Admin:

Comments on (many) project proposals on gradescope
Pset #4 out later today

Today:

Pedersen Commitments
PK encryption
El Gamal PK encryption
Semantic Security
DDH (Decision-Diffie-Hellman)
IND-CCA2
Cramer-Shoup PK enc

Readings:

Paar: Ch 6, 7, 8
Katz: Ch 10
Pedersen Commitment Scheme

Recall: \( \text{Commit}(x) \rightarrow "\text{commitment to } x" \)

\( \text{Reveal}(c) \rightarrow "\text{opens commitment, reveals } x" \)

Properties:
- **Hiding**: \( \text{Commit}(x) \) reveals nothing about \( x \)
- **Binding**: Can only open in one way (can’t change \( x \))
- **Non-malleability**: Can’t produce commitment to e.g., \( x+1 \) from commitment to \( x \).

**Setup**: \( p, q \) large primes s.t. \( q \mid p-1 \) (e.g., \( p \) "safe prime"

- \( g \) generator of order-\( q \) subgroup of \( \mathbb{Z}_p^* \)
- (e.g., if \( p \) safe then \( \langle g \rangle = \mathbb{Z}_p^* \))
- \( h = g^a \) a secret \( h \) generates \( \langle g \rangle \) as well

**Commit** (\( x \)):
- \( x \in \mathbb{Z}_q \) (i.e., \( 0 \leq x < q \))
- Sender chooses random \( r \in \mathbb{Z}_q \)
- \( \text{Commit}(x) = c = g^x h^r \mod p \)

**Reveal**:
- Sender reveals \( x \) and \( r \)
- Receiver verifies that \( c = g^x h^r \mod p \)
Pedersen commitment (cont.)

Hiding: Given $c = g^x h^r$

Can in principle be opened to any $x' \in \mathbb{Z}_p$, for some $r'$

$g^{x'} h^{r'} = g^{x^*} h^r$

$g^x g^{ar} = g^{x^*} g^{ar'}$

$g^{x+ar} = g^{x^*+ar'}$

$x + ar = x' + ar' \pmod{p}$

$r' = (x-x')/a + r$

$r'$ is prime so $a \neq x'$

Binding: If sender can reveal two ways

$c = g^{x^*} h^{r'}$

$x + ar = x' + ar'$

$a = (x-x')/(r'-r)$

$r' \neq r$ since $x \neq x'$

$\text{gcd}(r'-r, g)$ is prime

$= \text{discrete log of } h, \text{ base } g, \text{ mod } p$

Non-malleable: No.

If $c = \text{Commit}(x) = g^x h^r$

then $c' = \text{Commit}(x') = g^{x'} (g^x h^r) = g^{x^*+1} h^r$

(Some applications don’t need non-malleability)
Public-key encryption:

Let $\lambda = \text{"security parameter" }$ (i.e. "key size")

Then $1^\lambda = \underbrace{11\ldots 1}_\lambda \; \lambda$'s in a row, length $= \lambda$

Need three algorithms:

1. $\text{Keygen } \left( 1^\lambda \right) \rightarrow (PK, SK)$

2. $E(PK, m) \rightarrow c$

   Encryption takes $m \in \text{message space } M$

   to $c \in \text{cipher text space } C$

   (with given public key $PK$)

   Encryption may be randomized.

3. $D(SK, c) \rightarrow m$

   Decryption is deterministic

s.t. (Correctness condition)

$(\forall(PK, SK))(\forall m) \; D(SK, E(PK, m)) = m$
El-Gamal PK encryption  (Taheer El Gamal, 1984)

Let $G = \langle g \rangle$ be a cyclic group with generator $g$. 
(Keygen may output description of $g$ & $G$ given $\lambda$.)

**Keygen:**
- Pick $x$ at random from $[0, |G| - 1]$
- Let $SK = x$.
- Let $PK = g^x$
- Output $(PK, SK)$ ($\&$ description of $G$, if needed)

**Encryption:**
- Pick $k$ at random from $[0, |G| - 1]$
- Assume message $m$ represented as element of $G$.
- Let $y = g^k$ be PK of recipient
- Output $c = (g^k, m \cdot y^k)$ as ciphertext

**Decryption:**
- Let $c = (a, b)$ be received ciphertext
- Let $m = b / a^x$. Output $m$.

[Correctness follows since $a^x = g^{kx} = g^x y^k = y^k$.]
ElGamal encryption related to DH key exchange:

Alice: \[y = g^x \quad \text{(via PKI?)} \]

\[a = g^k \]

\[\text{DH Key} = (g^x)^x \]

\[= g^{kx} \]

Bob: \[\text{DH Key} \]

\[= (g^x)^k = g^{kx} \]

\[b = m \cdot (\text{DH Key}) \]

Encrypt by multiplying by DH Key.

Decrypt by dividing by DH Key.
How to define security for PK encryption?

We'll see two definitions:

1. "semantic security" (Goldwasser & Micali)

2. "adaptive chosen ciphertext attack" (ACCA) secure (as to IND-CCA we saw for symmetric encryption)

"Game" definition of semantic security:

**Phase I ("Find"):**

- Examiner generates (PK, SK) using Keygen(1^λ)
- Examiner sends PK to Adversary
- Adversary computes for polynomial (in λ) time, then outputs two messages m₀, m, of same length, and "state information" s. [m₀ ≠ m, required]

**Phase II ("Guess"):**

- Examiner picks b ∈ {0, 1}^*, computes c* = E(PK, m_b)
- Examiner sends c*, s to Adversary
- Adversary computes for polynomial (in λ) time, then outputs b̂ (his "guess" for b).

Adversary "wins" game if b̂ = b.
**Def:** A PK encryption scheme is semantically secure if \( \text{Prob[Adv wins]} \leq \frac{1}{2} + \text{negligible} \)

**Fact:** In order for a PK encryption scheme to be semantically secure, it must necessarily be randomized. *(Randomized encryption is necessary but not sufficient for semantic security.)*

Is El Gamal PK encryption semantically secure?

*More precisely: it can't be stateless & deterministic.*

*It may be randomized, or stateful, or both.*
DDH (Decision Diffie-Hellman Assumption):

Given a group \( G \) with generator \( g \):

It is hard/infeasible to decide whether a given triple of elements was generated as

\[
(g^a, g^b, g^c) \quad [a, b, c \text{ random}]
\]

or as

\[
(g^a, g^b, g^{ab}) \quad [a, b \text{ random}]
\]

That is, if DDH holds in a group, you can't even recognize the DH key \( g^{ab} \) when it is given to you! (You can't distinguish it from a random element.)

Recall: CDH = Computing \( g^{ab} \) from \( g^a \) & \( g^b \) is hard

Theorem: \( \text{DDH} \Rightarrow \text{CDH} \)

Proof: If \( \neg \text{CDH} \), then \( \neg \text{DDH} \) (contrapositive).

If you can compute \( g^{ab} \) from \( g^a \) and \( g^b \) (i.e. \( \neg \text{CDH} \)), then you can decide if given third element is \( g^{ab} \) (i.e. \( \neg \text{DDH} \)). \( \blacksquare \)
Theorem (Tsiounis & Yung):

El Gamal is semantically secure in \( G \):

\[ \iff \]

DDH holds in \( G \)

- Semantic security may not be enough for some applications.
- El Gamal is malleable:
  
  Given \( E(m) = (g^k, m \cdot y^k) \)
  
  it is easy to produce \( E(2m) = (g^k, (2m) \cdot y^k) \)
  
  without knowing \( m \)!

- More generally, El Gamal is homomorphic:

  Given \( c_1 \in E(m_1) = (g^{r_1}, m_1 \cdot y^{r_1}) \)
  
  & given \( c_2 \in E(m_2) = (g^{r_2}, m_2 \cdot y^{r_2}) \)
  
  can produce \( c_1 \cdot c_2 = (g^{r_1 + r_2}, (m_1 \cdot m_2) \cdot y^{r_1 + r_2}) \)
  
  \( \in E(m_1 \cdot m_2) \)

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

- Special case: multiplying by \( E(1) = (g^s, y^s) \)
  
  re-randomizes encryption.
- What is stronger notion of security for PK encryption?
  (e.g. one that excludes malleability...)

- "IND-CCA2 secure" (ACCA secure = secure under adaptive chosen ciphertext attack)

- IND-CCA secure defn we saw for symmetric enc.

- Similar to semantic security defn, except that
  Adv allowed access to decryption oracle, too.
  (He has PK so access to encryption oracle already there.)
  (As before, may not use oracle to decrypt challenge ciphertext during "guess" phase.)
IND-CCA2 (ACCA) Security games:

**Phase I ("Find")**:
- Examiner generates \((PK, SK)\) using Keygen \((1^\lambda)\)
- Examiner sends \(PK\) to Adversary
- Adversary computes for polynomial \((\text{in } \lambda)\) time, having access to a decryption oracle \(D(SK, c)\)
  then outputs two messages \(m_0, m_1\), of same length, and "state information" \(s\). \([m_0 \neq m_1, \text{ required}]\)

**Phase II ("Guess")**:
- Examiner picks \(b \leftarrow \mathcal{R} \{0, 1\}\), computes \(c_* = E(PK, m_b)\)
- Examiner sends \(c_*, s\) to Adversary
- Adversary computes for polynomial \((\text{in } \lambda)\) time, having access to a decryption oracle \(D(SK, c_*)\)
  expect on input \(c_*\)
  then outputs \(\hat{b}\) (his "guess" for \(b\)).

Adversary wins if \(\hat{b} = b\).

**Def**: PK encryption method is IND-CCA2 secure (ACCA-secure) if
\[
\text{Prob}[\text{Adv wins}] < \frac{1}{2} + \text{negligible}
\]
How to make El Gamal IND-CCA2 secure?

- Cramer-Shoup method is such an extension of El Gamal.
- Let \( G_q \) be a group of prime order \( q \)
  
  \( (e.g., \ G_q = \mathbb{Q}_p, \ \text{where} \ p = 2^q + 1, \ p & q \text{ prime}). \)

- **Keygen:**

  \[
  g_1, g_2 \xleftarrow{R} G_q
  \]

  \[
  x_1, x_2, y_1, y_2, z \xleftarrow{R} \mathbb{Z}_q
  \]

  \[
  c = g_1^{x_1} \cdot g_2^{x_2}
  \]

  \[
  d = g_1^{y_1} \cdot g_2^{y_2}
  \]

  \[
  h = g_1^z
  \]

  **PK** = \((g_1, g_2, c, d, h)\)

  **H** = hash fn mapping \( G_q^3 \) to \( \mathbb{Z}_q \)

  **SK** = \((x_1, x_2, y_1, y_2, z)\)
**Enc(m)** [where \( m \in G_q \)]:

- \( r \leftarrow \mathbb{Z}_q \)
- \( u_1 = g_1^r \)
- \( u_2 = g_2^r \)
- \( e = h^r \cdot m \)
- \( \alpha = H(u_1, u_2, e) \)
- \( v = c^r \cdot d^{r \alpha} \)
- **cipher text** = \((u_1, u_2, e, v)\)

**Decrypt** \((u_1, u_2, e, v)\):

- \( \alpha = H(u_1, u_2, e) \)
- **Check** : \( u_1^{x_1 + y_1 \alpha} \cdot u_2^{x_2 + y_2 \alpha} \overset{?}{=} v \)
- If not equal, reject
- else output \( m = e / u_1^{x_1} \)

**Note**:
- \( u_1^{x_1} u_2^{x_2} = g_1^{rx_1} g_2^{rx_2} = c^r \)
- \( u_1^{y_1} u_2^{y_2} = d^r \)
- \( u_1^z = g_1^{rz} = h^r \)
Theorem: Cramer-Shoup is IND-CCA2 secure (i.e. secure against adaptive chosen ciphertext) if

1. DDH holds in Gq
2. H satisfies a certain condition (x "target collision resistance")

Thus, our strongest notion of security for PK encryption is in fact achievable, albeit at some cost in terms of speed & complexity.