1 Last lecture, and plan for today's lecture

Last lecture, we first defined semantic security. Why should we use this as our definition for security against 'read-only' adversaries? Because Goldwasser and Micali proved that this is the 'best' security definition against read-only adversaries (they gave a formal argument for this, but that is beyond the scope of our course).

Next, we observed that **no deterministic scheme can satisfy semantic security**. Therefore, our encryption scheme needs to be randomized.

Then, we constructed an encryption scheme with message space $\mathcal{M} = \{0,1\}^{\ell_{\text{msg}}}$, and randomized encryption algorithm. This encryption scheme uses a pseudorandom function $F : \mathcal{X} \times \mathcal{K} \to \mathcal{M}$. Of course, we need a formal proof of security to be convinced of security. This is the main focus of today's lecture.

2 Proof of security

We will prove the following theorem. Let us set up some notations first. Let $\mathcal{E}$ denote our construction from last class. For any adversary $\mathcal{A}$, let $\mathsf{adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}$ denote $p_0 - p_1$, where p_b is the probability that adversary $\mathcal{A}$ outputs 0 in world-b when playing the $\mathsf{semantic\ security}$ game against $\mathcal{E}$.[1] Note that this will be a function of the security parameter.

Theorem 16.01. Suppose $F : \mathcal{X} \times \mathcal{K} \to \mathcal{M}$ is a secure pseudorandom function, and suppose $|\mathcal{X}|$ is superpolynomial in the security parameter. Then, for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n, $\mathsf{adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}(n) \leq \mu(n)$.

Proof intuition: The proof intuition comes from the 'toy scenario', or more formally, Observation 15.01 from Lecture 15. Currently, the challenger sends $(x_i, m_{i,b} \oplus F(x_i,k))$ for each i. If we could replace $m_{i,b} \oplus F(x_i,k)$ with a completely random string (and for each query the challenger chooses the random string independently) then the adversary would have no idea whether $b = 0$ or $b = 1$. In order to get from $m_{i,b} \oplus F(x_i,k)$ to a random string, we will need a sequence of hybrids.

Proof. Our proof will proceed via a sequence of hybrids, where world-0 and world-1 correspond to the two-world formulation.

- **World-0**
  - **Setup:** Challenger chooses PRF key $k \leftarrow \mathcal{K}$.
  - **Queries:** Adversary sends polynomially many queries. The i^{th} query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$, computes $\mathsf{ct}_i = F(x_i,k) \oplus m_{i,0}$ and sends (x_i, ct_i) to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.
- **Hybrid-1**
  - **Setup:** Challenger chooses uniformly random function $f \leftarrow \mathsf{Func}[\mathcal{X}, \mathcal{M}]$.
  - **Queries:** Adversary sends polynomially many queries. The i^{th} query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$, computes $\mathsf{ct}_i = f(x_i) \oplus m_{i,0}$ and sends (x_i, ct_i) to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.
- **Hybrid-2**
  - **Setup:** Challenger chooses nothing during setup. It initializes an empty table $\mathcal{T} = \{\}$.

[1] Here, $\mathsf{adv}^{\mathsf{ss}}_{\mathcal{A},\mathcal{E}}$ denotes the 'advantage' that $\mathcal{A}$ has in world-0, as compared to world-1.

- **Queries:** Adversary sends polynomially many queries. The i^{th} query consists of two messages $(m_{i,0}, m_{i,1})$.
  Challenger chooses $x_i \leftarrow \mathcal{X}$. If $\exists y_i$ s.t. $(x_i, y_i) \in \mathcal{T}$, it sends $(x_i, y_i \oplus m_{i,0})$ to the adversary.
  If $(x_i, y) \notin \mathcal{T}$ for all y, it chooses random $y_i \leftarrow \mathcal{M}$ and adds (x_i, y_i) to $\mathcal{T}$. It sends $(x_i, y_i \oplus m_{i,0})$ to the adversary.
- **Guess:** After polynomially many queries, the adversary sends its guess b'.

- **Hybrid-3**
  - **Setup:** Challenger chooses nothing during setup. It does not initialize an empty table $\mathcal{T} = \{\}$.
  - **Queries:** Adversary sends polynomially many queries. The i^{th} query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$ without replacement.
    It chooses random $y_i \leftarrow \mathcal{M}$ and sends $(x_i, y_i \oplus m_{i,0})$ to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.

- **Hybrid-4**
  - **Setup:** Challenger chooses nothing during setup. It does not initialize an empty table $\mathcal{T} = \{\}$.
  - **Queries:** Adversary sends polynomially many queries. Each query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$ without replacement.
    It chooses random $y_i \leftarrow \mathcal{M}$ and sends $(x, y_i \oplus m_{i,1})$ to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.

- **Hybrid-5**
  - **Setup:** Challenger chooses nothing during setup. It initializes an empty table $\mathcal{T} = \{\}$.
  - **Queries:** Adversary sends polynomially many queries. Each query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$. If $\exists y_i$ s.t. $(x_i, y_i) \in \mathcal{T}$, it sends $(x_i, y_i \oplus m_{i,1})$ to the adversary.
    If $(x_i, y_i) \notin \mathcal{T}$ for all y_i, it chooses random $y_i \leftarrow \mathcal{M}$ and adds (x_i, y_i) to $\mathcal{T}$. It sends $(x_i, y_i \oplus m_{i,1})$ to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.

- **Hybrid-6**
  - **Setup:** Challenger chooses uniformly random function $f \leftarrow \mathsf{Func}[\mathcal{X}, \mathcal{M}]$.
  - **Queries:** Adversary sends polynomially many queries. Each query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$, computes $\mathsf{ct}_1 = f(x_i) \oplus m_{i,1}$ and sends (x_i, ct_1) to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.

- **World-1**
  - **Setup:** Challenger chooses uniformly random key $k \leftarrow \mathcal{K}$.
  - **Queries:** Adversary sends polynomially many queries. Each query consists of two messages $(m_{i,0}, m_{i,1})$.
    Challenger chooses $x_i \leftarrow \mathcal{X}$, computes $\mathsf{ct}_1 = F(x_i, k) \oplus m_{i,1}$ and sends (x_i, ct_1) to the adversary.
  - **Guess:** After polynomially many queries, the adversary sends its guess b'.

Summary of hybrids and proof idea: The number of hybrids here can be a bit intimidating. However, there's not much happening in these hybrids. The purpose of hybrid-1 and hybrid-6 is to replace the pseudorandom function with a truly random function. This makes the hybrids similar to our 'toy-scenario'. Next, we are simply using the definition of a random function. If a function is completely random, then the value at one input does not influence the value at another input. As a result, the random function can be chosen 'on-the-fly'. This is what happens in hybrid-2 and hybrid-5.

Unfortunately, we are not done yet. In hybrid-2 and hybrid-5, if x is repeated, then the adversary can learn the XOR of the corresponding messages. But luckily, $\mathcal{X}$ is superpolynomial in size, and therefore sampling with and without replacement are indistinguishable (thanks to the 'birthday bound'). Therefore, in hybrid-3 and hybrid-4, we can sample the x values without replacement.

Analysis: Let p_0 and p_1 denote the probability that $\mathcal{A}$ outputs 0 in world-0 and world-1 respectively. Let $p_{\text{hyb},i}$ denote the probability of outputting 0 in hybrid-i.

Claim 16.01. Assuming F is a secure PRF, $p_0 \approx p_{\text{hyb},1}$. [2]

Proof. The only difference between World-0 and Hybrid-1 is that the challenger picks a uniformly random key k and uses $F(\cdot, k)$ in World-0, and uses a uniformly random function $f(\cdot)$ in Hybrid-1. Therefore, if there exists a p.p.t. adversary $\mathcal{A}$ that breaks the semantic security of $\mathcal{E}$, then there exists a reduction algorithm $\mathcal{B}$ that breaks PRF security.

For each query $(m_{i,0}, m_{i,1})$ received from $\mathcal{A}$, the reduction algorithm chooses a uniformly random $x_i \leftarrow \mathcal{X}$ and sends x_i to the PRF challenger. The challenger returns y_i, and the reduction algorithm sends $(x, y \oplus m_{i,0})$ to the adversary $\mathcal{A}$.

Finally, after polynomially many queries, the adversary sends its guess b', which the reduction algorithm forwards to the challenger.

Check that the probability of $\mathcal{B}$ winning the PRF game is exactly $1/2 + (p_0 - p_{\text{hyb},1})/2$. □

Claim 16.02. For any adversary $\mathcal{A}$, $p_{\text{hyb},1} = p_{\text{hyb},2}$.

Proof. This follows from the definition of a random function. In one case, the challenger chooses the entire truth table at once, while in the other case, it chooses the random function 'on-the-fly'. □

Claim 16.03. For any adversary $\mathcal{A}$ that makes q queries, $p_{\text{hyb},2} - p_{\text{hyb},3} \leq q^2/|\mathcal{X}|$.

Proof. The only difference in the two hybrids happens if some x is sampled twice. Using the birthday bound, we know that this happens with probability at most $q^2/|\mathcal{X}|$, where q is the number of queries made by the adversary. Since the adversary is a polynomial time adversary, q is also polynomial in the security parameter. Finally, note that $|\mathcal{X}|$ is superpolynomial in the security parameter, and therefore $q^2/|\mathcal{X}|$ is negligible in the security parameter.

□

Claim 16.04. For any adversary $\mathcal{A}$, $p_{\text{hyb},3} = p_{\text{hyb},4}$.

Proof. This part relies on Observation 15.01 from Lecture 15 (Shannon OTP satisfies semantic security if different random key is used for each query).

Suppose there exists an adversary $\mathcal{A}$ such that $p_{\text{hyb},3} > p_{\text{hyb},4}$. We will show that this implies the existence of an algorithm $\mathcal{B}$ that wins the Semantic-Security-Multiple-Keys game with Shannon's OTP scheme with probability $1/2 + (p_{\text{hyb},3} - p_{\text{hyb},4})/2$ (thereby contradicting Observation 15.01).

[2] For ease of readability, I am skipping the usual *"for any p.p.t. $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that ... $p_0 - p_{\text{hyb},1} \leq \mu(n)$"*, and instead using the $\approx$ notation to denote the same.

For each query $(m_{i,0}, m_{i,1})$ received from $\mathcal{A}$, $\mathcal{B}$ sends the query to Semantic-Security-Multiple-Keys challenger. It receives ct_i. It then samples $x_i \leftarrow \mathcal{X}$ **without replacement** and sends (x_i, ct_i) to $\mathcal{A}$. Finally, after all the queries, the adversary sends a bit b', which the reduction forwards to the challenger.

Check the following:
if challenger chose $b = 0$, then the adversary is getting exactly what it would get in Hybrid-3, else it is participating in Hybrid-4. Therefore, the probability of $\mathcal{B}$ winning the Semantic-Security-Multiple-Keys game is $1/2 + (p_{\text{hyb},3} - p_{\text{hyb},4})/2$.

□

Claim 16.05. For any adversary $\mathcal{A}$, $p_{\text{hyb},4} - p_{\text{hyb},5} \leq q^2/|\mathcal{X}|$.

Same as proof of Claim 16.03, follows from the birthday bound.

Claim 16.06. For any adversary $\mathcal{A}$, $p_{\text{hyb},5} = p_{\text{hyb},6}$.

Same as proof of Claim 16.02.

Claim 16.07. Assuming F is a secure PRF, $p_{\text{hyb},6} \approx p_1$.

Same as proof of Claim 16.01. The reduction will be slightly different here. On receiving y_i for the i^{th} query to the challenger, it sends $(x_i, y_i \oplus m_{i,1})$ to the adversary.

Putting together all the above hybrids, we get our theorem.

□

2.1 Key takeaways from this construction/proof

- We saw that if $F : \mathcal{X} \times \mathcal{K} \to \mathcal{M}$ is a secure PRF, and $|\mathcal{X}|$ is superpolynomial in size, then we have an encryption scheme that is secure against 'read-only' adversaries.
- If F is an insecure PRF, then we might have an actual attack. For instance, suppose F is a really bad PRF whose output always ends with a 0. Then you can choose two messages m_0, m_1 that differ in their last bit, and that'll be a distinguishing attack.

  See Question 16.01 below, it describes another insecure PRF. Find an attack against the resulting encryption scheme.
- There is another potential vulnerability: if $|\mathcal{X}|$ is not large enough. Note that the 'birthday bound' also gives a lower bound for the probability of repetitions. If q queries are made, then with probability $\Omega(q^2/|\mathcal{X}|)$, there will be two indices i, j such that $x_i = x_j$.

  This leads to the following attack: choose $\sqrt{|\mathcal{X}|}$ pairs of messages $\{(m_{i,0}, m_{i,1})\}_i$ such that for all $i \neq j$, $m_{i,0} \oplus m_{j,0} \neq m_{i,1} \oplus m_{j,1}$. We know that, with decent probability, there will exist two indices $i \neq j$ such that $x_i = x_j = x^*$. As a result, the i^{th} (resp. j^{th}) ciphertext is $(x^*, F(x^*, k) \oplus m_{i,b})$ (resp. $(x^*, F(x^*, k) \oplus m_{j,b})$). The adversary can therefore figure out the bit b (by XORing the second component of the i^{th} and j^{th} ciphertexts).

  Such attacks have been implemented in practice. The reason is that in practice, we tend to focus on the PRF key size being reasonably large, but ignore the domain size. The 3DES is one such PRP that is widely used in practice. It has $\mathcal{X} = \{0,1\}^{64}$ and $\mathcal{K} = \{0,1\}^{168}$. As a result, if this is used for encryption, then only 2^{32} queries are enough for breaking semantic security.

3 Handling unbounded length messages

In the above construction, we fixed our message space to be $\{0,1\}^{\ell_{\text{msg}}}$, and used a PRF with co-domain $\{0,1\}^{\ell_{\text{msg}}}$. In practice, we are given a PRF with fixed co-domain (for instance, in the case of AES, the co-domain is $\{0,1\}^{128}$). How can we handle unbounded length messages?

Suppose we fix our message space to be $\{0,1\}^*$, and our encryption algorithm should handle messages of any length. The first thing to note is that once the encryption scheme is fixed, there will exist different message with different length ciphertexts. As a result, if we use the semantic security definition discussed in last class, then the adversary can always win. **The ciphertext size leaks the size of underlying message.**

As a result, we need to modify the security game, and force the adversary to send equal length messages for each query. The updated security game is as follows.

Semantic Security

1. **Setup:** Challenger chooses an encryption key $k \leftarrow \mathcal{K}$ and a bit $b \leftarrow \{0,1\}$.
2. **Challenge encryption queries:** Adversary sends polynomially many challenge encryption queries (adaptively). The i^{th} challenge pair consists of two messages $m_{i,0}, m_{i,1}$ such that $|m_{i,0}| = |m_{i,1}|$. The challenger sends $\mathsf{ct}_i = \mathsf{Enc}(m_{i,b}, k)$ to the adversary.
   Note that the bit b and key k were chosen during setup, and the same bit and key are used for all queries.
3. **Guess:** The adversary sends its guess b', and wins the security game if $b = b'$.

Figure 1: Semantic Security Game

Here are a few approaches discussed in class:

3.1 Attempts discussed in class

1. *Break message into blocks, encrypt each block separately using fresh randomness, but same key*

   This approach is provably secure. In practice, this approach is sometimes avoided, since it requires sampling fresh randomness for each block.

   Note that we could not do the same if the scheme satisfied only No-Query-Semantic-Security. Why is it so?

2. *Instead of Shannon OTP, use some other encryption scheme that supports unbounded length messages with short keys*

   We will pick one random $x \leftarrow \mathcal{X}$, then compute an encryption key $k' = F(x,k)$, and use this key for encrypting the unbounded length message.

   For instance, recall the deterministic counter mode encryption that we saw in Lecture 11 (Section 2).

   Check that the resulting ciphertext for encryption of $m = m_1 \,||\, \ldots \,||\, m_t$ will be

   $$\Big(x, m_1 \oplus F(1,k') \,||\, \ldots \,||\, m_t \oplus F(t,k')\Big) \text{ where } k' = F(x,k).$$

4 Lecture summary, plan for next lecture, additional resources

Summary: In this lecture, we discussed the proof of security for our encryption scheme. The construction uses a keyed function F mapping some input domain $\mathcal{X}$ to the message space $\mathcal{M}$. The proof of security

relied on the PRF security of F, as well as the domain $\mathcal{X}$ being sufficiently large. If either of these two does not hold, then we have an attack on the encryption scheme.

Next, we discussed how to handle messages of arbitrary length. Two of the approaches discussed in class:

- break the message into small chunks, and encrypt each chunk separately using the same key
- combine F with an encryption scheme that handles unbounded length messages and offers No-Query-Semantic-Security

Next lecture: This lecture concludes our discussion of encryption schemes secure against 'read-only' adversaries. We will start with a summary of this topic, and then start discussing security against stronger adversaries.

Relevant sections from textbook [Boneh-Shoup]: Our proof in Section 2 is a simplification of the one given in Section 5.4.1.

5 Questions

Question 16.01. Suppose $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ satisfies the following property: after seeing n evaluations of the PRF on random inputs $\{x_i, F(x_i, k)\}$, it is possible to recover the key k. Show that if F is used to construct an encryption scheme, then the resulting encryption scheme is not semantically secure. Describe your attack.

Solution sketch: Send n queries with $m_{i,0} = m_{i,1} = 0^n$. This will give n pairs $\{x_i, F(x_i, k)\}_i$. Recover k from the queries, then send two distinct messages (m_0, m_1). The challenger sends $(x, m_b \oplus F(k, x))$. Since we know k, we can recover m_b.

Question 16.02. Suppose $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ is a secure PRF, and $\mathcal{E}_1 = (\mathsf{Enc}_1, \mathsf{Dec}_1)$ is an encryption scheme with message space $\{0,1\}^*$, key space $\{0,1\}^n$ and satisfies No-Query-Semantic-Security. Show that the following encryption scheme, obtained by combining F and $\mathcal{E}_1$, satisfies Semantic Security.

- KeyGen: choose a PRF key $k \leftarrow \{0,1\}^n$.
- $\mathsf{Enc}(m, k)$: Choose $x \leftarrow \{0,1\}^n$, compute $k' = F(x, k)$ and output $(x, \mathsf{Enc}_1(m, k'))$.
- $\mathsf{Dec}((\mathsf{ct}_1, \mathsf{ct}_2), k)$: Compute $k' = F(\mathsf{ct}_1, k)$ and output $\mathsf{Dec}_1(\mathsf{ct}_2, k')$.

Show that $\mathcal{E}$ satisfies Semantic Security, assuming F is a secure PRF and $\mathcal{E}_1$ satisfies No-Query-Semantic-Security. The proof will go through a sequence of hybrids, which will be very similar to our hybrids in Section 2. The only part that needs to change is Claim 16.04 (you will need to use Observation 15.02 from Lecture 15).

Solution sketch: The detailed proof is described in Section 5.4.1 of the textbook. Using the security of the underlying encryption scheme $\mathcal{E}_1$, we can conclude that $p_{\text{hyb},3} - p_{\text{hyb},4} \leq \mathsf{negl}$ (use Observation 15.02 from Lecture 15).

Question (**). In Lecture 11, we saw a few other encryption schemes satisfying No-Query-Semantic-Security (in particular, OFB mode and CFB mode). Propose randomized variants of the same which satisfy Semantic Security.

Solution sketch: The textbook contains randomized variants of counter-mode (Section 5.4.2) and CBC mode (Section 5.4.3). You can refer to the detailed security proof in the textbook. For the OFB mode, the construction and proof will be similar to the construction/proof of counter mode (described in Section 5.4.2).