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
  Optimal Asymmetric Encryption Padding
- Digital Signatures
  - Def
  - Hash & Sign
  - RSA

Recall: El-Gamal Encryption Scheme

$$\left.\begin{aligned} PK &= (g, y = g^x) \\ SK &= x \end{aligned}\right]$$

PP: prime p
$g \in \mathbb{Z}_p^*$ is s.t.
$\langle g \rangle$ is a ~~prime order~~ subgroup of prime order q

$Enc(m)$: $g^k, y^k \cdot m$

$Dec(u, e) = e / u^x \mod p$

Thm: CPA (semantic) secure assuming DDH.

* El-Gamal is <u>not</u> CCA secure!

$$\mathrm{Enc}(m) \xrightarrow{\text{easy}} \mathrm{Enc}(2m)$$

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

More generally, <u>El-Gamal is homomorphic</u>:

$$\mathrm{Enc}(m_1), \mathrm{Enc}(m_2) \xrightarrow{\text{easy}} \mathrm{Enc}(m_1 \cdot m_2)$$

$$(g^{k_1}, y^{k_1} \cdot m_1)\ (g^{k_2}, y^{k_2} \cdot m_2) \longmapsto g^{k_1+k_2},\ y^{k_1+k_2} \cdot m_1 m_2$$

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

* <u>Special case</u>: <u>El-Gamal is rerandomizable</u>

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

- Malleability can be a feature, but can be viewed as a form of insecurity.

Stronger security notion: <u>CCA2</u>

(Security under adaptive chosen ciphertext attacks).

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

<u>CCA2 Security game</u>

Phase I ("Find"):

- Challenger generates $(PK, SK) \leftarrow KeyGen(1^\lambda)$ & send PK to adversary
- Adversary has <u>access to decryption oracle $Dec(SK, \cdot)$</u> runs in poly(λ) time, and outputs two msgs m_0 & m_1 of same length, and "state information" s.

Phase II (Guess):

- Challenger chooses $b \leftarrow \{0,1\}$ & computes $c_b \leftarrow Enc(PK, m_b)$, and sends c_b to adv.
- Adv. given (c_b, s) and given <u>access to decryption oracle $Dec(SK, \cdot)$ except on c_b</u>, outputs $\hat{b}$ (his guess for b).

Adv wins iff $\hat{b} = b$.

<u>Def</u>: Encryption is CCA2 secure if

$$\Pr[\text{Adv wins}] \leq \frac{1}{2} + negl(\lambda)$$

<u>Cramer-Shoup</u>: Extended El-Gamal to be CCA2 secure.

<u>Intuition</u>: Add to the ciphertext a "test".

The decryption checks if ciphertext "passed the test".

If not, decryption outputs $\perp$

If test passed then the decrypted msg is output.

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

<u>Cramer-Shoup Enc</u>:

Let p be a large prime & $q \mid p-1$ a large prime.

Let $g \in \mathbb{Z}_p^*$ an element of order q.

<u>KeyGen</u>: $\boxed{g_1}, g_2 \leftarrow \langle g \rangle$

EG $\boxed{h = g_1^z}$

$x_1, x_2, y_1, y_2 \leftarrow \mathbb{Z}_q$

$$\left.\begin{aligned} c &= g_1^{x_1} g_2^{x_2} \\ d &= g_1^{y_1} g_2^{y_2} \end{aligned}\right\} \text{ needed for the "test"}$$

$$PK = (g_1, \underbrace{g_2\ c, d}_{\text{test}}, h)$$

$$SK = (z, \underbrace{x_1, x_2, y_1, y_2}_{\text{test}})$$

$Enc(m)$ $[m \in \langle g \rangle]$

$$EG \quad \boxed{\begin{aligned} u_1 &= g_1^r \\ e &= h^r \cdot m \end{aligned}}$$

$$u_2 = g_2^r$$

$$\alpha = H(u_1, u_2, e)$$

$$v = c^r d^{r\alpha}$$

output: $(\underbrace{\boxed{u_1, e}}_{EG}, \underbrace{u_2, v}_{\text{test}})$

$Dec(\underbrace{\boxed{u_1, e}}_{EG}, u_2, v)$

<u>Check</u>: $u_1^{x_1 + y_1 \alpha} \cdot u_2^{x_2 + y_2 \alpha} \stackrel{?}{=} v$

If not reject. Else output $\boxed{m = e / u_1^z}$ EG

<u>Thm</u>: Cramer-Shoup is CCA2 secure if

- DDH holds in $\langle g \rangle$
- 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 77]

First public key encryption scheme!

Follows the <u>Diffie-Hellman model</u>:

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

  msg space (M), ciphertext space (C)

  Here $|M| = |C|$

- $\text{Enc}(PK, \cdot)$ is an efficiently computable (<u>deterministic</u>) map from M to C

  $c = \text{Enc}(PK, m)$ is unique ciphertext for m.

- $\text{Dec}(SK, \cdot)$ is efficiently computable inverse: $\text{Dec}(SK, \text{Enc}(PK, m)) = m \quad \forall m \in M$.

- It is <u>hard</u> 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]:

<u>KeyGen</u>: • Sample two large primes p, q (eg. $\lambda = 1024$ bits each).

$n = p \cdot q$

• Sample $e \leftarrow \mathbb{Z}^*_{\varphi(n)}$

Compute $d = e^{-1} \bmod \varphi(n)$

(using Euclid's alg).

Recall $\varphi(n) = |\mathbb{Z}^*_n| = (p-1)(q-1)$

$\gcd(e, \varphi(n)) = 1$.

$PK = (n, e)$

$SK = d$.

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

<u>Encrypt</u>: $Enc((n,e), m) = m^e \bmod n$

<u>Decrypt</u>:

$$\mathrm{Dec}((d,n), c) = c^d \bmod n.$$

<u>Correctness</u>: If $M = C = \mathbb{Z}_n^*$ then

$\forall m \in \mathbb{Z}_n^*$

$$\mathrm{Dec}(SK, \mathrm{Enc}(PK, m)) =$$

$$\mathrm{Dec}(SK, m^e \bmod n) = m^{ed} \bmod n = m^{1+c\cdot\varphi(n)}$$

$$= m \bmod n.$$

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

Enough to prove:

$$\left.\begin{aligned} & m^{ed} = m \bmod p \\ \& \quad & m^{ed} = m \bmod q \end{aligned}\right\} \text{exercize (follows from Fermat's Thm).}$$

Recall CTR:

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

$\forall x, y \in \mathbb{Z}_n$

$$(x = y \bmod n) \text{ iff } (x = y \bmod p \ \& \ x = y \bmod q.)$$

Security of RSA

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

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

Knowing $|\mathbb{Z}_n^*| \equiv$ knowing p, q.

How hard is factoring?

- Can be done it time $2^{O((\lg n)^3 \cdot (\lg \lg n)^{2/3})}$
- 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

$$f_{n,e} : \mathbb{Z}_n \longleftrightarrow \mathbb{Z}_n$$
$$a \longmapsto a^e \bmod n$$

Can be inverted ~~xx~~ efficiently given trapdoor d s.t. $ed = 1 \bmod \varphi(n)$.

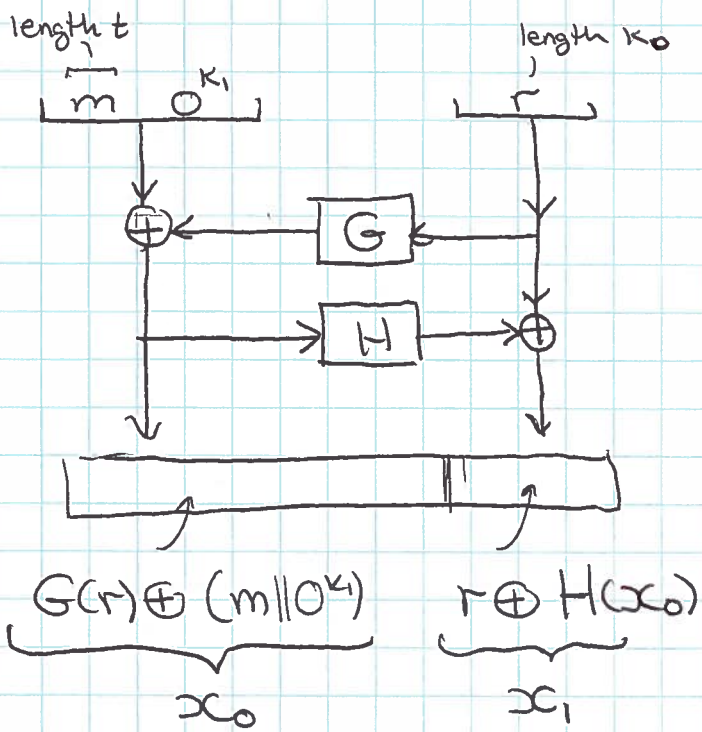
Assumed to be one-way (RSA Assumption).

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.



$G : \{0,1\}^{k_0} \longrightarrow \{0,1\}^{t+k_1}$

$H : \{0,1\}^{t+k_1} \longrightarrow \{0,1\}^{k_0}$

* 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 msg

- 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

- $\text{KeyGen}(1^\lambda) \longrightarrow (PK, SK)$
  verification key — secret key.
- $\text{Sign}(SK, m) \longrightarrow \sigma_{SK}(m)$ (may be randomized)
- $\text{Verify}(PK, m, \sigma) = 0/1$ (acc/reject).

Correctness: $\forall m$ $(PK, SK) \leftarrow \text{KeyGen}(1^\lambda)$

$$\Pr\left[\text{Verify}(PK, m, \text{sign}(SK, m)) = 1\right] = 1$$