Stronger notions of security for encryption and pseudorandom functions

Recommended reading:

KL, Section 3.4 + stream cipher part of Section 3.3.1 + Section 3.5 + Section 3.6

Back to encryption. :)

We have seen how to construct computationally secure encryption schemes with short keys using PRGs. We also saw that the existence of OWFs is a sufficient (and necessary) condition for the existence of PRGs.

BUT..... the definition of computationally secure encryption leaves a lot to be desired.

In particular, it only guarantees security for sending one message. Ideally, Alice and Bob would like to communicate back and forth using the same key. How can we formalize security in this setting?

Def (Computational security under multiple encryptions).

An encryption scheme (Enc, Dec, Gen) is computationally secure under multiple encryptions if for any PPT adv A there is a negl. function $\varepsilon(n)$ such that A wins the following game with probability at most $\frac{1}{2} + \varepsilon(n)$.

The game is played by A and a challenger, parameterized by the security parameter n:

→ $A(1^n)$ chooses two message vectors $\vec{m}_0, \vec{m}_1$ with the same number of messages and such that $|m_0^i| = |m_1^i|$ for all i, where m_b^i is the i-th message in $\vec{m}_b$; and sends $(\vec{m}_0, \vec{m}_1)$ to the challenger

→ the challenger samples $k \leftarrow Gen(1^n)$ and $b \leftarrow \{0,1\}$, computes $\vec{c}$ with $c^i = Enc(k, m_b^i)$, and sends $\vec{c}$ to A.

→ A outputs b' and wins iff $b' = b$.

Recall our previous encryption scheme

$$Enc(k, m) = G(k) \oplus m.$$

This scheme is not secure under multiple encryptions!

Consider an adversary that chooses

$$\vec{m}_0 = (0^{\ell(n)}, 0^{\ell(n)})$$

$$\vec{m}_1 (0^{\ell(n)}, 1^{\ell(n)})$$

If $b=0$ then $c^1 = c^2$. $\Rightarrow$ A wins game with

If $b=1$ then $c^1 \neq c^2$. prob. 1.

More generally...

Thm: No deterministic (stateless) encryption scheme is computationally secure under multiple encryptions. → Homework

Even encrypting 1-bit messages is now non-trivial. :)

How can we construct encryption schemes secure under multiple encryptions?

→ Stateful encryption (aka stream ciphers)

We have seen how, using an efficient permutation f with a hardcore predicate P, we can generate polynomially many pseudorandom bits starting from a short secret seed $s \leftarrow \{0,1\}^n$.

$$s_0 = s$$
$$s_1 = f(s_0), \quad \sigma_1 = P(s_0)$$
$$s_2 = f(s_1), \quad \sigma_2 = P(s_1)$$
$$\vdots$$
$$s_\ell = f(s_{\ell-1}), \quad \sigma_\ell = P(s_{\ell-1}).$$

If you're interested in learning about stream ciphers that are used in practice, start with the insecure RC4.

Alice and Bob start with the same secret state s_0. After transmitting ℓ bits they are in state s_ℓ. If Alice wants to transmit the $(\ell+1)$st bit $b_{\ell+1}$ to Bob, she computes $s_{\ell+1} = f(s_\ell)$ and $\sigma_{\ell+1} = P(s_\ell)$, and sends $\sigma_{\ell+1} \oplus b_{\ell+1}$ to Bob. To decrypt, Bob computes $s_{\ell+1} = f(s_\ell)$ and $\sigma_{\ell+1} = P(s_\ell)$.

The scheme above works, but requires Alice & Bob to keep state. If some transmission packet is dropped then they lose synchronization and cannot communicate.

they need to keep track of how many bits have already been transmitted

We prefer stateless encryption schemes.

Ideally, we would like to have stateless encryption schemes. Let's explore how we can make the scheme above less stateful.

Obs 1: Bob does not need to keep state if Alice appends a counter to the transmissions. More precisely, when sending the i-th bit Alice transmits $(i, b_i \oplus r_i)$.

Obs 2: The important thing above is that we never re-use the same part of the "key stream". The counter helps us avoid that. But what if we replace the counter by a "random" $r \leftarrow R$

More precisely, everytime Alice wishes to encrypt a bit b, she samples $r \leftarrow \mathcal{R}$ and transmits $(r, b \oplus \sigma_r)$.

This doesn't work, because $\sigma_1, \sigma_2, \ldots, \sigma_r$ are computed sequentially and efficiently,

and so there are only poly(n)-many possible values for r. This means that the probability that we re-use an r is non-negligible!

So, intuitively, we need an object that given an n-bit seed generates exponentially many possible r's.

These are called pseudorandom functions!

Security against chosen-plaintext attacks

Before we dive deeper into pseudorandom functions, we will upgrade our notion of security to handle

Chosen-plaintext attacks

These are attacks where an adversary gets to see encryptions of plaintexts of their choice before trying to break the encryption scheme.

Chosen-plaintext attacks are practically relevant:

→ They include known-plaintext attacks, where the adversary knows the messages that are being encrypted.

→ They've been used as far back as WW2

→ Let give the example of a terminal that encrypts and sends commands to a remote server. We'd like for encryption to remain secure even if the adversary gets to use the terminal before the honest party.

How can we model such attacks?

We consider a ciphertext distinguishing game where the adversary gets to observe encryptions of messages of their choice before choosing m_0, m_1!

(Enc, Dec, Gen) an encryption scheme. The CPA ciphertext dist. game is played between an adv A and a challenger, parameterized by n, and an encryption oracle $\mathcal{O}$

→ A key $k \leftarrow Gen(1^n)$ is generated

→ $A(1^n)$ can send messages m to $\mathcal{O}$ and gets back $Enc(k, m)$

→ A chooses m_0, m_1 and sends to challenger

→ Challenger samples $b \leftarrow \{0,1\}$ and sends back $c = Enc(k, m_b)$

→ A interacts more with $\mathcal{O}$, and outputs $b' \in \{0,1\}$.

A wins iff $b' = b$.

Def. (CPA-secure encryption) A secret-key encryption scheme is CPA-secure if for any PPT adv A there is a negl. function $\varepsilon(n)$ such that $A(1^n)$ wins the game above with probability $\leq \frac{1}{2} + \varepsilon(n)$.

For CPA-security there is no distinction between security for one or multiple encryptions.

Thm: If a scheme is CPA-secure, then it is CPA-secure "under multiple encryptions"

Proof: Homework

This means that CPA-security is at least as strong as security under multiple encryptions. In fact, it is strictly stronger (homework).

Constructing CPA-secure encryption: Pseudorandom functions (PRFs)

Intuitively, a PRF is a "keyed" function that looks random to PPT adversaries.

A keyed function F is of the form $F: \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$.
F is efficient if there exists a poly-time det. algo that computes $F(k,x)$ given (k,x).
We'll write $F_k(x) = F(k,x)$ to emphasize that k is a fixed secret key.

For simplicity, we'll assume that F is length-preserving.
If $|k| = n$, then $F_k(\cdot)$ only accepts inputs x with $|x| = n$ and $|F_k(x)| = n$.

How can we formalize PRFs?

Intuition: For $k \leftarrow \{0,1\}^n$, no PPT adv. interacting with F_k can distinguish it from a random function $f_n: \{0,1\}^n \to \{0,1\}^n$

More formally, fix a key $k \in \{0,1\}^n$ ↵ Suppose that the adversary gets to interact with an oracle $\mathcal{O}$. There are two options:

→ Either $\mathcal{O}$ is $\mathcal{O}_{real,k}(\cdot)$, which given x as input answers with $F_k(x)$.

→ or $\mathcal{O}$ is $\mathcal{O}_{ideal}(\cdot)$ which given x either outputs $y \leftarrow \{0,1\}^n$ if x hasn't been queried before, or answers consistently with prior answer to input x. This means that $\mathcal{O}$ is a random function.

We denote the behavior of A given access to oracle $\mathcal{O}$ by $A^{\mathcal{O}}$.

Def (PRFs). F is a PRF if for any PPT adversary A there is a negl. function $\varepsilon(n)$ such that

$$\left| \Pr_{k \leftarrow \{0,1\}^n}\left[A^{\mathcal{O}_{real,k}(\cdot)}(1^n) = 1 \right] - \Pr\left[A^{\mathcal{O}_{ideal}(\cdot)}(1^n) = 1 \right] \right| \leq \varepsilon(n)$$

Example: Is $F_k(x) = k \oplus x$ a PRF?

Answer: No. Consider $A^{\mathcal{O}}$ that queries $\mathcal{O}$ on distinct x_1, x_2 and gets back y_1, y_2.

If $\mathcal{O} = \mathcal{O}_{real,k}$, then $y_1 \oplus y_2 = (k \oplus x_1) \oplus (k \oplus x_2) = x_1 \oplus x_2$

If $\mathcal{O} = \mathcal{O}_{ideal}$ then $y_1 \oplus y_2$ is unif. random in $\{0,1\}^n$.

Suppose that A outputs 1 iff $Y_1 \oplus Y_2 = X_1 \oplus X_2$. Then

$$\left| \Pr\left(A^{\mathcal{O}_{real,k}(\cdot)}(1^n) = 1\right) - \Pr\left(A^{\mathcal{O}_{ideal}(\cdot)}(1^n) = 1\right) \right|$$

$$= |1 - 2^{-n}| \rightarrow \text{non-negl.}$$

CPA-secure encryption from PRFs

We'll combine previous ideas with PRFs to get CPA-secure encryption!

If F is a PRF, we define:

- $\text{Gen}(1^n)$ samples $k \leftarrow \{0,1\}^n$.
- $\text{Enc}(k, m)$ samples $r \leftarrow \{0,1\}^n$ & outputs $c = (r, F_k(r) \oplus m)$
- Dec is obvious

Thm: The encryption scheme above is CPA-secure if F is a PRF.

Proof: The proof proceeds in two main steps. First, we show that this is true when we pretend F is a random function. Second, we show that we can replace the random function by F without messing things up.

1) Consider the idealized encryption scheme $\tilde{\Pi} = (\widetilde{Enc}, \widetilde{Dec}, \widetilde{Gen})$ that is exactly like (Enc, Dec, Gen) but where F is replaced by a uniformly random function f.

Let A be a PPT adv for the CPA game, and let $q(n)$ be an upper bound on the number of queries that $A(1^n)$ makes to the encryption oracle. We claim that

$$\Pr(A \text{ wins CPA game for } \tilde{\Pi}) \leq \frac{1}{2} + \frac{q(n)}{2^n}$$

→ negligible since $q(n) = poly(n)$.

To see this, let $M_1, \ldots, M_{q(n)}$ be the messages queried by A. For each M_i, A gets $(R_i, f^*(R_i) \oplus M_i)$ from the encryption oracle.

A then chooses M_0, M_1 and gets back $(R^*, f^*(R^*) \oplus M_b)$ from the challenger.

If $R^* \neq R_i$ for all $i \in \{1, \ldots, q(n)\}$, then $f^*(R^*)$ is unif. random and independent of $f^*(R_1), \ldots, f^*(R_{q(n)})$.

This means that $f^*(R^*) \oplus M_b$ is independent of A's view $(R_1, \ldots, R_{q(n)}, M_1, \ldots, M_{q(n)}, M_0, M_1)$, and so A will win with probability $= 1/2$ in this case. So,

$$\Pr[A \text{ wins game}] \leq \Pr(R^* = R_i \text{ for some } i \in \{1, \ldots, q(n)\}) + \Pr(R^* \neq R_i \ \forall i \in \{1, \ldots, q(n)\}) \cdot \frac{1}{2}$$

$$\leq \frac{1}{2} + \Pr(R^* = R_i \text{ for some } i \in \{1, \ldots, q\})$$

$$\xleftarrow{\text{union bound}} \leq \frac{1}{2} + \sum_{i=1}^{q(n)} \underbrace{\Pr(R^* = R_i)}_{= 2^{-n}} = \frac{1}{2} + \frac{q(n)}{2^n}.$$

2) We will show that

$$(*)\quad \left| \Pr\left(A(1^n) \text{ wins CPA-game for } \pi = (Enc, Dec, Gen)\right) - \Pr\left(A(1^n) \text{ wins CPA game for } \tilde{\pi} = (\tilde{Enc}, \tilde{Dec}, \tilde{Gen})\right) \right| \in \varepsilon(n)$$

for every PPT A and negligible $\varepsilon(n)$.

Suppose not. Then, there is a polynomial $p(n)$ s.t. for infinitely many n we have $(*) \geq 1/p(n)$.

We design a PPT adv $\bar{A}$ that uses A to break the PRF F.

$\bar{A}^{\mathcal{O}}(1^n)$ emulates the CPA challenger & uses $\mathcal{O}$ to implement the encryption oracle:

→ It runs $A(1^n)$. Whenever A asks an encryption query M_i, $\bar{A}$ samples $R_i \leftarrow \{0,1\}^n$ and queries $\mathcal{O}(R_i)$ to get back Z_i, & sends back $(R_i, Z_i \oplus M_i)$ to A.

→ Once A chooses m_0, m_1 to be sent to the challenger, $\bar{A}$ samples $b \leftarrow \{0,1\}$, samples $R^* \leftarrow \{0,1\}^n$, queries $\mathcal{O}(R^*)$ to get Z^*, & sends back $(R^*, Z^* \oplus m_b)$.

→ If A outputs $b' = b$, then $\bar{A}$ outputs 1. Else $\bar{A}$ outputs 0.

Now we bound $\left| \Pr_{k \leftarrow \{0,1\}^n}[\bar{A}^{\mathcal{O}_{real,k}}(1^n)=1] - \Pr[\bar{A}^{\mathcal{O}_{ideal}}(1^n)=1] \right|$

i) If $\mathcal{O} = \mathcal{O}_{ideal}$ then the interaction above corresponds to the CPA game for $\tilde{\Pi}$,

so $\Pr[\bar{A}^{\mathcal{O}_{ideal}}(1^n)=1] = \Pr(A \text{ wins CPA game for } \tilde{\Pi})$.

ii) If $\mathcal{O} = \mathcal{O}_{real,k}$, then the interaction above corresponds to the CPA game for Π, so

$$\Pr_{k \leftarrow \{0,1\}^n}[\bar{A}^{\mathcal{O}_{real,k}}(1^n)=1] = \Pr(A \text{ wins CPA game for } \Pi)$$

Therefore,

$$\left| \Pr_{k \leftarrow \{0,1\}^n}[\bar{A}^{\mathcal{O}_{real,k}}(1^n)=1] - \Pr[\bar{A}^{\mathcal{O}_{ideal}}(1^n)=1] \right| \geq \frac{1}{p(n)},$$

non-negligible

which contradicts the fact that F is a PRF.

$$\Rightarrow \Pr(A \text{ wins CPA game for } \Pi) \leq \frac{1}{2} + \frac{q(n)}{2^n} + \varepsilon(n)$$

negligible!

∎