A more powerful adversary
Security against chosen-plaintext attacks

Foundations of Cryptography
Computer Science Department
Wellesley College

Fall 2016

Table of contents

Security against chosen-plaintext attacks (CPA)

- Our adversaries have been completely passive, merely listening in our conversations.
- Today we study a more powerful type of adversarial attack, called a *chosen-plaintext attack*.
- Here the adversary is allowed to ask for encryptions of multiple messages* chosen adaptively.

*Formally the adversary, denoted $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$, has access to an *encryption oracle*, viewed as a "black-box" that encrypts messages of $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$'s choice using the secret key k that is unknown to $\mathcal{A}^{\mathsf{Enc}_k(\cdot)}$.

Another experiment

The experiment is defined for any private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, any adversary $\mathcal{A}$, and any value n for the security parameter: *The CPA indistinguishability experiment* $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$

1. A key k is generated by running $\mathsf{Gen}(1^n)$.
2. The adversary $\mathcal{A}$ is given 1^n and oracle access to $\mathsf{Enc}_k(\cdot)$, and outputs a pair of messages $m_0, m_1 \in \mathcal{M}$ of the same length.
3. A random bit $b \leftarrow \{0,1\}$ is chosen. A *challenge ciphertext* $c \leftarrow \mathsf{Enc}_k(m_b)$ is computed and given to $\mathcal{A}$.
4. The adversary $\mathcal{A}$ continues to have oracle access to $\mathsf{Enc}_k(\cdot)$, outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. We write $\mathsf{PrivK}^{\mathsf{eav}}_{\mathcal{A},\Pi}(n) = 1$ if the output is 1 and in this case we say that $\mathcal{A}$ *succeeded*.

*CPA-secure**

Definition 3.22. An private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable encryption under a chosen-plaintext attack* if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n),$$

where the probability is taken over the random coins used by $\mathcal{A}$, as well as the random coins used by the experiment (for choosing the key, the random bit b, and any random coins used in the encryption process).

*Notice that this scheme encompasses *known plaintext attacks*. Certainly any scheme that has indistinguishable encryptions under a chosen-plaintext attack also has indistinguishable encryption in the presence of an eavesdropper. Why?

*Well, that's just plain impossible**

- Consider an adversary $\mathcal{A}$ that outputs (m_0, m_1) and then receives the challenge ciphertext $c \leftarrow \mathsf{Enc}_k(m_b)$.
- Since $\mathcal{A}$ has access to $\mathsf{Enc}_k(\cdot)$, it can obtain $c_0 \leftarrow \mathsf{Enc}_k(m_0)$ and $c_1 \leftarrow \mathsf{Enc}_k(m_1)$
- The adversary now does a simple comparison: If $c = c_0$ then it must be that $b = 0$; if $c = c_1$, then $b = 1$.

*What's wrong with this strategy?

Real world chosen-plaintext attacks

- In May 1942, US Navy cryptanalysts discovered that Japan was planning an attack in the Central Pacific by intercepting a message containing the ciphertext fragment "AF" which they believed corresponded to "Midway island".
- Unfortunately, their superiors in Washington were unconvinced, so they devised the following plan: US forces at Midway send a plaintext message that their freshwater supplies were low.
- The Japanese intercepted this message and reported to their superiors the "AF" was low on water.

CPA security for multiple encryptions

The definition for indistinguishable encryptions under a chosen-plaintext can easily be extended to indistinguishable multiple encryptions in the same way that indistinguishability encryption in the presence of an eavesdropper was.

The text takes a somewhat simpler approach that can model attackers that can adaptively choose plaintexts to be encrypted, even after observing previous ciphertext.

The attacker has access to a *"left-or-right" oracle $LR_{k,b}$* that, on input a pair of equal-length messages m_0, m_1, computes the ciphertext $c \leftarrow \mathrm{Enc}_k(m_b)$ and returns c.*

*Here b is a random bit chosen at the beginning of the experiment.

Left-or-right oracles

"Left-or-right" oracles generalize the previous definition of multiple-message security (Definition 3.19) because instead of outputting the lists $\vec{M}_0 = (m_{0,1}, \ldots, m_{0,t})$ and $\vec{M}_1 = (m_{1,1}, \ldots, m_{1,t})$ the attacker can now sequentially query $\mathsf{LR}_{k,b}(m_{0,1}, m_{1,1}), \ldots \mathsf{LR}_{k,b}(m_{0,t}, m_{1,t})$.

This also encompasses the attacker's access to an oracles, since the attacker can simply query $\mathsf{LR}_{k,b}(m, m)$ to obtain $\mathsf{Enc}_k(m)$.

*Here b is a random bit chosen at the beginning of the experiment.

More formally

The LR-oracle experiment $\mathsf{PrivK}^{\mathsf{LR-cpa}}_{\mathcal{A},\Pi}(n)$*:*

1. A key k is generated by running $\mathsf{Gen}(1^n)$.
2. A uniform bit $b \in \{0, 1\}$ is chosen.
3. The adversary $\mathcal{A}$ is given input 1^n and oracle access to $\mathsf{LR}_{k,b}(\cdot, \cdot)$,
4. The adversary $\mathcal{A}$ outputs a bit b'.
5. The output of the experiment is defined to be 1 if $b' = b$, and 0 otherwise. In the former case, we say that $\mathcal{A}$ *succeeds*.

Indistinguishable encryption under chosen-plaintext

Definition 3.23. An private-key encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$ has *indistinguishable multiple encryptions under a chosen-plaintext attack*, if for all probabilistic polynomial-time adversaries $\mathcal{A}$ there exists a negligible function negl such that

$$\Pr[\mathsf{PrivK}^{\mathsf{LR-cpa}}_{\mathcal{A},\Pi}(n) = 1] \leq \frac{1}{2} + \mathsf{negl}(n),$$

where the probability is taken over the random coins used by $\mathcal{A}$, as well as the random coins used by the experiment.

Good news

Theorem 3.24. Any private-key encryption scheme that is CPA-secure is also CPA-secure for multiple encryptions.

More Good News. Given any CPA-secure *fixed-length* encryption scheme $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, it is possible to construct a CPA-secure encryption scheme $\Pi' = (\mathsf{Gen'}, \mathsf{Enc'}, \mathsf{Dec'})$ for *arbitrary-length* messages quite easily.

.

Keyed functions: Some definitions

A *keyed function* F is a two-input function $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ where the first input is called the *key* and denoted k, and the second input is just called *the input*.

The key k will be chosen and then *fixed*, and we will then be interested in the single input function $F_k(x) \stackrel{\text{def}}{=} F(k,x)$.

We assume that F is *length-preserving*, i.e., $|F_k(x)| = |x| = |k|$, and *efficient*, i.e., there is a deterministic polynomial-time algorithm that computes $F(k,x)$.

Keyed functions: Some observations

- A keyed function F induces a natural distribution on functions given by choosing a random key $k \leftarrow \{0,1\}^n$ and then considering the resulting function F_k.
- Intuitively, we call F *pseudorandom* if the function F_k (for a randomly chosen k) is indistinguishable in polynomial time from a function chosen uniformly at random from the set of all functions from $\{0,1\}^n$ to $\{0,1\}^n$, denoted Func_n.
- This week's puzzler: How big is Func_n?

Constructing pseudorandom functions: A daunting task

- We wish to construct a keyed function F such that F_k (for $k \leftarrow \{0,1\}^n$ chosen uniformly at random) is indistinguishable from f (for $f \leftarrow \mathsf{Func}_n$).
- There are at most 2^n functions in the former set and exactly $2^{n \cdot 2^n}$ functions in the second. Despite this, the "behavior" of these function must look the same to any polynomial-time distinguisher.
- What "behavior" are we talking about? Well, we could require that every polynomial-time distinguisher D that receives a description of the pseudorandom function F_k output 1 with "almost" the same probability as when it receives a description of a random function f.

Pseudorandom Functions

Definition 3.25. Let $F : \{0,1\}^* \times \{0,1\}^* \to \{0,1\}^*$ be an efficient length-preserving, keyed function. We say that F is a *pseudorandom function* if for all probabilistic polynomial-time distinguishers D, there exists a negligible function negl such that:

$$\left|\Pr[D^{F_k(\cdot)}(1^n) = 1] - \Pr[D^{f(\cdot)}(1^n) = 1]\right| \leq \mathsf{negl}(n),$$

where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random and f is chosen uniformly at random from the set of functions mapping n-bit strings to n-bit strings.

Pseudorandom Function, Not

We gain some intuition about what a psuedorandom function might look like by examining one that it not.

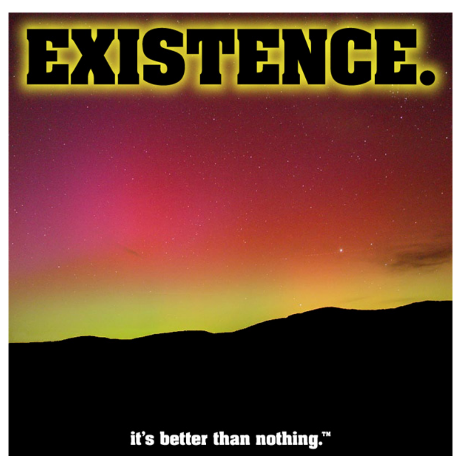
Example 3.26. Define the keyed, length-preserving function F by $F(k,x) = k \oplus x$. Note that for any input x, the value $F_k(x)$ is uniformly distributed (which k is).*

Define a distinguisher D that queries its oracle $\mathcal{O}$ on arbitrary, distinct points x_1, x_2 to obtain values $y_1 = \mathcal{O}(x_1)$ and $y_2 = \mathcal{O}(x_2)$, and outputs 1 if and only if $y_1 \oplus y_2 = x_1 \oplus x_2$. If $\mathcal{O} = F_k$, then D outputs 1 in all cases. On the other hand, if $\mathcal{O} = f$ for f chosen uniformly from Func_n, then probability that $f(x_1) \oplus f(x_2) = x_1 \oplus x_2$ is 2^{-n}. The difference is $|1 - 2^{-n}|$ is not negligible.

*We're off to a good start.

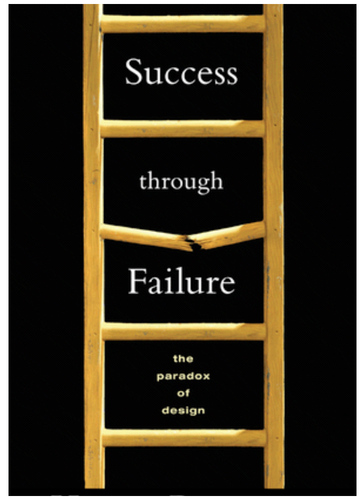
On the existence of pseudorandom functions

- You guessed it, we don't really know.
- However, very efficient primitives called *block ciphers* are widely believed to be pseudorandom functions. (More on this soon.)
- Also, it is known that pseudorandom functions exist if and only if pseudorandom generators do.



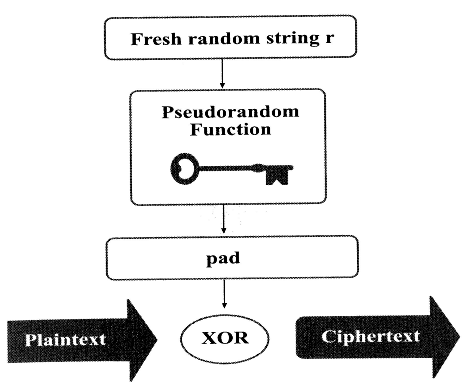
First shot

- We define $\mathsf{Enc}_k(m) = F_k(m)$.
- Since $f(m)$ for a *random function* is a random string and F_k is supposed to "look like" a random function, then we expect $F_k(m)$ reveals no information about m.
- There is however a rub.



Next shot

We encrypt by applying the pseudorandom function to a *random value* r (rather than the plaintext) and XORing the result with the plaintext.



This is another instance of XORing a pseudorandom "pad" with the message, except this time an *independent* pseudorandom pad is used each time.

The encryption scheme

Construction 3.30. Let F be a pseudorandom function. Define a private-key encryption scheme for messages of length n as follows:

- Gen: On input 1^n, choose $k \leftarrow \{0,1\}^n$ uniformly at random and output it as the key.
- Enc: On input a key $k \in \{0,1\}^n$ and a message $m \in \{0,1\}^n$, choose $r \leftarrow \{0,1\}^n$ uniformly at random and output the ciphertext
$$c := \langle r, F_k(r) \oplus m \rangle.$$
- Dec: On input a key $k \in \{0,1\}^n$ and a ciphertext $c = \langle r, s \rangle$, output the plaintext message
$$m := F_k(r) \oplus s.$$

Proving our construction is CPA-secure

Theorem 3.31. If F is a pseudorandom function, then Construction 3.30 is a fixed-length private-key encryption scheme for messages of length n that has indistinguishable encryption under a chosen-plaintext attack.
Proof. Define a modified encryption scheme $\widetilde{\Pi} = (\widetilde{\mathsf{Gen}}, \widetilde{\mathsf{Enc}}, \widetilde{\mathsf{Dec}})$ that is exactly the same as $\Pi = (\mathsf{Gen}, \mathsf{Enc}, \mathsf{Dec})$, except that a truly random function f is used in place of F_k.

Fix an arbitrary PPT adversary $\mathcal{A}$, and let $q(n)$ be a polynomial upper bound on the number of queries that $\mathcal{A}(1^n)$ makes to its encryption oracle. We first show that

$$\left| \Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1 \right] - \Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 \right] \right| \leq \mathsf{negl}(n).$$

Proof by reduction

The basic idea of the proof should be familiar by now:

- We use $\mathcal{A}$ to construct a distinguisher D for the pseudorandom funuction F. The distinguisher is given oracle access to some function $\mathcal{O}$ and its goal is to determine whether this function is "pseudorandom" or "random".
- D emulates experiment $\mathsf{PrivK}^{\mathsf{cpa}}$ for $\mathcal{A}$. If $\mathcal{A}$ succeeds D guesses its oracle must be a pseudorandom function. If $\mathcal{A}$ fails, then D guesses its oracle must be truly random.

The distinguisher D

Distinguisher D: D is given input 1^n and access to an oracle $\mathcal{O} : \{0,1\}^n \to \{0,1\}^n$.

1. Run $\mathcal{A}(1^n)$. Whenever $\mathcal{A}$ queries its oracle on message m, answer as follows:
    1.1 Choose $r \leftarrow \{0,1\}^n$ uniformly at random.
    1.2 Query $\mathcal{O}(r)$ and obtain response y.
    1.3 Return the ciphertext $\langle r, y \oplus m \rangle$ to $\mathcal{A}$.
2. When $\mathcal{A}$ outputs messages $m_0, m_1 \in \{0,1\}^n$, choose a random bit $b \leftarrow \{0,1\}$ and then:
    2.1 Choose $r \leftarrow \{0,1\}^n$ uniformly at random.
    2.2 Query $\mathcal{O}(r)$ and obtain response y.
    2.3 Return the ciphertext $\langle r, y \oplus m_b \rangle$ to $\mathcal{A}$.
3. Continue answering any encryption oracle queries of $\mathcal{A}$ as before. Eventually $\mathcal{A}$ outputs a bit b'. Output 1 if $b' = b$, and output 0 otherwise.

We establish our first claim

1. If D's oracle is a pseudorandom function, then the view of $\mathcal{A}$ when run as a sub-routine by D is distributed identically to the view of $\mathcal{A}$ in experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n)$. Thus,

$$\Pr_{k \leftarrow \{0,1\}^n}\left[D^{F_k(\cdot)}(1^n) = 1\right] = \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1],$$

where $k \leftarrow \{0,1\}^n$ is chosen uniformly at random.

2. If D's oracle is a random function then $\mathcal{A}$ view when run as a sub-routine by D is distributed identically to the view of $\mathcal{A}$ in experiment $\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n)$. Thus,

$$\Pr_{f \leftarrow \mathsf{Func}_n}\left[D^{f(\cdot)}(1^n) = 1\right] = \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1],$$

where $f \leftarrow \mathsf{Func}_n$ is chosen uniformly at random.

Since F is pseudorandom there exists a negligible function negl for which

$$\left|\Pr\left[D^{F_k(\cdot)}(1^n) = 1\right] - \Pr\left[D^{f(\cdot)}(1^n) = 1\right]\right| \leq \mathsf{negl}(n).$$

Our second claim

Next we show that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1\right] \leq \frac{1}{2} + \frac{q(n)}{2^n}.$$

where recall that $q(n)$ is polynomial upper bound on the number of queries $\mathcal{A}$ makes to its encryption oracle.

Establishing our second claim

Every time a message m is encrypted, a random $r \leftarrow \{0,1\}^n$ is chosen and the ciphertext is set equal to $\langle r, f(r) \oplus m \rangle$. Let r_c denote the random string used when generating the challenge ciphertext $c = \langle r_c, f(r_c) \oplus m_b \rangle$. There are two cases:

1. *The value r_c is used by the encryption oracle to answer at least one of $\mathcal{A}$'s queries:* $\mathcal{A}$ is in the money since whenever the oracle returns a ciphertext $\langle r, s \rangle$, the adversary learns the value of $f(r) = s \oplus m$. Probability of this happening in $q(n)$ queries is at most $q(n)/2^n$.
2. *The value r_c is never used by the encryption oracle to answer any of $\mathcal{A}$'s queries:* As far as $\mathcal{A}$ is concerned, the value $f(r_c)$ that is XORed with m is completely random, and so $\mathcal{A}$ outputs $b' = b$ with probability exactly $1/2$.

Probabilities of success

Let Repeat denote the event the r_c is used by the encryption oracle to answer at least one of $\mathcal{A}$'s queries.* We have:

$$
\begin{aligned}
&\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1] = \\
&= \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 \wedge \mathsf{Repeat}] + \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 \wedge \overline{\mathsf{Repeat}}] \\
&\leq \Pr[\mathsf{Repeat}] + \Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1 \mid \overline{\mathsf{Repeat}}] \\
&\leq \frac{q(n)}{2^n} + \frac{1}{2}.
\end{aligned}
$$

*As on the previous slide, the probability of Repeat is at most $q(n)/2^n$, and the probability that $\mathcal{A}$ success if Repeat does not occur is exactly $1/2$.

Success

Our first result implies that

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \mathsf{negl}(n) + \Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1\right]$$

while our second result states that

$$\Pr[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\widetilde{\Pi}}(n) = 1] \leq \frac{q(n)}{2^n} + \frac{1}{2}.$$

Putting these two together yields

$$\Pr\left[\mathsf{PrivK}^{\mathsf{cpa}}_{\mathcal{A},\Pi}(n) = 1\right] \leq \mathsf{negl}(n) + \frac{q(n)}{2^n} + \frac{1}{2}.$$