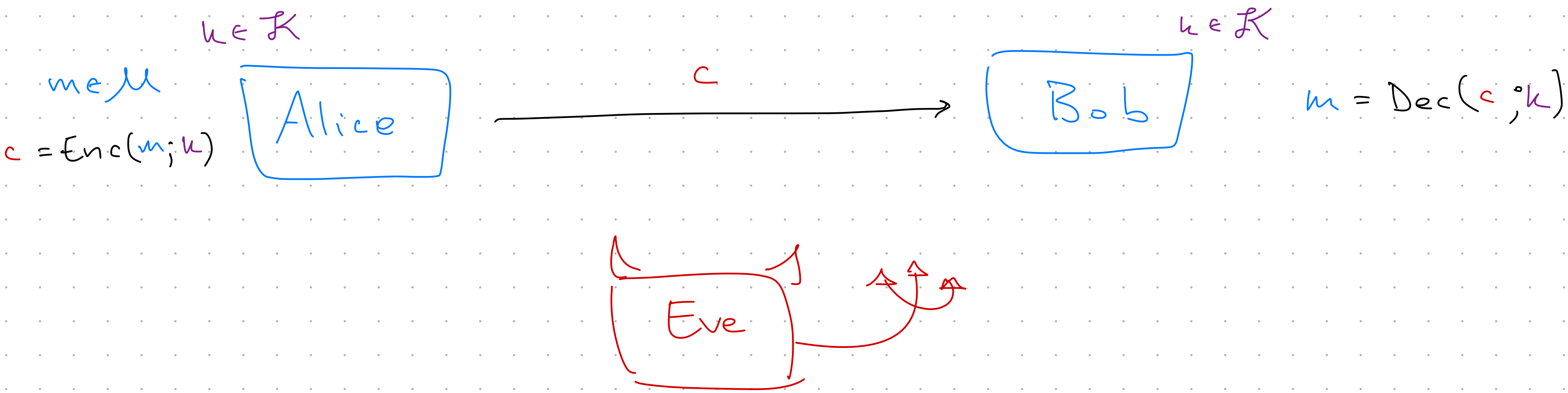
2 May 2025

Diffie-Hellman Key Exchange

Plan

* Key Exchange
* Announcements
* Diffie-Hellman & Public Key Cryptography

Motivation: Secure Communication



Encryption Scheme

$k \leftarrow Gen()$

$c \leftarrow Enc(m; k)$

$m \leftarrow Dec(c; k)$

1. **Functionality**

   $Dec(Enc(m; k); k) = m$

2. **Secrecy**

   Eve shouldn't learn about m given c

One-Time Pad

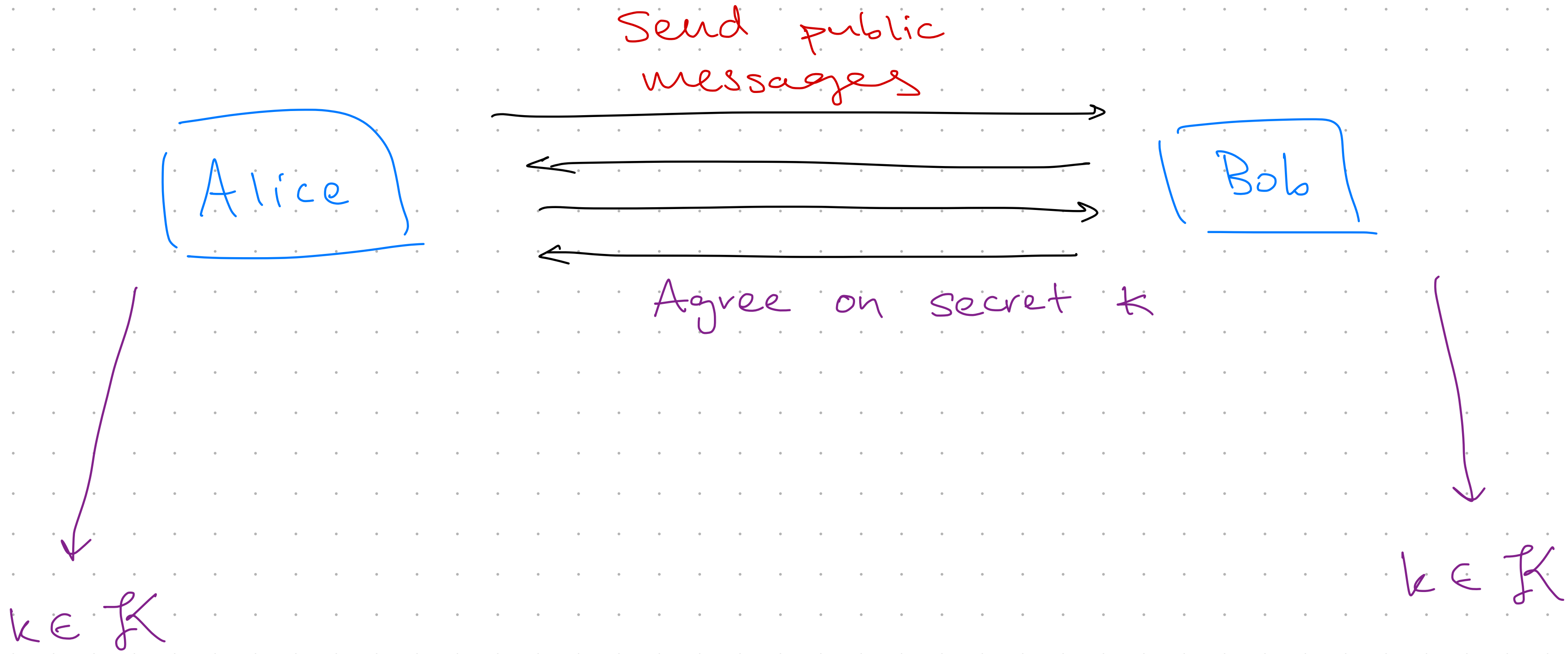
$$c = m \oplus k$$

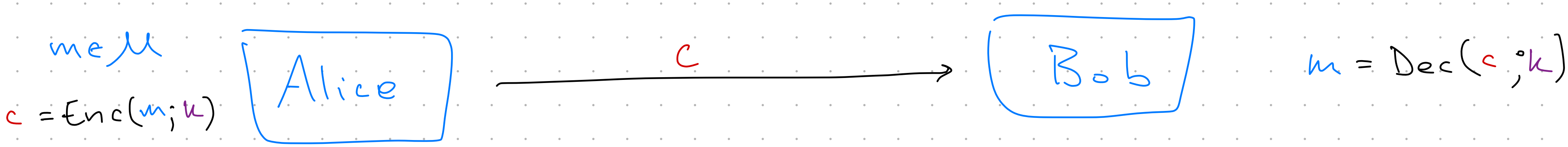
$\Rightarrow$ leaks No information about m

Problem: Infeasible!

↳ Requires a new key for every message.

Idea Key Exchange





Idea Key Exchange

Send public messages

Alice ⇄ Bob

Agree on secret k

$k \in \mathcal{K}$

$k \in \mathcal{K}$

Eve

Eve witnesses every transmission

↳ Learns Nothing

$m \in \mathcal{M}$

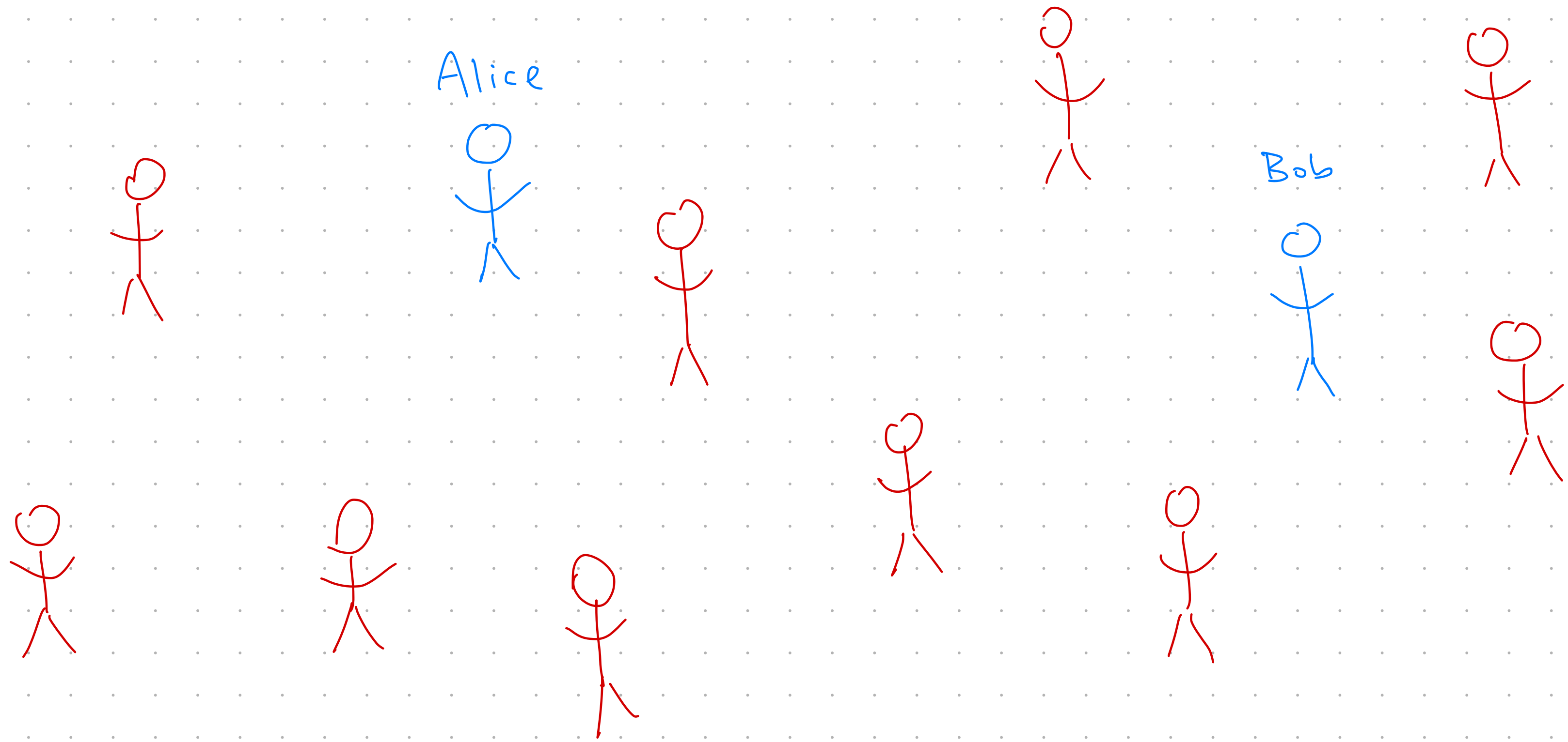
$c = Enc(m; k)$

Alice $\xrightarrow{c}$ Bob

$m = Dec(c; k)$

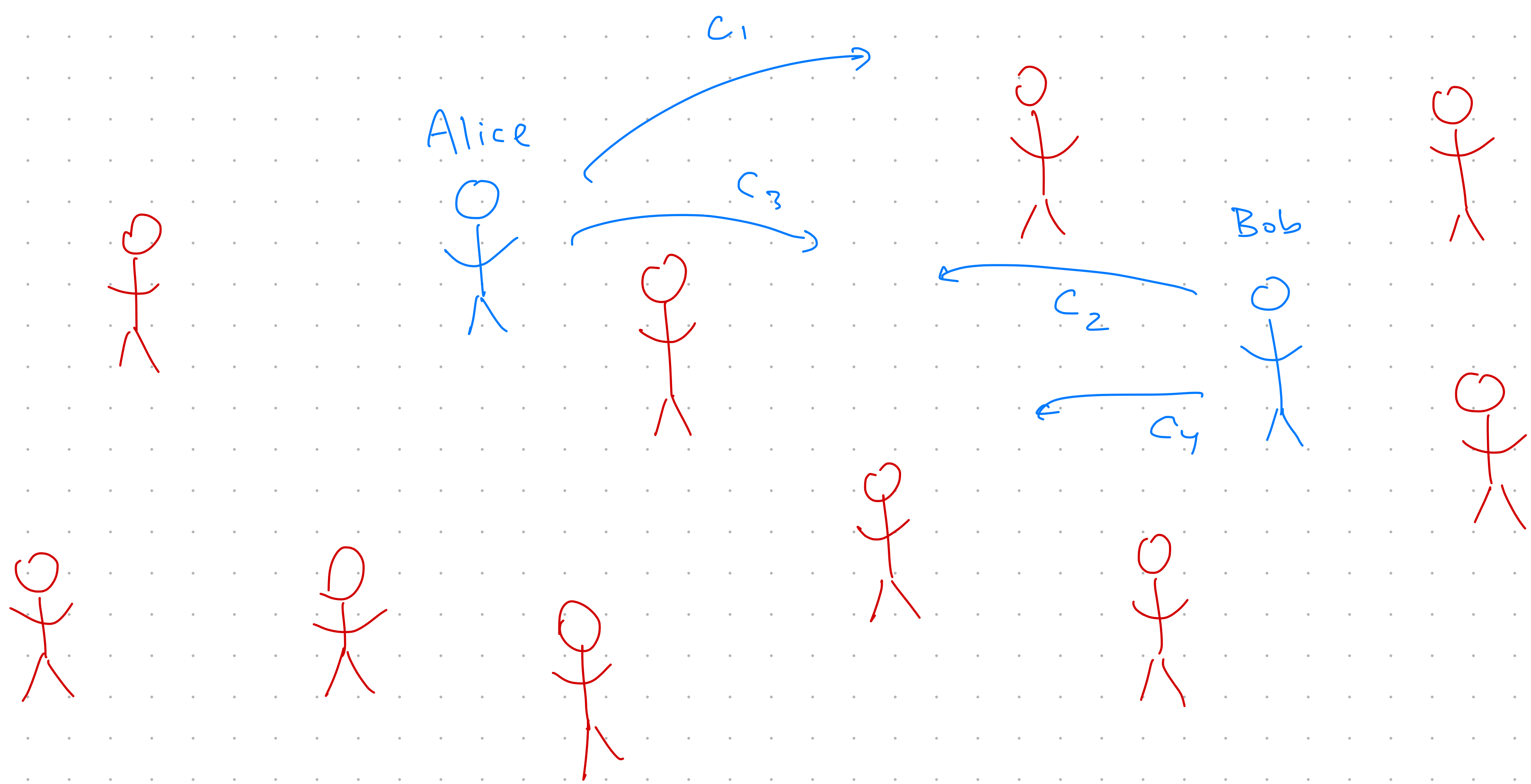
Dinner Party Model

* Alice & Bob meet at a party (No shared setup!)



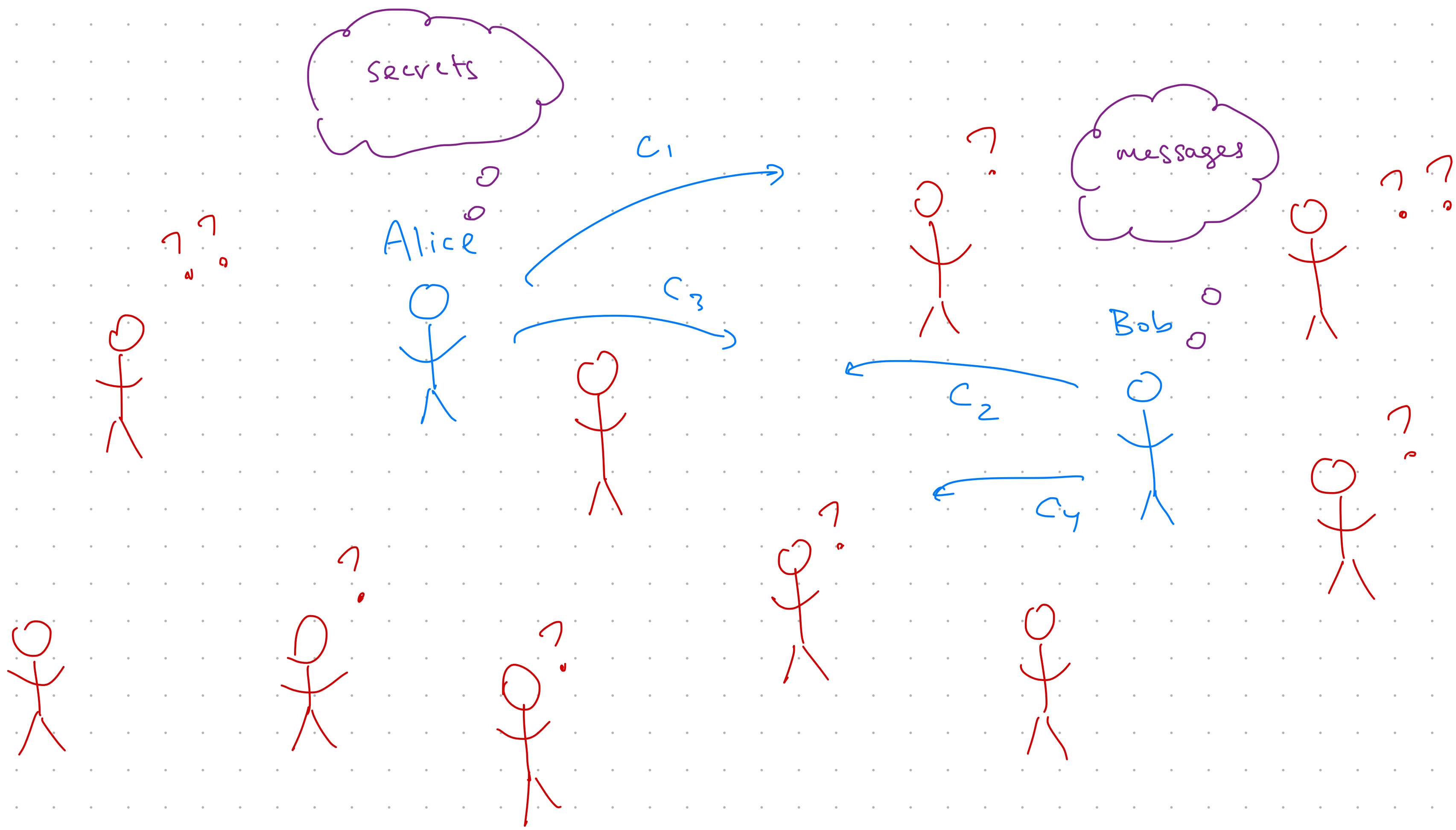
Dinner Party Model

* Alice & Bob meet at a party (No shared setup!)
* Short public messages (Everyone can listen)



Dinner Party Model

* Alice & Bob meet at a party (No shared setup!)
* Shout public messages (Everyone can listen)
* Alice & Bob communicate secretly (only A&B learn messages)



Key Exchange Protocol

* Alice & Bob send public messages
* After interacting, each outputs some $k \in \mathcal{K}$

① Functionality Alice & Bob agree on key

② Secrecy. Eve learns nothing about k

Key Exchange Protocol

* Alice & Bob send public messages
* After interacting, each outputs some $k \in \mathcal{K}$

① Functionality Alice & Bob agree on key

② Secrecy. Eve learns nothing about k

$\forall$ polynomial-time algs $\mathcal{A}$,

$\mathcal{A}$ cannot distinguish k from random

even given transcript of messages

Key Exchange Protocol

* Alice & Bob send public messages
* After interacting, each outputs some $k \in \mathcal{K}$

① Functionality Alice & Bob agree on key

② Secrecy. Eve learns nothing about k

$\forall$ polynomial-time algs $\mathcal{A}$,

$\mathcal{A}$ cannot distinguish k from random

even given transcript of messages

Is Key Exchange Possible??

Announcements

* HW 9 optional for 4820
  - ↳ bonus point based on completion
* Final Exam
  - ↳ May 13
  - ↳ Cumulative
  - ↳ See upcoming Ed post for details
* Recitation: Saturday 1-2p (on Crypto)
* Review Session:

  Fri, May 9

  6-9p in Phillips 101

Diffie - Hellman Protocol (Intuition)

Alice

Bob

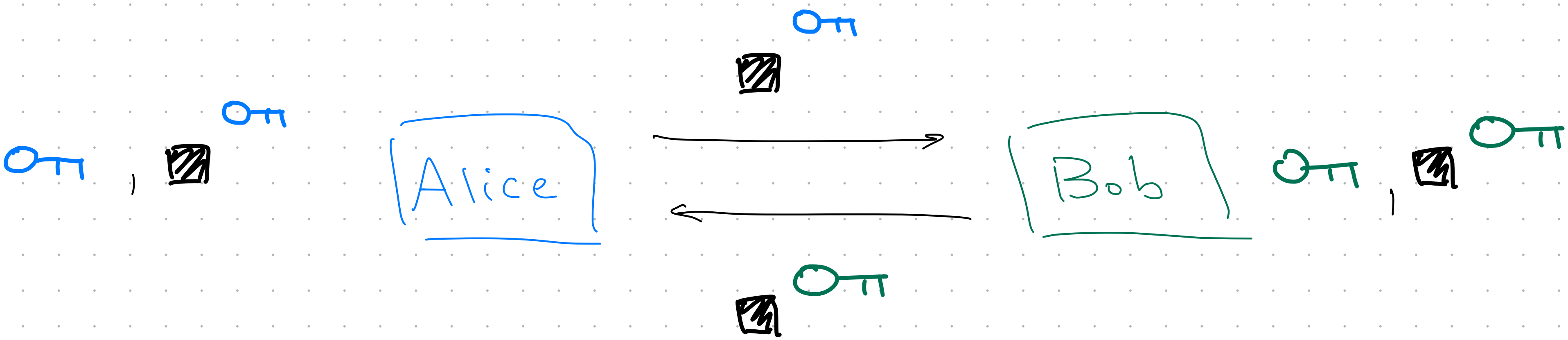
Random key

Random key

Diffie-Hellman Protocol (Intuition)

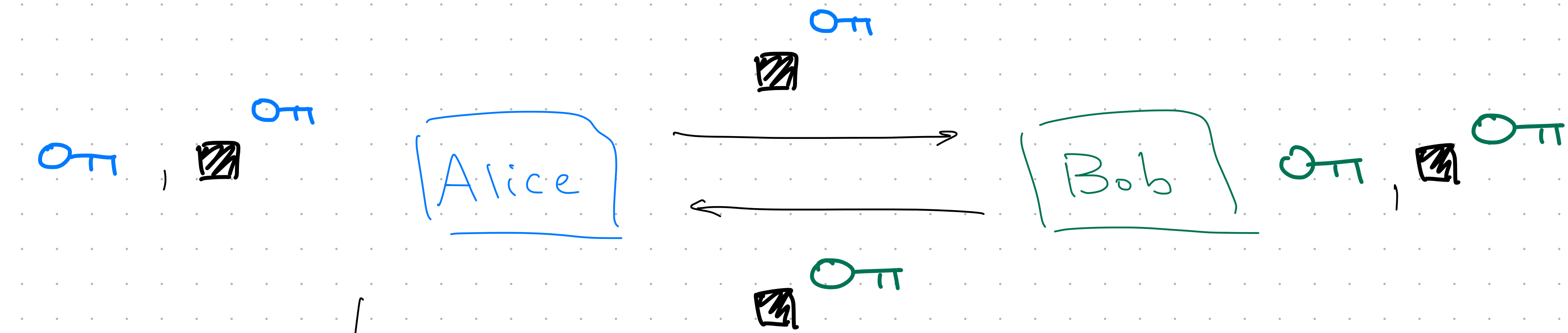
Hide the key w/

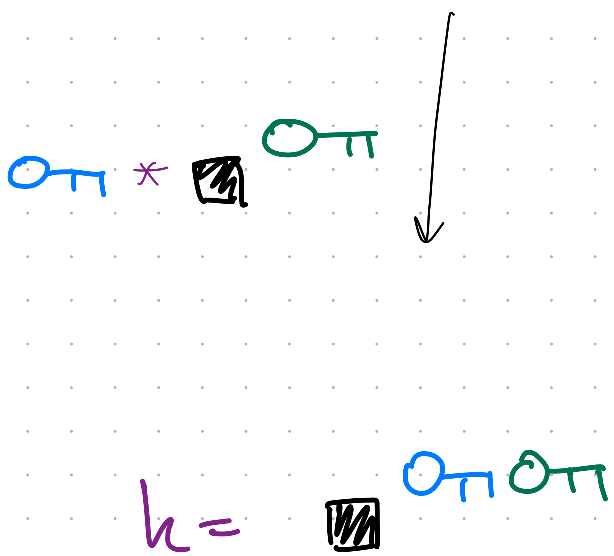
Diffie-Hellman Protocol (Intuition)



Exchange "hidden" keys

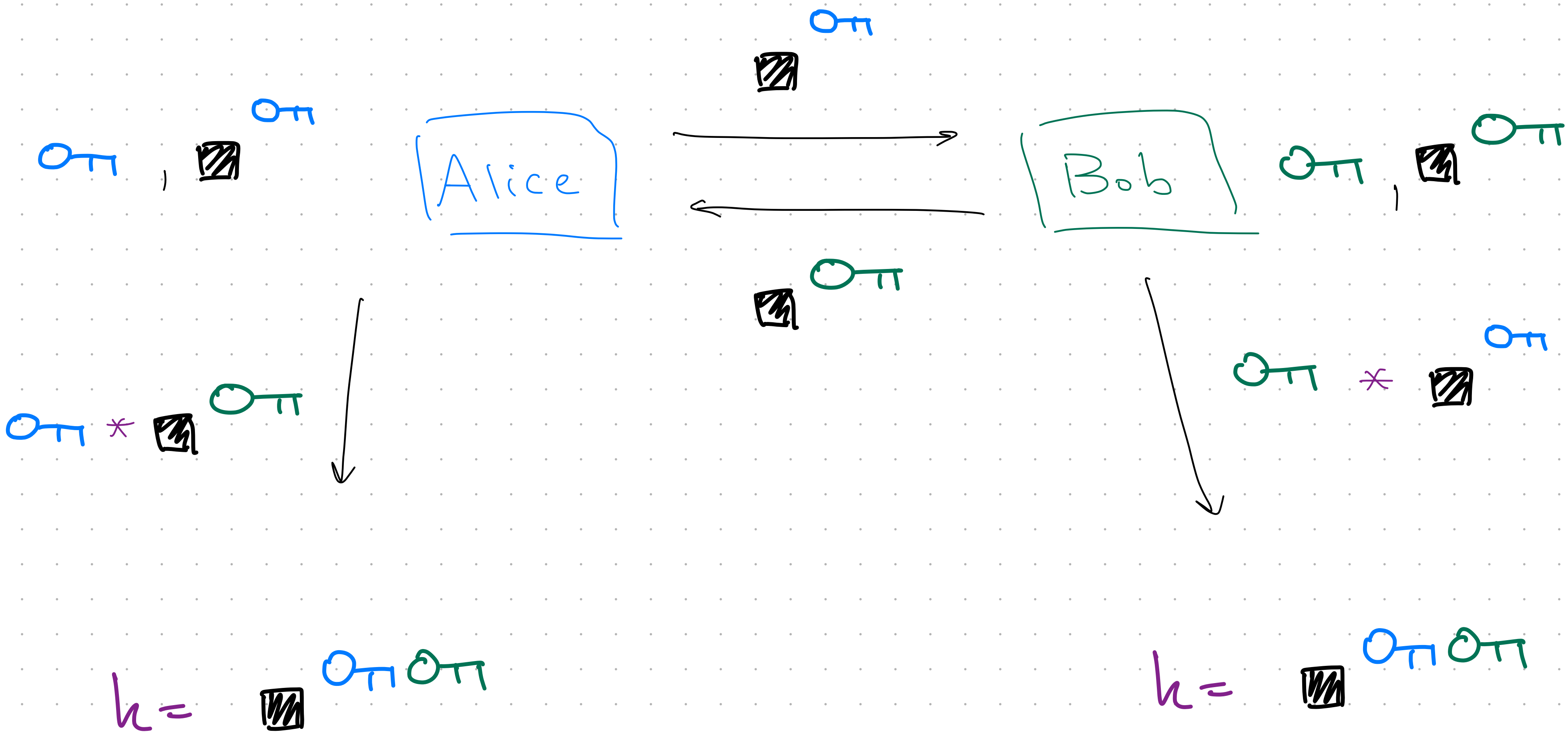
Diffie-Hellman Protocol (Intuition)



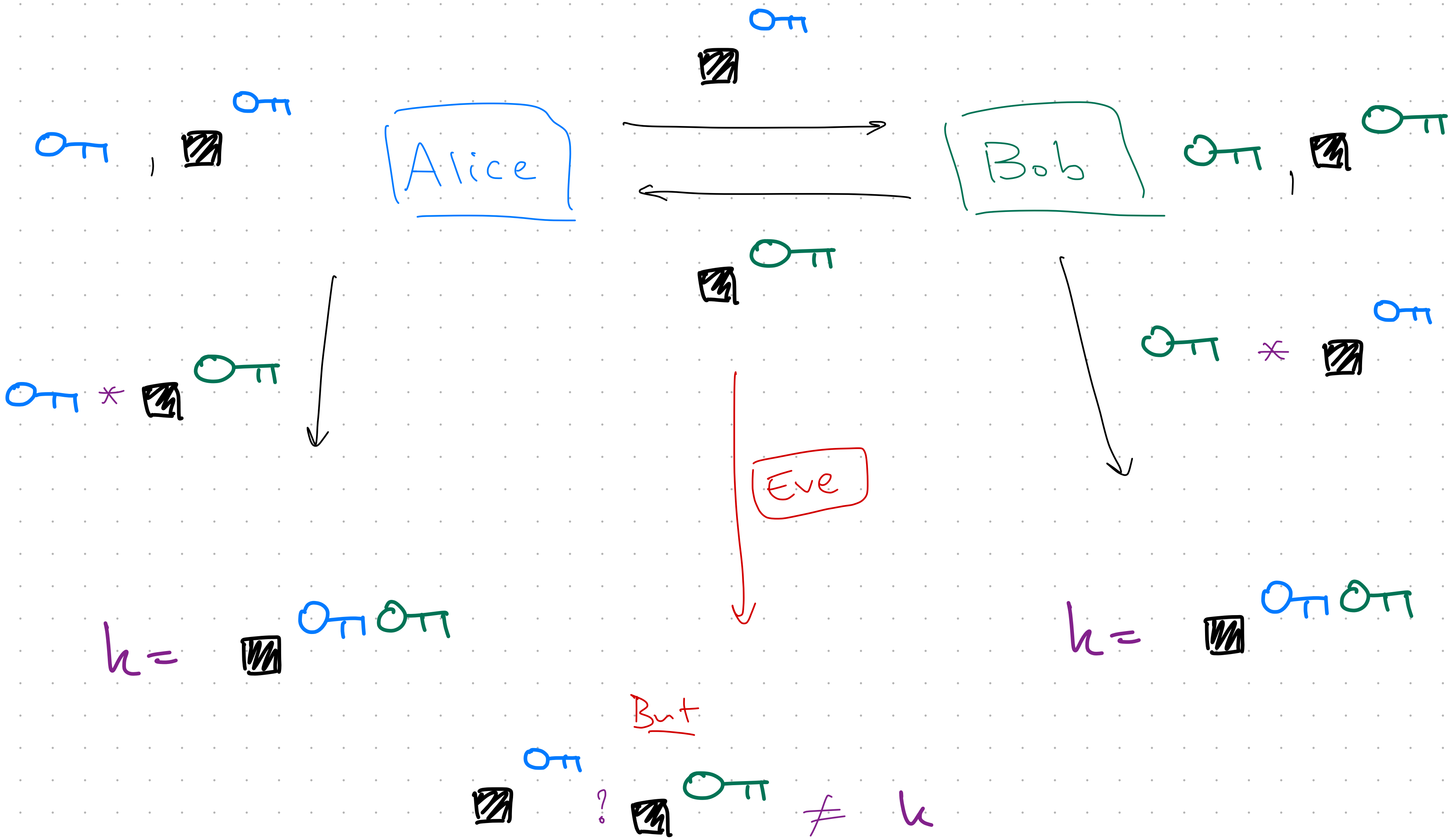


Combine known key
w/ hidden key

Diffie-Hellman Protocol (Intuition)



Diffie-Hellman Protocol (Intuition)



Two hidden keys cannot be combined correctly.

One-way Function: Modular Exponentiation

Compute $f(x) = g^x \pmod{p}$

Group "generator" $g \in \{2, 3, \ldots, p-1\}$

Exponent $x \in \{0, 1, \ldots, p-1\}$

Modulus large prime p.

One-way Function : Modular Exponentiation

$$\boxed{\underline{\text{Compute}} \qquad f(x) = g^x \pmod{p}}$$

Group "generator" $g \in \{2, 3, \ldots, p-1\}$

Exponent $x \in \{0, 1, \ldots, p-1\}$

Modulus large prime p.

Inversion. Discrete Log Problem.

Given $y \in \{0, 1, \ldots, p-1\}$,

find x s.t. $g^x \pmod{p} = y$.

One-way Function: Modular Exponentiation

$$\boxed{\text{Compute} \quad f(x) = g^x \pmod{p}}$$

Group "generator" $g \in \{2, 3, \ldots, p-1\}$

Exponent $x \in \{0, 1, \ldots, p-1\}$

Modulus large prime p.

Inversion. Discrete Log Problem.

Given $y \in \{0, 1, \ldots, p-1\}$,

find x s.t. $g^x \pmod{p} = y$.

Conjecture Discrete Log $\notin P$.

Diffie - Hellman Protocol :

Random secret keys

$a \xleftarrow[r]{} \{0, \dots, p-1\}$

$b \xleftarrow[r]{} \{0, 1, \dots, p-1\}$

Alice

Bob

Diffie-Hellman Protocol:

$a \leftarrow_r \{0, \dots, p-1\}$

$A = g^a \pmod p$ $\boxed{\text{Alice}}$

$\boxed{\text{Bob}}$ $b \leftarrow_r \{0, 1, \dots, p-1\}$

$B = g^b \pmod p$

"hide" keys w/
Modular exponentiation

Diffie-Hellman Protocol:

$a \xleftarrow{r} \{0, ..., p-1\}$

$b \xleftarrow{r} \{0, 1, ..., p-1\}$

$A = g^a \pmod{p}$

$$\boxed{\text{Alice}} \underset{B}{\overset{A}{\rightleftarrows}} \boxed{\text{Bob}}$$

$B = g^b \pmod{p}$

Exchange hidden keys

Diffie - Hellman Protocol :

$a \leftarrow_r \{0, ..., p-1\}$

$A = g^a \pmod p$

$$\begin{tikzcd} \boxed{\text{Alice}} \arrow[r, shift left, "A"] & \boxed{\text{Bob}} \arrow[l, shift left, "B"] \end{tikzcd}$$

$b \leftarrow_r \{0, 1, ..., p-1\}$

$B = g^b \pmod p$

$B^a = (g^b \pmod p)^a$

$k = g^{ab} \pmod p$

Combine known a w/ B via modular exponentiation to obtain shared key

Diffie - Hellman Protocol :

$a \leftarrow_r \{0, \ldots, p-1\}$

$b \leftarrow_r \{0, 1, \ldots, p-1\}$

$A = g^a \pmod p$ $\quad$ $\boxed{\text{Alice}}$ $\xrightarrow{A}$ $\boxed{\text{Bob}}$ $\quad$ $B = g^b \pmod p$

$\boxed{\text{Alice}} \xleftarrow[B]{} \boxed{\text{Bob}}$

$$B^a = \left(g^b \pmod p\right)^a \qquad\qquad A^b = \left(g^a \pmod p\right)^b$$

$$k = g^{ab} \pmod p \qquad\qquad k = g^{ab} \pmod p$$

Diffie - Hellman Protocol :

$a \leftarrow_r \{0, \dots, p-1\}$

$b \leftarrow_r \{0, 1, \dots, p-1\}$

$A = g^a \pmod p$ $\boxed{\text{Alice}}$ $\xrightarrow{A}$ $\boxed{\text{Bob}}$ $B = g^b \pmod p$

$\boxed{\text{Alice}}$ $\xleftarrow[B]{}$ $\boxed{\text{Bob}}$

$B^a = \left(g^b \pmod p\right)^a$

$A^b = \left(g^a \pmod p\right)^b$

$\boxed{\text{Eve}}$

$k = g^{ab} \pmod p$

$k = g^{ab} \pmod p$

But

Computing $g^{ab} \pmod p$ from g^a and g^b seems HARD!

Diffie-Hellman

* Allows for key agreement

$$\boxed{A} \xrightarrow{g^a} \boxed{B} \qquad g^{ab} \qquad (\text{all mod } p)$$

$$\boxed{A} \xleftarrow[g^b]{} \boxed{B}$$

Diffie - Hellman

* Allows for key agreement

$$\boxed{A} \underset{g^b}{\overset{g^a}{\rightleftarrows}} \boxed{B} \qquad g^{ab} \qquad (\text{all mod } p)$$

<u>Subtle Issue</u>: "Proof" of security does <u>NOT</u> reduce Discrete Log to breaking DH.

Diffie-Hellman

* Allows for key agreement

$$\boxed{A} \underset{g^b}{\overset{g^a}{\rightleftarrows}} \boxed{B} \qquad g^{ab} \qquad (\text{all mod } p)$$

Subtle Issue: "Proof" of security does NOT reduce Discrete Log to breaking DH.

New Assumption: Decisional Diffie-Hellman

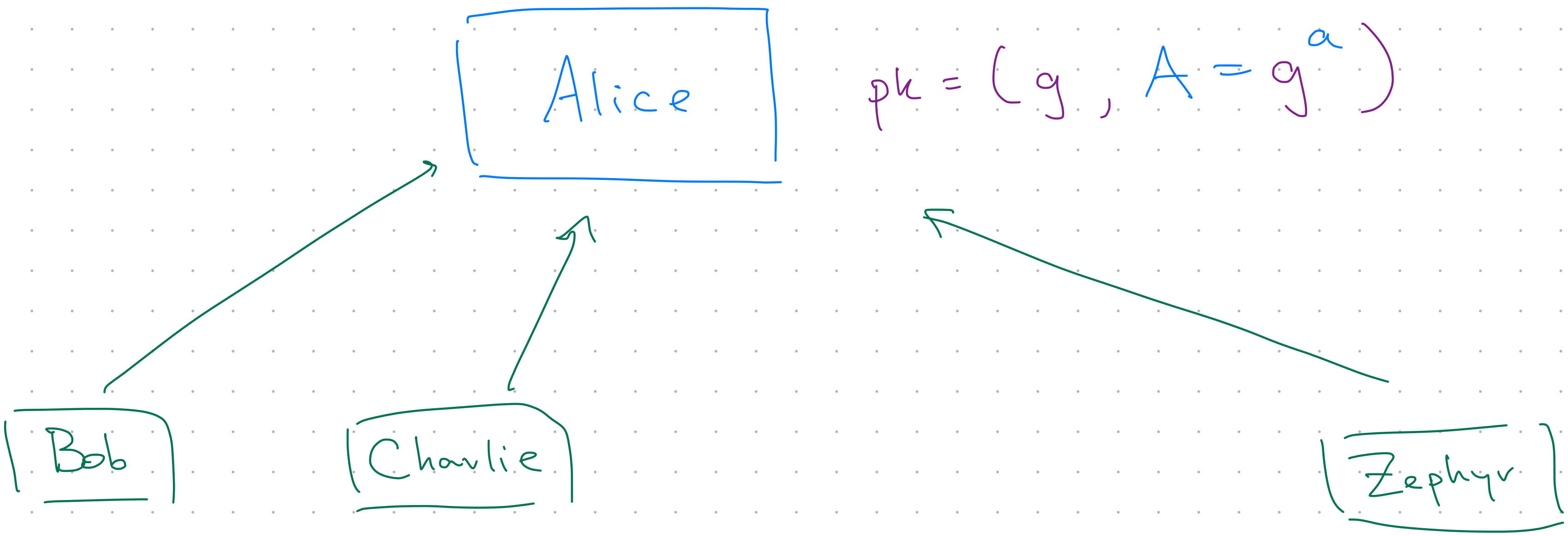
$a, b, c \leftarrow$ random

$$A(g^a, g^b, g^c) \approx A(g^a, g^b, g^{ab})$$

for all poly-time algs. A.

Public-Key Cryptography

* Alice publishes her public key
* Anyone can send secure message to Alice



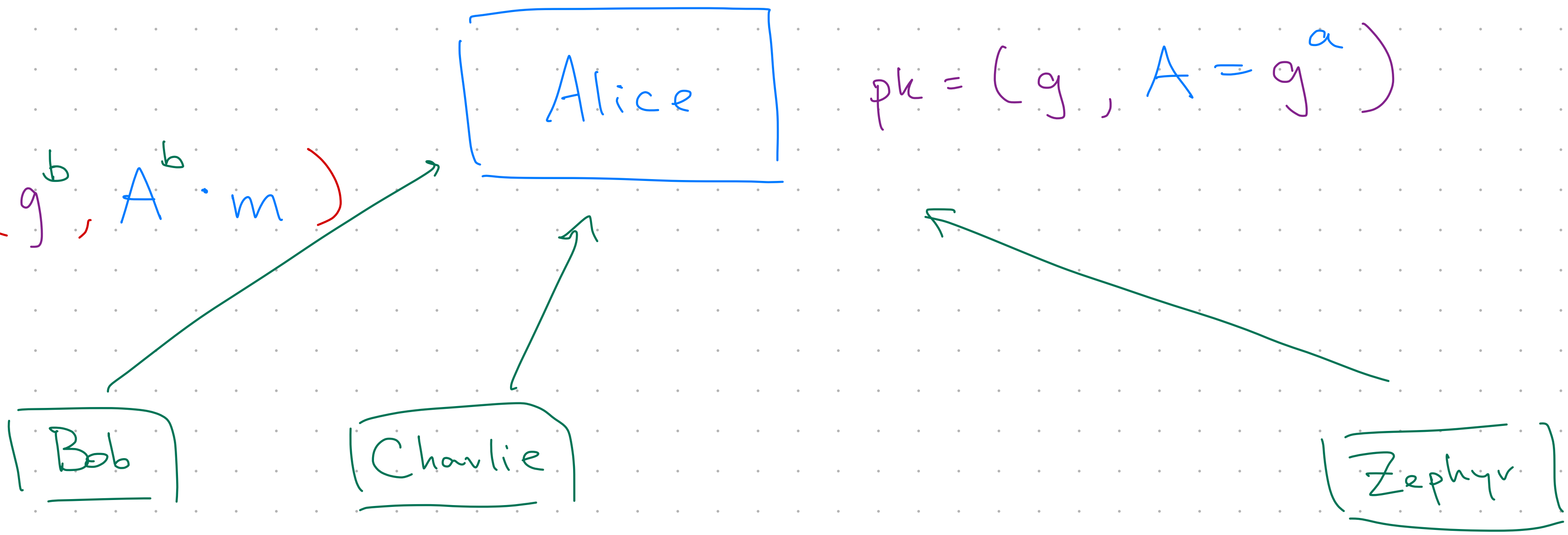
$pk = (g, A = g^a)$

Public-Key Cryptography

* Alice publishes her <u>public key</u>
* Anyone can send secure message to Alice

$$c = (g^b, A^b \cdot m)$$

$$pk = (g, A = g^a)$$

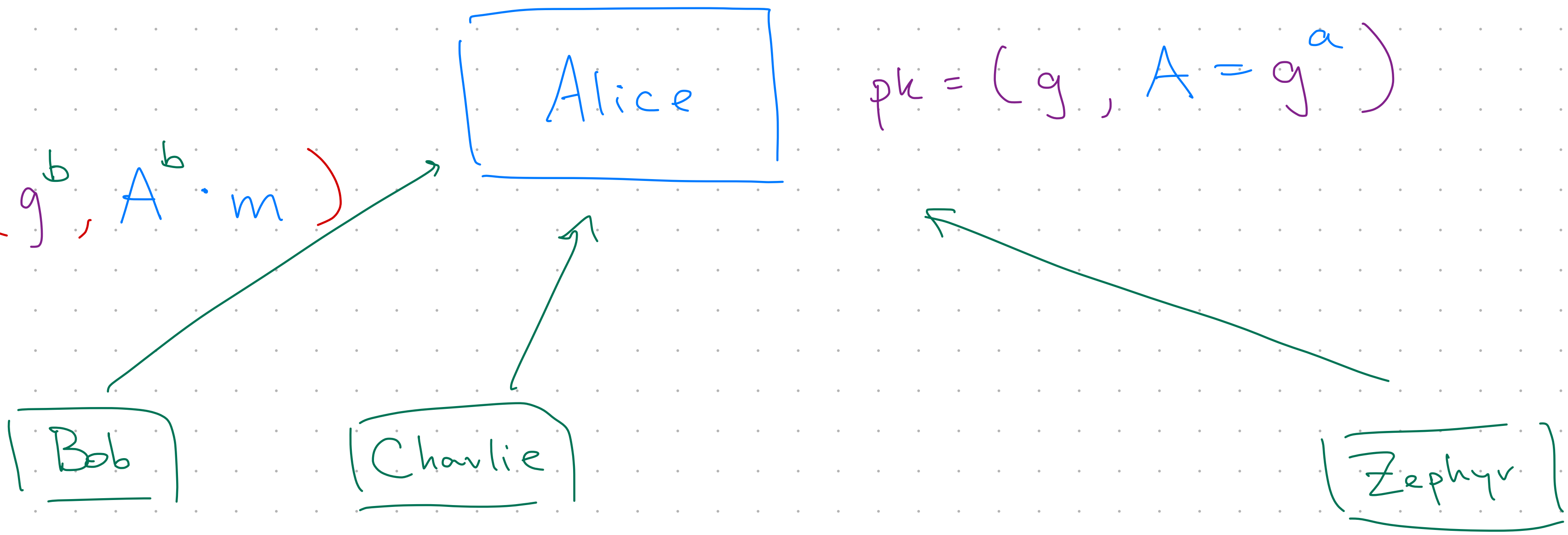


Public-Key Cryptography

* Alice publishes her public key
* Anyone can send secure message to Alice

$pk = (g, A = g^a)$

$c = (g^b, A^b \cdot m)$



$$\mathrm{Dec}(c; pk) = \left(\left(g^b\right)^a\right)^{-1} \cdot \left(A^b \cdot m\right) = m$$

El Gamal Encryption