1 Last lecture, and plan for today's lecture

Last lecture, we saw a few security games. One of them (Attempt 3) is the 'gold-standard' definition for security against 'read-only' attacks. Today, we will spend some time with this definition, and then see a construction that achieves this definition.

2 Semantic security game with multiple keys

One of the security games proposed in the last class (Attempt 1, Section 3.1 in Lecture 14) involved multiple encryption keys, one for each challenge query. Let us recall that security game.

Semantic-Security-Multiple-Keys

1. **Setup:** Challenger chooses 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 $m_{i,0}, m_{i,1}$.
   The challenger chooses $k_i \leftarrow \mathcal{K}$ and sends $\mathsf{ct}_i = \mathsf{Enc}(m_{i,b}, k_i)$ to the adversary.
   Note that the bit b was chosen during setup, and the same bit is 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 with multiple keys

As mentioned in class, the definition based on this game is equivalent to the one based on No-Query-Semantic-Security game. The proof goes via a hybrid argument, which is summarised in Figure 2.

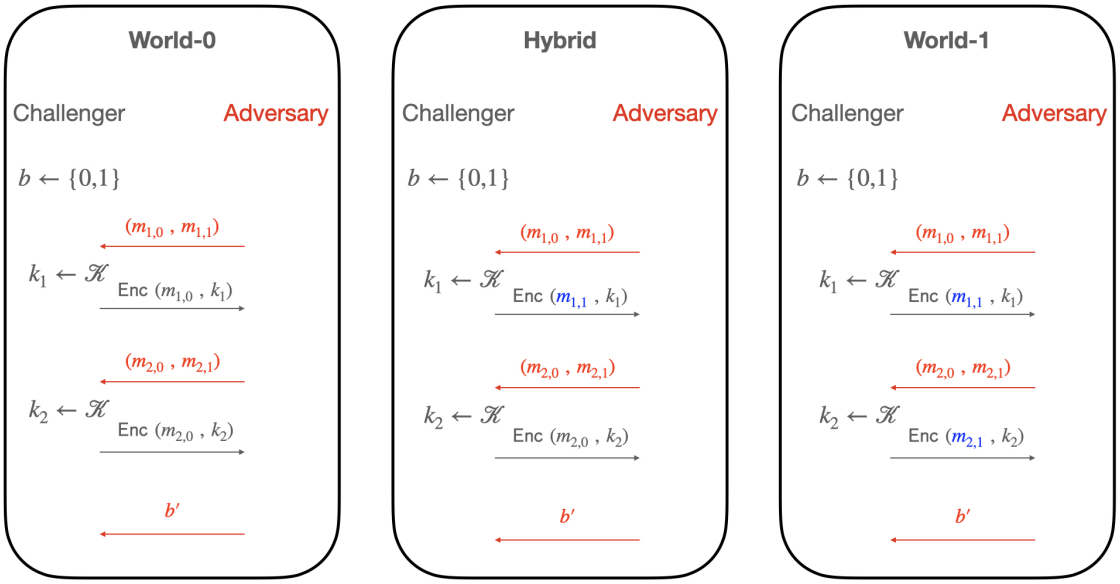

Figure 2: Proof sketch for showing that Semantic-Security-Multiple-Keys game is equivalent to No-Query-Semantic-Security game. If there exists an adversary that can distinguish between world-0 and hybrid-world, then there exists a reduction algorithm that breaks No-Query-Semantic-Security. Similarly, if there exists an adversary that can distinguish between hybrid-world and world-1, then there exists a reduction algorithm that breaks No-Query-Semantic-Security.

A formal proof should contain a reduction, showing that if the probabilities in world-0 and hybrid world are far apart, then there exists a reduction algorithm that breaks the No-Query-Semantic-Security. Similarly, if the probabilities in hybrid world and world-1 are far apart, then there exists a reduction algorithm that breaks the No-Query-Semantic-Security. I am including one of the reductions here, the second one is pretty similar.

Claim 15.01. Suppose there exists a p.p.t adversary $\mathcal{A}$ such that $p_0 - p_{\text{hyb}} = \epsilon$. Then there exists p.p.t. reduction algorithm $\mathcal{B}$ that breaks the No-Query-Semantic-Security.

Proof. The reduction algorithm receives the first query $(m_{1,0}, m_{1,1})$ and sends this to the No-Query-Semantic-Security challenger. It receives ct_1 in response, which it sends to the adversary.

On receiving the second query $(m_{2,0}, m_{2,1})$, the reduction algorithm chooses a uniformly random key $k_2 \leftarrow \mathcal{K}$, computes $\mathsf{ct}_2 = \mathsf{Enc}(m_{2,0}, k_2)$ and sends ct_2 to the adversary.

Finally, the adversary sends its guess b', which the reduction forwards to the challenger.

Analysis: If the challenger encrypts $m_{1,0}$, then $\mathcal{A}$ is participating in world-0. Otherwise $\mathcal{A}$ is participating in the hybrid world. As a result,

$$
\begin{aligned}
&\Pr\left[\mathcal{B} \text{ wins }\right] \\
&= \Pr\left[\mathcal{B} \text{ sends } 0 \wedge \text{ Challenger chose } 0\right] + \Pr\left[\mathcal{B} \text{ sends } 1 \wedge \text{ Challenger chose1}\right] \\
&= \frac{1}{2}\Pr\left[\mathcal{B} \text{ sends } 0 \mid \text{ Challenger chose } 0\right] + \frac{1}{2}\Pr\left[\mathcal{B} \text{ sends } 1 \mid \text{ Challenger chose1}\right] \\
&= \frac{1}{2}\Pr\left[\mathcal{B} \text{ sends } 0 \mid \text{ Challenger chose } 0\right] + \frac{1}{2}\left(1 - \Pr\left[\mathcal{B} \text{ sends } 0 \mid \text{ Challenger chose1}\right]\right) \\
&= \frac{1}{2}\Pr\left[\mathcal{A} \text{ sends } 0 \text{ in world-0}\right] + \frac{1}{2}\left(1 - \Pr\left[\mathcal{A} \text{ sends } 0 \text{ in Hybrid-world}\right]\right) \\
&= \frac{1}{2} + \frac{1}{2}(p_0 - p_{\text{hyb}})
\end{aligned}
$$

□

The above sketch gives us the following observations.

Observation 15.01. Let $\mathcal{E}$ denote the Shannon's One-Time-Pad encryption scheme. Then, for any adversary $\mathcal{A}$,

$$\Pr\left[\mathcal{A} \text{ wins the Semantic-Security-Multiple-Keys game }\right] = 1/2.$$

This follows from the fact that Shannon's OTP is perfectly secure.

Observation 15.02. Let $\mathcal{E}$ be any encryption scheme satisfying No-Query-Semantic-Security. Then, for any adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

$$\Pr\left[\mathcal{A} \text{ wins the Semantic-Security-Multiple-Keys game }\right] \leq 1/2 + \mu(n).$$

Main takeaway from this section: This security definition is not good enough when the adversary has access to multiple ciphertexts. For instance, Shannon's OTP will satisfy this definition, but we know that Shannon's OTP is not secure if the adversary sees two ciphertexts. This should also make sense intuitively, since the challenger is choosing a fresh key for each query. Instead, the challenger should choose one key, and use it for all the queries.

Qn: The number of hybrids in this proof depend on the number of queries made by the adversary. How can we assume that we know the number of queries made by $\mathcal{A}$?

Ans: This is an excellent question. For this course, you can assume that, given an adversary, we know the number of queries made by the adversary. Here's a short explanation why that's the case (discuss with me offline if this explanation doesn't make sense). We only require an upper bound on the number of queries made by the adversary. Therefore, the reduction can 'guess' i s.t. that the number of queries is between 2^i and 2^{i+1}. Of course, all i's are not equally likely. The probabilities of guesing i should degrade as i increases.

3 The security game we will use: Semantic security game

We will use the following security game, which is the simplest one to use, and at the same time, capture all these 'read-only' attacks. This was proposed by Shafi Goldwasser and Silvio Micali in 1984. They also proved that this is the best definition to use, as far as 'read-only' attacks are concerned. However, this proof is beyond the scope of this course. We will 'assume' that this is the best definition for security against 'read-only' adversaries.

The security game for this definition is described in Figure 3. In this game, the challenger chooses a key k, and uses the same key for either encrypting $m_{i,0}$ or $m_{i,1}$. The adversary gets to send polynomially many queries, and at the end, it must send its guess.

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 $m_{i,0}, m_{i,1}$. 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 3: Semantic Security Game

Since all our encryption algorithms have been deterministic (so far), there exist adversaries that can win the semantic security game with probability 1. The adversary can first send $m_{1,0} = m_{1,1} = m_1$. It receives a ciphertext ct_1. Next, it queries for $m_{2,0} = m_1$ and $m_{2,1} \neq m_1$. Since the encryption scheme is deterministic, if the second ciphertext ct_2 is equal to ct_1, then the adversary can conclude that $m_{2,0}$ was encrypted. Else, the adversary guesses that $m_{2,1}$ was encrypted.

Therefore, in order to satisfy security against all 'read-only' adversaries, the encryption algorithm must also be randomized.

Qn: Does this mean that there is no encryption scheme with deterministic encryption, and also satisfying semantic security? Ans: Indeed, we cannot have a semantically secure encryption scheme with deterministic encryption. For deterministic encryption, the 'gold-standard' security definition looks quite different (and is weaker than semantic security).

Qn: Where does the randomness for encryption and key generation come from?

Ans: We will assume that these algorithms have some source of randomness. Without randomness, we cannot have any cryptography. What is the exact source of randomness - this is beyond the scope of this course, talk to me offline.

Given the security game, the security definition is very similar to our previous security definitions.

Definition 15.01. An encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ is said to satisfy Semantic Security if, for any p.p.t. adversaries $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ such that for all n,

$$\Pr\left[\mathcal{A} \text{ wins the semantic security game}\right] \leq 1/2 + \mu(n)$$

where the probability is over the choice of key k, randomness used in Enc, and the adversary's randomness.

Qn: The security definition in Section 2 is weaker than the semantic security definition. What about the relation between Attempt 2 (from last lecture) and semantic security?
Ans: These two are incomparable, and neither implies the other.

4 Secure PRFs $\implies$ Semantically secure encryption

Before we see a formal encryption scheme that satisfies the above definition, let us build our intuition with 'randomized encryption schemes'. If the encryption scheme is randomized, are we guaranteed perfect correctness? Let us consider the following 'toy scenario'.

Toy scenario: Alice and Bob want to communicate securely over an insecure channel. They both share a huge database of random strings $y_1, y_2, \ldots$. Assume all strings have the same length, and the length is **exactly equal to the length of messages** that Alice/Bob want to communicate. The adversary does not have access to this database of strings, and can only see the communication between Alice and Bob. Can Alice and Bob securely communicate?

Attempts discussed in class:

1. Alice/Bob use the first string y_1 for the first message, use y_2 for the second message and so on. This approach works, but Alice and Bob need to keep track of the 'order' of ciphertexts. They also need to make sure that they don't use the same string.

2. Suppose Alice wants to send message m to Bob. She first converts m to a decimal number i (that is, the binary representation of i is equal to m). She then sends $y_i \oplus m$. There are a couple of issues here: first of all, if Alice sends the same ciphertext twice, then the adversary knows that the same message was encrypted in both ciphertexts. Second, how does Bob decrypt this?

3. Suppose Alice wants to send a message m to Bob. She picks a random index i, and sends $(i, y_i \oplus m)$ to Bob. Alice needs to communicate i, otherwise Bob does not know which string to use for recovering the message. Similarly, note that Alice **should not** send the string y_i, otherwise the adversary can learn the message.

   Caution: if Alice/Bob use the same string for two different messages, then the adversary can learn the XOR of the underlying messages. For instance, suppose Alice uses the same index i for encrypting messages m and m'. As a result, the adversary sees $(i, y_i \oplus m)$ and $(i, y_i \oplus m')$. The adversary, given these two ciphertexts, can compute $m \oplus m'$.

   However, if the index is chosen at random, and the set of strings is very large, then with high probability, Alice/Bob will not use the same string.

Attempt 3 seems to be a reasonable scheme that's secure against read-only attacks. However, it requires Alice and Bob to share a huge database of random strings. Luckily, we have an efficient replacement to huge random databases: **pseudorandom functions/permutations**! Instead of sharing a huge database

of random strigs, Alice and Bob can share a PRF key k. The evaluation $F(i,k)$ looks like the i^{th} row of our uniformly random database.

4.1 Formal construction 1: PRFs with large enough co-domain $\implies$ semantically secure encryption schemes

Let $\mathcal{M} = \{0,1\}^{\ell_{\text{msg}}}$ be the message space for which we want to construct a secure encryption scheme. Let $F : \mathcal{X} \times \mathcal{K} \to \mathcal{M}$ be a secure pseudorandom function. Consider the following encryption scheme $\mathcal{E} = (\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$:

- KeyGen: The key generation algorithm chooses a PRF key $k \leftarrow \mathcal{K}$.
- $\mathsf{Enc}(m,k)$: Choose a uniformly random $x \leftarrow \mathcal{X}$, compute $y = F(x,k)$. Output $(x, y \oplus m)$.
- $\mathsf{Dec}(\mathsf{ct} = (\mathsf{ct}_1, \mathsf{ct}_2), k)$: Output $\mathsf{ct}_2 \oplus F(\mathsf{ct}_1, k)$.

Qn: What does it mean for the key generation algorithm to choose a PRF key $k \leftarrow \mathcal{K}$?
Ans: Suppose $\mathcal{K} = \{0,1\}^n$. The above means that the key generation algorithm picks a uniformly random n bit string, and sets that as the secret key for our scheme.

Clearly, the above scheme satisfies correctness. We want to show that this scheme satisfies semantic security (Definition 15.01). Before seeing the formal proof (we will discuss it in the next lecture), let us intuitively understand what are the main ideas being used here.

- First, we will rely on PRF security. If Alice and Bob use the huge database of random strings, then the communication seemed secure in the 'toy scenario'.
- Second, it is necessary that the string x_i does not repeat for different queries. If the set $\mathcal{X}$ is small, then there is a decent chance that the sampled x_i will repeat. However, if $\mathcal{X}$ is much larger than the number of adverarial encryption queries, then with high probability, the strings x_i will not repeat.
- Finally, we crucially used the security of Shannon's OTP.

There is one last thing to address: the PRF's output space needs to be as large as the message space. Recall, from a theoretical as well as practical perspective, this may be a problem. Maybe we have a PRF that has a fixed output space $\{0,1\}^n$ (as is the case for AES).

Luckily, we have already resolved this issue. Last lecture, we discussed how to extend the co-domain of a PRF. There are other ways of addressing this, and we will discuss it next lecture (if time permits).

4.1.1 Malleability attacks

Note that the above scheme is not secure against malleability attacks. There are a couple of different malleability attacks possible here:

- if the adversary replaces x with x', then the decryptor will not get back the correct message.
- if the adversary XORs a string s to the second ciphertext component, then the decryptor learns $m \oplus s$ instead of m.

5 Lecture summary, plan for next lecture, additional resources

Summary: Here are the main take-aways from this lecture:

- We saw two security games for encryption. The first one (Attempt 1) is equivalent to the No-Query-Semantic-Security definition, and hence it is very weak to capture all 'read-only' attacks.

- Next, we discussed the semantic security definition, which is what we'll use for encryption. This forces the encryption scheme to be randomized.
- To build intuition with randomized encryption, we discussed a 'toy scenario' where Alice and Bob share a large database of random strings.
- Next, we discussed how to use a PRF/PRP to replace the large database with a short PRF/PRP key.

Next Lecture: We will see the formal security proof of our construction.

Relevant sections from textbook [Boneh-Shoup]: Sections 5.1, 5.2, 5.3. We discussed a simplification of the construction given in Section 5.4.1.