

6.857 Computer and Network Security  
Lecture 13

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 SchemeRecall:  $\text{Commit}(x) \rightarrow$  "commitment to  $x$ " $\text{Reveal}(c) \rightarrow$  "opens commitment, reveals  $x$ "Properties: Hiding:  $\text{Commit}(x)$  reveals nothing about  $x$ Binding: Can only open in one way (can't change  $x$ )Nonmalleability(?): Can't produce commitment to e.g.  $x+1$  from commitment to  $x$ .values  
can be  
chosen by  
receiverSetup:  $p, q$  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  $\langle g \rangle = \mathbb{Q}_p = \text{squares mod } p$ ) $h = g^a$  a secretCommit( $x$ ):  $x \in \mathbb{Z}_q$  (i.e.  $0 \leq x < q$ )Sender chooses random  $r \in \mathbb{Z}_q$  $\text{Commit}(x) = c = g^x h^r \pmod{p}$ Reveal: Sender reveals  $x$  and  $r$ Receiver verifies that  $c = g^x h^r \pmod{p}$

Pedersen commitment (cont.)

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

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

$$\left. \begin{aligned} g^x h^r &= g^{x'} h^{r'} \\ g^x g^{ar} &= g^{x'} g^{ar'} \\ g^{x+ar} &= g^{x'+ar'} \end{aligned} \right\} \pmod{p}$$

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

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

$\leftarrow g$  is prime so  $a^{-1} \in \mathbb{B}$   
 $r' \neq r$  since  $x \neq x'$

Binding: If sender can reveal two ways

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

$$x+ar = x'+ar'$$

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

$\leftarrow r' \neq r$  &  $g$  is prime  
 = discrete log of  $h$ , base  $g$ , mod  $p$   $\square$

Non-malleable: Nope.

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

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

(Some applications don't need non-malleability)

"Perfectly Hiding"  
 (Adversary could have  $\infty$  computational power...)

"Computationally Binding"  
 (Sender can't compute  $a$ )

Public-key encryption:

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

Then  $1^\lambda = \underbrace{11\dots1}_\lambda$   $\lambda$  1's in a row. Length =  $\lambda$

Need three algorithms:

$$\textcircled{1} \text{ Keygen}(1^\lambda) \rightarrow (PK, SK)$$

$$\textcircled{2} E(PK, m) \rightarrow c$$

Encryption takes  $m \in$  message space  $M$

to  $c \in$  ciphertext space  $C$

(with given public key  $PK$ )

Encryption may be randomized.

$$\textcircled{3} D(SK, c) \rightarrow m$$

Decryption is deterministic

s.t. (Correctness condition)

$$(\forall (PK, SK)) (\forall m) D(SK, E(PK, m)) = m$$

El-Gamal PK encryption (Taher 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  $[0 \dots |G|-1]$

Let  $SK = x$ .

Let  $PK = g^x$

Output  $(PK, SK)$  (& description of  $G$ , if needed)

Encryption:

Pick  $k$  at random from  $[0 \dots |G|-1]$

Assume message  $m$  represented as element of  $G$ .

Let  $y = g^x$  be PK of recipient

Output  $c = (g^k, m \cdot y^k)$  as ciphertext

Decryption:

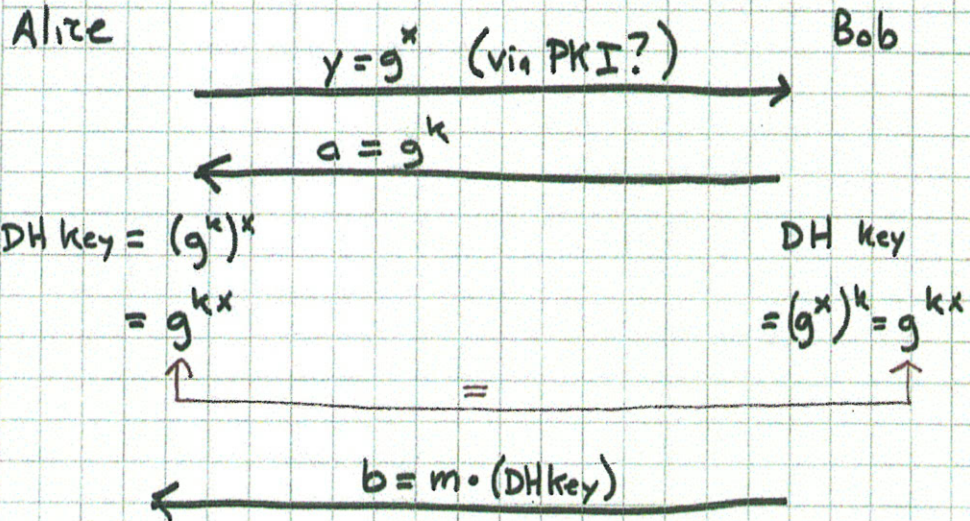
Let  $c = (a, b)$  be received ciphertext

Let  $m = b / a^x$ . Output  $m$ .

[Correctness follows since  $a^x = g^{kx} = g^{xk} = y^k$ .]

randomized!

E) Gamal encryption related to DH key exchange:



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  
( $\approx$  to IND-CCA we saw for symmetric encryption)

"Game" definition of semantic security:

Phase I ("Find"):

- Examiner generates  $(PK, SK)$  using  $\text{Keygen}(1^\lambda)$
- Examiner sends  $PK$  to Adversary
- Adversary computes for polynomial (in  $\lambda$ ) time, then outputs two messages  $m_0, m_1$ , of same length, and "state information"  $s$ . [ $m_0 \neq m_1$ , required]

Phase II ("Guess"):

- Examiner picks  $b \xleftarrow{R} \{0, 1\}$ , computes  $c_b = E_{SK}(PK, m_b)$
- Examiner sends  $c_b, s$  to Adversary
- Adversary computes for polynomial (in  $\lambda$ ) 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.)

for stateless encryption

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.)

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.  $\neg DDH$ ).  $\square$

Recall:  
 $CDH \equiv$   
 Computing  $g^{ab}$   
 from  $g^a$  &  $g^b$   
 is hard

Theorem (Tsionnis & Yung):

El Gamal is semantically secure in  $G$



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  $E(am) = (g^k, (a \cdot m) \cdot y^k)$

without knowing  $m$ !

- More generally, El Gamal is homomorphic:

Given  $c_1 \in E(m_1) = (g^r, m_1 \cdot y^r)$

& given  $c_2 \in E(m_2) = (g^s, m_2 \cdot y^s)$

can produce  $c_1 \cdot c_2 = (g^{r+s}, (m_1 \cdot m_2) \cdot y^{r+s})$

$\in E(m_1 \cdot m_2)$

- 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)  
 $\approx$  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"):

new =>

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

Phase II ("Guess"):

new => {

- Examiner picks  $b \leftarrow_R \{0, 1\}$ , computes  $c_b = E(PK, m_b)$
- Examiner sends  $c_b, s$  to Adversary
- Adversary computes for polynomial (in  $\lambda$ ) time, having access to a decryption oracle  $D(SK, \cdot)$  except on input  $c_b$  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

$$\text{Prob}[\text{Adv wins}] \leq \frac{1}{2} + \text{negligible}$$

### How to make El Gamal IND-CCA2 secure?

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

$$g_1, g_2 \xleftarrow{R} G_g$$

$$x_1, x_2, y_1, y_2, z \xleftarrow{R} \mathbb{Z}_g$$

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

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

$$h = g_1^z$$

EG

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

$$H = \text{hash fn mapping } G_g^3 \text{ to } \mathbb{Z}_g$$

$$SK = (x_1, x_2, y_1, y_2, z)$$

• Enc(m) [where  $m \in G_q$ ]:

$$r \xleftarrow{R} \mathbb{Z}_q$$

EG

$$u_1 = g_1^r$$

EG

$$u_2 = g_2^r$$

$$e = h^r \cdot m$$

EG

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

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

$$\text{ciphertext} = (\underline{u_1}, \underline{u_2}, \underline{e}, v)$$

EG

• Decrypt( $u_1, u_2, e, v$ ):

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

$$\text{Check: } u_1^{x_1 + y_1 \alpha} u_2^{x_2 + y_2 \alpha} \stackrel{?}{=} v$$

If not equal, reject

else output  $m = e / u_1^z$

EG

Note:  $u_1^{x_1} u_2^{x_2} = g_1^{rx_1} g_2^{rx_2} = c^r$

$$u_1^{y_1} u_2^{y_2} = d^r$$

$$u_1^z = g_1^{rz} = h^r$$

EG

Theorem: Cramer-Shoup is IND-CCA2 secure (i.e. secure against adaptive chosen ciphertexts) if

- ① DDH holds in  $G_g$
- ②  $H$  satisfies a certain condition ( $\approx$  "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.

MIT OpenCourseWare  
<http://ocw.mit.edu>

6.034: Introduction to Algorithms  
Spring 2014

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