PRPs and PRFs

Final exam: March 22, 3 hours in a 24 hour window (or 3:30pm in Gates B01)
Quick Recap

A **block cipher** is a pair of efficient algs. \((E, D)\):

![Diagram showing a block cipher with inputs and outputs labeled.](image)

Canonical examples:

1. **AES**: \(n=128\) bits, \(k = 128, 192, 256\) bits
2. **3DES**: \(n=64\) bits, \(k = 168\) bits  (historical)
Block Ciphers Built by Iteration

R(k,m) is called a round function

3DES: n=48, AES128: n=10, AES256: n=14
AES: an iterated Even-Mansour cipher

\[ \begin{align*}
\text{input} & \oplus k_0 \rightarrow \pi \\
\text{invertible} & \\
\text{key expansion:} & \\
\text{key} & \rightarrow k_1 \rightarrow k_2 \rightarrow \cdots \rightarrow k_{d-1} \rightarrow k_d \\
\text{output} & \\
\end{align*} \]

single round EM

\[ \pi : \{0,1\}^n \rightarrow \{0,1\}^n \text{ invertible function} \]
AES128: 10 rounds of EM

input

(1) ByteSub
(2) ShiftRow
(3) MixColumn

inverted

k₀

key

16 bytes

key expansion: 16 bytes → 176 bytes

10 rounds

k₁

k₂

k₉

k₁₀

output

4

4

4

10 rounds of EM
The permutation $\pi$

1. **ByteSub**: a 1 byte S-box. 256 byte table. (invertible)

2. **ShiftRows**:

3. **MixColumns**:
Recall the AES pledge

I promise that I will not implement AES myself in production code, even though it might be fun. This agreement will remain in effect until I learn all about side-channel attacks and countermeasures to the point where I lose all interest in implementing AES myself.
## Performance
(no HW acceleration)

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Block/key size</th>
<th>Speed (MB/sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ChaCha20</td>
<td>- / 256</td>
<td>643</td>
</tr>
<tr>
<td>3DES</td>
<td>64 / 168</td>
<td>30</td>
</tr>
<tr>
<td>AES128</td>
<td>128 / 128</td>
<td>163</td>
</tr>
<tr>
<td>AES256</td>
<td>128 / 256</td>
<td>115</td>
</tr>
</tbody>
</table>
AES-NI: AES hardware instructions

AES instructions (Intel, AMD, ARM, …)

• **aesenc, aesenclast**: do one round of AES
  
  128-bit registers: \( \text{xmm1}=\text{state}, \quad \text{xmm2}=\text{round key} \)

  ```
  aesenc \text{xmm1, xmm2}; \text{ puts result in xmm1}
  ```

• **aesdec, aesdeclast**: one round of AES\(^{-1}\)

• **aeskeygenassist**: performs AES key expansion

Claim 1: 14 x speed-up over OpenSSL on same hardware

Claim 2: constant time execution
AES-NI: parallelism and pipelining

- Intel Skylake (old): 4 cycles for one aesenc
  - fully pipelined: can issue one instruction every cycle

- Intel Icelake (2019): vectorized aesenc (vaesenc)
  - vaesenc: compute aesenc on four blocks in parallel
  - fully pipelined

Implications:

- AES128 encrypt a single block takes 40 cycles (10 rounds)
- AES128 encrypt 16 blocks on Icelake takes 43 cycles
AES128 encrypt on Icelake

To encrypt 16 blocks do: \( m_0, \ldots, m_{15} \in \{0,1\}^{128} \)

\[
\begin{array}{cccc}
m_0 & m_1 & m_2 & m_3 \\
\end{array}
\]

(vaesenc)

\[
\begin{array}{cccc}
m_4 & m_5 & m_6 & m_7 \\
\end{array}
\]

(vaesenc)

\[
\begin{array}{cccc}
m_8 & m_9 & m_{10} & m_{11} \\
\end{array}
\]

(vaesenc)

\[
\begin{array}{cccc}
m_{12} & m_{13} & m_{14} & m_{15} \\
\end{array}
\]

(vaesenc)

\[
\begin{array}{cccc}
m_0' & m_1' & m_2' & m_3' \\
\end{array}
\]

(vaesenc)

\[
\begin{array}{cccc}
m_4' & m_5' & m_6' & m_7' \\
\end{array}
\]

(vaesenc)

... finish all 10 rounds after 43 cycles
PRPs and PRFs

Topics:
1. Abstract block ciphers: PRPs and PRFs
2. Security models for encryption
3. Analysis of CBC and counter mode
PRPs and PRFs

• Pseudo Random Function (PRF) defined over (K,X,Y):
  \[ F: K \times X \rightarrow Y \]
such that exists “efficient” algorithm to evaluate \( F(k,x) \)

• Pseudo Random Permutation (PRP) defined over (K,X):
  \[ E: K \times X \rightarrow X \]
such that:
  1. Exists “efficient” algorithm to evaluate \( E(k,x) \)
  2. The function \( E(k, \cdot) \) is one-to-one
  3. Exists “efficient” inversion algorithm \( D(k,x) \)
Running example

• Example PRPs: 3DES, AES, ...

  AES128: \( K \times X \rightarrow X \) where \( K = X = \{0,1\}^{128} \)

  DES: \( K \times X \rightarrow X \) where \( X = \{0,1\}^{64} \), \( K = \{0,1\}^{56} \)

  3DES: \( K \times X \rightarrow X \) where \( X = \{0,1\}^{64} \), \( K = \{0,1\}^{168} \)

• Functionally, any PRP is also a PRF.
  – A PRP is a PRF where \( X=Y \) and is efficiently invertible
  – A PRP is sometimes called a block cipher
Secure PRFs

• Let $F: K \times X \rightarrow Y$ be a PRF

$$F_{\text{uns}[X,Y]}: \text{the set of all functions from } X \text{ to } Y$$

$$S_F = \{ F(k, \cdot) \text{ s.t. } k \in K \} \subseteq F_{\text{uns}[X,Y]}$$

• **Intuition**: a PRF is **secure** if a random function in $F_{\text{uns}[X,Y]}$ is indistinguishable from a random function in $S_F$
Secure PRFs

- Let $F: K \times X \rightarrow Y$ be a PRF

Funs$[X,Y]$: the set of all functions from $X$ to $Y$

$S_F = \{ F(k,\cdot) \text{ s.t. } k \in K \} \subseteq Funs[X,Y]$

**Intuition:** a PRF is **secure** if a random function in $Funs[X,Y]$ is indistinguishable from a random function in $S_F$
Secure PRF: definition

• For \( b=0,1 \) define experiment \( \text{EXP}(b) \) as:

  - \( b=0: \) \( k \leftarrow K, \ f \leftarrow F(k,\cdot) \)
  - \( b=1: \) \( f \leftarrow \text{Funs}[X,Y] \)

• **Def:** \( F \) is a **secure PRF** if for all “efficient” \( \mathcal{A} \):

  \[
  \text{Adv}_{\text{PRF}}[\mathcal{A},F] = \left| \Pr[\text{EXP}(0) = 1] - \Pr[\text{EXP}(1) = 1] \right|
  \]

  is “negligible.”
An example

Let $K = X = \{0,1\}^n$.

Consider the PRF: $F(k, x) = k \oplus x$ defined over $(K, X, X)$

Let's show that $F$ is insecure:

Adversary $\mathcal{A}$:
1. choose arbitrary $x_0 \neq x_1 \in X$
2. query for $y_0 = f(x_0)$ and $y_1 = f(x_1)$
3. output `0' if $y_0 \oplus y_1 = x_0 \oplus x_1$, else `1'

$\Pr[\text{EXP}(0) = 0] = 1$

$\Pr[\text{EXP}(1) = 0] = 1/2^n$

$\Rightarrow \text{Adv}_{\text{PRF}}[\mathcal{A}, F] = 1 - (1/2^n)$ (not negligible)
Secure PRP

• For $b=0,1$ define experiment $\text{EXP}(b)$ as:

  - $\text{Chal.}$
    - $b=0$: $k \leftarrow K$, $f \leftarrow E(k, \cdot)$
    - $b=1$: $f \leftarrow \text{Perms}[X]$
  - $\text{Adv. } \mathcal{A}$
    - $x_i \in X$
    - $f(x_i)$

  - $b' \in \{0,1\}$

• **Def:** $E$ is a secure PRP if for all “efficient” $\mathcal{A}$:

$$\text{Adv}_{\text{PRP}}[\mathcal{A}, E] = \left| \Pr[\text{EXP}(0) = 1] - \Pr[\text{EXP}(1) = 1] \right|$$

is “negligible.”
Example secure PRPs

- **Example secure PRPs:** 3DES, AES, ...

  AES256: \( K \times X \rightarrow X \)  
  where \( X = \{0,1\}^{128} \)  
  \( K = \{0,1\}^{256} \)

- **AES256 PRP Assumption** (example):

  For all \( \mathcal{A} \) s.t. \( \text{time}(\mathcal{A}) < 2^{80} \) :  
  \( \text{Adv}_{\text{PRP}}[\mathcal{A}, \text{AES256}] < 2^{-40} \)
The PRP-PRF Switching Lemma

Any secure PRP is also a secure PRF.

**Lemma:** Let $E$ be a PRP over $(K, X)$. Then for any $q$-query adversary $\mathcal{A}$:

$$|\text{Adv}_{\text{PRF}}[\mathcal{A}, E] - \text{Adv}_{\text{PRP}}[\mathcal{A}, E]| < \frac{q^2}{2|X|}$$

$\Rightarrow$ Suppose $|X|$ is large so that $\frac{q^2}{2|X|}$ is “negligible”

Then $\text{Adv}_{\text{PRP}}[\mathcal{A}, E]$ “negligible” $\Rightarrow$ $\text{Adv}_{\text{PRF}}[\mathcal{A}, E]$ “negligible”
Using PRPs and PRFs

• **Goal**: build “secure” encryption from a PRP.

• Security is always defined using two parameters:

1. What “**power**” does adversary have?
   
   examples:
   
   • Adv sees only one ciphertext (one-time key)
   
   • Adv sees many PT/CT pairs (many-time key, CPA)

2. What “**goal**” is adversary trying to achieve?
   
   examples:
   
   • Fully decrypt a challenge ciphertext.
   
   • Learn info about PT from CT (semantic security)
Incorrect use of a PRP

Electronic Code Book (ECB):

PT:  \[ m_1 \quad m_2 \quad \ldots \quad \]

CT:  \[ c_1 \quad c_2 \quad \ldots \quad \]

Problem:
  – if \( m_1 = m_2 \) then \( c_1 = c_2 \)
In pictures

An example plaintext

Encrypted with AES in ECB mode

(courtesy B. Preneel)
Modes of Operation for One-time Use Key

Example application:

Encrypted email. New key for every message.
Semantic Security for one-time key

- $E = (E, D)$ a cipher defined over $(K, M, C)$
- For $b=0,1$ define $\text{EXP}(b)$ as:

\[
\text{Adv}_{\text{SS}}[\mathcal{A}, E] = \left| \Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1] \right|
\]

is “negligible.”
Semantic security (cont.)

Sem. Sec. ⇒ no “efficient” adversary learns “info” about PT from a **single** CT.

Example: suppose efficient $A$ can deduce LSB of PT from CT. Then $E = (E,D)$ is not semantically secure.

Then $\text{Adv}_{SS}[B, E] = 1 \implies E$ is not sem. sec.
Note: ECB is not Sem. Sec.

ECB is not semantically secure for messages that contain two or more blocks.

If $c_1 = c_2$ output 1, else output 0

Then $\text{Adv}_{SS}[\mathcal{A}, \text{ECB}] = 1$
Examples of sem. sec. systems:

1. $\text{Adv}_{SS}[\mathcal{A}, \text{OTP}] = 0$ for all $\mathcal{A}$

2. Deterministic counter mode from a PRF $F$:
   - $E_{\text{DETCCTR}}(k,m) =$
   
   $m[0]$  $m[1]$  …  $m[L]$

   $\oplus$

   $F(k,0)$  $F(k,1)$  …  $F(k,L)$

   $c[0]$  $c[1]$  …  $c[L]$

   - Stream cipher built from PRF (e.g. AES)
Det. counter-mode security

**Theorem:** For any $L > 0$.

If $F$ is a secure PRF over $(K,X,X)$ then $E_{\text{DETCTR}}$ is sem. sec. cipher over $(K,X^L,X^L)$.

In particular, for any adversary $\mathcal{A}$ attacking $E_{\text{DETCTR}}$ there exists a PRF adversary $B$ s.t.:

$$\text{Adv}_{\text{SS}}[\mathcal{A}, E_{\text{DETCTR}}] = 2 \cdot \text{Adv}_{\text{PRF}}[B, F]$$

$\text{Adv}_{\text{PRF}}[B, F]$ is negligible (since $F$ is a secure PRF)

$$\Rightarrow \text{Adv}_{\text{SS}}[\mathcal{A}, E_{\text{DETCTR}}] \text{ must be negligible.}$$
Modes of Operation for Many-time Key

Example applications:
1. File systems: Same AES key used to encrypt many files.
2. IPsec: Same AES key used to encrypt many packets.
Semantic Security for many-time key  

(CPA security)

Cipher $E = (E,D)$ defined over $(K,M,C)$. For $b=0,1$ define $\text{EXP}(b)$ as:

$$\text{Adv}_{\text{CPA}}[\mathcal{A},E] = |\Pr[\text{EXP}(0)=1] - \Pr[\text{EXP}(1)=1]|$$

is "negligible."
Security for many-time key

Fact: stream ciphers are insecure under CPA.

- More generally: if $E(k,m)$ always produces same ciphertext, then cipher is insecure under CPA.

If secret key is to be used multiple times $\Rightarrow$

given the same plaintext message twice, the encryption alg. must produce different outputs.
Nonce-based Encryption

nonce $n$: a value that changes from msg to msg
(k,n) pair never used more than once

- **method 1**: encryptor chooses a random nonce, $n \leftarrow \mathcal{N}$
- **method 2**: nonce is a counter (e.g. packet counter)
  - used when encryptor keeps state from msg to msg
  - if decryptor has same state, need not send nonce with CT
Construction 1: CBC with random nonce

Cipher block chaining with a random IV (IV = nonce)

note: CBC where attacker can predict the IV is not CPA-secure. HW.
**CBC: CPA Analysis**

**CBC Theorem:** For any $L>0$,  
If $E$ is a secure PRP over $(K,X)$ then  
$E_{CBC}$ is a sem. sec. under CPA over $(K, X^L, X^{L+1})$.

In particular, for a $q$-query adversary $A$ attacking $E_{CBC}$  
there exists a PRP adversary $B$ s.t.:  

$$\text{Adv}_{\text{CPA}}[A, E_{CBC}] \leq 2 \cdot \text{Adv}_{\text{PRP}}[B, E] + 2 \frac{q^2 L^2}{|X|}$$

**Note:** CBC is only secure as long as  
$q^2L^2 \ll |X|$  

# messages enc. with key  
max msg length
Construction 1’: CBC with unique nonce

Cipher block chaining with unique IV  \((IV = \text{nonce})\)

unique IV means:  \((\text{key}, IV)\) pair is used for only one message

\[
\begin{align*}
\text{IV} & \quad \text{m}[0] \quad \text{m}[1] \quad \text{m}[2] \quad \text{m}[3] \\
\text{E}(k_2, \cdot) & \quad \text{E}(k_1, \cdot) \quad \text{E}(k_1, \cdot) \quad \text{E}(k_1, \cdot) \\
\text{IV}' & \quad \text{IV}' \quad \text{IV}' \quad \text{IV}' \\
\text{c}[0] & \quad \text{c}[1] \quad \text{c}[2] \quad \text{c}[3] \\
\text{ciphertext} & \quad \text{ciphertext} \quad \text{ciphertext} \quad \text{ciphertext}
\end{align*}
\]
A CBC technicality: padding

TLS 1.0: if need n-byte pad, n>0, use: 

pad is removed during decryption

if no pad needed, add a dummy block
Construction 2: rand ctr-mode

F: PRF defined over \((K,X,Y)\) where \(X = \{0,1,\ldots,2^n-1\}\) and \(Y = \{0,1\}^n\)

\((e.g., \ n=128)\)

IV - chosen at random for every message

note: parallelizable (unlike CBC)
Why is this CPA secure?

CPA security holds as long as intervals do not intersect

- \( q \) msgs, \( L \) blocks each \( \Rightarrow \) \( \Pr[\text{intersection}] \leq 2 q^2 L / |X| \)

needs to be negligible
rand ctr-mode: CPA analysis

Randomized counter mode: random IV.

**Counter-mode Theorem:** For any L > 0,
If F is a secure PRF over (K,X,X) then
\( E_{\text{CTR}} \) is a sem. sec. under CPA over \((K,X^L,X^{L+1})\).

In particular, for a q-query adversary A attacking \( E_{\text{CTR}} \)
there exists a PRF adversary B s.t.:

\[
\text{Adv}_{\text{CPA}}[A, E_{\text{CTR}}] \leq 2 \cdot \text{Adv}_{\text{PRF}}[B, F] + 2q^2 L / |X|
\]

**Note:** ctr-mode only secure as long as \( q^2 \cdot L \ll |X| \)

Better then CBC!
An example

\[ \text{Adv}_{\text{CPA}}[A, E_{\text{CTR}}] \leq 2 \cdot \text{Adv}_{\text{PRF}}[B, E] + 2 q^2 L / |X| \]

\( q = \# \) messages encrypted with \( k \), \( L = \) length of max msg

Suppose we want \( \text{Adv}_{\text{CPA}}[A, E_{\text{CTR}}] \leq 1/2^{31} \)

• Then need: \( q^2 L / |X| \leq 1/2^{32} \)

• AES: \( |X| = 2^{128} \Rightarrow q L^{1/2} < 2^{48} \)

So, after \( 2^{32} \) CTs each of \( \text{len} \ 2^{32} \), must change key
(total of \( 2^{64} \) AES blocks)
Construction 2’: nonce ctr-mode

To ensure $F(k,x)$ is never used more than once, choose IV as:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>IV:</td>
<td>nonce</td>
<td>00000000</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>96 bits</td>
<td>32 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV+1:</td>
<td>nonce</td>
<td>00000001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IV+2:</td>
<td>nonce</td>
<td>00000002</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IV starts at 0 for every msg.
## Comparison: ctr vs. CBC

<table>
<thead>
<tr>
<th></th>
<th>CBC</th>
<th>ctr mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>required primitive</td>
<td>PRP</td>
<td>PRF</td>
</tr>
<tr>
<td>parallel processing</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>security</td>
<td>$q^2 L^2 \ll</td>
<td>X</td>
</tr>
<tr>
<td>dummy padding block</td>
<td>Yes*</td>
<td>No</td>
</tr>
<tr>
<td>1 byte msgs (nonce-based)</td>
<td>16x expansion</td>
<td>no expansion</td>
</tr>
</tbody>
</table>

* for CBC, dummy padding block can be avoided using ciphertext stealing
Summary

PRPs and PRFs: a useful abstraction of block ciphers.

We examined two security notions:

1. Semantic security against one-time.
2. Semantic security against many-time CPA.

Note: neither mode ensures data integrity.

Stated security results summarized in the following table:

<table>
<thead>
<tr>
<th>Goal</th>
<th>Power</th>
<th>one-time key</th>
<th>Many-time key (CPA)</th>
<th>CPA and CT integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sem. Sec.</td>
<td>steam-ciphers det. ctr-mode</td>
<td>rand CBC</td>
<td>rand ctr-mode</td>
<td>later</td>
</tr>
</tbody>
</table>
Attacks on block ciphers

**Goal:** distinguish block cipher from a random permutation

- if this can be done efficiently then block cipher is broken

Harder goal:

find key $k$ given many $c_i = E(k, m_i)$ for random $m_i$
(1) Linear and differential attacks
[BS’89,M’93]

Given many \((m_i, c_i)\) pairs, can recover key much faster than exhaustive search

**Linear cryptanalysis** (overview): let \(c = DES(k, m)\)

Suppose for random \(k, m:\)

\[
\Pr\left[ m[i_1] \oplus \cdots \oplus m[i_r] \oplus c[j_j] \oplus \cdots \oplus c[j_v] = k[l_1] \oplus \cdots \oplus k[l_u] \right] = \frac{1}{2} + \varepsilon
\]

For some \(\varepsilon\).

For DES, this exists with \(\varepsilon = 1/2^{21} \approx 0.0000000477\) !!!
Linear attacks

\[ \Pr[ m[i_1] \oplus \cdots \oplus m[i_r] \oplus c[j] \oplus \cdots \oplus c[j_v] = k[l_1] \oplus \cdots \oplus k[l_u] ] = \frac{1}{2} + \varepsilon \]

**Thm:** given \( \frac{1}{\varepsilon^2} \) random pairs \((m, c=\text{DES}(k, m))\) then

\[ k[l_1] \oplus \cdots \oplus k[l_u] = \text{MAJ}[ m[i_1] \oplus \cdots \oplus m[i_r] \oplus c[j] \oplus \cdots \oplus c[j_v] ] \]

with prob. \( \geq 97.7\% \)

\[ \Rightarrow \text{ with } \frac{1}{\varepsilon^2} \text{ inp/out pairs can find } k[l_1] \oplus \cdots \oplus k[l_u] \text{ in time } \approx \frac{1}{\varepsilon^2} \]
Linear attacks

For DES, $\varepsilon = 1/2^{21} \Rightarrow$

with $2^{42}$ inp/out pairs can find $k[l_1] \oplus \ldots \oplus k[l_u]$ in time $2^{42}$

Roughly speaking: can find 14 key “bits” this way in time $2^{42}$

Brute force remaining $56 - 14 = 42$ bits in time $2^{42}$

Attack time: $\approx 2^{43} (<< 2^{56})$ with $2^{42}$ random inp/out pairs
Lesson

A tiny bit of linearly leads to a $2^{42}$ time attack.

⇒ don’t design ciphers yourself  !!
(2) Side channel attacks on software AES

Attacker measures the time to compute AES128(k,m) for many random blocks m.

- Suppose that the 256-byte S table is not in L1 cache at the start of each invocation
  \[ \Rightarrow \text{time to encrypt reveals the order in which S entries are accessed} \]
  \[ \Rightarrow \text{leaks info. that can compromise entire key} \]

Lesson: don’t implement AES yourself!

Mitigation: AES-NI or use vetted software (e.g., BoringSSL)
(3) Quantum attacks

Generic search problem:

Let $f: X \to \{0,1\}$ be a function.

Goal: find $x \in X$ s.t. $f(x)=1$.

Classical computer: best generic algorithm time $= O(|X|)$

Quantum computer [Grover ’96]: time $= O(|X|^{1/2})$

(requires a long running quantum computation)
Quantum exhaustive search

Given $m, c = E(k,m)$ define

$$f(k) = \begin{cases} 1 & \text{if } E(k,m) = c \\ 0 & \text{otherwise} \end{cases}$$

Grover $\Rightarrow$ quantum computer can find $k$ in time $O( |K|^{1/2} )$

AES128: quantum key recovery time $\approx 2^{64}$

Adversary has access to a quantum computer $\Rightarrow$

encrypt data using a cipher with 256-bit keys (AES256)
THE END
Recap

**Secure PRF:** \( F : K \times X \rightarrow Y \) and
\[
\{ f(x) = F(k,x) \text{ for } k \leftarrow K \} \text{ is indist. from random } f \text{ in } \text{Funs}[X,Y]
\]

**Secure PRP:** \( E : K \times X \rightarrow X \), efficiently invertible, and
\[
\{ \pi(x) = E(k,x) \text{ for } k \leftarrow K \} \text{ is indist. from random } \pi \text{ in } \text{Perms}[X]
\]

How to use a secure PRF and a secure PRP for encryption?

- One-time key (semantic security): det. CTR-mode
- Many-time key (CPA security):
  - nonce-based CBC, nonce-based CTR mode