Lecture 8
Outline

Deterministic Encryption
Format-Preserving Encryption
Number Theory
Deterministic Encryption
Deterministic Encryption

Server

\[ k_1, k_2 \]

```
<table>
<thead>
<tr>
<th>Alice</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

??

Database (untrusted)

```
<table>
<thead>
<tr>
<th>Bob</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

encrypted database
Deterministic Encryption

Later:

Retrrieve record \( E(k_1, \text{“Alice”}) \)

<table>
<thead>
<tr>
<th>Alice</th>
<th>data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob</td>
<td>data</td>
</tr>
</tbody>
</table>

encrypted database

det. enc. enables later lookup
Deterministic Encryption

The problem: attacker can tell when two ciphertexts encrypt the same message \( \Rightarrow \) leaks information

Leads to significant attacks when message space \( M \) is small.
Deterministic Encryption is NOT CPA-secure

The problem: attacker can tell when two ciphertexts encrypt the same message $\Rightarrow$ leaks information

Leads to significant attacks when message space $M$ is small.

Attacker wins CPA game:

$\begin{align*}
&b \\
&k \leftarrow K \\
&m_0, m_0 \in M \\
&c_0 \leftarrow E(k, m_0) \\
&m_0, m_1 \in M \\
&c \leftarrow E(k, m_b) \\
&\text{output } 0 \text{ if } c = c_0
\end{align*}$
Deterministic Encryption

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

$\text{Def: } E \text{ is sem. sec. under det. CPA if for all efficient } A,$

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

is negligible.

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

\[
\text{Chal.} \quad k \leftarrow K
\]

\[
\text{Adv.} \quad b' \in \{0,1\}
\]

for $i=1,\ldots,q$:

$m_{i,0}, m_{i,1} \in M : |m_{i,0}| = |m_{i,1}|$

$c_i \leftarrow E(k, m_{i,b})$

where $m_{1,0}, \ldots, m_{q,0}$ are distinct and $m_{1,1}, \ldots, m_{q,1}$ are distinct
Is counter mode with a fixed IV det. CPA secure?

IV is set to a fixed string. (not randomized)

Derive IV as det. function of msg.
Construction: Synthetic IV (SIV)

Let \((E, D)\) be a CPA-secure encryption. \(E(k, m ; r) \rightarrow c\)

Let \(F:K \times M \rightarrow R\) be a secure PRF

Define: \(E_{\text{det}}((k_1,k_2), m) = \text{First compute } r = F(k_2, m)\)

\[
\text{output } E(k_1, m; r) \downarrow \text{IV}
\]

**Thm:** \(E_{\text{det}}\) is sem. sec. under det. CPA.

Proof sketch: distinct msgs. \(\Rightarrow\) all r’s are indist. from random

Well suited for messages longer than one AES block \((16 \text{ bytes})\)
Deterministic Authenticated Encryption

**Goal:** det. CPA security and ciphertext integrity

⇒ **DAE: deterministic authenticated encryption**

Consider a SIV special case: **SIV-CTR**

SIV where cipher is counter mode with rand. IV

\[
\text{Enc}(k_1, k_2, m) \xrightarrow{\text{PRF } F} \text{message} \xrightarrow{\text{CTR mode with PRF } F_{\text{ctr}}} \text{ciphertext}
\]
Deterministic Authenticated Encrypted Encryption

**Thm:** If $F$ is a secure PRF and $\text{CTR}$ from $F_\text{ctr}$ is CPA-secure, then SIV-CTR from $(F, F_\text{ctr})$ provides DAE.

**Given:** $IV' \parallel \text{ciphertext', first decrypt, apply } F(k_1, \text{message'), check } IV'$.
Construction 2: Use a PRP

Let \((E, D)\) be a secure PRP. \(E: K \times X \rightarrow X\)

**Thm:** \((E,D)\) is semi-secure under deterministic CPA.

Proof sketch: let \(f: X \rightarrow X\) be a truly random invertible func.

- In \(\text{EXP}(0)\) adv. sees: \(f(m_{1,0}), \ldots, f(m_{q,0})\)
- In \(\text{EXP}(1)\) adv. sees: \(f(m_{1,1}), \ldots, f(m_{q,1})\)

**Using AES:** Det. CPA secure encryption for 16 byte messages.

Longer messages?? Need PRPs on larger msg spaces...
PRP-based Deterministic Authenticated Encryption

Let $(E, D)$ be a secure PRP. $E: K \times (X \times \{0,1\}^n) \rightarrow X \times \{0,1\}^n$

**Thm:** $1/2^n$ is negligible $\Rightarrow$ PRP-based enc. provides DAE

Proof sketch: suffices to prove ciphertext integrity

But then $\Pr[ \text{LSB}_n(\pi^{-1}(c)) = 0^n ] \leq 1/2^n$
Format-Preserving Encryption
Encrypting Credit Card Numbers

Credit card format: $\text{bbbb bbnn nnnn nnnn}$ (≈ 42 bits)

Goal: end-to-end encryption

Intermediate processors expect to see a credit card number

$\Rightarrow$ encrypted credit card should look like a credit card
Format Preserving Encryption

Given $0 < s \leq 2^n$, build a PRP on $\{0, \ldots, s-1\}$ from a secure PRF $F: K \times \{0,1\}^n \rightarrow \{0,1\}^n$ (e.g. AES)

Then to encrypt a credit card number: $(s = \text{total # credit cards})$

1. map given CC# to $\{0, \ldots, s-1\}$
2. apply PRP to get an output in $\{0, \ldots, s-1\}$
3. map output back a to CC#
Step 1: From $\{0,1\}^n$ to $\{0,1\}^t$

Want PRP on $\{0,\ldots,s-1\}$. Let $t$ be such that $2^{t-1} < s \leq 2^t$.

Method: Feistel with $F' : K \times \{0,1\}^{t/2} \rightarrow \{0,1\}^{t/2}$

(better to use 7 rounds a la Patarin, Crypto’03)
Step 2: From \( \{0,1\}^t \) to \( \{0, \ldots, s-1\} \)

Given PRP \((E,D): K \times \{0,1\}^t \rightarrow \{0,1\}^t\)

we build \((E’,D’): K \times \{0,\ldots,s-1\} \rightarrow \{0,\ldots,s-1\}\)

\[ E'(k, x): \text{on input } x \in \{0,\ldots,s-1\} \text{ do:} \]
\[
    y \leftarrow x; \quad \text{do } \{ y \leftarrow E(k, y) \} \quad \text{until } y \in \{0,\ldots,s-1\}; \quad \text{output } y
\]

Expected \# iterations: 2

\[ 2^t \geq s > 2^{t-1} \]

\[ 10^{16} - 1 \]

\( s \) may not be a power of 2.
Security

Step 2 is tight: For all A, there exists B: \( \text{PRP}_{\text{adv}}[A,E] = \text{PRP}_{\text{adv}}[B,E'] \)

Intuition: For all sets \( Y \subseteq X \), applying the transformation to a random perm. \( \pi: X \rightarrow X \) gives a random perm. \( \pi': Y \rightarrow Y \)

Step 1: same security as Feistel construction

note: no integrity
Number Theory

Public Key Encryption
Notation

From here on:
• \( N \) denotes a positive integer.
• \( p \) denote a prime.

Notation: \( \mathbb{Z}_N = \{0, 1, 2, \ldots, N-1\} \)

Can do addition and multiplication modulo \( N \)

\[ \forall a, b, \quad (a + b) \mod N \in \mathbb{Z}_N \]
Modular arithmetic

Examples:  let $N = 12$

$$9 + 8 = 5 \quad \text{in } \mathbb{Z}_{12}$$

$$5 \times 7 = 11 \quad \text{in } \mathbb{Z}_{12}$$

$$5 - 7 = 10 \quad \text{in } \mathbb{Z}_{12}$$

$$-2 \mod 12 = 10$$

Arithmetic in $\mathbb{Z}_N$ works as you expect, e.g. $x \cdot (y+z) = x \cdot y + x \cdot z$ in $\mathbb{Z}_N$
Greatest common divisor

**Def:** For ints. $x, y$: $\text{gcd}(x, y)$ is the greatest common divisor of $x, y$

**Example:** $\text{gcd}(12, 18) = 6$

**Fact:** for all ints. $x, y$ there exist ints. $a, b$ such that

$$a \cdot x + b \cdot y = \text{gcd}(x, y) \mod N.$$  

$a, b$ can be found efficiently using the extended Euclid alg.

If $\text{gcd}(x, y) = 1$ we say that $x$ and $y$ are **relatively prime** $(\mod N)$.
Modular inversion

Over the rationals, inverse of $2$ is $\frac{1}{2}$. What about $\mathbb{Z}_n$?

**Def:** The inverse of $x$ in $\mathbb{Z}_N$ is an element $y$ in $\mathbb{Z}_N$ s.t. $x \cdot y = 1$ in $\mathbb{Z}_N$.

$y$ is denoted $x^{-1} \pmod{N}$.

Example: let $N$ be an odd integer. The inverse of $2$ in $\mathbb{Z}_N$ is $\frac{N+1}{2}$.

Why $\frac{N+1}{2}$?

$2 \cdot \left(\frac{N+1}{2}\right) \mod N = (N+1) \mod N \equiv 1 \mod N$. 

Modular inversion

Which elements have an inverse in \( \mathbb{Z}_N \)?

**Lemma**: \( x \) in \( \mathbb{Z}_N \) has an inverse if and only if \( \gcd(x, N) = 1 \)

**Proof**: 

\[
\gcd(x, N) = 1 \quad \Rightarrow \quad \exists \ a, b: \ a \cdot x + b \cdot N = 1 \quad \Rightarrow \quad a \cdot X \equiv 1 \pmod{\mathbb{Z}_N}
\]

\[
\gcd(x, N) > 1 \quad \Rightarrow \quad \forall a: \ gcd(a \cdot x, N) > 1 \quad \Rightarrow \quad a \cdot x \neq 1 \quad \text{in} \quad \mathbb{Z}_N
\]

\[
\gcd(x, N) = 2 \quad \Rightarrow \quad \forall a: \ a \cdot x \text{ is even} \quad \Rightarrow \quad \frac{a \cdot x}{a \cdot x} \neq \frac{\text{odd}}{b \cdot N + 1}
\]

\( \exists x \) is not invertible \( \mod N \).
More notation

**Def:** \( \mathbb{Z}_N^* \) = (set of invertible elements in \( \mathbb{Z}_N \)) = 
= \{ x \in \mathbb{Z}_N : \gcd(x,N) = 1 \}

Examples:

1. for prime \( p \), \( \mathbb{Z}_p^* = \mathbb{Z}_p \setminus \{0\} = \{1, 2, \ldots, p - 1\} \)

2. \( \mathbb{Z}_{12}^* = \{1, 5, 7, 11\} \) [all are invertible]

For \( x \) in \( \mathbb{Z}_N^* \), can find \( x^{-1} \) using extended Euclid algorithm.
Solving modular linear equations

Solve: \[ a \cdot x + b = 0 \] in \( \mathbb{Z}_N \)

\[
\Rightarrow a \cdot x = -b \Rightarrow x = -b \cdot a^{-1} = a^{-1}(-b)
\]

Solution: \( x = -b \cdot a^{-1} \) in \( \mathbb{Z}_N \)

Find \( a^{-1} \) in \( \mathbb{Z}_N \) using extended Euclid. Run time: \( O(\log^2 N) \)

What about modular quadratic equations? (next time..)