Today:
- Pedersen Commitment
- PK Encryption
- El Gamal PK encryption
- Semantic security
- DDH (Decision Diffie-Hellman)
- IND-CCA2
- Cramer-Shoup PK encryption

Readings:
- Paar and Pelzl, Chapters 6, 7, 8
- Katz and Lindell, Chapter 10
Pedersen Commitment Scheme

Recall: Commit(x) → “commitment to x”
Reveal(c) → “opens commitment, reveals x”

Properties:
- Hiding: 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+ from commitment to x.

Setup:
- p, q large primes s.t. q | p-1 (e.g., p “safe prime”)
- g generator of order-q subgroup of \( Z_p^* \)
  (e.g., if p safe then \( <g> = \mathbb{Z}_p^* = \text{quatsum mod } p \))
- \( h = g^a \) a secret

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 \( r' \)

\[
\begin{align*}
g^{x' h^r} & = g^{x' h^{r'}} \\
g^x g^{ar} & = g^{x' g^{ar'}} \\
g^{x + ar} & = g^{x' + ar'} \\
x + ar & = x' + ar' \pmod{q}
\end{align*}
\]

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

(\( q \) is prime so \( a \) exists)

\( r' \neq r \) since \( x \neq x' \)

**Binding:** If sender can reveal two ways

\[
\begin{align*}
c & = g^{x' h^r} = g^{x' h^{r'}} \\
x + ar & = x' + ar'
\end{align*}
\]

\[
a = (x - x')/(r' - r)
\]

(\( r' \neq r \) & \( q \) is prime)

= discrete log of \( h \), base \( g \), \ modulo \( p \)

**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 $\lambda$ = "security parameter" (i.e. "key size")

Then $1^\lambda = 11\ldots1$ \(\lambda\) 1's in a row. Length = $\lambda$

Need three algorithms:

1. Keygen ($1^\lambda$) $\rightarrow$ (PK, SK)

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

   Encryption takes \(m\) e message space \(M\)
   to \(c\) e ciphertext space \(C\)
   (with given public key PK)

   Encryption may be randomized.

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

   Decryption is deterministic

   s.t. (Correctness condition)

   \[ (\forall (PK, SK))(\forall m) \quad D(SK, E(PK, m)) = m \]
El-Gama\textsuperscript{a} PK encryption (Taher El Gama\texttextsuperscript{a}, 1984)

Let \( G = \langle g \rangle \) be a cyclic group with generator \( g \).

(\text{Keygen} \text{ may output description of } g \text{ & } G \text{ given } \lambda. )

\textbf{Keygen:}

Pick \( x \) at random from \([0 \ldots |G|-1]\)

Let \( SK = x \).

Let \( PK = g^x \)

Output \((PK, SK) \quad (\& \text{description of } G, \text{if needed})\)

\textbf{Encryption:}

Pick \( k \) at random from \([0 \ldots |G|-1]\)

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

\textbf{Decryption:}

Let \( c = (a, b) \) be received ciphertext

Let \( m = b / a^k \). Output \( m \).

[Correctness follows since \( a^k = g^k = g^{xk} = y^k. \)]
El Gamal encryption related to DH key exchange:

Alice \[ y = g^x \quad \text{(via PK1?)} \quad \rightarrow \quad \text{Bob} \]

\[ a = g^k \]

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

(9 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, m₁ of same length, and "state information" s. \([m₀ ≠ m₁ \text{ required}]\)

Phase II ("Guess"): 
- Examiner picks \(b ∈ \{0, 1\}^s\), computes \(c ← E(PK, m_b)\)
- Examiner sends \(c, s\) to Adversary
- Adversary computes for polynomial (in λ) time, then outputs \(\hat{b}\) (his "guess" for b).

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

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* and 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.)

**Theorem:** $DDH \Rightarrow CDH$

**Proof:** If $\neg CDH$, then $\neg DDH$ (contrapositive).

If you can compute $g^{ab}$ from $g^a$ and $g^b$ (i.e. $\neg CDH$) then you can decide if given third element is $g^{ab}$ (i.e. $\neg DDH$).

---

**Recall:**

CDH $\equiv$ Computing $g^{ab}$ from $g^a \& g^b$ is hard.
Theorem (Tsounis & Yung):
El Gamal is semantically secure in $G$ if and only if DDH holds in $G$.

- Semantic security may not be enough for some applications.
- El Gamal is malleable:
  Given $E(m) = (g^k, m, y^k)$, it is easy to produce $E(2m) = (g^k, (2m) y^k)$ without knowing $m$.
- More generally, El Gamal is homomorphic:
  Given $c_1 \in E(m_1) = (g^r, m_1, y_1^r)$ and $c_2 \in E(m_2) = (g^s, m_2, y_2^s)$,
can produce $c_1 \cdot c_2 = (g^{r+s}, (m_1 m_2) y_1^r y_2^s) \in E(m_1 m_2)$.
- Product of ciphertexts yields an encryption of the 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 Game:

Phase I ("Find"):
- Examiners generates (PK, SK) using Keygen(1^n)
- Examiners sends PK to Adversary
- Adversary computes for polynomial (in n) time, having access to a decryption oracle D(SK, c)
then outputs two messages m₀, m₁ of same length, and "state information" s. [m₀ ≠ m₁, required]

Phase II ("Guess"):
- Examiners picks b ∈ {0, 1}, computes cᵦ = E(PK, m_b)
- Examiners sends cᵦ, s to Adversary
- Adversary computes for polynomial (in n) time, having access to a decryption oracle D(SK, c)
except on input cᵦ,
then outputs ₀ (his "guess" for b).
Adversary wins if ₀ = 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. $G_q = \mathbb{Z}_p$, where $p=2q+1$, $p$ & $q$ prime).

- Keygen's

  $g_1, g_q \leftarrow_R G_q$

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

  $c = g_1^{x_1} g_2^{x_2}$

  $d = g_1^{y_1} g_2^{y_2}$

  $h = g_1^z$

  $PK = (g_1, g_2, c, d, h)$

  $H = \text{hash fn mapping } G_q^3 \text{ to } \mathbb{Z}_q$

  $SK = (x_1, x_2, y_1, y_2, z)$
• **Enc (m) [where \( m \in G_0 \):**

\[
\begin{align*}
    r & \leftarrow R \mathbb{Z}_q \\
    u_1 & = g_1^r \\
    u_2 & = g_2^r \\
    e & = h^{r \cdot m} \\
    \alpha & = H(u_1, u_2, e) \\
    v & = c^r d^{-\alpha} \\
    \text{ciphertext} & = (u_1, u_2, e, v)
\end{align*}
\]

\( \text{EG} \)

• **Decrypt \((u_1, u_2, e, v)\):**

\[
\begin{align*}
    \alpha & = H(u_1, u_2, e) \\
    \text{Check: } u_1^{x_1 + y_1 \alpha} \cdot u_2^{x_2 + y_2 \alpha} & \overset{?}{=} v \\
    \text{If not equal, reject} \\
    \text{else output } m & = e^{u_1^2}
\end{align*}
\]

\( \text{EG} \)

Note:
\[
\begin{align*}
    u_1^{x_1} u_2^{x_2} & = g_1^{r x_1} \cdot g_2^{r x_2} = c^r \\
    u_1^{x_1} u_2^{x_2} & = d^r \\
    u_1^z & = g_1^{r e} = h^r
\end{align*}
\]

\( \text{EG} \)
Theorem: Cramer-Shoup is IND-CCA2 secure (i.e., secure against adaptive chosen ciphertexts) if

1. DDH holds in \( G_q \)
2. \( H \) satisfies a certain condition
   (\( \% \) "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.
6.857 Network and Computer Security
Spring 2014

For information about citing these materials or our Terms of Use, visit: http://ocw.mit.edu/terms.