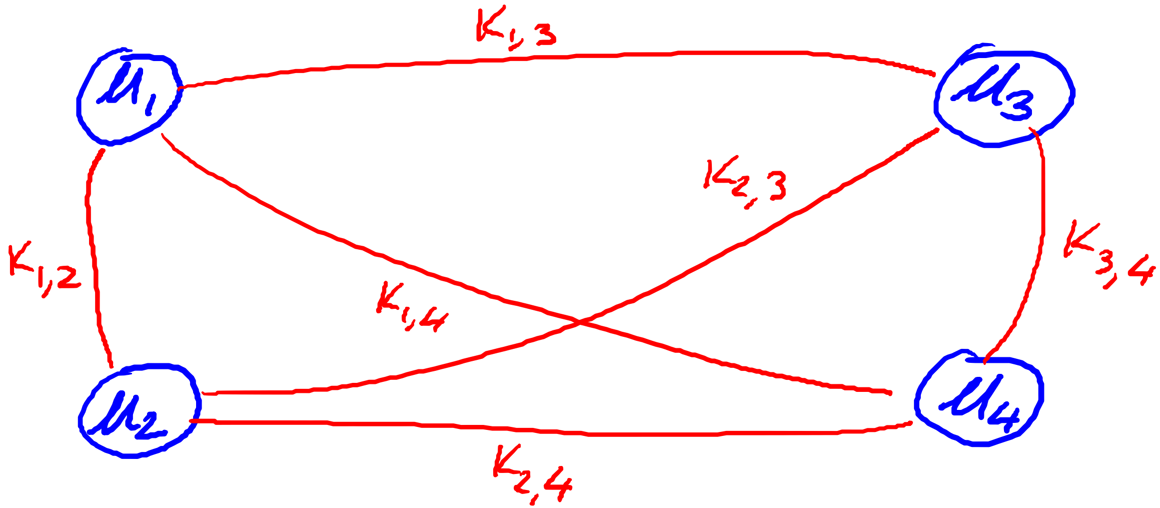
Basic key exchange

Trusted 3rd parties

Key management

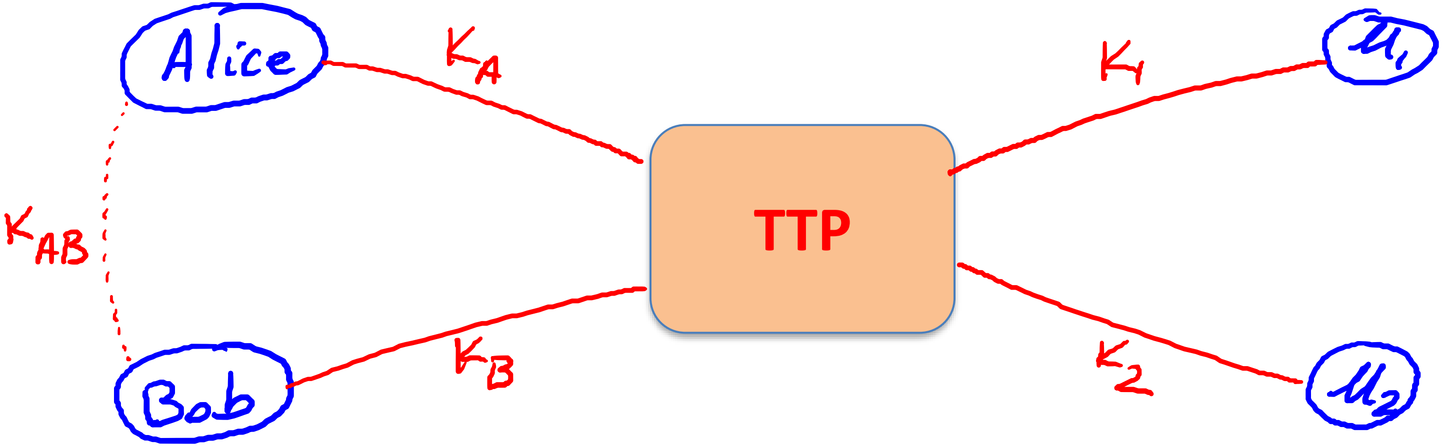
Problem: n users. Storing mutual secret keys is difficult



Total: O(n) keys per user

A better solution

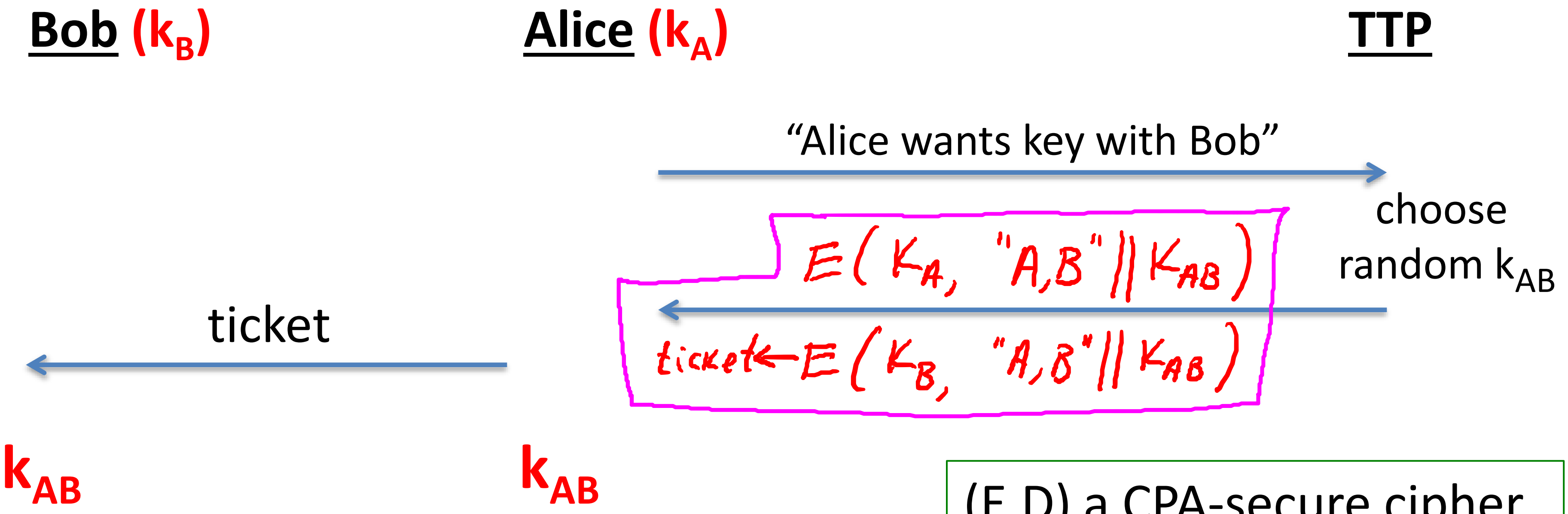
Online Trusted 3rd Party (TTP)



Every user only remembers one key.

Generating keys: a toy protocol

Alice wants a shared key with Bob. Eavesdropping security only.



Bob **(k_B)**

Alice **(k_A)**

TTP

"Alice wants key with Bob"

choose random k_{AB}

$E(K_A, \text{"A,B"} \| K_{AB})$

$\text{ticket} \leftarrow E(K_B, \text{"A,B"} \| K_{AB})$

ticket

k_{AB}

k_{AB}

(E,D) a CPA-secure cipher

Generating keys: a toy protocol

Alice wants a shared key with Bob. Eavesdropping security only.

Eavesdropper sees: $E(k_A,$ “A, B” ll k_{AB}) ; $E(k_B,$ “A, B” ll k_{AB})

(E,D) is CPA-secure ⇒

eavesdropper learns nothing about k_{AB}

Note: TTP needed for every key exchange, knows all session keys.

Toy protocol: insecure against active attacks

Example: insecure against replay attacks

Attacker records session between Alice and merchant Bob

- For example a book order

Attacker replays session to Bob

- Bob thinks Alice is ordering another copy of book

Key question

Can we generate shared keys without an **online** trusted 3rd party?

Answer: yes!

Starting point of public-key cryptography:

- Merkle (1974), Diffie-Hellman (1976), RSA (1977)
- More recently: ID-based enc. (BF 2001), Functional enc. (BSW 2011)

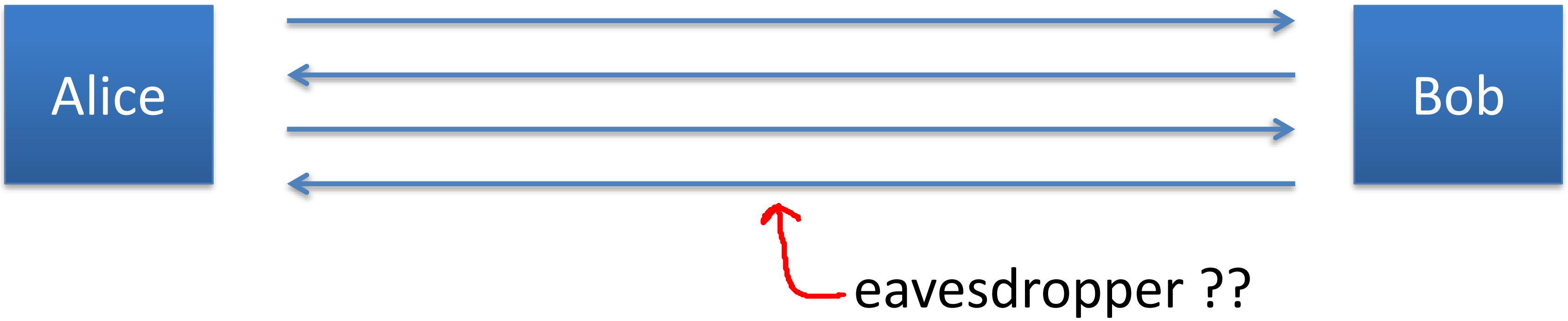
Basic key exchange

The Diffie-Hellman protocol

Key exchange without an online TTP?

Goal: Alice and Bob want shared secret, unknown to eavesdropper

- For now: security against eavesdropping only (no tampering)



Can this be done with an exponential gap?

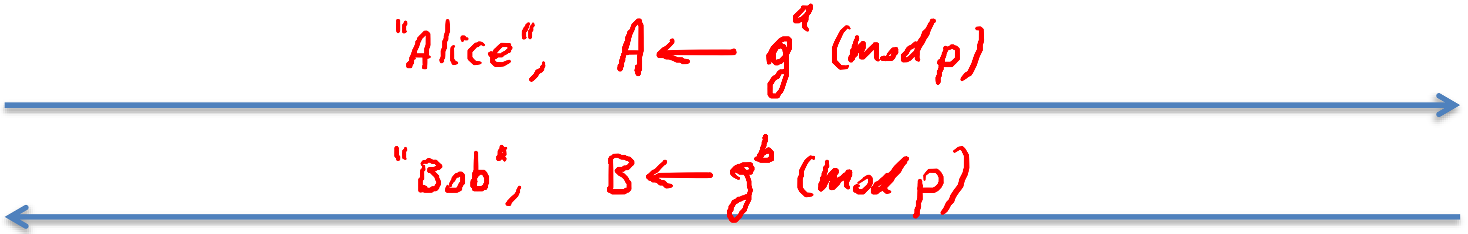
The Diffie-Hellman protocol (informally)

Fix a large prime p (e.g. 600 digits)

Fix an integer g in {1, …, p}

Alice **Bob**

choose random **a** in {1,…,p-1} choose random **b** in {1,…,p-1}



"Alice", $A \leftarrow g^a \pmod p$

"Bob", $B \leftarrow g^b \pmod p$

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

Security (much more on this later)

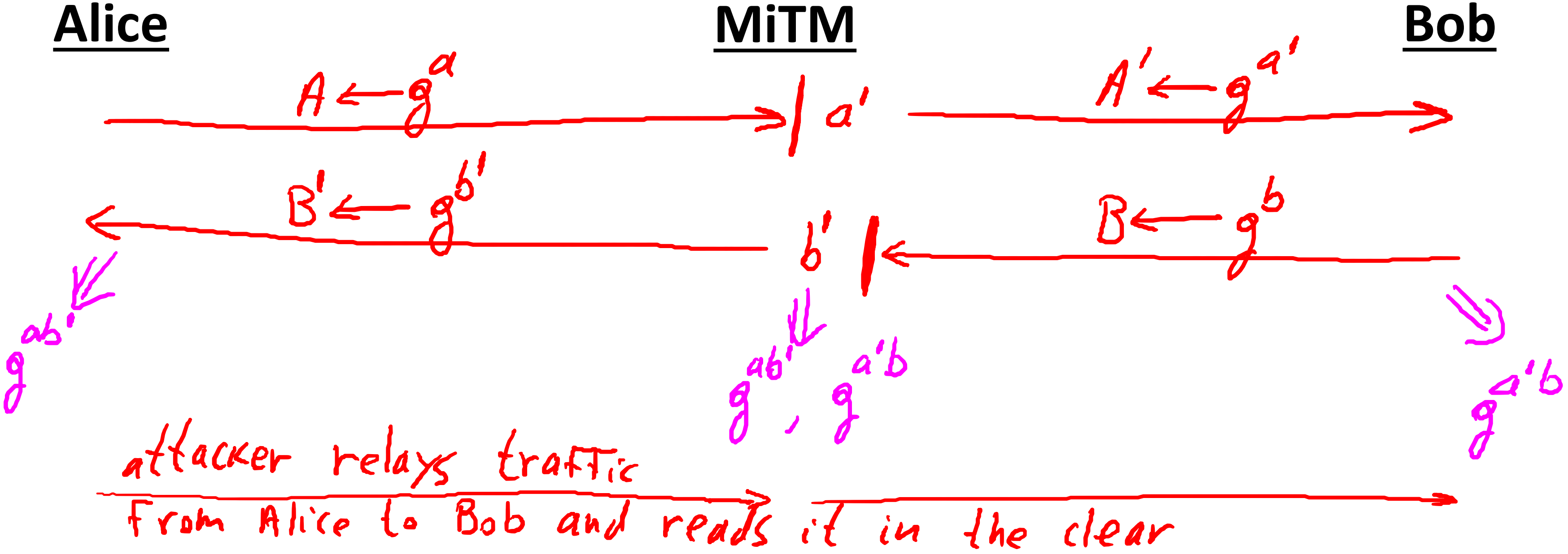
Eavesdropper sees: p, g, $A=g^a \pmod p$, and $B=g^b \pmod p$

Can she compute $g^{ab} \pmod p$??

More generally: define $DH_g(g^a, g^b) = g^{ab} \pmod p$

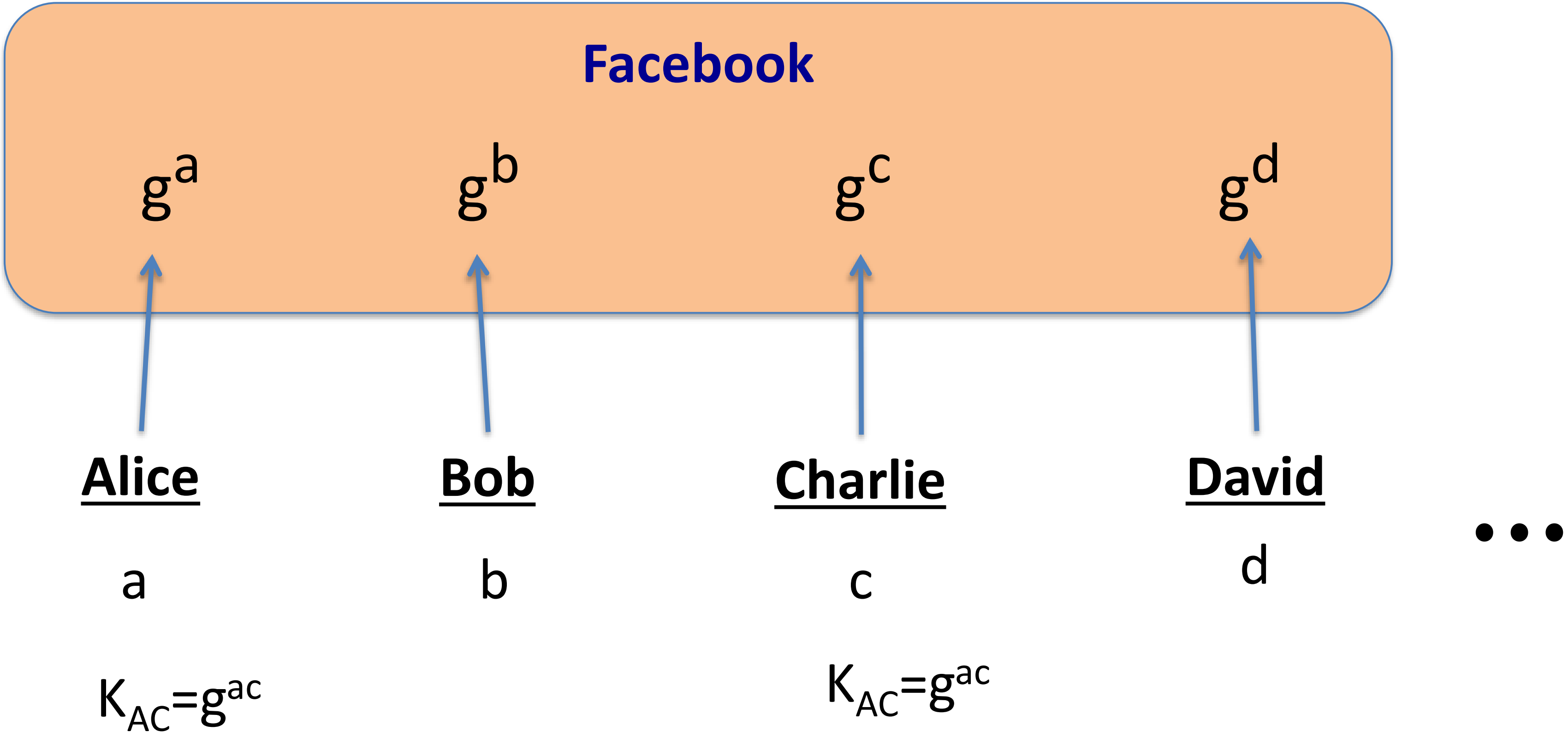
Insecure against man-in-the-middle

As described, the protocol is insecure against **active** attacks



attacker relays traffic from Alice to Bob and reads it in the clear

Another look at DH



Facebook

g^a g^b g^c g^d

Alice **Bob** **Charlie** **David** …

a b c d

$K_{AC}=g^{ac}$ $K_{AC}=g^{ac}$

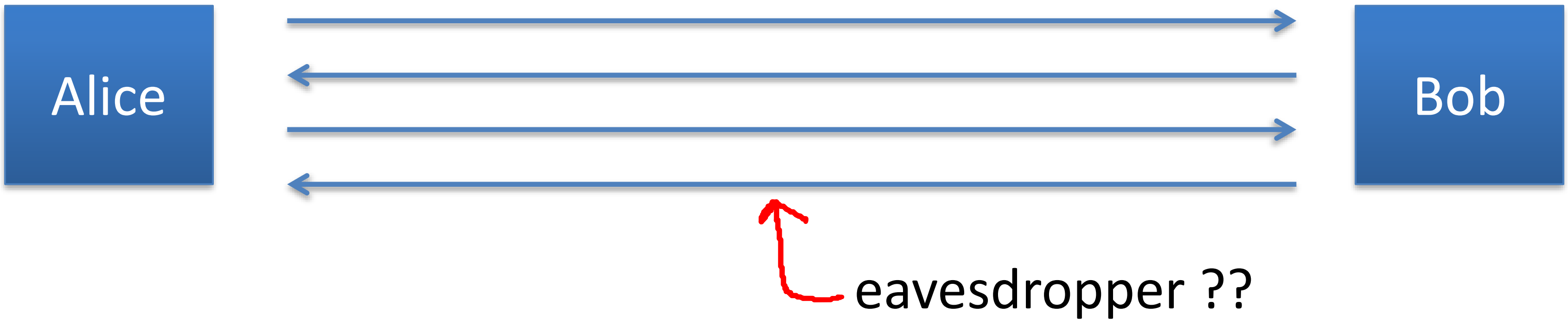
Basic key exchange

Public-key encryption

Establishing a shared secret

Goal: Alice and Bob want shared secret, unknown to eavesdropper

- For now: security against eavesdropping only (no tampering)



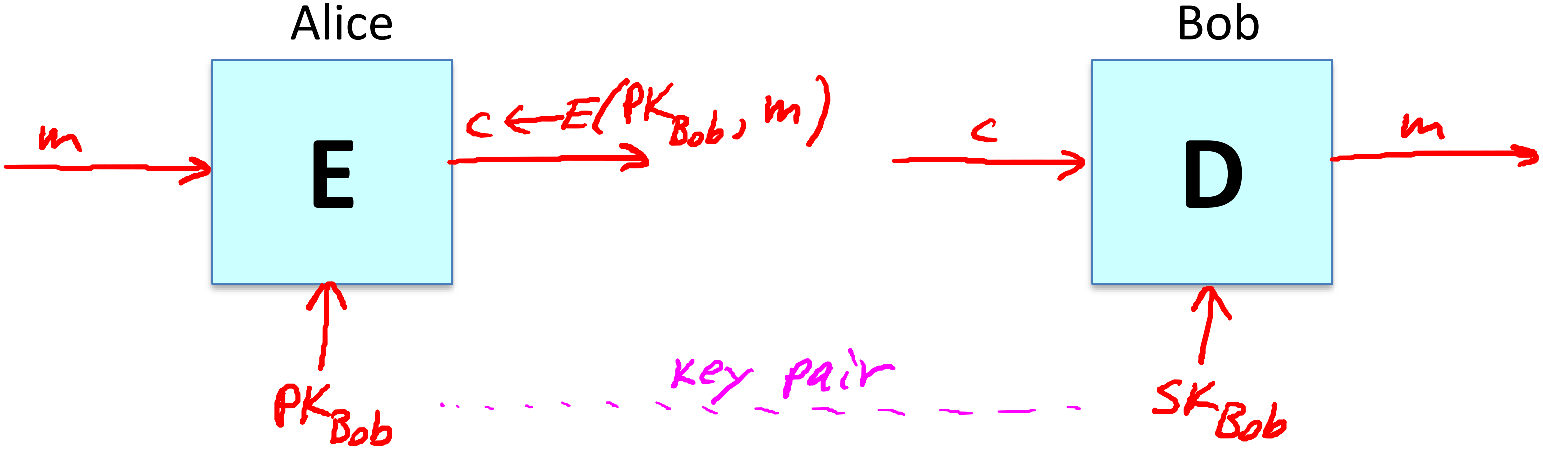
Alice

Bob

eavesdropper ??

This segment: a different approach

Public key encryption



PK: public key, SK: secret key

Public key encryption

Def: a public-key encryption system is a triple of algs. (G, E, D)

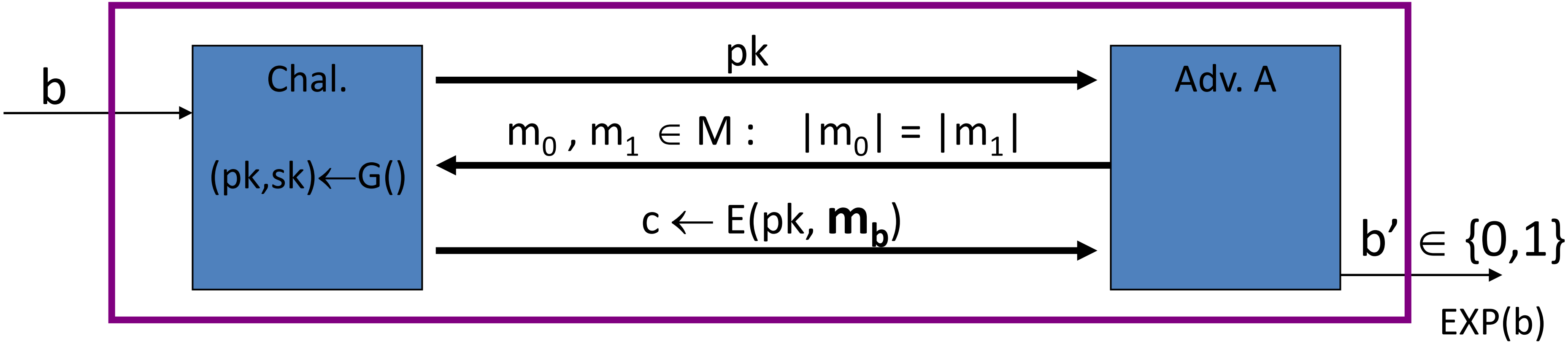
- G(): randomized alg. outputs a key pair (pk, sk)
- E(pk, m): randomized alg. that takes $m \in M$ and outputs $c \in C$
- D(sk,c): det. alg. that takes $c \in C$ and outputs $m \in M$ or $\perp$

Consistency: $\forall$(pk, sk) output by G :

$$\forall m \in M: \quad D(sk, E(pk, m)) = m$$

Semantic Security

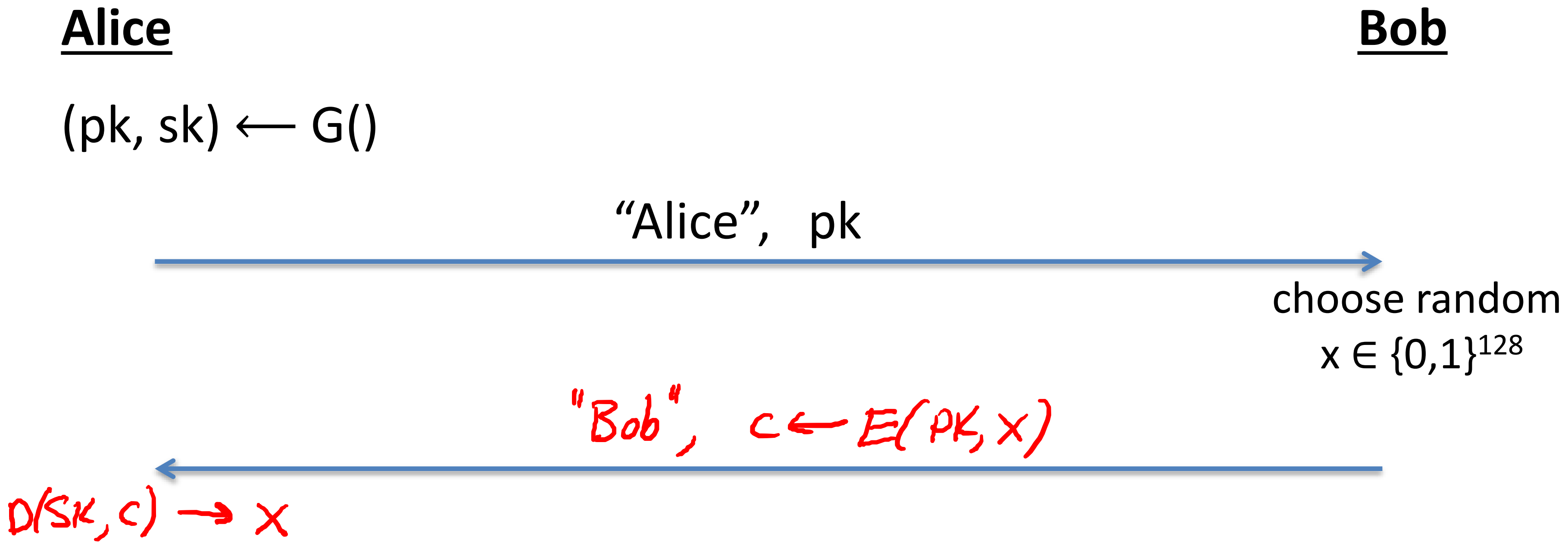
For b=0,1 define experiments EXP(0) and EXP(1) as:



Def: $\mathbb{E}$ =(G,E,D) is sem. secure (a.k.a IND-CPA) if for all efficient A:

$$\mathrm{Adv}_{SS}[A,\mathbb{E}] = \left| \Pr[\mathrm{EXP}(0)=1] - \Pr[\mathrm{EXP}(1)=1] \right| < \text{negligible}$$

Establishing a shared secret



Alice

Bob

$(pk, sk) \longleftarrow G()$

"Alice", pk

choose random

$x \in \{0,1\}^{128}$

"Bob", $c \leftarrow E(pk, x)$

$D(sk, c) \rightarrow x$

x: shared secret

Security (eavesdropping)

Adversary sees **pk,** **E(pk, x)** and wants **x** $\in M$

Semantic security ⇒

adversary cannot distinguish

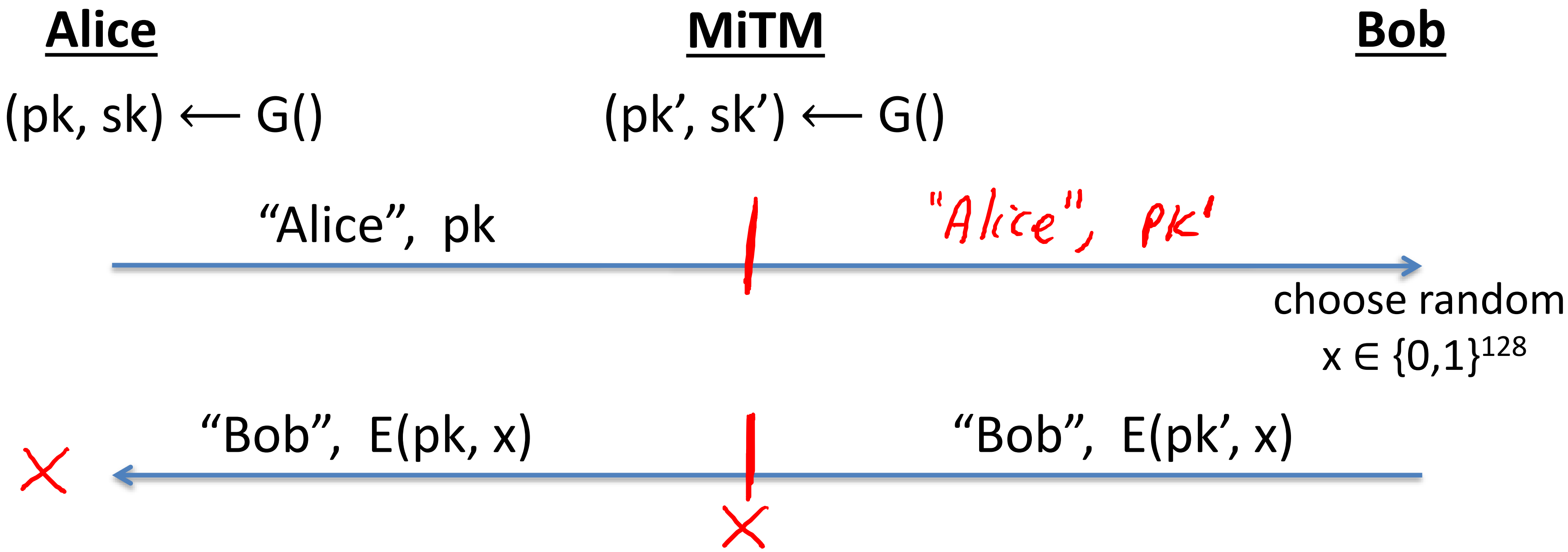
$\{$ pk, E(pk, x), x $\}$ from $\{$ pk, E(pk, x), rand$\in$M $\}$

⇒ can derive session key from x.

Note: protocol is vulnerable to man-in-the-middle

Insecure against man in the middle

As described, the protocol is insecure against **active** attacks



Alice **MiTM** **Bob**

$(pk, sk) \longleftarrow G()$

$(pk', sk') \longleftarrow G()$

"Alice", pk

"Alice", pk'

choose random

$x \in \{0,1\}^{128}$

"Bob", $E(pk', x)$

"Bob", $E(pk, x)$