Cryptography

Lecture 08

Pseudorandom Functions and Permutations

Keyed functions

- Let $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ be an efficient, deterministic algorithm
  - Define $F_k(x) = F(k, x)$
  - The first input is called the key
- Choosing a uniform $k \in \{0,1\}^n$ is equivalent to choosing the function $F_k : \{0,1\}^n \rightarrow \{0,1\}^n$
  - i.e. for fixed key length n, the algorithm F defines a distribution over functions in $Func_n$!

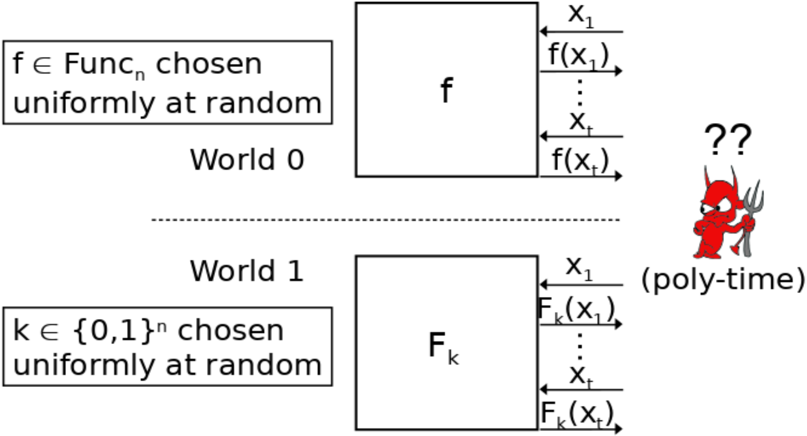
Note: A Keyed Perm requires F_k a perm and F_k^{-1} easy to compute.

Pseudorandom Functions (PRFs)

We define Pseudorandom Function informally.

A Pseudorandom Function is a keyed function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^n$ such that a PPT Eve cannot do well in the following game:

1. Alice picks $k \in \{0,1\}^n$ and hence picks F_k
2. Bob picks a function f uniformly at random from $func_n$.
3. Eve gets a black box for one of $\{F_k, f\}$.
4. Eve needs to determine which one.



f ∈ $Func_n$ chosen uniformly at random
World 0
f
x_1
$f(x_1)$
x_t
$f(x_t)$
??
(poly-time)
World 1
k ∈ $\{0,1\}^n$ chosen uniformly at random
F_k
x_1
$F_k(x_1)$
x_t
$F_k(x_t)$

Pseudorandom Permutations (PRPs)

We define Pseudorandom Permutation informally.

A Pseudorandom Permutation is a keyed function $F : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$ such that every F_k is a permutation and a PPT Eve cannot do well in the following game:

1. Alice picks $k \in \{0,1\}^n$ and hence picks F_k
2. Bob picks a permutation f uniformly from $perm_n$.
3. Eve gets a black box for one of $\{F_k, f\}$.
4. Eve needs to determine which one.

Note:

- For large enough n, a random permutation is indistinguishable from a random function
- So in Psuedorandom Function game Bob could pick a random permutation.

PRFunctions Yields PRGenerators

- PRF F immediately implies a PRG G:
  - Define $G(k) = F_k(0 \cdots 0) \mid F_k(0 \cdots 1) \mid \cdots F_k(1 \cdots 1)$
- PRF can be viewed as a PRG with random access to exponentially long output
  - The function F_k can be viewed as the $n2^n$-bit string $F_k(0 \ldots 0) \mid \cdots \mid F_K(1 \ldots 1)$

Do PRFs/PRPs exist? Theoretical Answer

A one-way function (perm) is function (perm): easy to compute, hard to invert.
A one-way function (perm) with a hard core predicate is a function (perm) that is easy to compute but hard to invert, and (say) the middle bit of $f^{-1}(x)$ is hard to compute.

Chapter 7 shows:
$\exists$ One way Perm $\implies$ $\exists$ one way perm with a hcp.
$\exists$ one way perm with hcp $\implies$ $\exists$ PRG with expanion 1
$\exists$ PRG with expa-1 $\implies$ $\exists$ PRG with expa-$p(n)$ any poly p.
$\exists$ PRG with expa-$2n$ $\implies$ $\exists$ PRF.
Note: One way func $\implies$ PRF also known but much harder.

Comment on Theoretical Answer

Could start with a function that we thing is a One Way Perm.
Can you think of one? Discuss

Comment on Theoretical Answer

Could start with a function that we thing is a One Way Perm. Can you think of one? Discuss

If p is a prime and g is a generator than $f(x) = g^x \pmod{p}$:

1. f is a perm.
2. If we think Discrete Log is hard then f is not invertible.

Comment on Theoretical Answer

Could start with a function that we thing is a One Way Perm.
Can you think of one? Discuss

If p is a prime and g is a generator than $f(x) = g^x \pmod{p}$:

1. f is a perm.
2. If we think Discrete Log is hard then f is not invertible.

DL hard $\implies$ f is one-way-perm $\implies \cdots \implies$ PRF.
Should we construct one this way?Discuss

Comment on Theoretical Answer

Could start with a function that we thing is a One Way Perm.
Can you think of one? Discuss

If p is a prime and g is a generator than $f(x) = g^x \pmod{p}$:

1. f is a perm.
2. If we think Discrete Log is hard then f is not invertible.

DL hard $\implies$ f is one-way-perm $\implies \cdots \implies$ PRF.
Should we construct one this way?Discuss
No: Too slow. But good for proof of concept.

Do PRFs/PRPs exist? Practical

- Block ciphers are practical constructions of pseudorandom permutations
- No asymptotics: $F : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^m$
  - n = "key length"
  - m = "block length"
- Hard to distinguish F_k from uniform $f \in Perm_m$ *even for attackers running in time* $\approx 2^n$

AES

- Advanced encryption standard (AES)
  - Standardized by NIST in 2000 based on a public, worldwide competition lasting over 3 years
  - Block length = 128 bits
  - Key length = 128, 192, or 256 bits
- Will discuss details later in the course
- Currently no reason to use anything else

Recall Comp CPA-security via a Game.

Π is an encryption system. n is a security param.

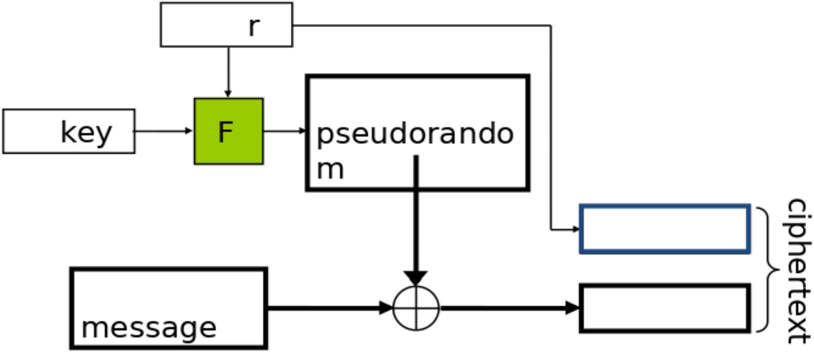
1. $k \leftarrow Gen(1^n)$. Eve does NOT know k.
2. Eve picks $m_0, m_1 \in \mathcal{M}$ $(|m_0| = |m_1|)$. Eve has BB for Enc_k.
3. $b \leftarrow \{0,1\}$, $c \leftarrow Enc_k(m_b)$
4. Π sends c to Eve.
5. Eve outputs $b\text{`} \in \{0,1\}$. Eve has BB for Enc_k.
6. If $b = b'$ then Eve *Wins!*

Π Comp CPA-secure if for all PPT Eve

$$\Pr[\text{Eve Wins}] \le \frac{1}{2} + \varepsilon(n)$$

CPA-secure encryption

- Let F be a keyed function
- $Gen(1^n)$: choose a uniform key $k \in \{0,1\}^n$
- $Enc_k(m)$
  - Choose uniform $r \in \{0,1\}^n$ (IV, Public)
  - Output ciphertext $< r, F_k(r) \oplus m >$
- $Dec_k(c_1, c_2)$: output $c_2 \oplus F_k(c_1)$
- Correctness is immediate


r
key
F
pseudorandom
message
ciphertext

Real-world security?

- What happens if an r is ever reused?
- What is the probability that the r used in some challenge ciphertext is also used for some other ciphertext?
- What happens to the bound if the r is chosen non-uniformly?

Real-world security?

- What happens if an r is ever reused?
- What is the probability that the r used in some challenge ciphertext is also used for some other ciphertext?
- What happens to the bound if the r is chosen non-uniformly?

Do Not Do Any Of These Things!

PROS and CONS?

PROS and CONS. Discuss

PROS and CONS?

PROS and CONS. Discuss

PRO If F is a pseudorandom function, then this scheme is CPA-secure

Intuition: If the scheme was not CPA-secure can use to predict F and hence F is not psuedorandom.

PRO Can use same key k for t messages, any t.

PROS and CONS?

PROS and CONS. Discuss

PRO If F is a pseudorandom function, then this scheme is CPA-secure

Intuition: If the scheme was not CPA-secure can use to predict F and hence F is not psuedorandom.

PRO Can use same key k for t messages, any t.

CON Only defined for encryption of n-bit messages

CON $Enc_k(m) = < r, F_k(r) \oplus m >$: n bit message requires $2n$ bits.

CAVEAT Can send long message break up into n-bit chunks.

CON To send t n-bits messages requires $2tn$ bits.

k

$c_1, \ldots, c_t$

k

$m_1, \ldots, m_t$

$c_1 \leftarrow \mathrm{Enc}_k(m_1)$

$\ldots$

$c_t \leftarrow \mathrm{Enc}_k(m_t)$

Sending Many Messages

Goal

The method:

$$Enc_k(m) = < r, F_k(r) \oplus m >$$

is secure but to send ONE n-bit message takes $2n$ bits.

Could send t n-bit messages with $2tn$ bits.

Goal: Send t n-bit message with $< (1+\epsilon)tn$ bits

Goal

The method:

$$Enc_k(m) = < r, F_k(r) \oplus m >$$

is secure but to send ONE n-bit message takes $2n$ bits.

Could send t n-bit messages with $2tn$ bits.

Goal: Send t n-bit message with $< (1+\epsilon)tn$ bits

securely!

Electronic Code Book (ECB) mode

1. $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
   - Send $(F_k(m_1), \ldots, F_k(m_t))$
2. Decryption? Discuss

Electronic Code Book (ECB) mode

1. $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
   - Send $(F_k(m_1), \ldots, F_k(m_t))$
2. Decryption? Discuss
   - Decryption requires F_k to be invertible. Thats fine.

Electronic Code Book (ECB) mode

1. $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
   - Send $(F_k(m_1), \ldots, F_k(m_t))$
2. Decryption? Discuss
   - Decryption requires F_k to be invertible. Thats fine.
3. To send t n-bit messages, send t n-bit messages. Only tn bits!

Electronic Code Book (ECB) mode

1. $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
   - Send $(F_k(m_1), \ldots, F_k(m_t))$
2. Decryption? Discuss
   - Decryption requires F_k to be invertible. Thats fine.
3. To send t n-bit messages, send t n-bit messages. Only tn bits!
4. Drawbacks

Electronic Code Book (ECB) mode

1. $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
   - Send $(F_k(m_1), \ldots, F_k(m_t))$
2. Decryption? Discuss
   - Decryption requires F_k to be invertible. Thats fine.
3. To send t n-bit messages, send t n-bit messages. Only tn bits!
4. Drawbacks This is idiotic! Deterministic!

Not CPA secure. Not EAV-secure. So why used?

Electronic Code Book (ECB) mode

Not CPA secure. Not EAV-secure. So why used?
(1) Was originally used before security was formalized

Electronic Code Book (ECB) mode

Not CPA secure. Not EAV-secure. So why used?
(1) Was originally used before security was formalized

(2) Used today because people are stupid

Electronic Code Book (ECB) mode

Not CPA secure. Not EAV-secure. So why used?
(1) Was originally used before security was formalized

(2) Used today because people are stupid

(3) Half of the apps in the Android App Store use this.

Electronic Code Book (ECB) mode

Not CPA secure. Not EAV-secure. So why used?
(1) Was originally used before security was formalized

(2) Used today because people are stupid

(3) Half of the apps in the Android App Store use this.

(I have an iphone)

Not just a theoretical problem!

Want that when we transmit a picture secretly, Eve learns nothing, sees a blank screen or all black or something like that.

Not just a theoretical problem!

Want that when we transmit a picture secretly, Eve learns nothing, sees a blank screen or all black or something like that.

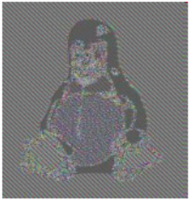
If we transmit a picture using ECB here is what Eve sees:

Not just a theoretical problem!

Want that when we transmit a picture secretly, Eve learns nothing, sees a blank screen or all black or something like that.

If we transmit a picture using ECB here is what Eve sees:

original



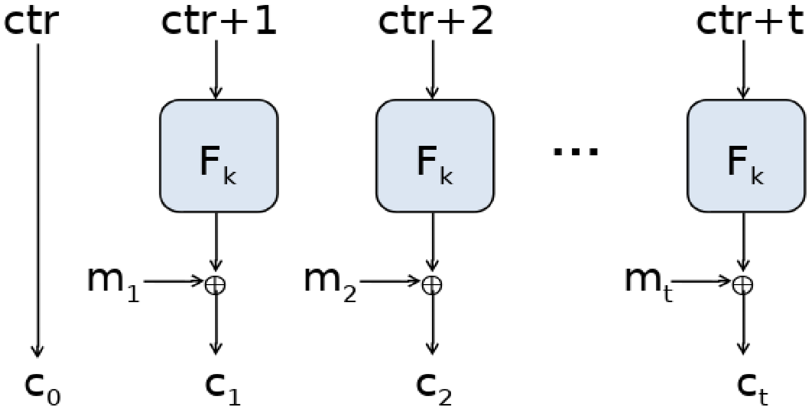
encrypted using ECB mode

(Taken from http://en.wikipedia.org and derived from images created by Larry Ewing (lewing@isc.tamu.edu) using The GIMP.)

Counter (CTR) Mode

- $Enc_k(m_1, \ldots, m_t)$ // note: t is arbitrary
  - Choose $c_0 \leftarrow \{0,1\}^n$
  - For $i = 1$ to t: $c_i = m_i \oplus F_k(c_0 + i \pmod{2^n})$
  - Output $c_0, c_1, \ldots, c_t$
- Decryption? Discuss
- Send t strings by sending one and add to it t times.
- To send t n-bit messages, send $t+1$ n-bit messages.

CTR mode



CTR mode

Theorem: if F is a pseudorandom function, then CTR mode is CPA-secure

Intuition: If CTR is not CPA-secure then can use that to show that to predict F, so F is not pseudorandom.

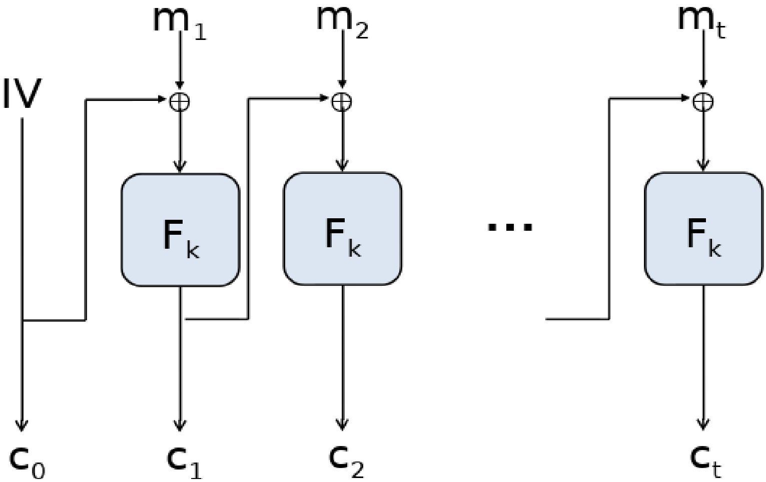
Cipher Block Chaining (CBC) Mode

- $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
  - Choose random $c_0 \leftarrow \{0,1\}^n$ (also called the IV)
  - For $i = 1$ to t: $c_i = F_k(m_i \oplus c_{i-1})$
  - Output $c_0, c_1, \ldots, c_t$
- Decryption? Discuss

Cipher Block Chaining (CBC) Mode

- $Enc_k(m_1, \ldots, m_t)$ //note t is arbitrary
  - Choose random $c_0 \leftarrow \{0,1\}^n$ (also called the IV)
  - For $i = 1$ to t: $c_i = F_k(m_i \oplus c_{i-1})$
  - Output $c_0, c_1, \ldots, c_t$
- Decryption? Discuss
  - Decryption requires F to be invertible
- Send t strings by sending one and $\oplus$.
- To send t n-bit messages, send $t + 1$ n-bit messages.

CBC mode



CBC mode

Theorem: If F is a pseudorandom permutation, the CBC mode is CPA-secure

Intuition: If CBC is not CPA-secure then can use that to show that to predict F, so F is not pseudorandom.