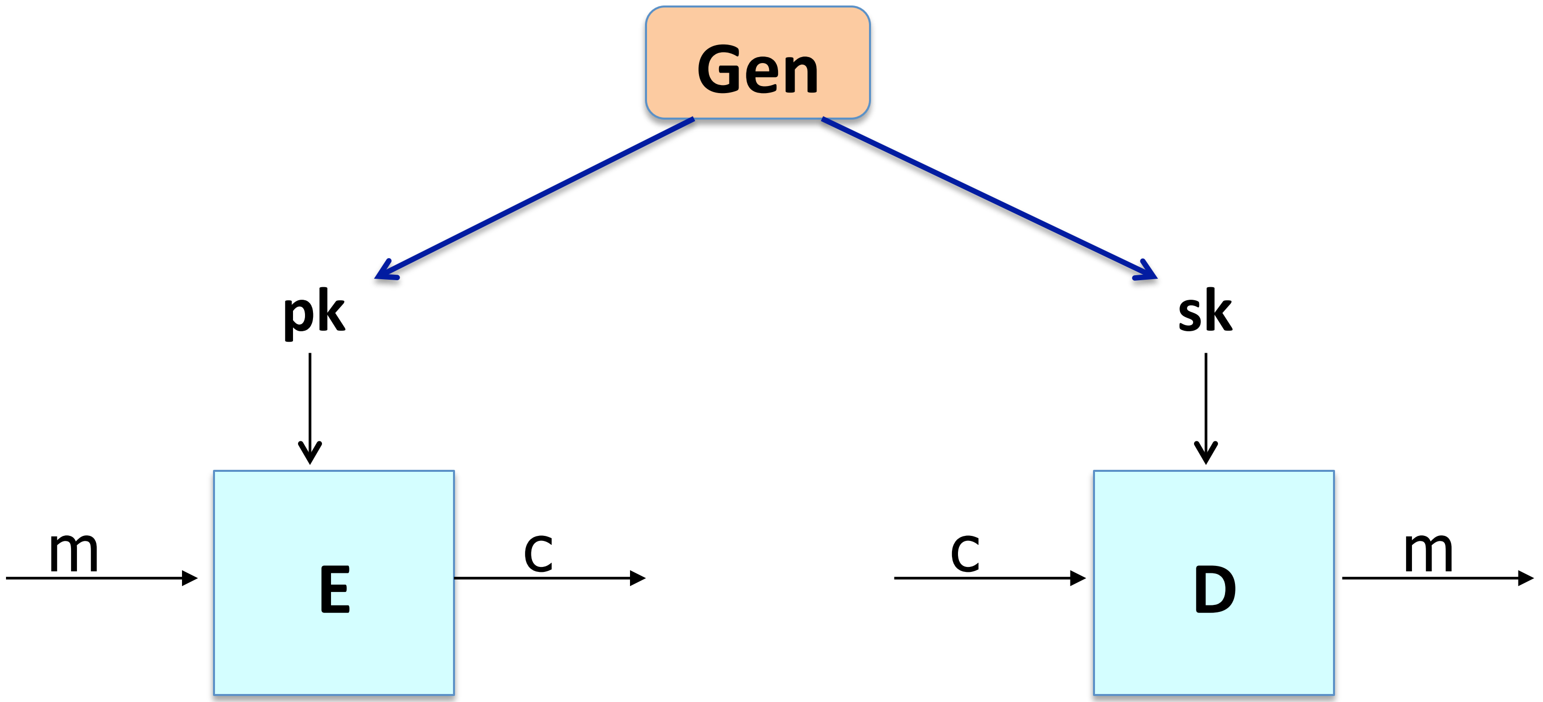
Public key encryption from Diffie-Hellman

The ElGamal Public-key System

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

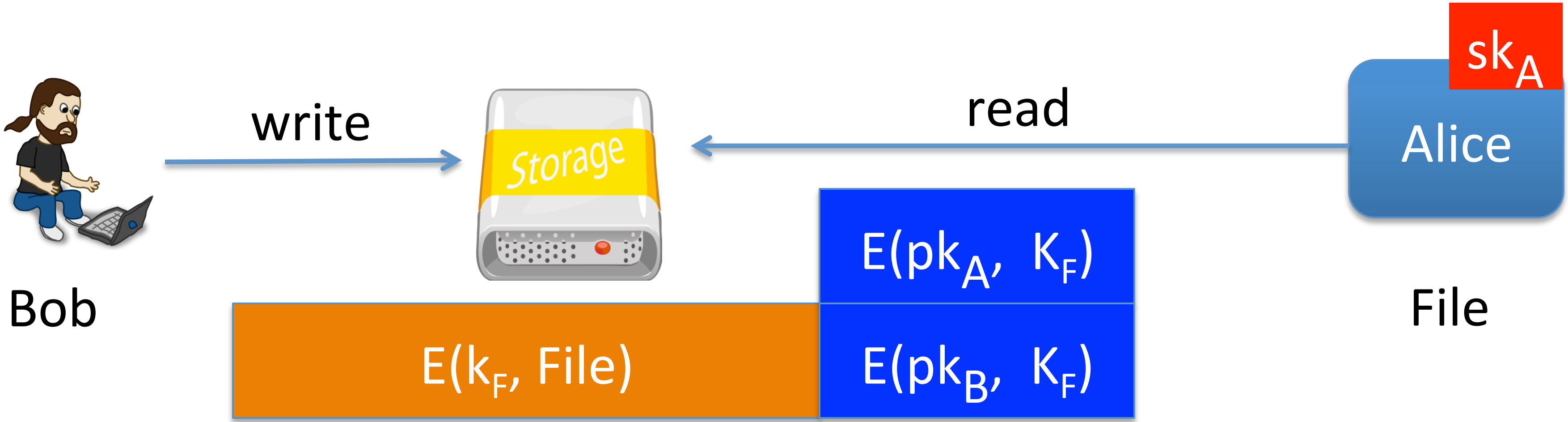


Recap: public-key encryption applications

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

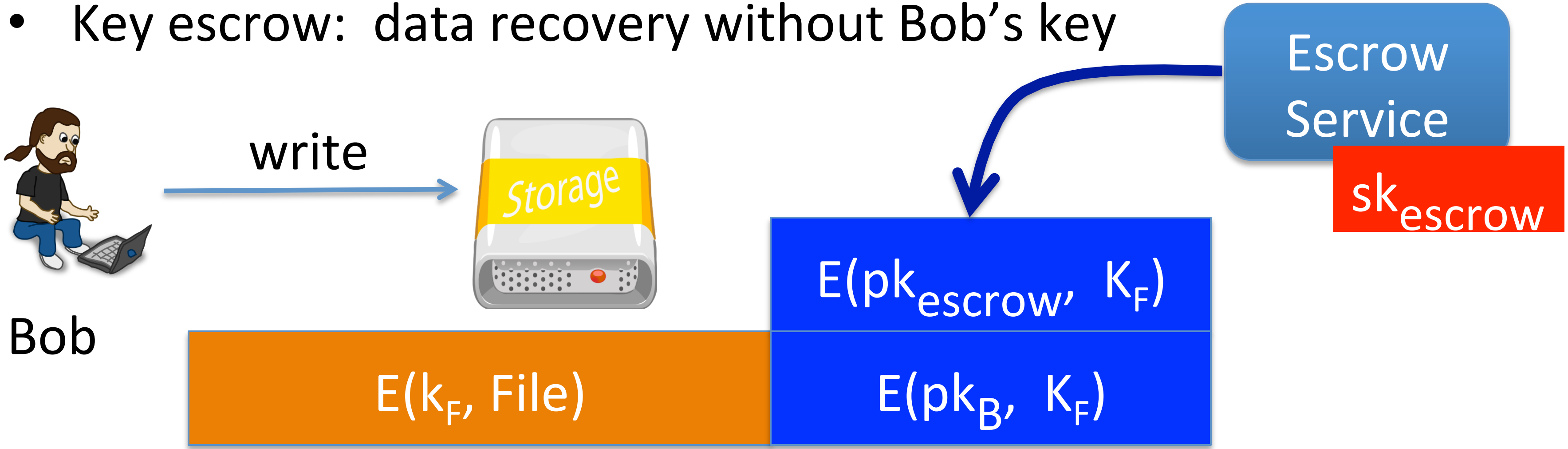


Recap: public-key encryption applications

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 **Bob**

choose random **a** in {1,…,n} choose random **b** in {1,…,n}

$A = g^a$

B =

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

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

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

← 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

choose random **a** in {1,…,n}

$A = g^a$

Treat as a public key

Bob

…ndom **b** in {1,…,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)
- $H: G^2 \longrightarrow K$ 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)

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

$v \longleftarrow u^a$

$k \longleftarrow$, $m \longleftarrow D_s(k, c)$

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 $[\, g^{(2^i)} ,\ h^{(2^i)} \quad \text{for} \quad i=1,\ldots,\log_2 n \,]$
– 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