Public key encryption from Diffie-Hellman

Public key encryption from Diffie-Hellman

ElGamal Security

Computational Diffie-Hellman Assumption

G: finite cyclic group of order n

Comp. DH (CDH) assumption holds in G if: $g,\ g^a,\ g^b \quad \not\Rightarrow \quad g^{ab}$

for all efficient algs. A:

$$\Pr\left[\ A(g, g^a, g^b) = g^{ab}\ \right] < \text{negligible}$$

where $g \longleftarrow \{\text{generators of } G\}$, $\quad a, b \longleftarrow Z_n$

Hash Diffie-Hellman Assumption

G: finite cyclic group of order n , $\quad$ $H: G^2 \longrightarrow K$ a hash function

Def: Hash-DH (HDH) assumption holds for (G, H) if:

$$\left(g,\ g^a,\ g^b,\ H(g^b, g^{ab})\right) \approx_p \left(g,\ g^a,\ g^b,\ R\right)$$

where $\quad g \longleftarrow \{\text{generators of } G\}$, $\quad a, b \longleftarrow Z_n$, $R \longleftarrow K$

H acts as an extractor: strange distribution on $G^2 \Rightarrow$ uniform on K

Suppose $K = \{0,1\}^{128}$ and

$H: G^2 \longrightarrow K$ only outputs strings in K that begin with 0

(i.e. for all x,y: $msb(H(x,y))=0$)

Can Hash-DH hold for (G, H) ?

- Yes, for some groups G
- No, Hash-DH is easy to break in this case
- Yes, Hash-DH is always true for such H

ElGamal is sem. secure under Hash-DH

KeyGen: $g \longleftarrow$ {generators of G} , $a \longleftarrow Z_n$

output $pk = (g, h=g^a)$, $sk = a$

E(pk=(g,h), m) : $b \longleftarrow Z_n$

$k \longleftarrow H(g^b, h^b)$, $c \longleftarrow E_s(k, m)$

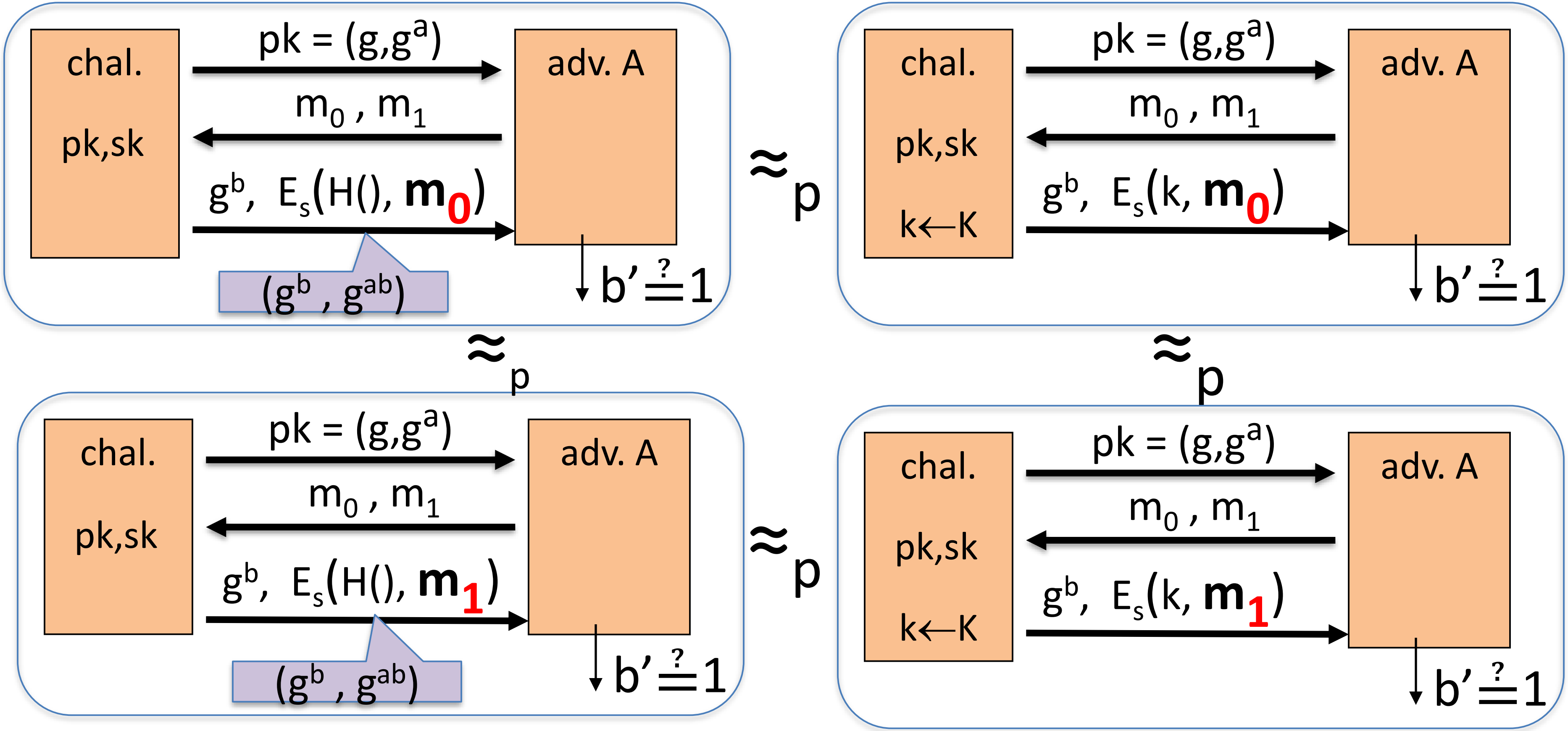
output (g^b, c)

D(sk=a, (u,c)) :

$k \longleftarrow H(u, u^a)$, $m \longleftarrow D_s(k, c)$

output m

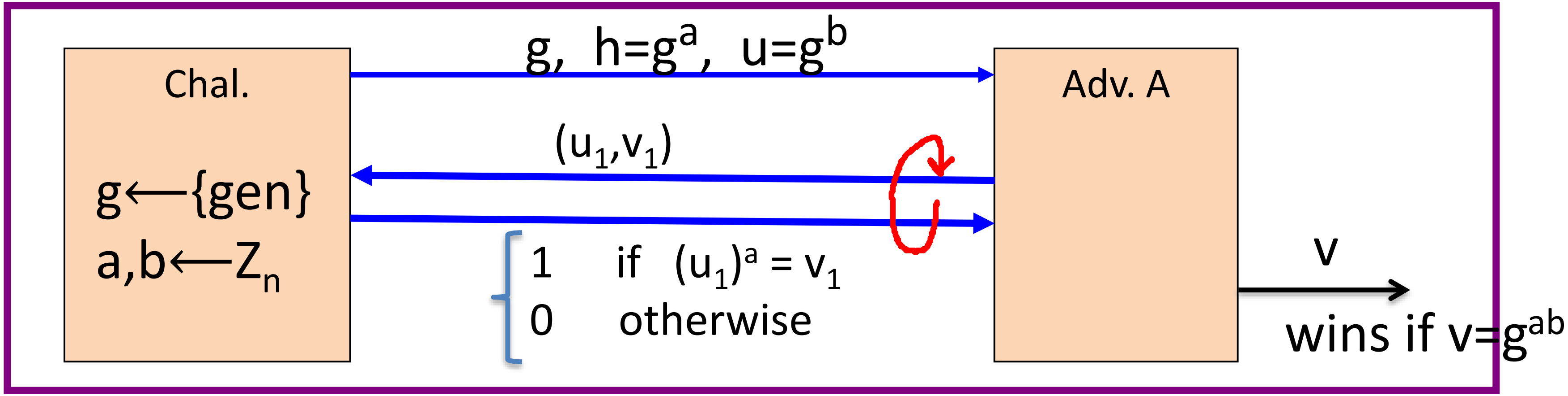
ElGamal is sem. secure under Hash-DH



ElGamal chosen ciphertext security?

To prove chosen ciphertext security need stronger assumption

Interactive Diffie-Hellman (IDH) in group G:



IDH holds in G if: **∀efficient A: Pr[A outputs g^{ab}] < negligible**

ElGamal chosen ciphertext security?

<u>Security Theorem</u>:

If **IDH** holds in the group G, **(E_s, D_s)** provides auth. enc.

and **H:** $G^2 \longrightarrow K$ is a "random oracle"

then **ElGamal** is CCA^{ro} secure.

Questions: (1) can we prove CCA security based on CDH?

(2) can we prove CCA security without random oracles?

End of Segment

Public key encryption from Diffie-Hellman

ElGamal Variants With Better Security

Review: ElGamal encryption

KeyGen: $g \longleftarrow$ {generators of G} , $a \longleftarrow Z_n$

output $pk = (g,\ h=g^a)$, $sk = a$

<u>E(pk=(g,h), m)</u> : $b \longleftarrow Z_n$

$k \longleftarrow H(g^b, h^b)$, $c \longleftarrow E_s(k, m)$

output (g^b, c)

<u>D(sk=a, (u,c))</u> :

$k \longleftarrow H(u, u^a)$, $m \longleftarrow D_s(k, c)$

output m

ElGamal chosen ciphertext security

Security Theorem:

If **IDH** holds in the group G, **(E_s, D_s)** provides auth. enc.

and **H:** $G^2 \longrightarrow K$ is a "random oracle"

then **ElGamal** is CCA^{ro} secure.

Can we prove CCA security based on CDH $(g, g^a, g^b \not\rightarrow g^{ab})$?

- Option 1: use group G where CDH = IDH (a.k.a bilinear group)
- Option 2: change the ElGamal system

Variants: twin ElGamal [CKS'08]

KeyGen: $g \longleftarrow \{\text{generators of } G\}$, $a1, a2 \longleftarrow Z_n$

output $pk = (g, h_1 = g^{a1}, h_2 = g^{a2})$, $sk = (a1, a2)$

E(pk=(g,h_1,h_2), m) : $b \longleftarrow Z_n$

$k \longleftarrow H(g^b, h_1{}^b, h_2{}^b)$

$c \longleftarrow E_s(k, m)$

output (g^b, c)

D(sk=(a1,a2), (u,c)) :

$k \longleftarrow H(u, u^{a1}, u^{a2})$

$m \longleftarrow D_s(k, c)$

output m

Chosen ciphertext security

Security Theorem:

If **CDH** holds in the group G, $(\mathbf{E_s}, \mathbf{D_s})$ provides auth. enc.

and **H:** $G^3 \longrightarrow K$ is a "random oracle"

then **twin ElGamal** is CCA^{ro} secure.

Cost: one more exponentiation during enc/dec

– Is it worth it? No one knows …

ElGamal security w/o random oracles?

Can we prove CCA security without random oracles?

- Option 1: use Hash-DH assumption in "bilinear groups"
  - Special elliptic curve with more structure [CHK'04 + BB'04]

- Option 2: use Decision-DH assumption in any group [CS'98]

Further Reading

- The Decision Diffie-Hellman problem. D. Boneh, ANTS 3, 1998
- Universal hash proofs and a paradigm for chosen ciphertext secure public key encryption. R. Cramer and V. Shoup, Eurocrypt 2002
- Chosen-ciphertext security from Identity-Based Encryption.
  D. Boneh, R. Canetti, S. Halevi, and J. Katz, SICOMP 2007
- The Twin Diffie-Hellman problem and applications.
  D. Cash, E. Kiltz, V. Shoup, Eurocrypt 2008
- Efficient chosen-ciphertext security via extractable hash proofs.
  H. Wee, Crypto 2010

Public key encryption from Diffie-Hellman

A Unifying Theme

One-way functions (informal)

A function $f: X \longrightarrow Y$ is one-way if

- There is an efficient algorithm to evaluate $f(\cdot)$, but

- Inverting f is hard:

  for all efficient A and $x \leftarrow X$:

$$\Pr\left[\, f\big(A(f(x))\big) = f(x) \,\right] < \text{negligible}$$

Functions that are not one-way: $f(x) = x$, $f(x) = 0$

Ex. 1: generic one-way functions

Let $f: X \longrightarrow Y$ be a secure PRG (where $|Y| \gg |X|$)

(e.g. f built using det. counter mode)

Lemma: f a secure PRG $\Rightarrow$ f is one-way

Proof sketch:

$$A \text{ inverts } f \quad \Rightarrow \quad B(y) = \begin{cases} 0 & \text{if } f(A(y)) = y \\ 1 & \text{otherwise} \end{cases} \quad \text{is a distinguisher}$$

Generic: no special properties. Difficult to use for key exchange.

Ex 2: The DLOG one-way function

Fix a finite cyclic group G (e.g $G = (Z_p)^*$) of order n

g: a random generator in G (i.e. $G = \{1, g, g^2, g^3, \ldots, g^{n-1}\}$)

Define: $f: Z_n \longrightarrow G$ as $\boxed{f(x) = g^x \in G}$

Lemma: Dlog hard in G $\Rightarrow$ f is one-way

Properties: $f(x), f(y) \Rightarrow f(x+y) = f(x) \cdot f(y)$

$\Rightarrow$ key-exchange and public-key encryption

Ex. 3: The RSA one-way function

- choose random primes $p,q \approx 1024$ bits. Set **N=pq**.
- choose integers **e , d** s.t. **$e \cdot d = 1 \pmod{\varphi(N)}$**

Define: f: $\mathbb{Z}_N^* \to \mathbb{Z}_N^*$ as $\boxed{f(x) = x^e \quad \text{in} \quad \mathbb{Z}_N}$

Lemma: f is one-way under the RSA assumption

Properties: $f(x \cdot y) = f(x) \cdot f(y)$ and **f has a trapdoor**

Summary

Public key encryption:

made possible by one-way functions
with special properties

homomorphic properties and trapdoors

End of Segment