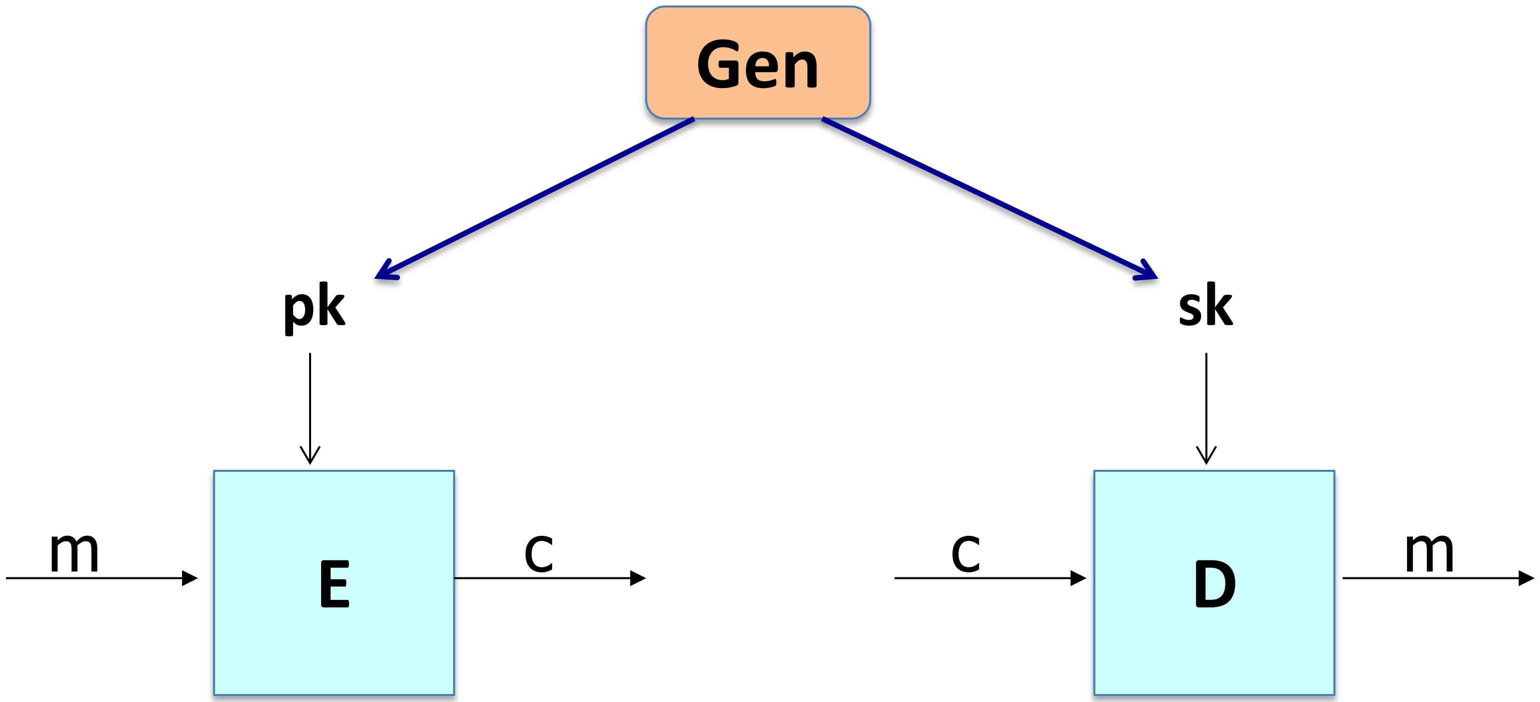
Public key encryption from Diffie-Hellman

This slide is made based the online course of Cryptography by Dan Boneh

Recap: public key encryption: (Gen, E, D)

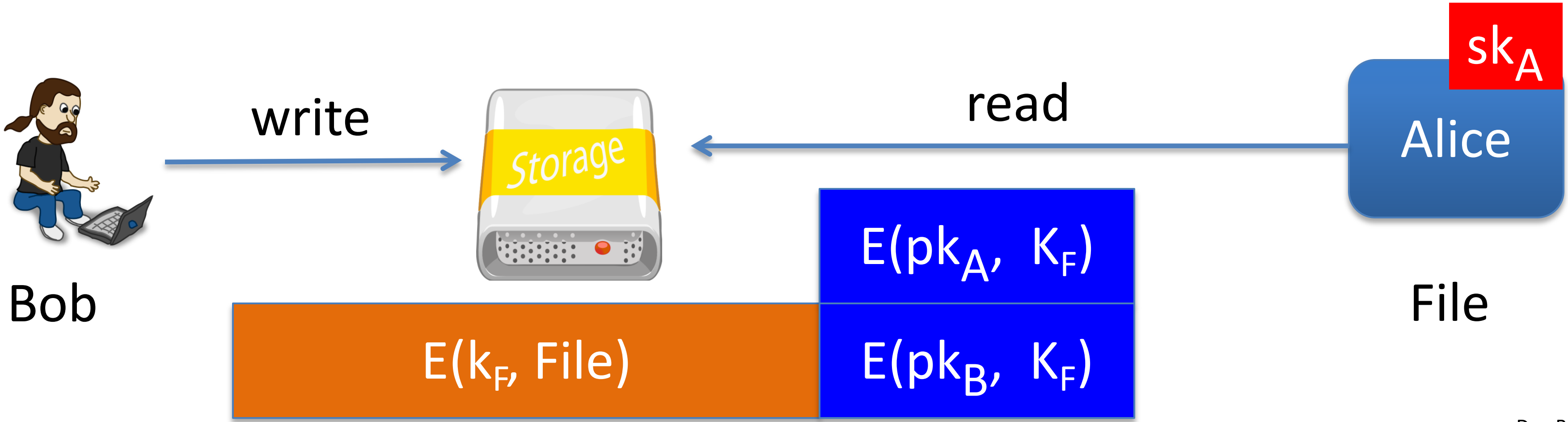


Recap: public-key encryption applications

Key exchange (e.g. in HTTPS)

Encryption in non-interactive settings:

- Secure Email: Bob has Alice's pub-key and sends her an email
- Encrypted File Systems

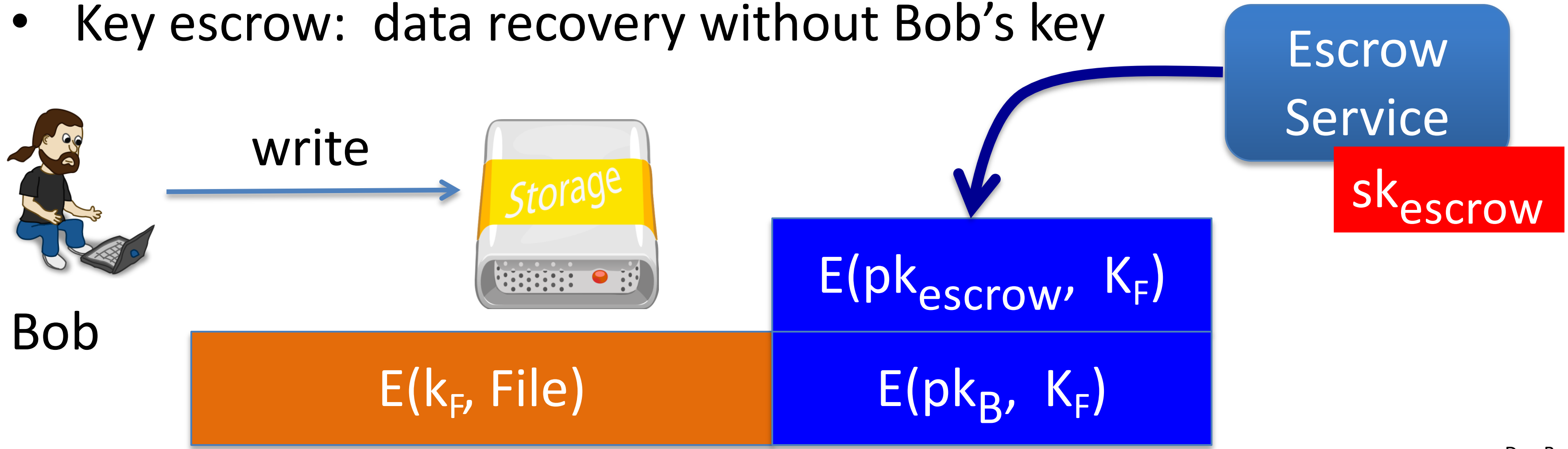


Recap: public-key encryption applications

Key exchange (e.g. in HTTPS)

Encryption in non-interactive settings:

- Secure Email: Bob has Alice's pub-key and sends her an email
- Encrypted File Systems
- Key escrow: data recovery without Bob's key



Constructions

This week: two families of public-key encryption schemes

- Previous lecture: based on trapdoor functions (such as RSA)
  - Schemes: ISO standard, OAEP+, …

- This lecture: based on the Diffie-Hellman protocol
  - Schemes: ElGamal encryption and variants (e.g. used in GPG)

Security goals: chosen ciphertext security

Review: the Diffie-Hellman protocol (1977)

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

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

Alice

choose random **a** in {1,…,n}

Bob

choose random **b** in {1,…,n}

$A = g^a$ $\longrightarrow$

$\longleftarrow$ $B = g^b$

$\mathbf{B^a} = (g^b)^a = \boxed{\mathbf{k_{AB} = g^{ab}}} = (g^a)^b = \mathbf{A^b}$

ElGamal: converting to pub-key enc. (1984)

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

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

Alice

Bob

choose random **a** in {1,…,n}

ndom **b** in {1,…,n}

$A = g^a$

Treat as a public key

compute $g^{ab} = A^b$,

derive symmetric key k ,

ct = [$B = g^b$, encrypt message m with k]

ElGamal: converting to pub-key enc. (1984)

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

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

Alice

Bob

choose random **a** in {1,…,n}

ndom **b** in {1,…,n}

Treat as a public key

$A = g^a$

compute $g^{ab} = A^b$,
derive symmetric key k ,
encrypt message m with k

ct = [$B = g^b$, compute $g^{ab} = A^b$, derive symmetric key k, encrypt message m with k]

To decrypt:
compute $g^{ab} = B^a$,
derive k, and decrypt

The ElGamal system (a modern view)

- G: finite cyclic group of order n
- (E_s, D_s) : symmetric auth. encryption defined over (K,M,C)
- $H: G^2 \longrightarrow K$ a hash function

We construct a pub-key enc. system (Gen, E, D):

- Key generation Gen:
  - choose random generator g in G and random a in Z_n
  - output sk = a , pk = (g, $h=g^a$)

The ElGamal system (a modern view)

- G: finite cyclic group of order n
- (E_s, D_s) : symmetric auth. encryption defined over (K,M,C)
- $H: G^2 \longrightarrow K$ a hash function

<u>E(pk=(g,h), m)</u> :

$b \xleftarrow{R} Z_n$, $u \longleftarrow g^b$, $v \longleftarrow h^b$

$k \longleftarrow H(u,v)$, $c \longleftarrow E_s(k, m)$

output (u, c)

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

$v \longleftarrow u^a$

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

output m

ElGamal performance

E(pk=(g,h), m) :

$b \longleftarrow Z_n$, $u \longleftarrow g^b$, $v \longleftarrow h^b$

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

$v \longleftarrow u^a$

Encryption: 2 exp. (fixed basis)

- Can pre-compute $[\, g^{(2^i)}\,,\ h^{(2^i)}$ for $i=1,\ldots,\log_2 n\,]$
- 3x speed-up (or more)

Decryption: 1 exp. (variable basis)

Next step: why is this system chosen ciphertext secure?

under what assumptions?

End of Segment

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

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

ElGamal is sem. secure under Hash-DH

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

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

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

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

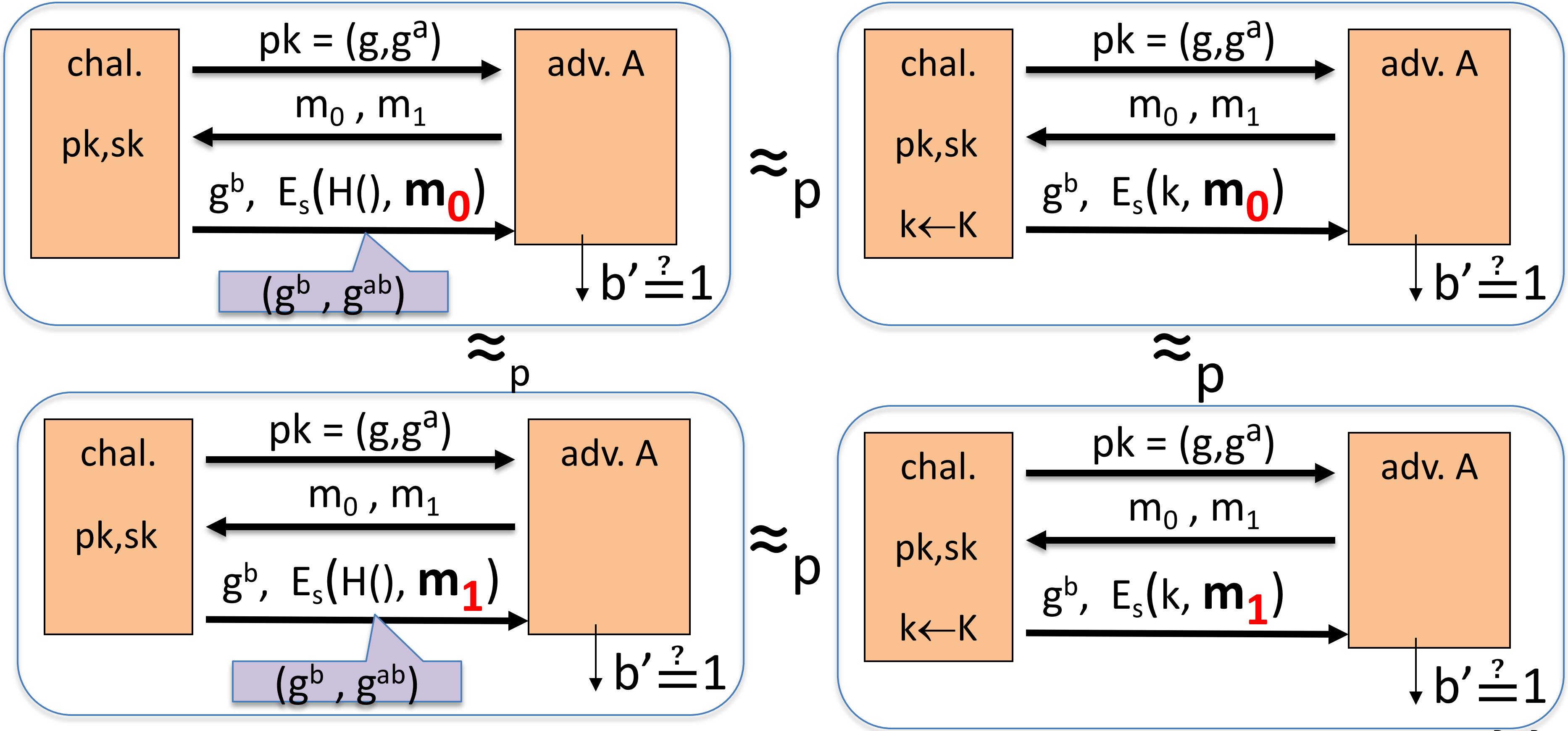
output (g^b, c)

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

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

output m

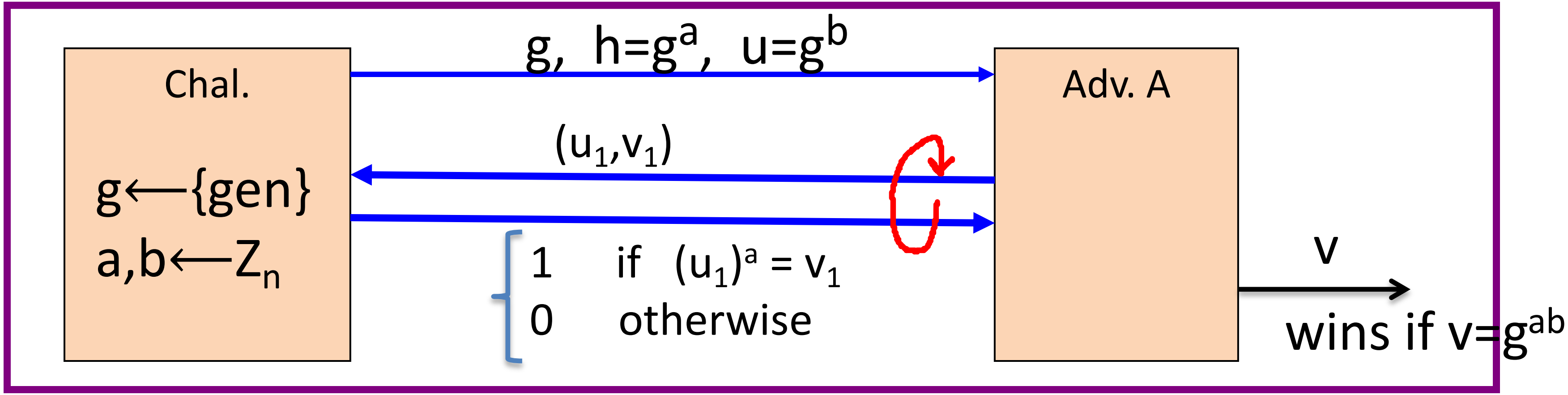
ElGamal is sem. secure under Hash-DH



ElGamal chosen ciphertext security?

To prove chosen ciphertext security need stronger assumption

Interactive Diffie-Hellman (IDH) in group G:



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

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.

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: $g \leftarrow \{\text{generators of } G\}$, $a \leftarrow Z_n$

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

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

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

output (g^b, c)

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

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

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: $g \longleftarrow$ {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, (E_s, 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