose a message distribution. If the adversary sends a distribution that outputs 0^n with probability 1, then the adversary 'knows' that the challenge ciphertext is an encryption of 0^n.