Announcements/Reminders

HW5 due Nov 10

PR2 due Dec 5
Previously on COS 433...
Trapdoor Permutations

Domain $X$

\[ \text{Gen}(): \text{outputs} \ (pk, sk) \]
\[ F(pk, x \in X) = y \in X \]
\[ F^{-1}(sk, y) = x \]

Correctness:
\[ \Pr[ \ F^{-1}(sk, \ F(pk, \ x)) = x : (pk, sk) \leftarrow \text{Gen}() \ ] = 1 \]

Correctness implies $F, F^{-1}$ are deterministic, permutations
Trapdoor Permutation Security

\[(sk, pk) \leftarrow \text{Gen()}
\]
\[x \leftarrow X
\]
\[y \leftarrow F(pk, x)
\]
\[\text{Adversary wins if } x = x'
\]

In other words, \(F(pk, \cdot)\) is a one-way function
Key Distribution from TDPs

\[(pk, sk) \leftarrow \text{Gen}()\]

\[x \leftarrow h( F^{-1}(sk, y) )\]

\[h( x )\]

\[h\] a hardcore bit for \( F(pk, \cdot) \)
Trapdoor Permutations from RSA

**Gen:**
- Choose random primes \( p,q \)
- Let \( N=pq \)
- Choose \( e,d \) s.t \( ed=1 \mod (p-1)(q-1) \)
- Output \( pk=(N,e), \ sk=(N,d) \)

**F(pk,x):** Output \( y = x^e \mod N \)

**F^{-1}(sk,y):** Output \( x = y^d \mod N \)
Key Distribution from DH

Everyone agrees on group $G$ of prime order $p$

$a \leftarrow \mathbb{Z}_p$  

$b \leftarrow \mathbb{Z}_p$
Key Distribution from DH

Everyone agrees on group $\textbf{G}$ or prime order $p$

$$a \leftarrow \mathbb{Z}_p$$
$$g^a \quad \text{g}^b$$
$$b \leftarrow \mathbb{Z}_p$$
Key Distribution from DH

Everyone agrees on group $G$ or prime order $p$

$a \leftarrow \mathbb{Z}_p$  \[ k = (g^b)^a = g^{ab} \]

$b \leftarrow \mathbb{Z}_p$  \[ k = (g^a)^b = g^{ab} \]
Key Distribution from DH

**Theorem:** If DDH holds on $G$, then the Diffie-Hellman protocol is secure

Proof:
- $(\text{Trans}, k) = (\langle g^a, g^b \rangle, g^{ab})$
- DDH means indistinguishable from $(\langle g^a, g^b \rangle, g^c)$

What if only CDH holds, but DDH is easy?
Today

Public Key Encryption
Digital signatures (if time)
Public Key Encryption
Public Key Encryption
Public Key Encryption

\[ c \leftarrow \text{Enc}(pk, m) \]
Public Key Encryption

\[ m \xleftarrow{} \text{Enc}(pk,m) \xrightarrow{} c \xleftarrow{} \text{Dec}(sk,c) \]
Public Key Encryption

\[ c \leftarrow \text{Enc}(pk,m) \]

\[ m \leftarrow \text{Dec}(sk,c) \]
PKE vs Key Agreement

Key agreement:

$k_{AB}$
PKE vs Key Agreement

Key agreement:

\[ k_{AB} \]
PKE vs Key Agreement

Key agreement:

Alice $k_{AB}$
Bob $k_{AC}$
Charlie $k_{AB}$
PKE vs Key Agreement

Key agreement:

For $n$ users, need $O(n^2)$ key exchanges
PKE vs Key Agreement

PKE:

$sk_A \rightarrow pk_A$
PKE vs Key Agreement

PKE:

$sk_A$

$pk_A$

$pk_B$

$sk_B$
PKE vs Key Agreement

PKE:

\[ \text{sk}_A \]
\[ \text{pk}_A \]
\[ \text{pk}_B \]
\[ \text{sk}_C \]

For \( n \) users, need \( O(n) \) public keys
PKE Syntax

Message space $\mathcal{M}$

Algorithms:
• $(sk, pk) \leftarrow \text{Gen}(\lambda)$
• $\text{Enc}(pk, m)$
• $\text{Dec}(sk, m)$

Correctness:
$\Pr[\text{Dec}(sk, \text{Enc}(pk, m)) = m: (sk, pk) \leftarrow \text{Gen}(\lambda)] = 1$
Security

One-way security

Semantic Security

CPA security

CCA Security
One-way Security

\[(sk, pk) \leftarrow \text{Gen()} \]
\[m \leftarrow M \]
\[c \leftarrow \text{Enc}(pk, m) \]

\[m' \]
Semantic Security

\[(sk, pk) \leftarrow \text{Gen}()\]

\[c \leftarrow \text{Enc}(pk, m_b)\]
CPA Security

\[(sk, pk) \leftarrow \text{Gen}() \]
\[c \leftarrow \text{Enc}(pk, m_b)\]
CCA Security

$(sk, pk) \leftarrow \text{Gen}()$

c $\leftarrow$ Enc$(pk, m_b)$

$pk$

$m$

$m_0, m_1$

$c^*$

$c \neq c^*$

$m$

$b'$
Question: Which two notions are equivalent?
One-Way Encryption from TDPs

\[ \text{Gen}_E() = \text{Gen}_{\text{TDP}}() \]

\[ \text{Enc}(pk,m): \text{Output } c = F(pk,m) \]

\[ \text{Dec}(sk,c): \text{Output } m' = F^{-1}(sk,c) \]
Semantically Secure Encryption from TDPs

Ideas?
Considerations

A single server often has to decrypt many ciphertexts, whereas each user only encrypts a few messages.

Therefore, would like to make decryption fast.
Considerations

Encryption running time:
• $O(\log e)$ multiplications, each taking $O(\log^2 N)$
• Overall $O(\log e \log^2 N)$

Decryption running time:
• $O(\log d \log^2 N)$

(Note that $ed \geq \Phi(N) \approx N$)
Considerations

Possibilities:
• \( e \) tiny (e.g. 3): fast encryption, slow decryption
• \( d \) tiny (e.g. 3): fast decryption, slow encryption
  • Problem?
• \( d \) relatively small (e.g. \( d \approx N^{0.1} \))
  • Turns out, there is an attack that works whenever \( d < N^{.292} \)

Therefore, need \( d \) to be large, but ok taking \( e=3 \)
Considerations

Chinese remaindering to speed up decryption:
• Let \( sk = (d_0, d_1) \) where
  \[ d_0 = d \mod (p-1), \quad d_1 = d \mod (q-1) \]

• Let \( c_0 = c \mod p, \quad c_1 = c \mod q \)
• Compute \( m_0 = c^{d_0} \mod p, \quad m_1 = c^{d_1} \mod q \)
• Reconstruct \( m \) from \( m_0, m_1 \)

Running time:
• \( r \log^3 p + r \log^3 q + O(\log^2 N) \approx r(\log^3 N)/4 \)
ElGamal

Group $G$ of order $p$, generator $g$
Message space = $G$

$Gen()$:  
• Choose random $a \leftarrow \mathbb{Z}_p^*$, let $h \leftarrow g^a$ 
• $pk=h$, $sk=a$

$Enc(pk, m \in \{0,1\})$:  
• $r \leftarrow \mathbb{Z}_p$ 
• $c = (g^r, h^r \times m)$

$Dec?$
**Theorem:** If DDH is hard in $G$, then ElGamal is CPA secure

Proof:
- Adversary sees $h = g^a, g^r, g^{ar} \times m_0$
- DDH: indistinguishable from $g^a, g^r, g^c \times m_0$
- Same as $g^a, g^r, g^c \times m_1$
- DDH again: indistinguishable from $g^a, g^r, g^{ar} \times m_0$
CCA-Secure Encryption

Non-trivial to construct with provable security

Most efficient constructions have heuristic security
CCA Secure PKE from TDPs

Let \((\text{Enc}_{\text{SKE}},\text{Dec}_{\text{SKE}})\) be a CCA-secure secret key encryption scheme.

Let \((\text{Gen},F,F^{-1})\) be a TDP

Let \(H\) be a hash function
CCA Secure PKE from TDPs

\[ \text{Gen}_{\text{PKE}}() = \text{Gen}() \]
\[ \text{Enc}_{\text{PKE}}(pk, m): \]
\[ \begin{align*}
\text{• Choose random } r \\
\text{• Let } c &\leftarrow F(pk, r) \\
\text{• Let } d &\leftarrow \text{Enc}_{\text{SKE}}(H(r), m) \\
\text{• Output } (c, d)
\end{align*} \]

\[ \text{Dec}_{\text{PKE}}(sk, (c, d)): \]
\[ \begin{align*}
\text{• Let } r &\leftarrow F^{-1}(sk, c) \\
\text{• Let } m &\leftarrow \text{Dec}_{\text{SKE}}(H(r), d)
\end{align*} \]
CCA Secure PKE from TDPs

**Theorem:** If \((\text{Enc}_{\text{SKE}}, \text{Dec}_{\text{SKE}})\) is a CCA-secure secret key encryption scheme, \((\text{Gen}, \text{F}, \text{F}^{-1})\) is a TDP, and \(H\) is modeled as a random oracle, then \((\text{Gen}_{\text{PKE}}, \text{Enc}_{\text{PKE}}, \text{Dec}_{\text{PKE}})\) is a CCA secure public key encryption scheme.
Theorem: For RSA TDP, if $G,H$ are modeled as a random oracles, then $(\text{Gen}_{\text{PKE}},\text{Enc}_{\text{PKE}},\text{Dec}_{\text{PKE}})$ is a CCA secure public key encryption scheme.
Insecure OAEP Variants

\[ c = F(pk, (m, O^t, y) ) \]

May contain \( m \) in the clear

- \( F(pk, (m, x, y) ) \)
  \[ = (m, F'(pk, (x, y) ) ) \]
Insecure OAEP Variants

\[ m \oplus O^t \oplus G \oplus F \rightarrow c \]

\[ F \rightarrow \]
Why padding?

All ciphertexts decrypt to valid messages
- Makes it hard to argue security
Digital Signatures
(aka public key MACs)
Message Authentication Codes

Goal: If Eve changed $m$, Bob should reject.
Problem

What if Alice and Bob have never met before to exchange key $k$?

Want: a public key version of MACs where Bob can verify without having Alice’s secret key
Message Integrity in Public Key Setting

Goal: If Eve changed \( m \), Bob should reject
Digital Signatures

Algorithms:

- \text{Gen()} \rightarrow (sk,pk)
- \text{Sign}(sk,m) \rightarrow \sigma
- \text{Ver}(pk,m,\sigma) \rightarrow 0/1

Correctness:

\[ \Pr[\text{Ver}(pk,m,\text{Sign}(sk,m))=1: (sk,pk) \leftrightarrow \text{Gen()}] = 1 \]
Security Notions?

Much the same as MACs, except adversary gets verification key
1-time Security For Signatures

\[(sk, pk) \leftarrow \text{Gen}()\]

\[
m \xrightarrow{\sigma} \sigma \leftarrow \text{Sign}(sk, m)
\]

Output 1 iff:
- \(m^* \neq m\)
- \(\text{Ver}(pk, m^*, \sigma^*) = 1\)

\[1\text{CMA-Adv}(\mathcal{A}) = \Pr[\ \text{outputs 1}]\]
Many-time Signatures

\[(m \sigma, m^*, \sigma^*)\]

Output 1 iff:
- \(m^* \not\in \{m_1, \ldots\}\)
- \(\text{Ver}(pk, m^*, \sigma^*) = 1\)

\[\text{CMA-Adv} = \Pr[\text{outputs 1}]\]
Strong Security

\[
\text{Output 1 iff:}
\begin{align*}
&\cdot (m^*, \sigma^*) \not\in \{(m_1, \sigma_1) \ldots\} \\
&\cdot \text{Ver}(pk, m^*, \sigma^*) = 1
\end{align*}
\]

\[
\text{CMA-Adv(\text{\textbullet\textbullet\textbullet})} = \Pr[ \text{\textbullet\textbullet\textbullet outputs 1}]
\]
Building Digital Signatures

Non-trivial to construct with provable security

Most efficient constructions have heuristic security
Signatures from TDPs?

\[ \text{Gen}_{\text{Sig}}() = \text{Gen}() \]

\[ \text{Sign}(sk,m) = F^{-1}(sk,m) \]

\[ \text{Ver}(pk,m,\sigma): F(pk, \sigma) == m \]
Signatures from TDPs

\( \text{Gen}_{\text{Sig}}() = \text{Gen}() \)

\( \text{Sign}(sk,m) = F^{-1}(sk, H(m)) \)

\( \text{Ver}(pk,m,\sigma): F(pk, \sigma) == H(m) \)

**Theorem:** If \((\text{Gen}, F, F^{-1})\) is a secure TDP, and \(H\) is “modeled as a random oracle”, then \((\text{Gen}_{\text{Sig}}, \text{Sign}, \text{Ver})\) is (strongly) CMA-secure
Basic Rabin Signatures

**Gen** \( \text{Sig}() \): let \( p,q \) be random large primes
\[
\text{sk} = (p,q), \quad \text{pk} = N = pq
\]

**Sign** \( \text{sk}, m \): Solve equation \( \sigma^2 = H(m) \mod N \)
using factors \( p,q \)
- Output \( \sigma \)

**Ver** \( \text{pk}, m, \sigma \): \( \sigma^2 \mod N = H(m) \)
Problems

\( H(m) \) might not be a quadratic residue
Can only sign roughly \( \frac{1}{4} \) of messages

Suppose adversary makes multiple signing queries on the same message
- Receives \( \sigma_1, \sigma_2, \ldots \) such that \( \sigma_i^2 \mod N = H(m) \)
- After enough tries, may get all 4 roots of \( H(m) \)
- Suppose \( \sigma_1 \neq \pm \sigma_2 \mod N \)
- Then \( \text{GCD}(\sigma_1-\sigma_2, N) \) will give a factor
One Solution

**Gen\textsubscript{Sig}()**: let \( p, q \) be primes, \( a, b, c \) s.t.
- \( a \) is a non-residue \( \text{mod} \ p \) and \( q \),
- \( b \) is a residue \( \text{mod} \ p \) but not \( q \),
- \( c \) is a residue \( \text{mod} \ q \) but not \( p \)

\( sk = (p, q, a, b, c) \), \( pk = (N = pq, a, b, c) \)

**Sign(sk,m):**
- Solve equation \( \sigma^2 \in \{1, a, b, c\} \times H(m) \text{ mod } N \)
- Output \( \sigma \)

**Ver(pk,m,\sigma):** \( \sigma^2 \text{ mod } N \in \{1, a, b, c\} \times H(m) \)
One Solution

Exactly one of $\{1,a,b,c\} \times H(m)$ is a residue $\text{mod } N$

$\Rightarrow$ Solution guaranteed to be found

Still have problem that multiple queries on same message will give different roots
One Solution

Possibilities:

• Have signer remember all messages signed

• Choose root that is itself a quadratic residue
  (if $-1$ is not a residue mod $p,q$, there will be exactly one)
Another Solution

**Gen\_Sig()**: let $p,q$ be random large primes
$$sk = (p,q), \quad pk = N = pq$$

**Sign(sk,m)**: Repeat until successful:
- Choose random $u \leftarrow \{0,1\}^\lambda$
- Solve equation $\sigma^2 = H(m,u) \mod N$
- Output $(u,\sigma)$

**Ver(pk,m,(u,\sigma))**: $\sigma^2 \mod N = H(m,u)$
Another Solution

In expectation, after 4 tries will have success

(Whp) Only ever get a single root of a given $H(m,u)$

**Theorem:** If factoring is hard and $H$ is modeled as a random oracle, then Rabin signatures are (weakly) CMA secure
Another Solution

**Sign**(sk,m): Repeat until successful:
- Choose random $u \leftarrow \{0,1\}^\lambda$
- Solve equation $\sigma^2 = H(m,u) \mod N$ using factors $p,q$, where $\sigma < (N-1)/2$
- Output $(u,\sigma)$

**Ver**(pk,m,(u,σ)): $\sigma^2 \mod N = = H(m,u) \land \sigma < (N-1)/2$

**Theorem:** If factoring is hard and $H$ is modeled as a random oracle, then Rabin signatures are strongly CMA secure