



# Public key encryption from Diffie-Hellman

---

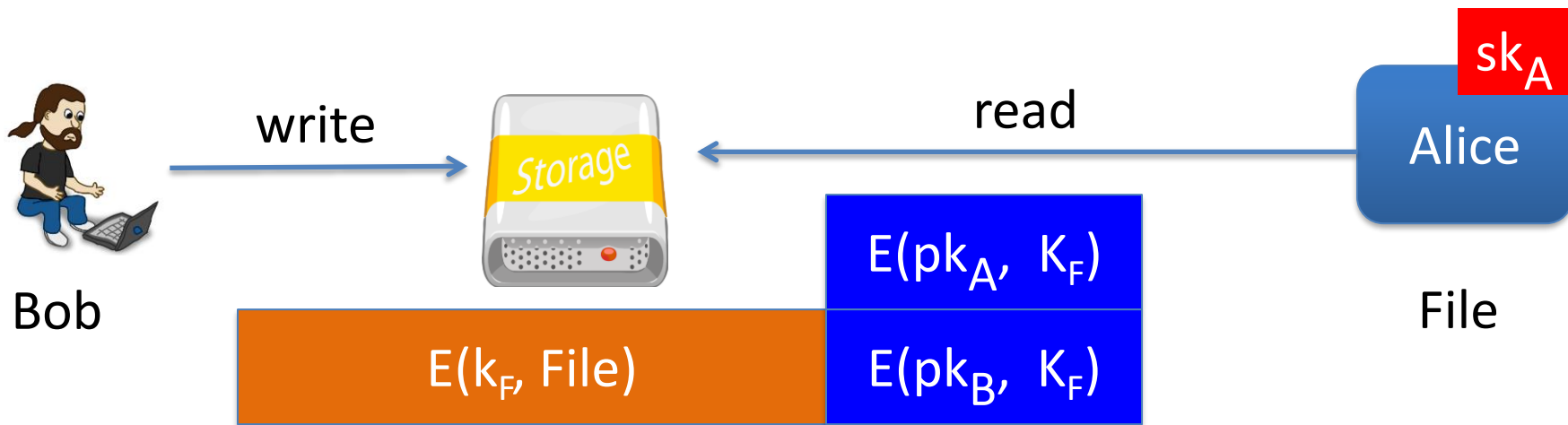
## The ElGamal Public-key System

# 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

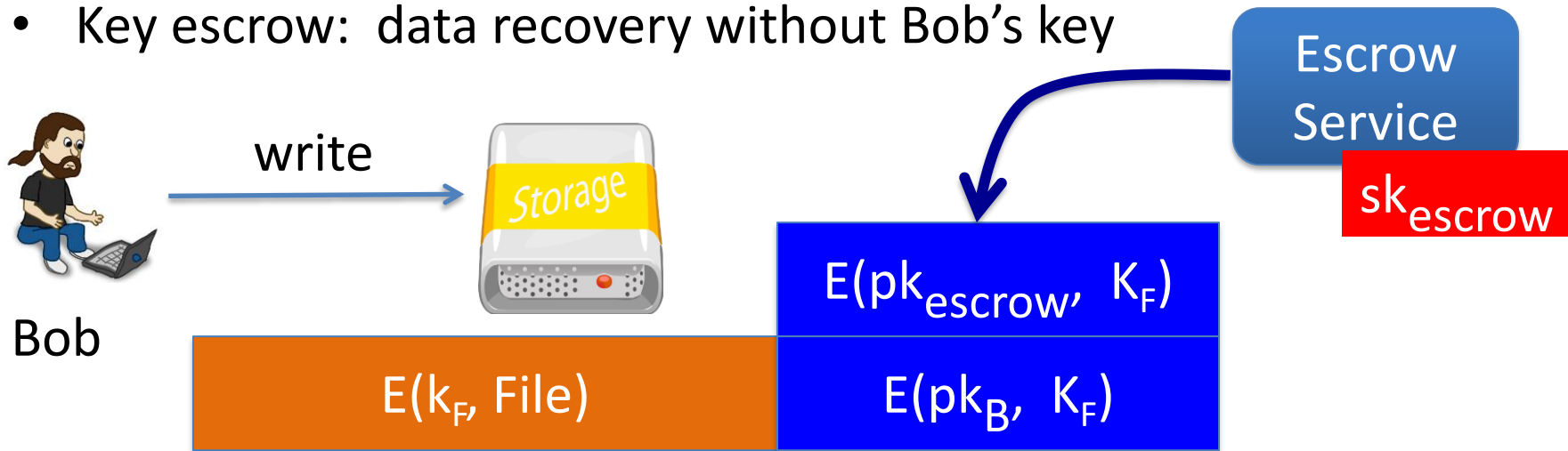


# Recap: public-key encryption applications

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



# Review: the Diffie-Hellman protocol (1977)

Fix a finite cyclic group  $G$  (e.g.  $G = (\mathbb{Z}_p)^*$ ) of order  $n$

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

Alice

choose random  $\mathbf{a}$  in  $\{1, \dots, n\}$

Bob

choose random  $\mathbf{b}$  in  $\{1, \dots, n\}$



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

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

Fix a finite cyclic group  $G$  (e.g.  $G = (\mathbb{Z}_p)^*$ ) of order  $n$

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

**Alice**

choose random  $\mathbf{a}$  in  $\{1, \dots, n\}$

$$A = g^a$$

Treat as a  
public key

**Bob**

choose random  $\mathbf{b}$  in  $\{1, \dots, n\}$

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

derive symmetric key  $k$ ,

encrypt message  $m$  with  $k$

ct = [  $B = g^b$ , 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  $(K, M, C)$
- $H: G^2 \rightarrow K$  a hash function

We construct a pub-key enc. system  $(\text{Gen}, E, D)$ :

- Key generation  $\text{Gen}$ :
  - choose random (not fixed) 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  $(K, M, C)$
- $H: G^2 \rightarrow K$  a hash function

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

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

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

output  $(u, c)$

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

$$v \leftarrow u^a$$

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

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

**Decryption:** 1 exp. (variable basis)





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:  $g, g^a, g^b \not\Rightarrow g^{ab}$

for all efficient algs.  $A$ :

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

where  $g \leftarrow \{\text{generators of } G\}$ ,  $a, b \leftarrow \mathbb{Z}_n$

# Hash Diffie-Hellman Assumption

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

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

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

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

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

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

$H: G^2 \rightarrow K$  only outputs strings in  $K$  that begin with 0  
( i.e. for all  $x,y$ :  $\text{msb}(H(x,y))=0$  )

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

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

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

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

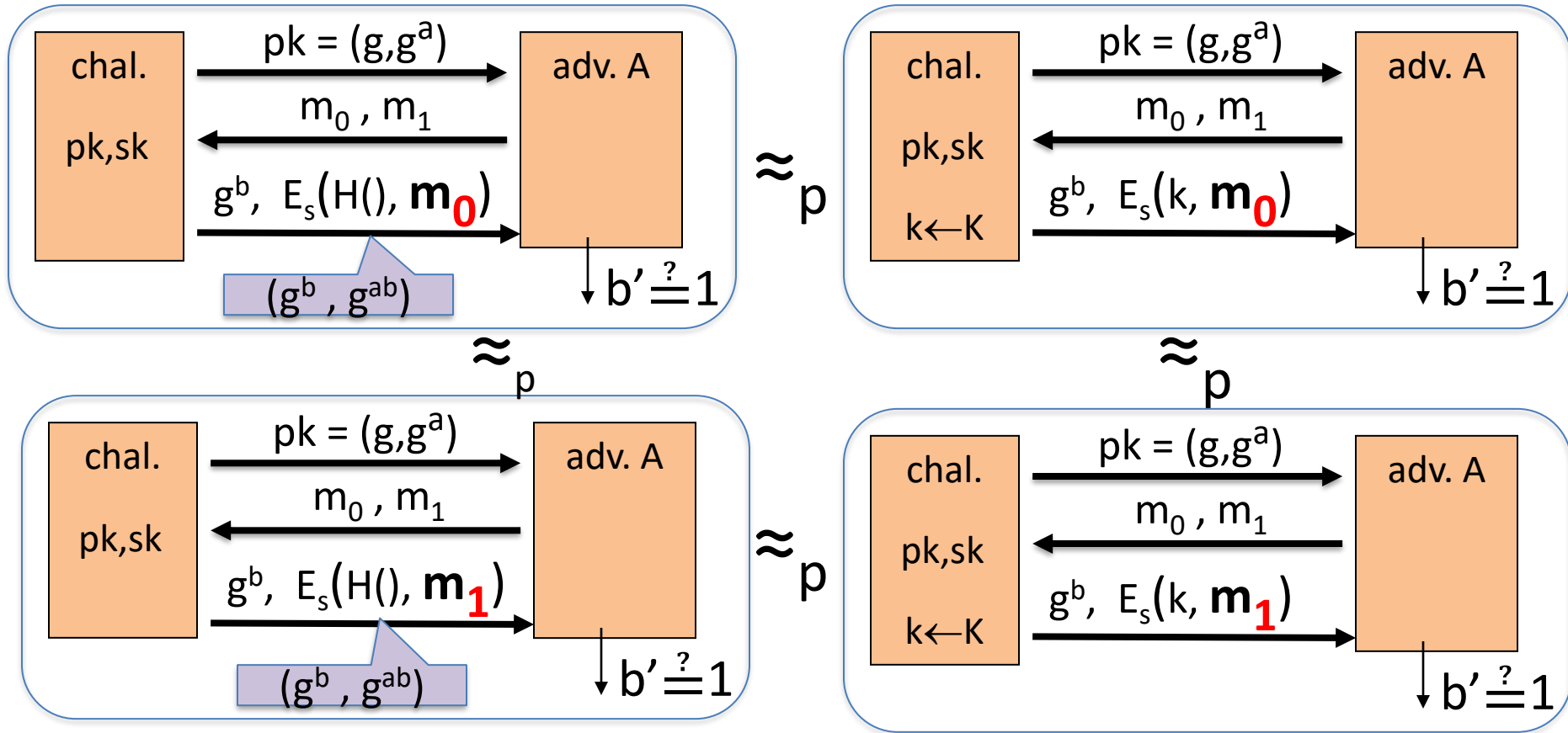
output  $(g^b, c)$

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

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

output  $m$

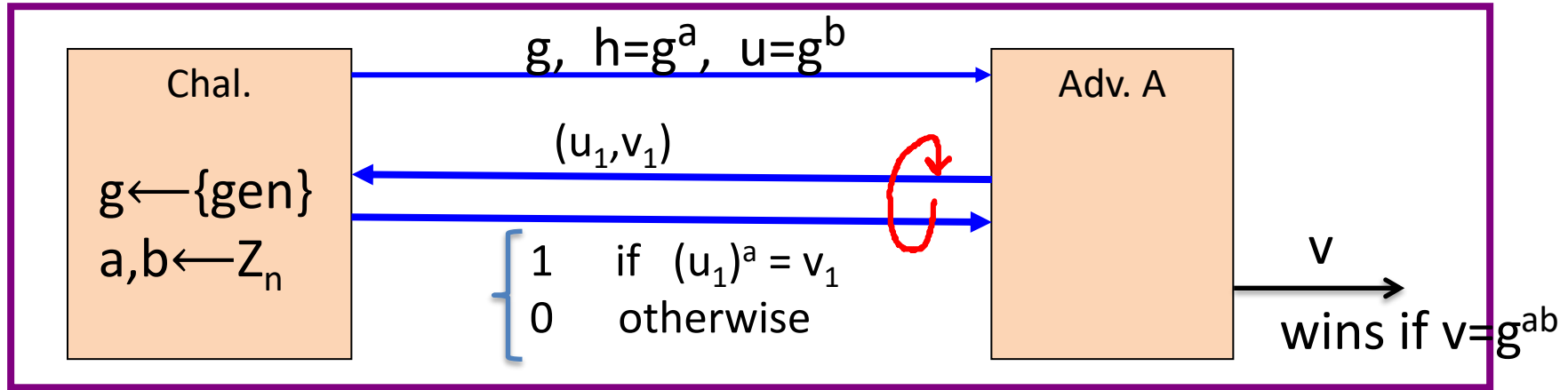
# ElGamal is sem. secure under Hash-DH



# 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:  $\Pr[\text{A outputs } g^{ab}] < \text{negligible}$**

# ElGamal chosen ciphertext security?

## Security Theorem:

If **IDH** holds in the group  $G$ ,  $(E_s, D_s)$  provides auth. enc.  
and  $H: G^2 \rightarrow K$  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?





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

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

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

output  $(g^b, c)$

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

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

output  $m$

# ElGamal chosen ciphertext security

## Security Theorem:

If **IDH** holds in the group  $G$ ,  $(E_s, D_s)$  provides auth. enc.  
and  $H: G^2 \rightarrow K$  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:**  $g \leftarrow \{\text{generators of } G\}$  ,  $a_1, a_2 \leftarrow Z_n$

output  $pk = (g, h_1=g^{a_1}, h_2=g^{a_2})$  ,  $sk = (a_1, a_2)$

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

$k \leftarrow H(g^b, h_1^b, h_2^b)$

$c \leftarrow E_s(k, m)$

output  $(g^b, c)$

**D(  $sk=(a_1,a_2)$ ,  $(u,c)$  ) :**

$k \leftarrow H(u, u^{a_1}, u^{a_2})$

$m \leftarrow D_s(k, c)$

output  $m$

# Chosen ciphertext security

## Security Theorem:

If **CDH** holds in the group  $G$ ,  $(E_s, D_s)$  provides auth. enc.  
and  $H: G^3 \rightarrow K$  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]



# Public key encryption from Diffie-Hellman

---

## A Unifying Theme

# One-way functions (informal)

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

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

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

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

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



# Ex. 1: generic one-way functions

Let  $f: X \rightarrow Y$  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:

$A$  inverts  $f \Rightarrow B(y) = \begin{cases} 0 & \text{if } f(A(y))=y \\ 1 & \text{otherwise} \end{cases}$  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.  $G = (\mathbb{Z}_p)^*$ ) of order  $n$

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

**Define:**  $f: \mathbb{Z}_n \rightarrow G$  as  $f(x) = g^x \in G$

**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:**  $f: \mathbb{Z}_N^* \rightarrow \mathbb{Z}_N^*$  as  $f(x) = x^e \text{ in } \mathbb{Z}_N$

**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