Previously

Exchanging keys and public key encryption
Public Key Distribution
Public Key Distribution
Public Key Distribution
Public Key Distribution
Public Key Encryption
Public Key Encryption
Public Key Encryption

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

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

\[ m \leftarrow \text{Dec}(sk, c) \]
Public Key Encryption

\[ m \xleftarrow{\text{Dec}(sk,c)} m \]
One-way Security

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

\[m'\]
Semantic Security

\[(sk, pk) \leftarrow \text{Gen}()\]
\[c \leftarrow \text{Enc}(pk, m_b)\]
Theorem: An encryption scheme \((\text{Gen}, \text{Enc}, \text{Dec})\) is semantically secure if and only if it is CPA secure.
CCA Security

$\text{Gen}(\cdot)$

$\text{Enc}(\cdot)$

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

$c \leftarrow \text{Enc}(pk,m_b)$

$c \neq c^*$

$m^* 

$m^*, m_{0,1}$

$c^*$

$\text{b}$
One-way Encryption from RSA

\textbf{Gen():}
\begin{itemize}
\item Choose random primes \( p, q \)
\item Let \( N = pq \)
\item Choose \( e, d \) s.t \( ed = 1 \mod (p-1)(q-1) \)
\item Output \( pk = (N, e), \ sk = (N, d) \)
\end{itemize}

\textbf{Enc}(pk, m): Output \( c = m^e \mod N \)

\textbf{Dec}(sk, c): Output \( m' = c^d \mod N \)
Goldwasser-Micali

\begin{itemize}
  \item \textbf{Gen()}: \\
    \begin{itemize}
    \item Choose random primes \( p,q \)
    \item Let \( N=pq \)
    \item Choose \( x \) a quadratic non-residue mod \( p \) and \( q \)
    \item Output \( pk=(N,x), sk=(p,q) \)
    \end{itemize}
  \end{itemize}

\begin{itemize}
  \item \textbf{Enc}(pk,\( m \in \{0,1\})\): \( r \leftarrow \mathbb{Z}_N^* \), \( c \leftarrow x^m r^2 \mod N \)
  \begin{itemize}
    \item If \( m=0 \), then \( c \) is a quadratic residue
    \item If \( m=1 \), then \( c \) is a non-residue
  \end{itemize}
\end{itemize}
ElGamal

Group $\mathbf{G}$ of order $p$, generator $g$
Message space = $\mathbf{G}$

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

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

$\text{Dec}$?
Today

Trapdoor Permutations: abstracting RSA

CCA-secure Encryption in the ROM

Begin: digital signatures
Trapdoor Permutations

Domain $\mathcal{X}$

$\text{Gen}()$: outputs $(pk,sk)$

$F(pk, x \in \mathcal{X}) = y \in \mathcal{X}$

$F^{-1}(sk, y) = x$

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

\[(sk, pk) \leftarrow \text{Gen}() \]
\[x \leftarrow X \]
\[y \leftarrow F(pk, x)\]
One-way Encryption from TDPs

\[
\begin{align*}
\text{Gen}_{\text{Enc}}() &= \text{Gen}() \\
\text{Enc}(pk,x) &= F(pk, x) \\
\text{Dec}(sk,c) &= F^{-1}(sk, c)
\end{align*}
\]

Thus, TDPs are special case of one-way encryption where \textbf{Enc} is deterministic and \textbf{C} = \textbf{M}
CPA-Secure Encryption from TDPs

Let $h$ be a hardcore bit for the one-way function $x \mapsto F(pk,x)$

$$\text{Enc}(pk,b) = F(pk,r), \ h(r) \oplus b$$

Constructing TDPs with hardcore bits?
- $F'(pk, (r,x)) = (r, F(pk,x))$
- $h(r,x) = r \oplus b$
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, c):** Output \( x = y^d \mod N \)
Caveats

RSA is not a true TDP as defined
• Why???
• What’s the domain?

Nonetheless, distinction is not crucial to most applications
Other TDPs?

For long time, none known
• Still interesting object:
  • Useful abstraction in protocol design
  • Maybe more will be discovered...

Using obfuscation:
• Let $\mathcal{P}$ be a PRP
• $sk = k$, $pk = \text{Obf}(\mathcal{P}(k, \cdot))$
Relaxation: Injective Trapdoor Functions

Domain $X$, range $Y$

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

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

Correctness implies $F$ is injective
Injective TDFs from DH

Notation:

Let \( A \in \mathbb{Z}_p^{n \times n} \)
\( g^A \in G^{n \times n}, (g^A)_{i,j} := g^{A_{i,j}} \)

Let \( H \in G^{n \times n}, v \in \mathbb{Z}_p^n \)
\( H^v \in G^n, (H^v)_{i} := \prod_j H_{i,j}^{v_j} \)

Note: \((g^A)^v_i = \prod_j g^{A_{i,j}v_j} = g^{(A \cdot v)_i}, \text{ so } (g^A)^v = g^{A \cdot v}\)
Injective TDFs from DH

Notation:

Let \( h \in G^n, A \in \mathbb{Z}_q^{n \times n} \)

\( A h \in G^n, (A h)_i := \prod_j h_j^{A_i,j} \)

\( A(g^v)? \)

\( (A(g^v))_i = \prod_j g^{A_i,j \cdot v_j} = g^{(A \cdot v)_i}, \text{ so } A(g^v) = g^{A \cdot v} \)
Injective TDFs from DH

First Attempt:

\[ \text{Gen}(): \text{ choose random } A \leftarrow \mathbb{Z}_q^{n \times n} \]

\[ \text{sk} = A, \text{ pk} = H = g^A \]

\[ F(\text{pk}, x \in \mathbb{Z}_q^n) = H^x \left( = g^{A \cdot x} \right) \]

\[ F^{-1}(\text{sk}, h): y \leftarrow A^{-1} h \left( = g^{A^{-1} \cdot A \cdot x} = g^x \right) \]

Then Dlog???????
Injective TDFs from DH

**Theorem:** If DDH holds, then \((\text{Gen}, F, F^{-1})\) is an injective TDF

\(\text{Gen}(): \) choose random \(A \leftarrow \mathbb{Z}_q^{n \times n}\)

\(sk = A, \ pk = H = g^A\)

\(F(pk, x \in \{0,1\}^n) = H^x (= g^{A \cdot x})\)

\(F^{-1}(sk, h): y \leftarrow A^{-1}h (= g^{A^{-1} \cdot A \cdot x} = g^x )\)

Then Dlog each component to recover \(x\)
Injective TDFs

Known constructions from most number theoretic problems

Useful in many cases where TDPs are used (but not all)
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 (we’ll pretend it’s a random oracle)
CCA Secure PKE from TDPs

\textit{Gen}_{PKE}() = \textit{Gen}()

\textit{Enc}_{PKE}(pk, m):
- Choose random \( r \)
- Let \( c_0 \leftarrow F(pk, r) \)
- Let \( c_1 \leftarrow \text{Enc}_{SKE}(H(r), m) \)
- Output \((c_0, c_1)\)

\textit{Dec}_{PKE}(sk, (c_0, c_1)):
- Let \( r \leftarrow F^{-1}(sk, c_0) \)
- Let \( m \leftarrow \text{Dec}_{SKE}(H(r), c_1) \)
CCA Secure PKE from TDPs

**Theorem:** If \((\text{Enc}_{\text{SKE}}, \text{Dec}_{\text{SKE}})\) is a CCA-secure secret key encryption scheme, \((\text{Gen}, F, F^{-1})\) 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.
Proof

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

$$r \leftarrow H^{-1}(sk, c_0)$$
$$m \leftarrow \text{Dec}_{SKE}(H(r), c_1)$$

$$(c_0, c_1)$$
Proof

Random $r^*$

$m_0^*, m_1^* \leftarrow F(pk, r^*)$
$c_0^* \leftarrow Enc_{SKE}(H(r^*), m_b^*)$

$c_1^* \leftarrow Enc_{SKE}(H(r^*), m_b^*)$
Proof

\[ (c_0, c_1) \neq (c_0^*, c_1^*) \]

\[ r \leftarrow F^{-1}(sk, c_0) \]
\[ m \leftarrow \text{Dec}_{\text{SKE}}(H(r), c_1) \]
Proof

Step 1: sample $H$ as follows:
- Choose a random function $H'$
- Let $H(x) = H'(F(pk, x))$

Since $F(pk, \cdot)$ is a permutation, all outputs of $H(x)$ are independent and uniform

Therefore, $H(x)$ is still a random oracle
Proof

\[ y = F(pk, x) \]

\[ z = H'(y) \]
Proof

\[ y = F(pk, r) \]

\[ z = H'(y) \]

\[ r \leftarrow F^{-1}(sk, c_0) \]

\[ m \leftarrow \text{Dec}_{SKE}(z, c_1) \]
Proof

\[ (c_0, c_1) \]

\[ \text{Dec}_{SKE}(z, c_1) \]

\[ m \leftrightarrow H \]

\[ H' \]
Proof

\[ y^* = F(pk, r^*) \]
\[ z^* = H'(y^*) \]
\[ z^* \]

\[ H \]
\[ H' \]

Random \( r^* \)

\[ c_0^* \leftarrow F(pk, r^*) \]
\[ c_1^* \leftarrow Enc_{SKE}(z^*, m_b^*) \]

\[ m_0^*, m_1^* \]

\[ (c_0^*, c_1^*) \]
Proof

\[ H \]

\[ H' \]

\[ m_0^*, m_1^* \]

\[ (c_0^*, c_1^*) \]

Random \( c_0^* \)

\[ c_1^* \leftarrow \text{Enc}_{\text{SKE}}(z^*, m_b^*) \]

Observation: now Charlie doesn’t need \textbf{sk} to run experiment
Proof

Consider two cases:

Case 1: adversary makes a RO query to $H$ on
\[ r^* = F^{-1}(sk, c_0^*) \]

Case 2: adversary never makes a RO query on $r^*$
Proof

Case 1: construct TDP adversary 🤖
Proof

Case 1: construct TDP adversary

\[ (c_0, c_1) \overset{m}{\leftarrow} \text{Dec}_{\text{SKE}}(z, c_1) \]

\[ H' \]

\[ c_0, z \]

\[ \text{pk}, y^* \]
Proof

Case 1: construct TDP adversary

\[ p_k, y^* \]

\[ H' \]

\[ m_0^*, m_1^* \]

\[ (c_0^* = y^*, c_1^*) \]

\[ c_1^* \leftarrow \text{Enc}_{\text{SKE}}(z^*, m_b^*) \]
Proof

Case 1: construct TDP adversary

Proof:

\begin{align*}
  &\text{If } y = y^*, \text{ output } x
\end{align*}
Proof

Case 2: construct \textbf{Enc}_{\text{SKE}} \text{ adversary}

\[ y = F(pk, x) \]
\[ z = H'(y) \]

If \( y = c_0^* \), abort
Proof

Case 2: construct $\text{Enc}_{\text{SKE}}$ adversary.

Proof:

Case 2: construct $\text{Enc}_{\text{SKE}}$ adversary.

Random $c_0^*$

$H'$

$m_0^*, m_1^*$

$(c_0^*, c_1^*)$
Case 2: construct $Enc_{SKE}$ adversary

Proof

$H'(c_0, c_1) \xrightarrow{m \leftarrow Dec_{SKE}(z, c_1)} c_0 \neq c_0^*$

If $c_0 = c_0^*$:

Random $c_0^*$
Proof

Case 2: construct $\text{Enc}_{\text{SKE}}$ adversary

Analysis:
• Effectively set $H'(c_0^*) = k$, where $k$ is (unknown) challenger key
• Answers all queries correctly, provided adversary never queries RO on $c_0^*$
• Therefore, breaks security of $\text{Enc}_{\text{SKE}}$ in case 2
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

\[ c = F(pk, (m, 0^t, y)) \]

May contain \( m \) in the clear

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

\[ m \oplus G(0^t) \Rightarrow r \Rightarrow F(pk) \Rightarrow c \]
Why padding?

All ciphertexts decrypt to valid messages
• Makes it hard to argue security
High Level Proof Sketch

\[ \text{pk, } y^* \]
Claim: For any valid ctxt c queried by adv, adv must have created c by running $\text{Enc}(pk, m; r)$. In this case, m can be decoded by looking at queries to G, H.
Advantages of RSA-OAEP

RSA domain is at least 2048 bits

In hybrid encryption, ciphertext overhead = 2048 bits

With OAEP (optimal asymmetric encryption padding), plaintext size can be, say 2048-256 bits with ciphertext size = 2048 bits
• Overhead = 256 bits
Digital Signatures
(aka public key MACs)
Message Integrity

Goal: If Eve changed $m$, Bob should reject
Syntax and Correctness

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

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

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

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

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

Output 1 iff:

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

\[1\text{CMA-Adv}(\mathcal{A}) = \Pr[\text{outputs 1}]\]
Unbounded Use MACs

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

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

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

\[\text{CMA-Adv}(\mathcal{A}) = \Pr[\text{outputs 1}]\]
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 CMA-secure
Signatures from Injective TDFs?

What goes wrong?
Next Time

More digital signatures