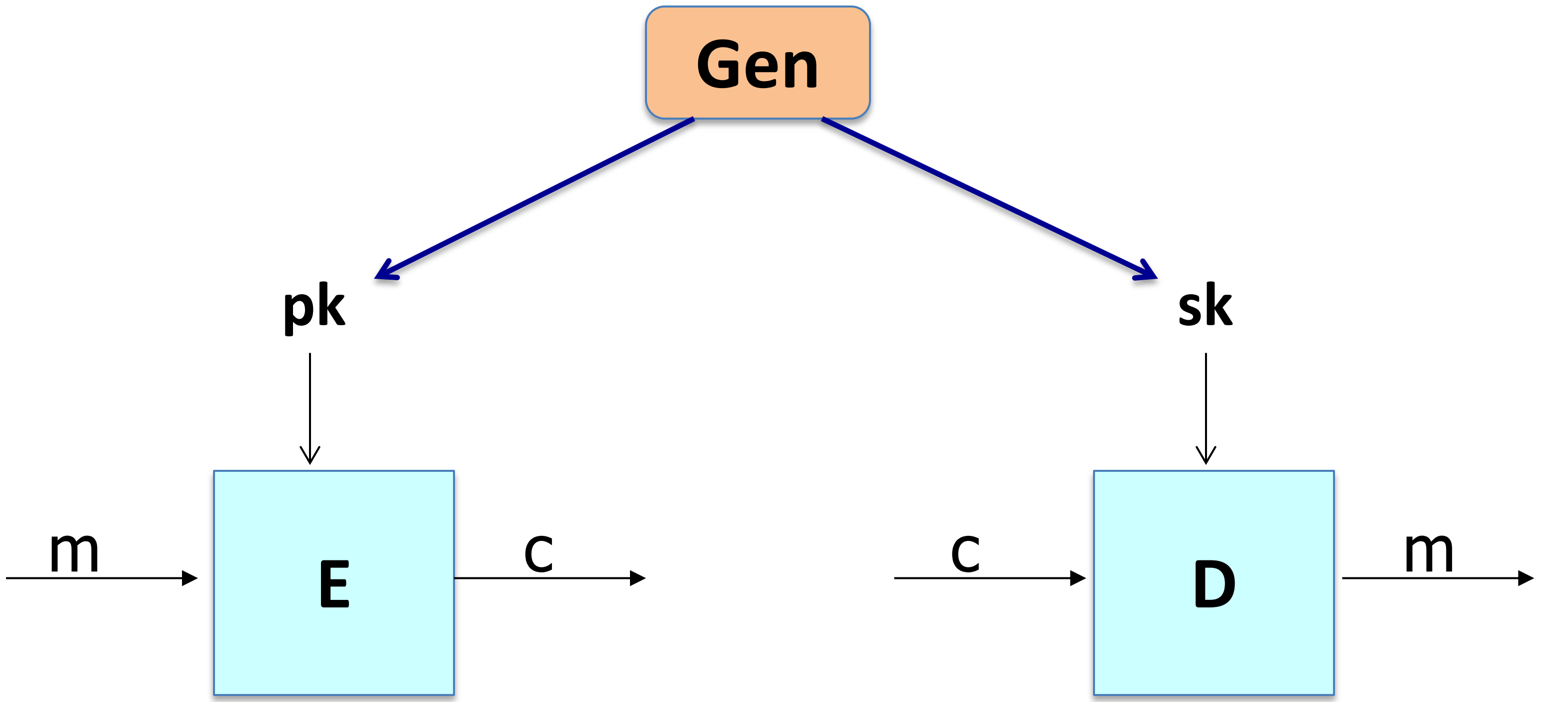
Public key encryption from Diffie-Hellman

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

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

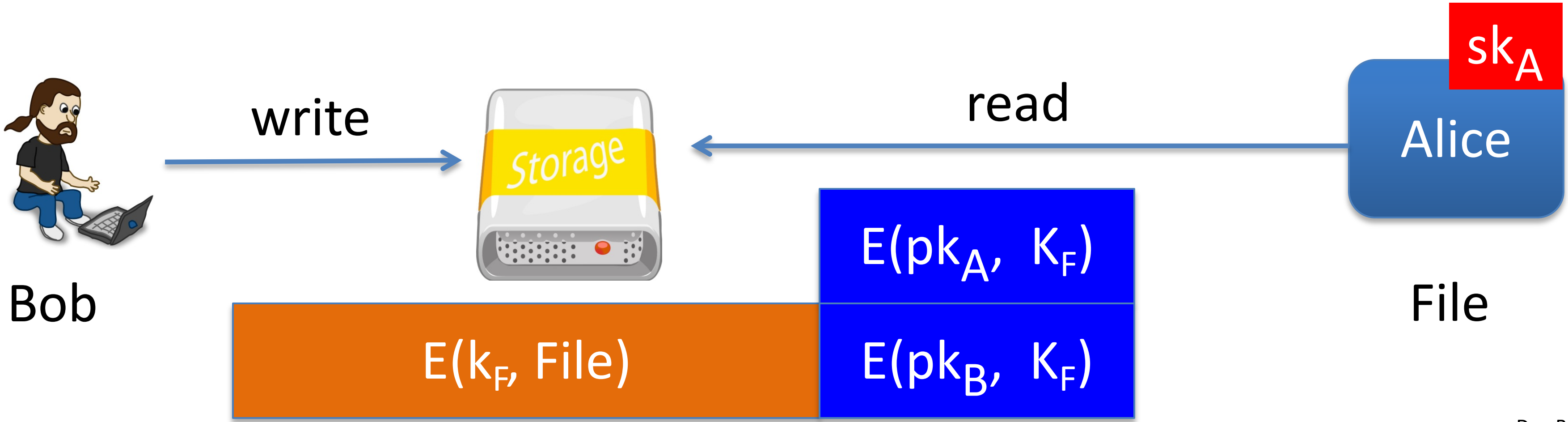


Recap: public-key encryption applications

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

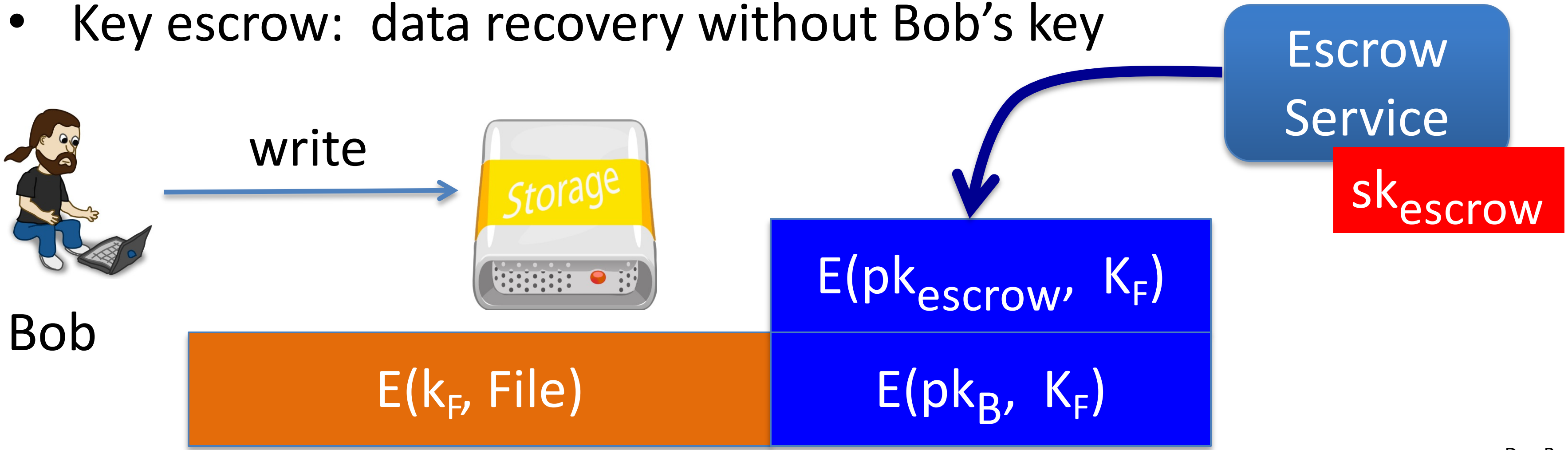


Recap: public-key encryption applications

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

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 **Bob**

choose random **a** in {1,…,n} 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. $G = \{1, g, g^2, g^3, \ldots, g^{n-1}\}$)

Alice | **Bob**

choose random **a** in {1,…,n}

Treat as a public key

…ndom **b** in {1,…,n}

$A = g^a$ →

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. $G = \{1, g, g^2, g^3, \ldots, g^{n-1}\}$)

<u>Alice</u> **<u>Bob</u>**

choose random **a** in {1,…,n}

Treat as a public key

ndom **b** in {1,…,n}

$A = g^a$

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

derive symmetric key k ,

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

To decrypt:

compute $g^{ab} = B^a$,

derive k, and decrypt

Traditional Elgamal Version

- Gen: on input 1^n run $\mathcal{G}(1^n)$ to obtain $(\mathbb{G}, q, g)$. Then choose a uniform $x \in \mathbb{Z}_q$ and compute $h := g^x$. The public key is $\langle \mathbb{G}, q, g, h \rangle$ and the private key is $\langle \mathbb{G}, q, g, x \rangle$. The message space is $\mathbb{G}$.
- Enc: on input a public key $pk = \langle \mathbb{G}, q, g, h \rangle$ and a message $m \in \mathbb{G}$, choose a uniform $y \in \mathbb{Z}_q$ and output the ciphertext

$$\langle g^y,\ h^y \cdot m \rangle.$$

- Dec: on input a private key $sk = \langle \mathbb{G}, q, g, x \rangle$ and a ciphertext $\langle c_1, c_2 \rangle$, output

$$\hat{m} := c_2 / c_1^x.$$

Theorem: *If the DDH problem is hard relative to $\mathcal{G}$, then the El Gamal encryption scheme is CPA-secure.*

Working in Subgroups of Z_p^*

- DL is hard in subgroup of Z_p^* with prime order q
- DDH is hard in the subgroup with prime order q

Let $p = rq + 1$ *with* p, q *prime. Then*

$$\mathbb{G} \stackrel{\text{def}}{=} \{[h^r \bmod p] \mid h \in \mathbb{Z}_p^*\}$$

is a subgroup of $\mathbb{Z}_p^*$ *of order* q.

If group Z_p^* is used, then traditional Elgamal is not secure in DDH assumption.

DDH doesn't hold in $\mathbb{Z}_p^*$

We show that the DDH assumption doesn't hold in $\mathbb{Z}_p^*$ by using a subset of $\mathbb{Z}_p^*$, called the quadratic residue which is defined as follows.

$$QR_p = \{f : \exists h \in \mathbb{Z}_p^* \quad s.t. \quad f = h^2\} = \{g^i : i \text{ is even}\}$$
$$f = h^2 = g^{2j \mod p-1} = g^i, i \text{ is even}$$

Claim: $f \in QR_p \iff f^{(p-1)/2} = 1$. This is easy to verify by looking at $f = g^i \Rightarrow f^{(p-1)/2} = g^{i(p-1)/2}$. If i is even, then this $f^{(p-1)/2} = f^{(p-1)} = 1$ and it is not 1 if i is odd.

xy	x is even	x is odd
y is even	even	even
y is odd	even	odd

$g^{xy} \in QR_p$? → ¾ if x or y is even

↓

Random case: 1/2

Distinguisher!

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

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

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

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

output (u, c)

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

$v \longleftarrow u^a$

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

output m

ElGamal performance

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

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

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

$v \leftarrow u^a$

Encryption: 2 exp. (fixed basis)

- Can pre-compute $[\ g^{(2^i)}\ ,\ h^{(2^i)}\ \ \text{for}\ \ i=1,\ldots,\log_2 n\ ]$
- 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

ElGamal encryption

- a cyclic group $\mathbb{G}$ of prime order q with generator $g \in \mathbb{G}$,
- a symmetric cipher $\mathcal{E}_s = (E_s, D_s)$, defined over $(\mathcal{K}, \mathcal{M}, \mathcal{C})$,
- a hash function $H : \mathbb{G}^2 \to \mathcal{K}$.

- the key generation algorithm runs as follows:

$$\begin{aligned} G() := \quad & \alpha \xleftarrow{R} \mathbb{Z}_q, \quad u \leftarrow g^\alpha \\ & pk \leftarrow u, \quad sk \leftarrow \alpha \\ & \text{output } (pk, sk); \end{aligned}$$

- for a given public key $pk = u \in \mathbb{G}$ and message $m \in \mathcal{M}$, the encryption algorithm runs as follows:

$$\begin{aligned} E(pk, m) := \quad & \beta \xleftarrow{R} \mathbb{Z}_q, \quad v \leftarrow g^\beta, \quad w \leftarrow u^\beta, \quad k \leftarrow H(v, w), \quad c \leftarrow E_s(k, m) \\ & \text{output } (v, c); \end{aligned}$$

- for a given secret key $sk = \alpha \in \mathbb{Z}_q$ and a ciphertext $(v, c) \in \mathbb{G} \times \mathcal{C}$, the decryption algorithm runs as follows:

$$\begin{aligned} D(sk,\ (v, c)\ ) := \quad & w \leftarrow v^\alpha, \quad k \leftarrow H(v, w), \quad m \leftarrow D_s(k, c) \\ & \text{output } m. \end{aligned}$$

Semantic security of ElGamal without random oracles

$$k \leftarrow H(v, w)$$

Random oracles: $\mathrm{SS^{ro}adv}[\mathcal{A}, \mathcal{E}_{\mathrm{EG}}] \leq 2Q \cdot \mathrm{CDHadv}[\mathcal{B}_{\mathrm{cdh}}, \mathbb{G}] + \mathrm{SSadv}[\mathcal{B}_{\mathrm{s}}, \mathcal{E}_{\mathrm{s}}]$.

random oracle version: the challenger uses ***O*** in place of ***H*** for all its computations, and in addition, the adversary is allowed to obtain the value of ***O*** at arbitrary input points of his choosing.

secure key derivation function

$$\mathrm{SSadv}[\mathcal{A}, \mathcal{E}_{\mathrm{EG}}] \leq 2 \cdot \mathrm{DDHadv}[\mathcal{B}_{\mathrm{ddh}}, \mathbb{G}] + 2 \cdot \mathrm{KDFadv}[\mathcal{B}_{\mathrm{kdf}}, H] + \mathrm{SSadv}[\mathcal{B}_{\mathrm{s}}, \mathcal{E}_{\mathrm{s}}].$$

Key derivation

key

secret data

$$k \leftarrow H(v, w)$$

Roughly speaking, the problem is this: we start with some secret data, and we want to convert it into an *n*-bit string that we can use as the key to some cryptographic primitive, like AES.

Intuitively, $H : \mathbb{G}^2 \rightarrow \mathcal{K}$ is a secure KDF if no efficient adversary can effectively distinguish between $(v, H(w))$ and (v, k), where v and w are randomly chosen from $\mathbb{G}$, and k is randomly chosen from $\mathcal{K}$.

secure key derivation

Attack Game 11.3 (secure key derivation). For a given hash function $F : \mathcal{X} \times \mathcal{Y} \to \mathcal{Z}$, and for a given adversary $\mathcal{A}$, we define two experiments.

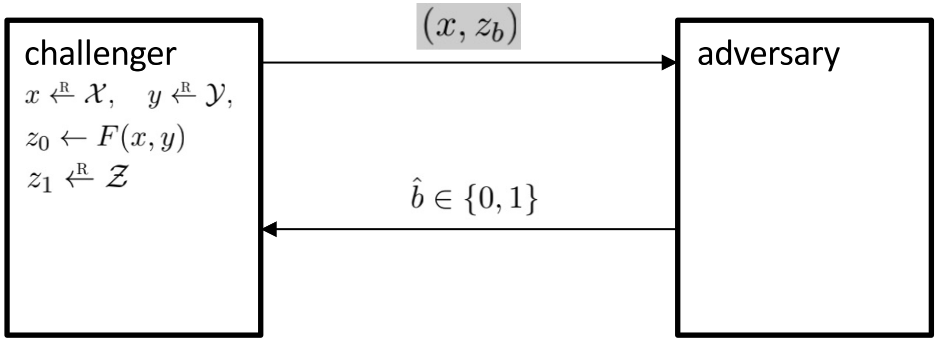
Experiment b **(**$b = 0, 1$**):**

- The challenger computes

$$x \xleftarrow{R} \mathcal{X}, \quad y \xleftarrow{R} \mathcal{Y}, \quad z_0 \leftarrow F(x, y), \quad z_1 \xleftarrow{R} \mathcal{Z},$$

  and sends (x, z_b) to the adversary.

- The adversary outputs a bit $\hat{b} \in \{0, 1\}$.



$$\mathsf{KDFadv}[\mathcal{A}, F] := \Big|\Pr[W_0] - \Pr[W_1]\Big|$$

Semantic security against chosen plaintext attack

Theorem 11.1. *If a public-key encryption scheme $\mathcal{E}$ is semantically secure, then it is also CPA secure.*

In particular, for every CPA adversary $\mathcal{A}$ that plays Attack Game 11.2 with respect to $\mathcal{E}$, and which makes at most Q queries to its challenger, there exists an SS adversary $\mathcal{B}$, where $\mathcal{B}$ is an elementary wrapper around $\mathcal{A}$, such that

$$\mathrm{CPA}\mathsf{adv}[\mathcal{A}, \mathcal{E}] = Q \cdot \mathrm{SS}\mathsf{adv}[\mathcal{B}, \mathcal{E}].$$

GAME 0

challenger

$\alpha, \beta \xleftarrow{R} \mathbb{Z}_q, \; u \leftarrow g^{\alpha}$

$v \leftarrow g^{\beta}, \; w \leftarrow g^{\alpha\beta}$

$k \leftarrow H(v, w)$

$b \xleftarrow{R} \{0,1\}$

$c \leftarrow E_{\mathrm{s}}(k, m_b)$

adversary

challenger $\xrightarrow{\text{public key } u}$ adversary

adversary $\xrightarrow{(m_0, m_1) \in \mathcal{M}^2}$ challenger

challenger $\xrightarrow{(v, c)}$ adversary

adversary $\xrightarrow{\hat{b}}$ challenger

$$\boxed{\mathrm{SSadv}^*[\mathcal{A}, \mathcal{E}_{\mathrm{EG}}] = |\Pr[W_0] - 1/2|}$$

GAME 1

challenger

$\alpha, \beta \xleftarrow{R} \mathbb{Z}_q,\ u \leftarrow g^{\alpha}$ $\quad \gamma \xleftarrow{R} \mathbb{Z}_q,$

$v \leftarrow g^{\beta},\ \cancel{w \leftarrow g^{\alpha\beta}}\ \ w \leftarrow g^{\gamma}$

$$k \leftarrow H(v, w)$$

$$b \xleftarrow{R} \{0,1\}$$

$$c \leftarrow E_s(k, m_b)$$

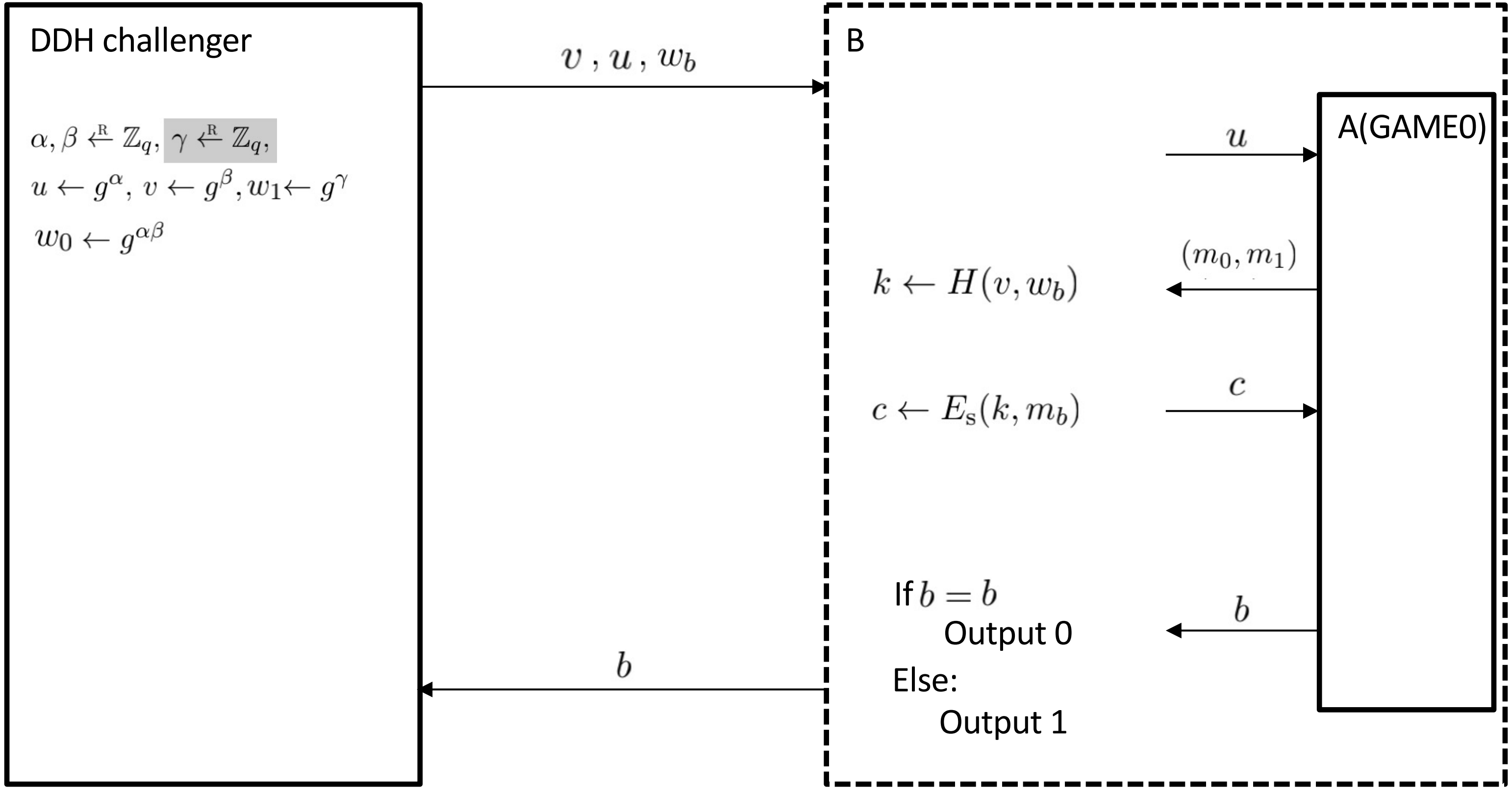
adversary

challenger $\xrightarrow{\text{public key } u}$ adversary

adversary $\xrightarrow{(m_0, m_1) \in \mathcal{M}^2}$ challenger

challenger $\xrightarrow{(v, c)}$ adversary

adversary $\xrightarrow{\hat{b}}$ challenger

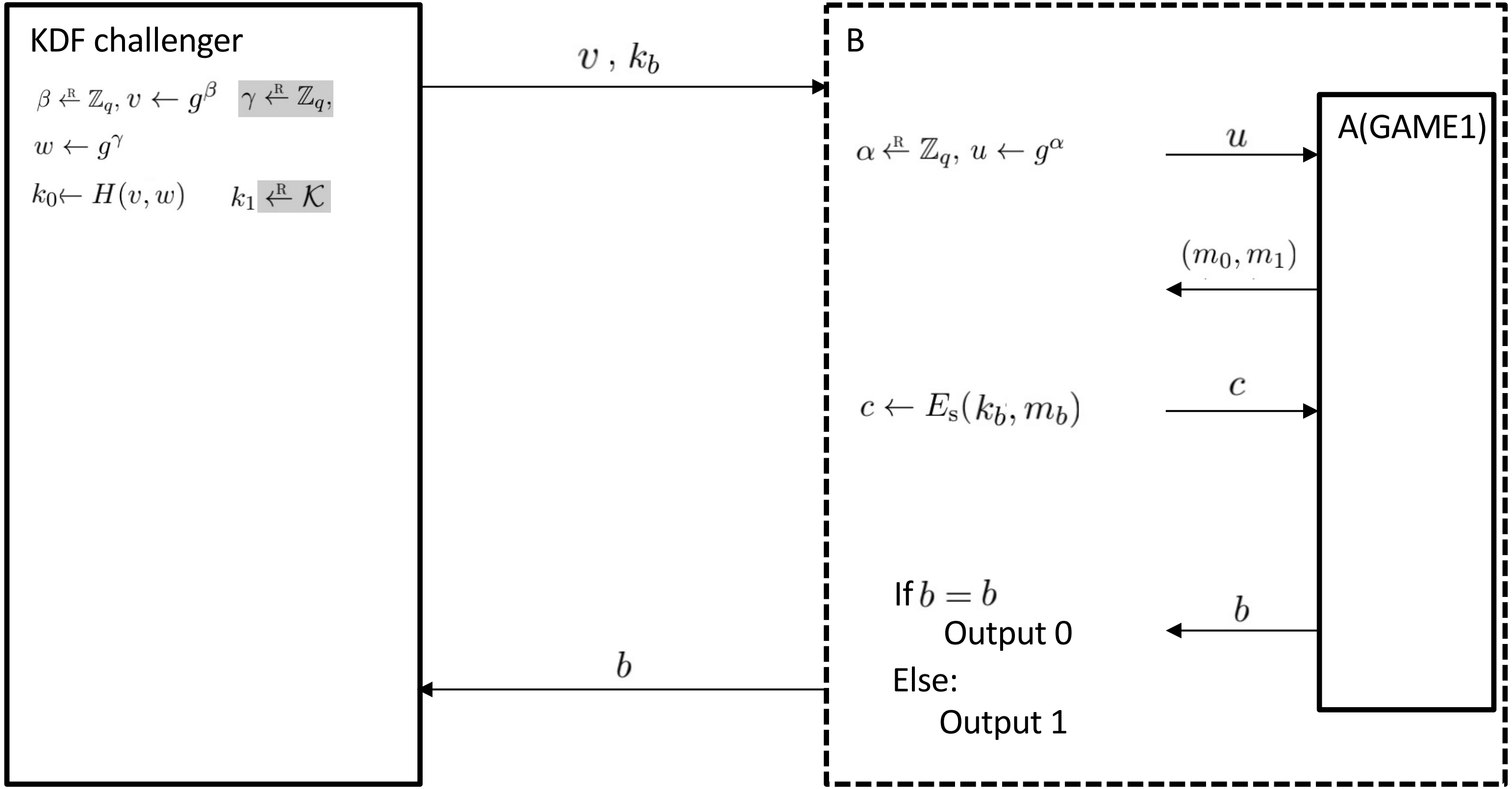


$$|\Pr[W_0] - \Pr[W_1]| = \text{DDHadv}[\mathcal{B}_{\text{ddh}}, \mathbb{G}].$$

GAME 2

challenger		adversary
$\alpha, \beta \xleftarrow{R} \mathbb{Z}_q,\ u \leftarrow g^{\alpha}\quad \boxed{\gamma \xleftarrow{R} \mathbb{Z}_q,}$	public key u $\longrightarrow$	
$v \leftarrow g^{\beta},\ \cancel{w \leftarrow g^{\alpha\beta}}\quad w \leftarrow g^{\gamma}$		
$\cancel{k \leftarrow H(v, w)}\quad \boxed{k \xleftarrow{R} \mathcal{K}}$	$\longleftarrow$ $(m_0, m_1) \in \mathcal{M}^2$	
$b \xleftarrow{R} \{0, 1\}$		
$c \leftarrow E_s(k, m_b)$	(v, c) $\longrightarrow$	
	$\longleftarrow$ $\hat{b}$	

$$\boxed{|\Pr[W_2] - 1/2| = \mathrm{SSadv}^*[\mathcal{B}_s, \mathcal{E}_s].}$$



$$|\Pr[W_1] - \Pr[W_2]| = \text{KDFadv}[\mathcal{B}_{\text{kdf}}, H].$$

$$\mathsf{SSadv}^*[\mathcal{A}, \mathcal{E}_{\mathrm{EG}}] = |\Pr[W_0] - 1/2|$$

$$|\Pr[W_0] - \Pr[W_1]| = \mathrm{DDH}\mathsf{adv}[\mathcal{B}_{\mathrm{ddh}}, \mathbb{G}].$$

$$|\Pr[W_2] - 1/2| = \mathsf{SSadv}^*[\mathcal{B}_{\mathrm{s}}, \mathcal{E}_{\mathrm{s}}].$$

$$|\Pr[W_1] - \Pr[W_2]| = \mathrm{KDF}\mathsf{adv}[\mathcal{B}_{\mathrm{kdf}}, H].$$

$$\Downarrow$$

$$\mathsf{SSadv}^*[\mathcal{A}, \mathcal{E}_{\mathrm{EG}}] \le \mathrm{DDH}\mathsf{adv}[\mathcal{B}_{\mathrm{ddh}}, \mathbb{G}] + \mathrm{KDF}\mathsf{adv}[\mathcal{B}_{\mathrm{kdf}}, H] + \mathsf{SSadv}^*[\mathcal{B}_{\mathrm{s}}, \mathcal{E}_{\mathrm{s}}].$$

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:

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

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

Hash Diffie-Hellman Assumption

G: finite cyclic group of order n , $H: G^2 \longrightarrow K$ 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 $g \longleftarrow \{\text{generators of } G\}$, $a, b \longleftarrow Z_n$, $R \longleftarrow K$

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> : $b \longleftarrow Z_n$

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

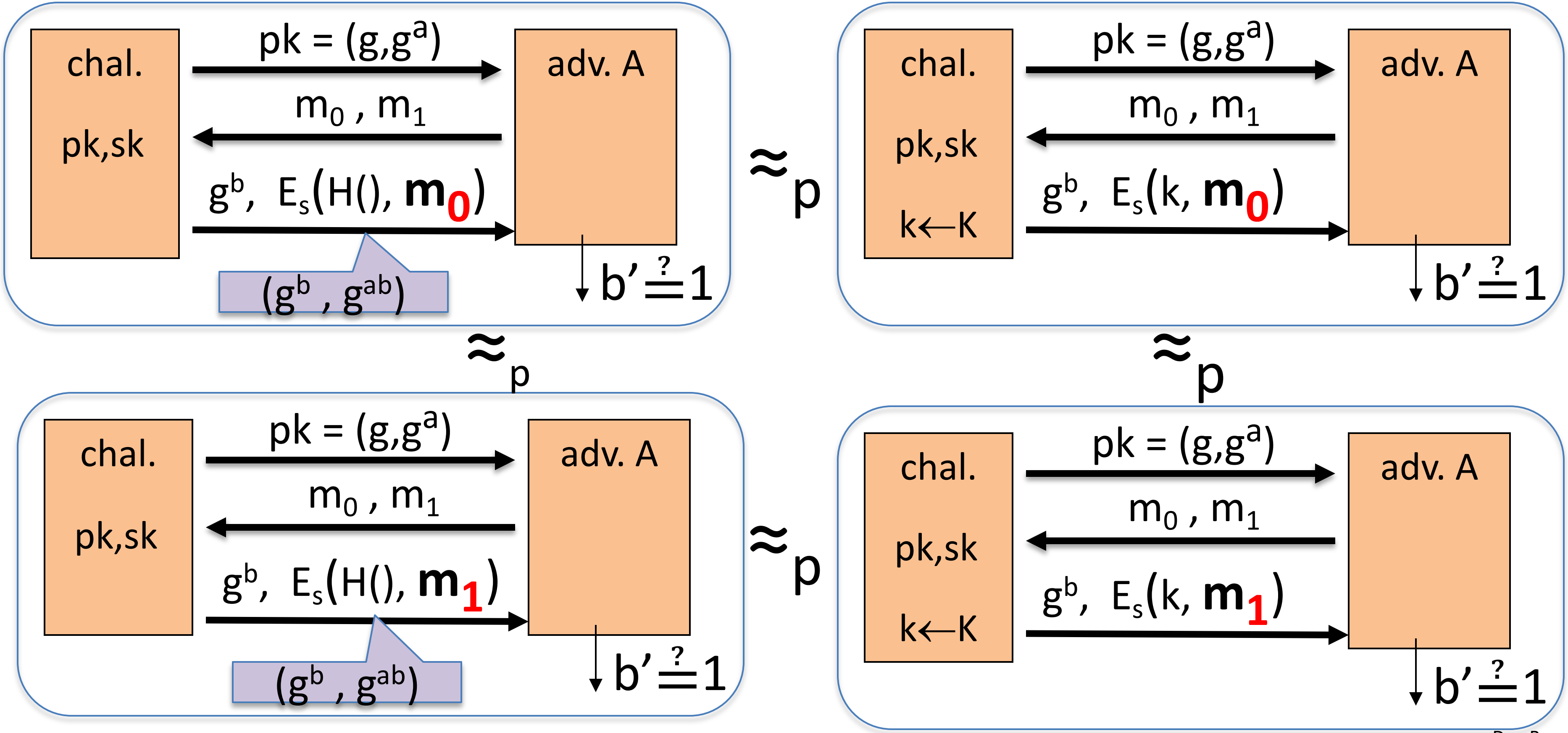
output (g^b, c)

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

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

output m

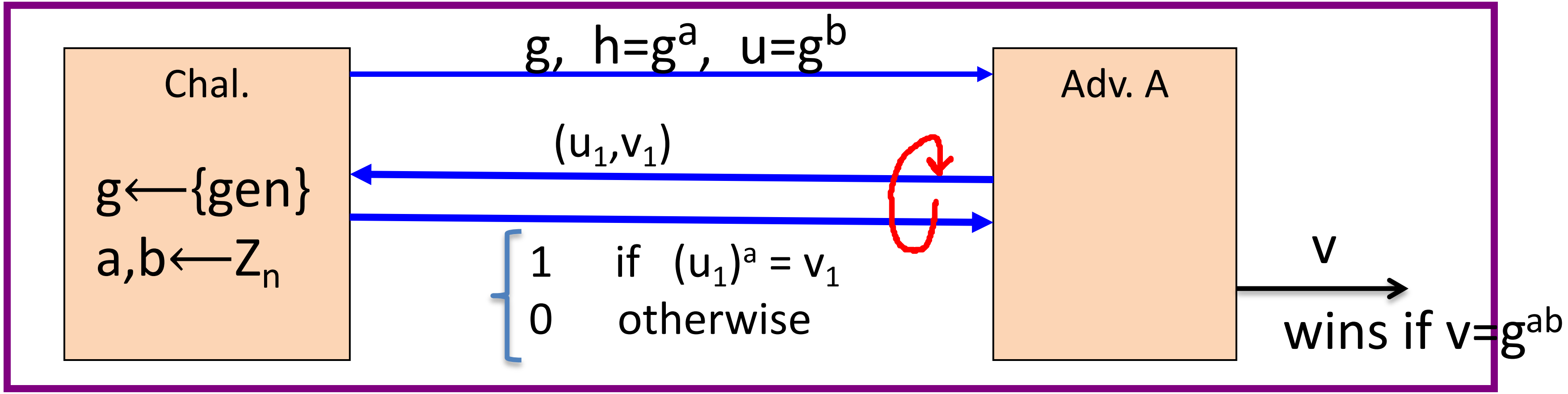
ElGamal is sem. secure under Hash-DH



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, $\mathbf{(E_s, D_s)}$ 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: $g \longleftarrow \{\text{generators of } G\}$, $a \longleftarrow Z_n$

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: $g \longleftarrow$ {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