1 Last lecture, and plan for today's lecture

Last lecture, we concluded our discussion on PRGs, and started with block ciphers/pseudorandom permutations. We started with the Advanced Encryption Standard (AES), one of the most popular block ciphers used in practice, and saw the ECB mode of encryption. This encryption scheme is not secure. Today, we will formally define what it means for a keyed function/permutation to be a *pseudorandom function/permutation*. We will also discuss a few other modes of encryption, whose proof of security will be discussed in the next lecture.

2 A quick summary, and the big picture

Since the first lecture, our prime focus has been symmetric key encryption schemes. There can be various security definitions for symmetric key encryption schemes. We would like to have a security definition that captures all kinds of real-world attacks: passive attacks (where the adversary only tries to learn something from the ciphertext), and active attacks (where the adversary tries to modify the ciphertext to 'fool' Alice/Bob). We haven't seen this definition yet, and even defining this notion is tricky (how do we capture tampering-style attacks)?

As a stepping stone, let us focus on 'read-only' (a.k.a passive) adversaries. The adversary sees a ciphertext, and tries to figure out what message was encrypted. Note that in the real world, a 'read-only' adversary could have seen a few ciphertexts exchanged between Alice and Bob, and may want to use this information to attack some particular ciphertext. As a result, we want to allow the adversary access to multiple ciphertexts. We haven't seen this definition either (but we will see it soon).

Instead, we focused our attention on even more restricted adversaries, where the adversary sees a ciphertext, and must determine whether it is an encryption of m_0 or m_1. We have already seen encryption schemes that satisfy this weak notion of security (and in particular, these encryption schemes did not satisfy the stronger notions of security discussed above). Eventually, we would like to have constructions for the strongest security definition, using the most basic building blocks.

3 Pseudorandom Functions (PRFs)/Pseudorandom Permutations (PRPs)

Block ciphers such as DES, AES came out of competitions organized by the National Institute of Standards and Technology (NIST). We will not study these objects in detail. Instead, we will study their properties abstractly. First, let us consider pseudorandom functions. These are keyed functions that 'look like' random functions if we use a random key.

Definition 09.01. A keyed function $F : \mathcal{X} \times \mathcal{K} \to \mathcal{Y}$ is a pseudorandom function (PRF) if, for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ s.t. for all n,

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

where the PRF security game is defined in Figure 1.

PRF-Game

1. Challenger chooses a bit $b \leftarrow \{0,1\}$. If $b = 0$, challenger chooses a key $k \leftarrow \mathcal{K}$ and sets function $F_0 \equiv F(\cdot, k)$. If $b = 1$, challenger chooses a truly random function F_1 from the set of all functions mapping $\mathcal{X}$ to $\mathcal{Y}$. [a]
2. The adversary sends polynomially many queries. For each query $x \in \mathcal{X}$, the challenger sends $F_b(x)$.
3. Finally, after polynomially many queries, the adversary sends a guess b' and wins if $b' = b$.

[a] Note: there are only $|\mathcal{K}|$ keys, but the number of functions from $\mathcal{X}$ to $\mathcal{Y}$ is much more. In some sense, this is similar to what we saw for PRGs: in one case, the set of possible outputs was equal to the size of input domain, while in the other case, it was equal to size of output domain.

Figure 1: PRF Security Game

Example 1: Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a secure PRF (as per Definition 09.01). Consider the function $F' : \{0,1\}^{2n} \times \{0,1\}^n \rightarrow \{0,1\}^n$ defined as follows:

$$F'(x \mid\mid y, k) = F(x,k) \oplus F(y,k)$$

for all $x, y \in \{0,1\}^n$, where $||$ denotes string concatenation. Show that F' does not satisfy Definition 09.01.

Solution: Note that $F'(x \mid\mid x, k) = 0^n$. However, for a randomly chosen function that maps $2n$ bits to n bits, the probability of the output being 0^n is $1/2^n$. Hence, the adversary queries on $(x \mid\mid x)$, and if it receives 0^n, it guesses that the function used was a pseudorandom function.

Example 2: Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a secure PRF (as per Definition 09.01). Consider the function $F' : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ defined as follows:

$$F'(x,k) = F(x,k) \oplus F(0^n, k)$$

for all $x \in \{0,1\}^n$. Show that F' does not satisfy Definition 09.01.

Solution: The adversary queries on $x = 0^n$. If it receives 0^n as the output, it guesses that the function chosen was pseudorandom. Note that if the challenger chose $b = 0$, then $F'(0^n) = 0^n$, and if $b = 1$, then the output of a random function, on input 0^n, will be 0^n with probability $1/2^n$.

(☕) **Challenge Question:** Construct a function F with appropriate key space, input and output space such that F is a secure PRF against adversaries that make at most 3 queries, but does not satisfy the general security definition (from Definition 1).

(☕) **Challenge Question:** An adversary is said to be non-adaptive if it sends all its queries at once, before seeing the responses. Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be a secure PRF. Construct a function F' (with appropriate domain, key space and range) such that F' is a secure PRF against non-adaptive adversaries, but does not satisfy the general security definition (from Definition 1).

Next, let us consider the abstraction that closely captures AES and DES. Here, we want keyed permutations that look like random permutations (if the key is random). Such functions are called pseudorandom permutations (PRPs).

Definition 09.02. A keyed function $F : \mathcal{X} \times \mathcal{K} \to \mathcal{X}$, together with an inverse function $F^{-1} : \mathcal{X} \times \mathcal{K} \to \mathcal{X}$, is a pseudorandom permutation (PRP) if the following correctness and security properties hold:

- Correctness: for any $k \in \mathcal{K}$, $x \in \mathcal{X}$, $F^{-1}\Big(F(x,k),k\Big) = x$.
- Security: for any p.p.t. adversary $\mathcal{A}$, there exists a negligible function $\mu(\cdot)$ s.t. for all n,

$$\Pr\Big[\mathcal{A} \text{ wins the PRP security game}\Big] \leq 1/2 + \mu(n),$$

where the PRP security game is defined in Figure 2.

PRP-Game

1. Challenger chooses a bit $b \leftarrow \{0,1\}$. If $b = 0$, challenger chooses a key $k \leftarrow \mathcal{K}$ and sets function $F_0 \equiv F(\cdot, k)$. If $b = 1$, challenger chooses a truly random function F_1 from the set of all permutations mapping $\mathcal{X}$ to $\mathcal{X}$. [a]
2. The adversary sends polynomially many queries. For each query $x \in \mathcal{X}$, the challenger sends $F_b(x)$.
3. Finally, after polynomially many queries, the adversary sends a guess b' and wins if $b' = b$.

[a] Again, there are only $|\mathcal{K}|$ keys, but the number of permutations from $\mathcal{X}$ to $\mathcal{X}$ is much more.

Figure 2: PRP Security Game

4 ECB mode encryption using a PRF/PRP

The ECB mode is a broken encryption scheme using a PRP (F, F^{-1}) with key space $\mathcal{K} = \{0,1\}^n$ and input/output domain $\mathcal{X} = \{0,1\}^n$. The scheme works as follows:

- $\mathsf{KeyGen}(1^n)$: chooses a random key $k \leftarrow \mathcal{K}$.
- $\mathsf{Enc}(m = m_1 \,||\, \ldots \,||\, m_\ell, k)$: let $m \in \{0,1\}^{n \cdot \ell}$ be a message (expressed as a concatenation of ℓ components, each being an n bit string). The encryption algorithm computes $\mathsf{ct}_i = F(m_i, k)$ for each $i \in [\ell]$, and outputs $\mathsf{ct} = (\mathsf{ct}_1 \,||\, \ldots \,||\, \mathsf{ct}_\ell)$.

This encryption mode does not satisfy No-Query-Semantic-Security, since encryption of $0^{n \cdot \ell}$ consists of ℓ repeated blocks of ciphertexts, while the encryption of a random message (with ℓ distinct blocks) consists of ℓ distinct ciphertext blocks.

4.1 Possible fixes to make the encryption no-query semantically secure

Here are a few possible fixes proposed in today's lecture:

1. permute the different chunks of the ciphertext: this does not work, due to the same reason why ECB mode does not offer No-Query-Semantic-Security.
2. choose ℓ different keys, and use the i^{th} key for the i^{th} message block: this idea leads to a secure encryption scheme, but the keys will need to be as large as the messages.
3. use a PRG to expand the single key k to ℓ keys $k_1 \,||\, k_2 \,||\, \ldots \,||\, k_\ell$, and then use each of these keys for encrypting the different blocks. This idea also leads to a secure encryption scheme, provided we have a secure PRG scheme. Therefore, we will need to construct a PRG from a PRF/PRP scheme. We will see this in the next lecture.
4. use m_{i-1} as a key for encrypting m_i. That is, $\mathsf{ct}_1 = F(m_1, k)$, and for all $i > 1$, $\mathsf{ct}_i = F(m_i, m_{i-1})$. This idea does not lead to a secure encryption scheme. Consider the following two messages: $m_0 = 0^n \,||\, 0^n$ and $m_1 = 0^n \,||\, 1^n$. Once the adversary receives challenge ciphertext $(\mathsf{ct}_1 \,||\, \mathsf{ct}_2)$, it knows that ct_2 is either equal to $F(0^n, 0^n)$ or $F(1^n, 0^n)$. Since F is deterministic, the adversary can easily check which case it is, and therefore break security.
5. compute $\mathsf{ct}_1 = F(m_1, k)$, and then use ct_{i-1} as key for the i^{th} ciphertext. That is, $\mathsf{ct}_i = F(m_i, \mathsf{ct}_{i-1})$. Again, note that the same attack as above will work. Once the adversary receives challenge ciphertext $(\mathsf{ct}_1 \,||\, \mathsf{ct}_2)$, it knows that ct_2 is either equal to $F(0^n, \mathsf{ct}_1)$ or $F(1^n, \mathsf{ct}_1)$.
6. compute $\mathsf{ct}_1 = F(m_1, k)$, and then use $(\mathsf{ct}_{i-1} \oplus k)$ as key for the i^{th} ciphertext. That is, $\mathsf{ct}_i = F(m_i, (\mathsf{ct}_{i-1} \oplus k))$. This scheme does not have any immediate vulnerabilities. However, note that the different blocks are using 'related keys', and this is not a good idea in general. It is plausible that AES satisfies security against related key attacks, but a general PRF may not guarantee that the above encryption scheme is secure.

(☕) **Challenge Question:** Let $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ be a secure PRF. Construct another secure PRF F' (you can choose domain and range appropriately) such that the encryption scheme defined above (Attempt 6) does not satisfy No-Query-Semantic-Security.

7. split the message m into 2ℓ blocks, each block having $n/2$ bits (therefore $m = m_1 \,||\, m_2 \,||\, \ldots \,||\, m_{2\ell}$). For each block, 'append' a counter (therefore, $m' = (1 \,||\, m_1) \,||\, (2 \,||\, m_2) \,||\, \ldots \,||(2\ell \,||\, m_{2\ell})$). Note that the counters are all padded to be of $n/2$ bits. Now, use the same PRF key k to encrypt each of the 2ℓ blocks of m'. That is, the ciphertext is $\mathsf{ct}_1 \,||\, \ldots \,||\, \mathsf{ct}_{2\ell}$, where $\mathsf{ct}_i = F(i||m_i, k)$. Intuitively, note that the previous attack no longer works — all blocks are now distinct. And indeed, it is possible to show that this is provably secure.

5 Lecture summary, plan for next lecture, additional resources

Summary We introduced the definition of PRFs/PRPs, an abstraction that captures popular encryption building blocks (AES and DES). We saw a few attempts towards building secure encryption from PRFs/PRPs.

Next Lecture: We will see a formal proof of security for some of the encryption modes (using PRFs/PRPs). Later, we will show that PRFs imply PRGs.

Relevant sections from textbook [Boneh-Shoup]: parts of Section 4.1 and 4.4.1.