Admin: Project proposals due Friday (March 23)
Pset #3 due Monday (March 26).

Today: Continue public key encryption
- CCA2 secure encryption
  - RSA
  - RSA + OAEP is CCA2 secure
  - OAEP = Optimal Asymmetric Encryption Padding
- Digital Signatures
  - Def
  - Hash & Sign
  - RSA

Recall: El-Gamal Encryption Scheme

\( \text{PK} = (g, y = g^x) \)
\( \text{SK} = x \)

\( \text{Enc}(m): g^x, y^x \cdot m \)

\( \text{Dec}(u, e) = e/u^x \mod p \)

Thin: CPA (semantic) secure assuming DDH.
* El-Gamal is not CCA secure!

\[ \text{Enc}(m) \xrightarrow{\text{easy}} \text{Enc}(2m) \]
\[ (g^k, y^k \cdot m) \xrightarrow{\text{easy}} (g^k, y^k \cdot 2m) \]
without knowing \( m \)!

More generally, El-Gamal is homomorphic:

\[ \text{Enc}(m_1), \text{Enc}(m_2) \xrightarrow{\text{easy}} \text{Enc}(m_1 \cdot m_2) \]
\[ (g^{k_1}, y^{k_1} \cdot m_1) (g^{k_2}, y^{k_2} \cdot m_2) \rightarrow g^{k_1+k_2}, y^{k_1+k_2} \cdot m_1 \cdot m_2 \]

* Product of ciphertexts yields an encryption of product of plaintexts.

* Special case: El-Gamal is rerandomizable

By multiplying by \( \text{Enc}(1) = (g^k, y^k) \) it rerandomizes the ciphertext.

* Malleability can be a feature, but can be viewed as a form of insecurity.
Stronger security notion: **CCA2**

(Security under adaptive chosen ciphertext attacks).

*Recall: We saw a similar def for symmetric enc.*

**CCA2 Security game**

**Phase I ("Find"):**
- Challenger generates \((PK, SK) \leftarrow \text{KeyGen}(1^n)\)
  & send \(PK\) to adversary
- Adversary has access to decryption oracle \(\text{Dec}(sk, \cdot)\)
  runs in \(\text{poly}(n)\) time, and outputs two msgs \(m_0, m_1\) of some length, and "state information \(s\).

**Phase II (Guess):**
- Challenger chooses \(b \leftarrow \{0, 1\}\) & computes \(C_b \leftarrow \text{Enc}(PK, m_b)\), and sends \(C_b\) to adv.
- Adv. given \((C_b, s)\) and given
  access to decryption oracle \(\text{Dec}(sk, \cdot)\)
  except on \(C_b\),
  outputs \(\hat{b}\) (his guess for \(b\)).
  Adv wins iff \(\hat{b} = b\).
Def: Encryption is CCA2 secure if
\[ P_{Adv[\text{wms}]} = \frac{1}{2} + \text{negl}(\lambda) \]

Cramer-Shoup: Extended El-Gamal to be CCA2 secure.

Intuition: Add to the ciphertext a "test".
The decryption checks if ciphertext "passed the test".
If not, decryption outputs ⊥
If test passed then the decrypted msg is output.

Idea: To pass the test one needs to know
the msg. (and hence dec. oracle is useless)

Cramer-Shoup Enc:

Let \( p \) be a prime & \( g \mid p-1 \) a large prime.
Let \( g \in \mathbb{Z}_p^* \) an element of order \( g \).

Key Gen: \[
\begin{align*}
G1, G2 &\leftarrow <g> \\
&g^2 = g^1
\end{align*}
\]
\[ x_1, x_2, y_1, y_2 \in \mathbb{Z}_p \]
\[ c = g_1^{x_1} g_2^{x_2} \]
\[ d = g_1^{y_1} g_2^{y_2} \] \{ needed for the "test" \}

\[ \text{PK} = (g_1, g_2, c, d, h) \]
\[ \text{SK} = (z, x_1, x_2, y_1, y_2) \]

\[ \text{Enc}(m) \ [m \in \langle g \rangle] \]

\[ E_G = g_1^r \]
\[ e = h^{z \cdot m} \]

\[ u_2 = g_2^e \]
\[ \alpha = H(u, u_2, e) \]
\[ \nu = c^r d^{r \alpha} \]

\text{output} : \ (u_1, e, u_2, \nu) \]

\[ \text{Dec} \ (u_1, e, u_2, \nu) \]

\[ \text{Check} : \ u_1^{x_1 + y_1 \alpha} = u_2^{x_2 + y_2 \alpha} \tag*{?} \]

If not reject, else output \[ m = e/\nu \]

\[ \text{EG} \]
**Thm:** Cramer-Shoup is CCA2 secure if
- DDH holds in $\mathbb{G} < g$
- $H$ is: "target collision resistant".

Thus, strongest notion of security is achievable,
    albeit at cost in terms of speed & complexity.

**RSA Encryption** [Rivest-Shamir-Adleman 78]

First public key encryption scheme!

Follows the Diffie-Hellman model:

- $\text{KeyGen}(1^{|\lambda|}) \rightarrow (PK, SK, M, C)$

Here $|M| = |C|$

- $\text{Enc}(PK, \cdot)$ is an efficiently computable
  (deterministic) map from $M$ to $C$
  
  $c = \text{Enc}(PK, m)$ is unique ciphertext for $m$.

- $\text{Dec}(SK, \cdot)$ is efficiently computable

  inverse: $\text{Dec}(SK, \text{Enc}(PK, m)) = m$ for $m \in M$. 
It is hard to decrypt given only $PK$ (without knowledge of $SK$). $SK$ is "trapdoor" information that enables inversion of (otherwise one-way) function $Enc(PK, \cdot)$.

**RSA Encryption Scheme**

[A trapdoor one-way function family]:

**KeyGen**: Sample two large primes $P, Q$ (e.g., $2=1024$ bits each).

$n = P \cdot Q$

- Sample $e \in \mathbb{Z}_n^*$
  - $\text{gcd}(e, \phi(n)) = 1$.
  - Recall $\phi(n) = (P-1)(Q-1)$.

Compute $d = e^{-1} \mod \phi(n)$ (using Euclid's alg).

$PK = (n, e)$

$SK = d$

$M = C = \mathbb{Z}_n$

**Encrypt**: $Enc((n, e), m) = m^e \mod n$.
Decrypt: \( \text{Dec} \left( d, n \right), C \) = \( C^d \mod n \).

Correctness: If \( M = C = \mathbb{Z}_n^* \) then

\[ \forall m \in \mathbb{Z}_n^* \]

\[ \text{Dec} \left( \text{sk}, \text{Enc} \left( \text{pk}, m \right) \right) = m^e \mod n \]

Correctness also holds for \( \mathbb{Z}_n \), via Chinese Remainder Thm (though we will not see msgps in \( \mathbb{Z}_n \setminus \mathbb{Z}_n^* \)).

Enough to prove:

\[ \begin{align*}
 m^e d &= m \mod q \\
 &\text{exercize (follows from Fermat's Thm)}
\end{align*} \]

Recall CRT:

For \( n = p \cdot q \) where \( p \) & \( q \) are distinct primes

\[ \forall x, y \in \mathbb{Z}_n \]

\[ x = y \mod n \] iff \( x = y \mod p \) & \( x = y \mod q \).
Security of RSA

If adv can factor $n$, then adv can compute $\varphi(n)$ and thus compute $d = e^{-1} \mod \varphi(n)$.

Key insight: Security relies on the fact that the size of the group $\mathbb{Z}_n^*$ is unknown.

Knowing $|\mathbb{Z}_n^*| = n$ is knowing $\varphi(n)$.

How hard is factoring?

- Can be done in time $2^{O((\log n)^3 \cdot (\log \log n)^{2/3})}$
- RSA keys of length 768 factored in 2009 can expect 1024-bit keys to be factored in near future.
- RSA keys of length 2048 are believed to be secure for a long time, unless there will be an algorithmic breakthrough.
- Can be done in poly time on quantum computer.

Is RSA semantically secure?

No! (It is not even randomized!)
RSA is a trapdoor permutation

\[ f_{n,e} : \mathbb{Z}_n \rightarrow \mathbb{Z}_n \]

\[ a \rightarrow a^e \mod n \]

Can be inverted efficiently given trapdoor d s.t. \( ed \equiv 1 \mod \phi(n) \).
Assumed to be one-way (RSA Assumption).

How to make RSA CCA2 secure?

**OAEP** = "Optimal asymmetric encryption padding"

[Bellare-Rogaway 94]

Idea: Apply RSA encryption to a randomized encoding of the msg.

![Diagram showing OAEP process]

\[ G : \mathbb{Z}_m^{t} \rightarrow \mathbb{Z}_m^{t+k_1} \]

\[ H : \mathbb{Z}_m^{t+k_1} \rightarrow \mathbb{Z}_m^{k_0} \]

*Similar to UFE of Desai for symmetric encryption.

Unbalanced Feistel Encryption.*
Digital Signatures

- Proposed by Diffie & Hellman in 1976
  (“New directions in Cryptography”)

- Idea: Signature depends on the \( m \).

- How to verify?
  - Each user has a pair of keys \((pk, sk)\)
  - \( pk \) is public, \( sk \) is kept secret.
  - Use \( pk \) to verify & use \( sk \) to sign.

- First implementation: RSA (1977)

**Def:** A digital signature scheme consists of 3 algorithms

- **KeyGen** \( (1^n) \rightarrow \hat{\text{(pk, sk)}} \)
  - Verification key
  - Secret key

- **Sign** \( (sk, m) \rightarrow \sigma_{sk}(m) \) (may be randomized)

- **Verify** \( (pk, m, \sigma) = 0/1 \) (acc/reject).

**Correctness:** \( \forall m \quad \nexists (pk, sk) \leftarrow \text{KeyGen}(1^n) \)

\[
\Pr\left[ \text{Verify}\left(pk, m, \text{sign}(sk, m) = 1 \right) \right] = 1
\]