CSE467: Computer Security

6. Asymmetric-key Encryption

Seongil Wi

Notification: Homework #1

- Programming assignment
- Due: April 4 (Friday), 11:59 PM

- Implementing encryption, decryption, signing program for the RSA cryptosystem

- Late submission will be assessed a penalty of 10% per day

Notification: Quiz #1

- Date: 3/31 (Mon.), Class time
- Scope
  - Everything learned in Cryptography!
- T/F problems
- Computation problems

Notification: Participation Points

- If you asked a question during class, please let me know your name and student ID

Notification: Hack Class101

- Find unknown security issues on Class101 websites!
- Instruction: https://bounty.class101.net/
  - Foreigners should use a translator
- Activity period: 03/03 ~ 06/18
- DO NOT try anything illegal!

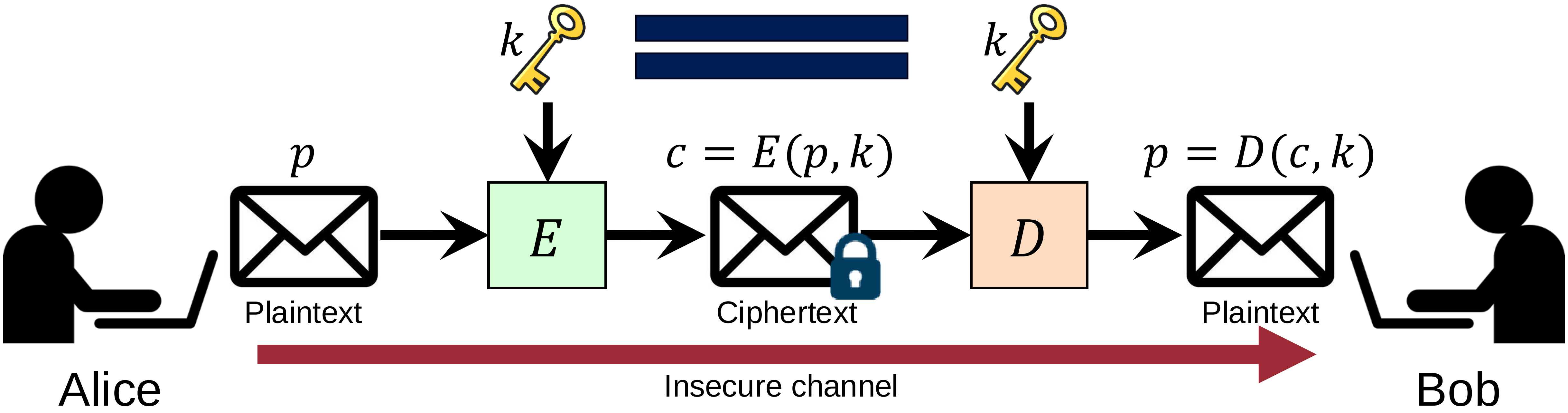
Hack Class101: Reported Bugs (Anonymized)

- Cross-Site Scripting (XSS) attacks
- Cross-Stie Request Forgery (CSRF) attacks
- Leak of the decryption key for paid video contents

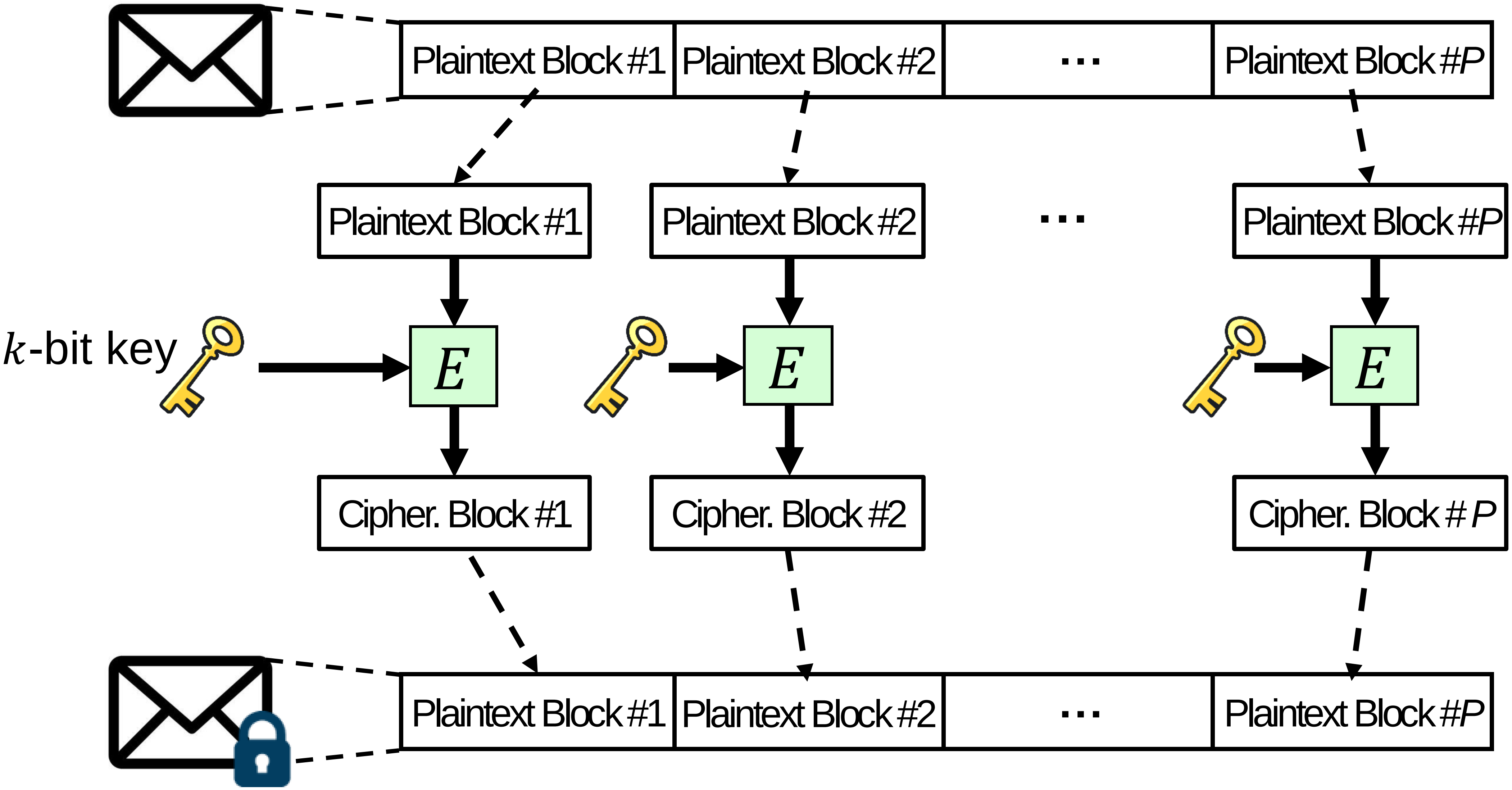
Your colleagues have already reported many vulnerabilities. I recommend getting involved in this activity as soon as possible ☺

Recap: Symmetric-key Encryption

- **Symmetric**: the encryption and decryption keys *are the same*

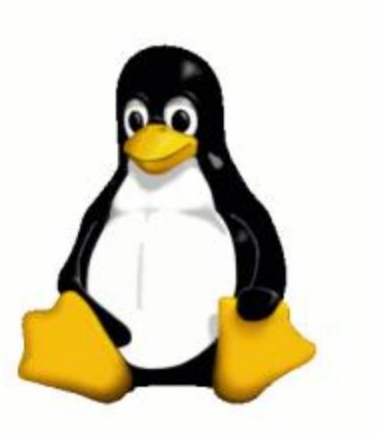


Recap: Practical Use of Block Cipher

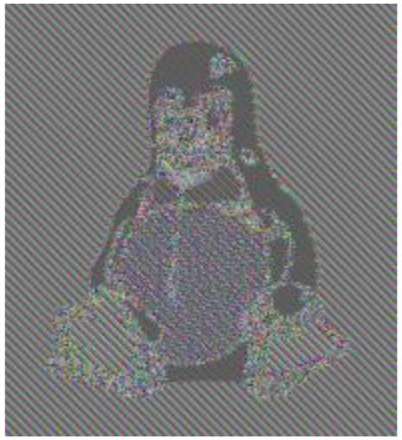


Recap: Problems for the Example

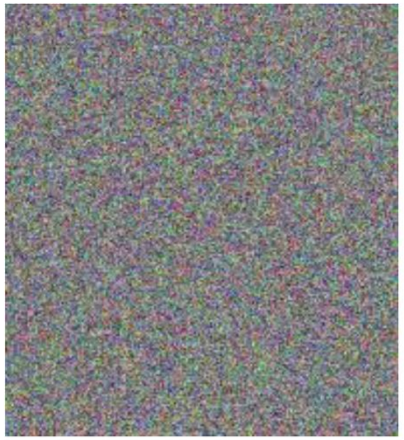
- Identical plaintext blocks → identical ciphertext blocks



Plaintext



Ciphertext of the Naive block cipher



Ciphertext we want!

How to generate different ciphertexts for the same plaintext?

Recap: Block Cipher Modes of Operation

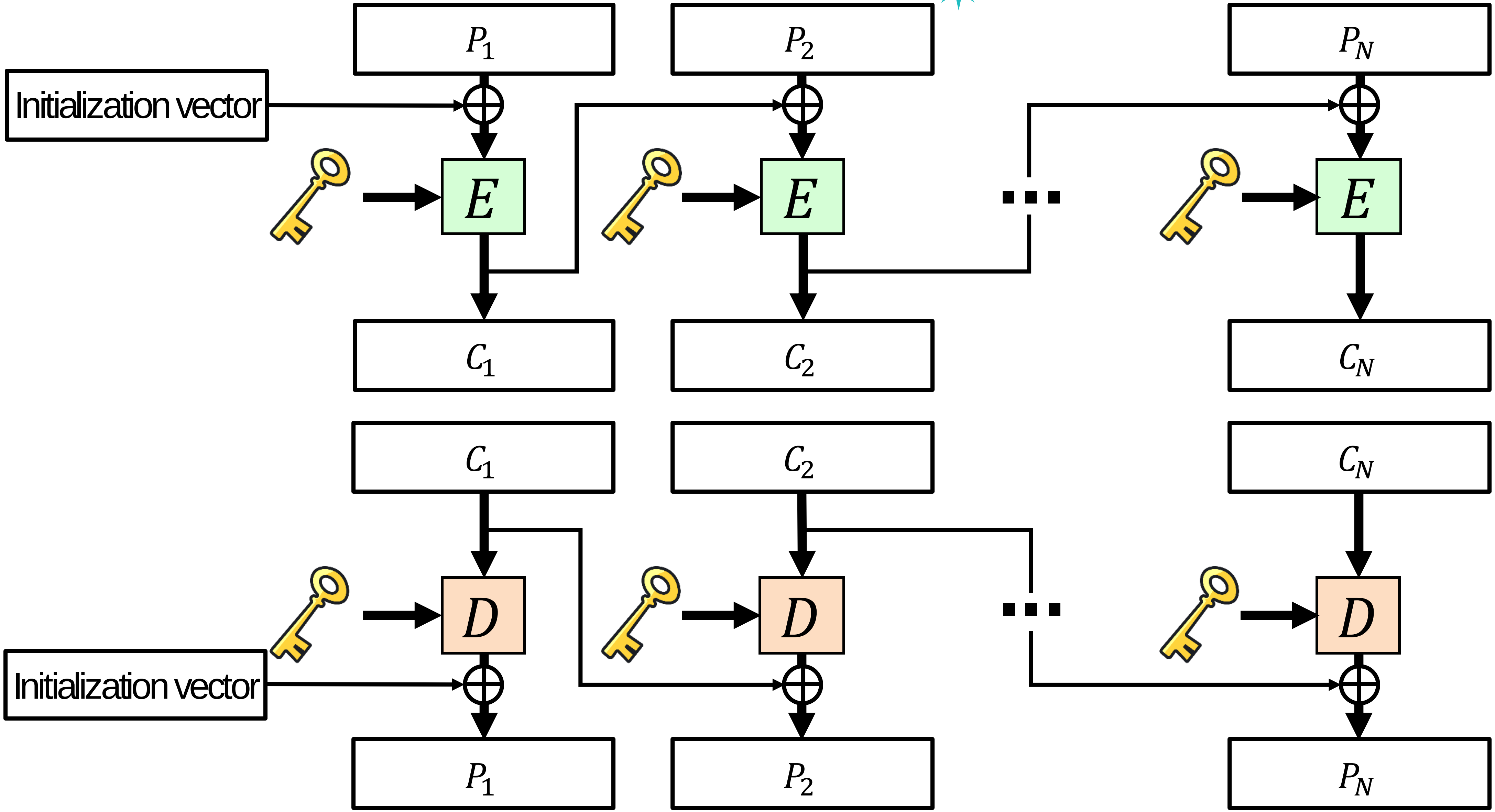
- Determine **how to repeatedly apply a single-block operation** to a sequence of blocks
- Different modes of operations
  - ECB: Electronic Code Book (The naïve one we've just discussed)
  - CBC: Cipher Block Chaining
  - CFB: Cipher FeedBack
  - OFB: Output FeedBack
  - CTR: CounTeR mode

Block cipher mode

Shell Command

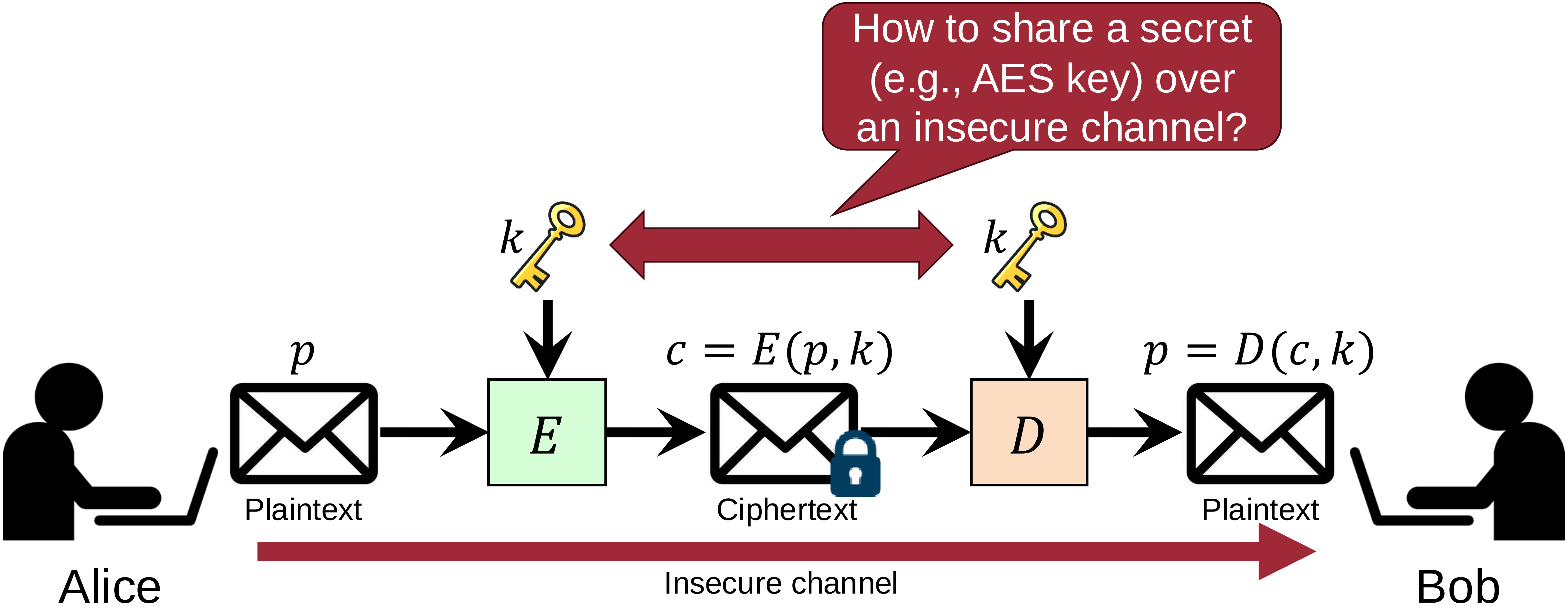
```
$ openssl enc -aes-128-cfb -e -in plain.bin -out cipher.bin -K
```

Recap: Cipher Block Chaining



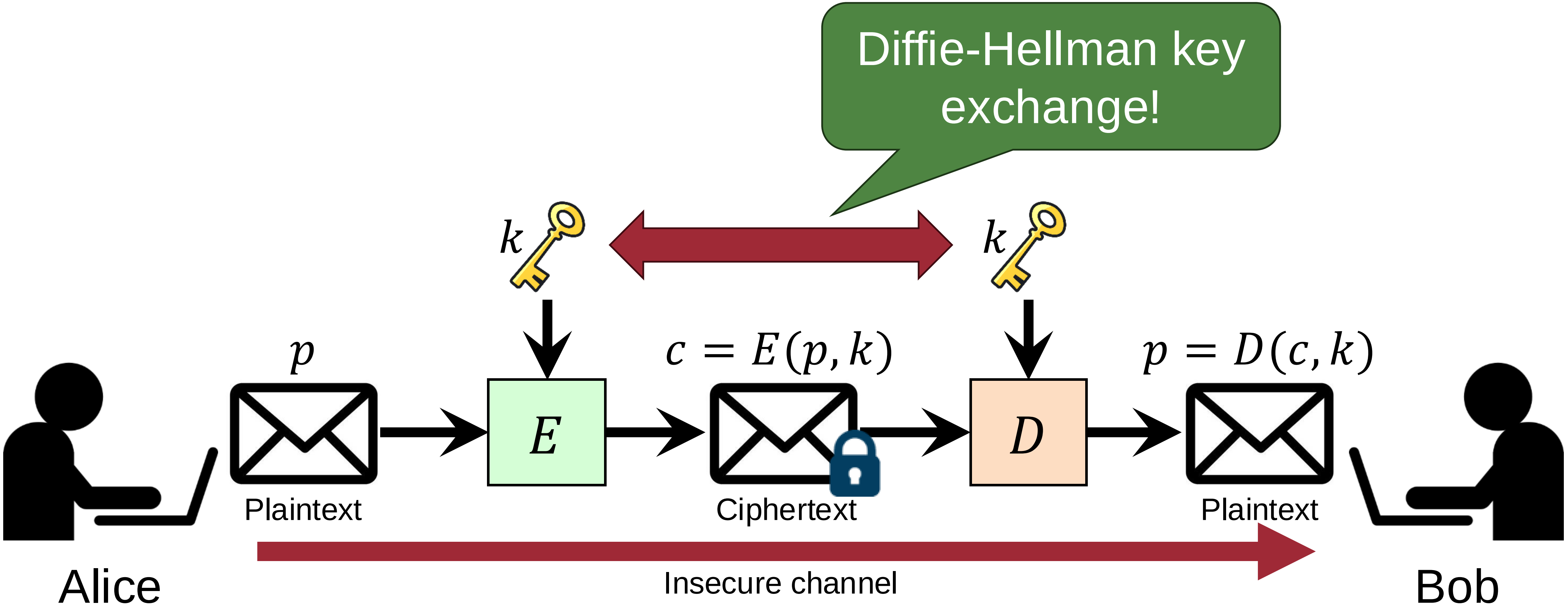
Recap: Symmetric-key Encryption

- **Symmetric**: the encryption and decryption keys *are the same*



Motivation of the Diffie-Hellman Key Exchange

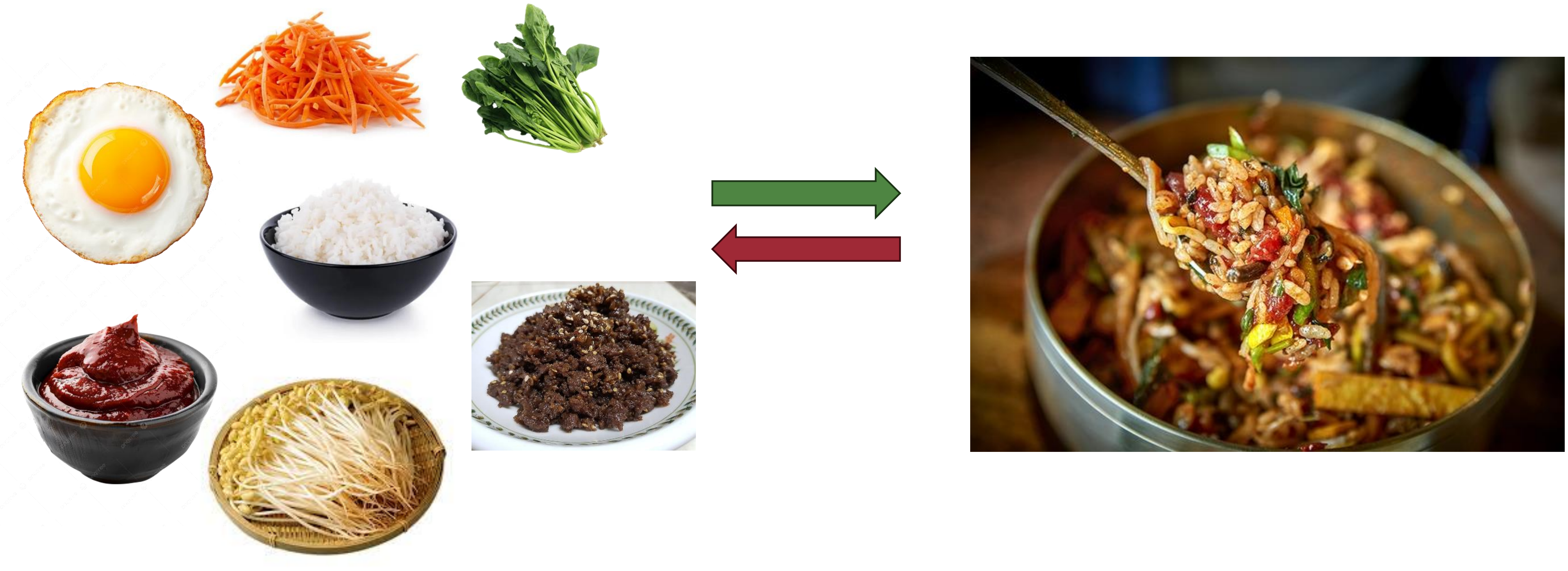
- **Symmetric**: the encryption and decryption keys *are the same*



Diffie-Hellman key exchange

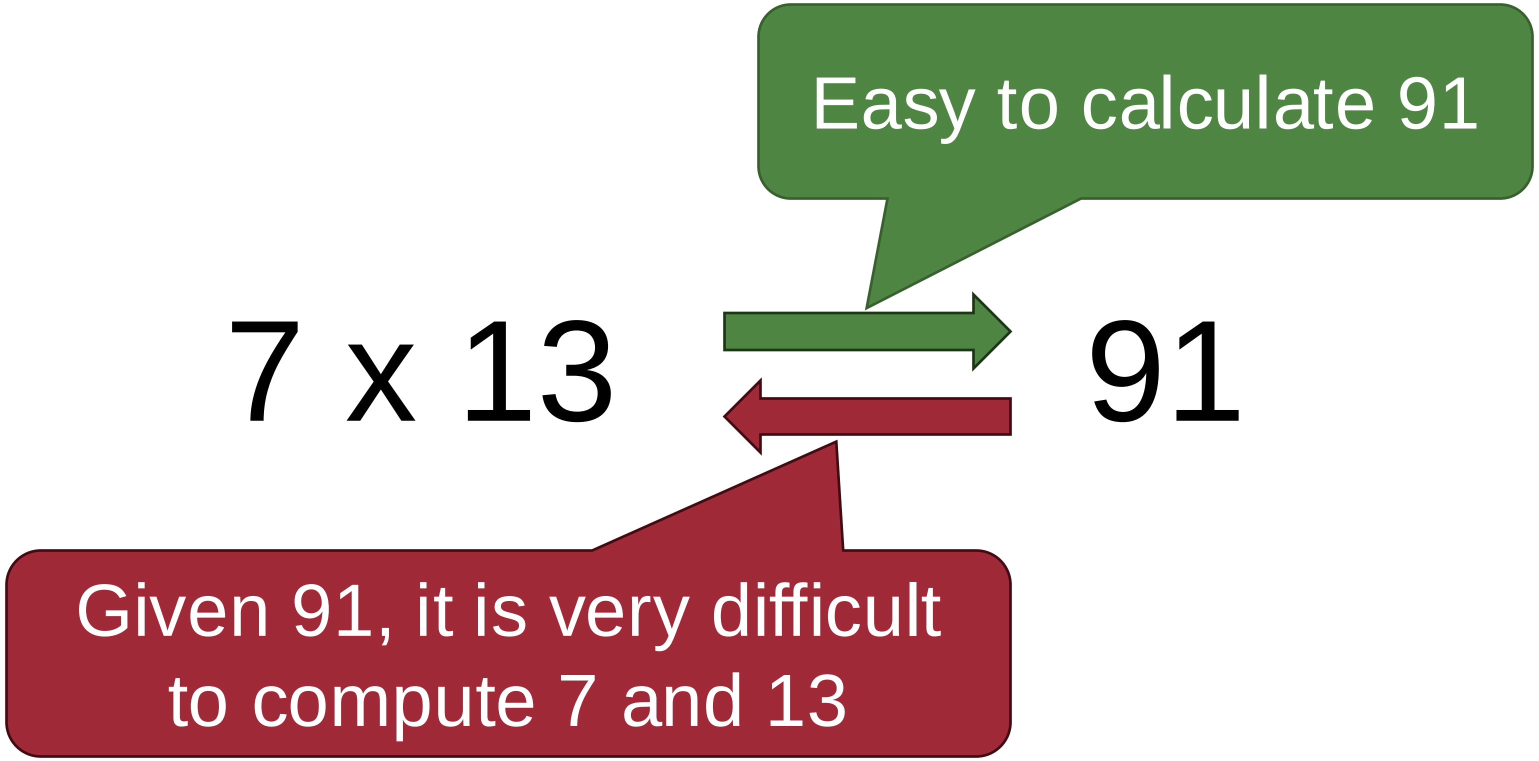
Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute



Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute



Integer Factorization Problem

Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute

Given $\underline{g, p, x}$, it is easy to calculate y

$$g^x \bmod p \longrightarrow y$$

Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute

$$g = 3$$
$$p = 5$$
$$x = 2$$

$$g^x \bmod p \longrightarrow y = ?$$

Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute

$g = 3$
$p = 5$
$x = 2$

Given $\underline{g, p, x}$, it is easy to calculate y

$$g^x \; mod \; p \longrightarrow y = 4$$

Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute

Given $\underline{g, p, x}$, it is easy to calculate y

$$g^x \bmod p \rightleftarrows y$$

Given $\underline{g, p, y}$, it is very difficult to compute x

Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute

$$g = 3$$
$$p = 5$$
$$x = ?$$

$$g^x \; mod \; p \longleftarrow y = 4$$

Given $\underline{g, p, y}$, it is very difficult to compute x

Discrete Logarithm Problem

Core Idea: One-way Function

- Easy in one direction, but hard in the reverse direction
  - f is easy to compute, but f^{-1} is difficult to compute

$$g = 3$$
$$p = 5$$
$$x = ?$$

$$g^{x} \; mod \; p \longleftarrow y = 4$$

There is no efficient algorithm known for computing discrete logarithms in general

Diffie-Hellman Key Exchange (1)

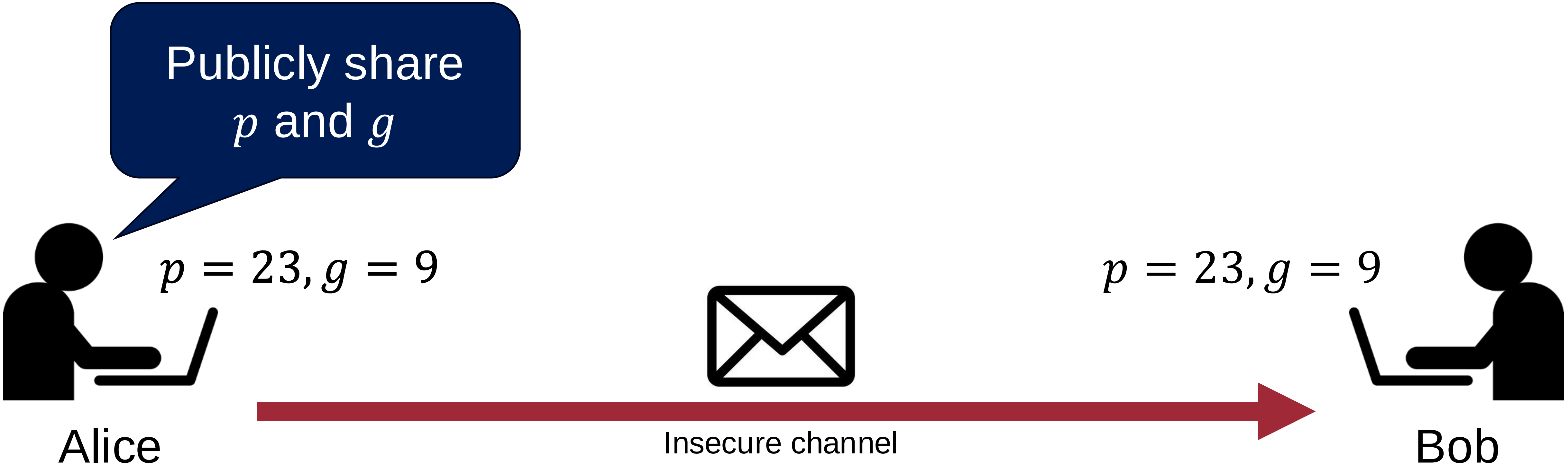
$$g^x \bmod p \rightleftarrows y$$

Pick two value:
Large prime p and
integer g

$p = 23, g = 9$

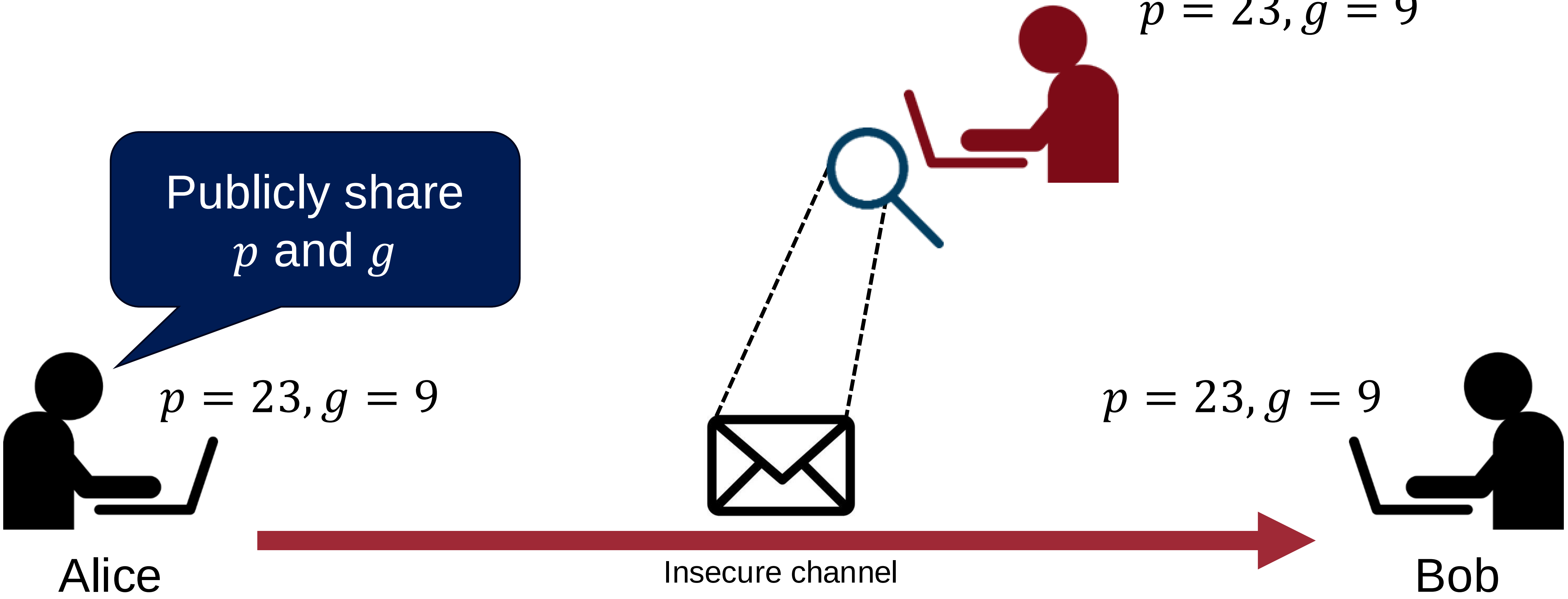
Diffie-Hellman Key Exchange (2)

$$g^x \bmod p \rightleftarrows y$$

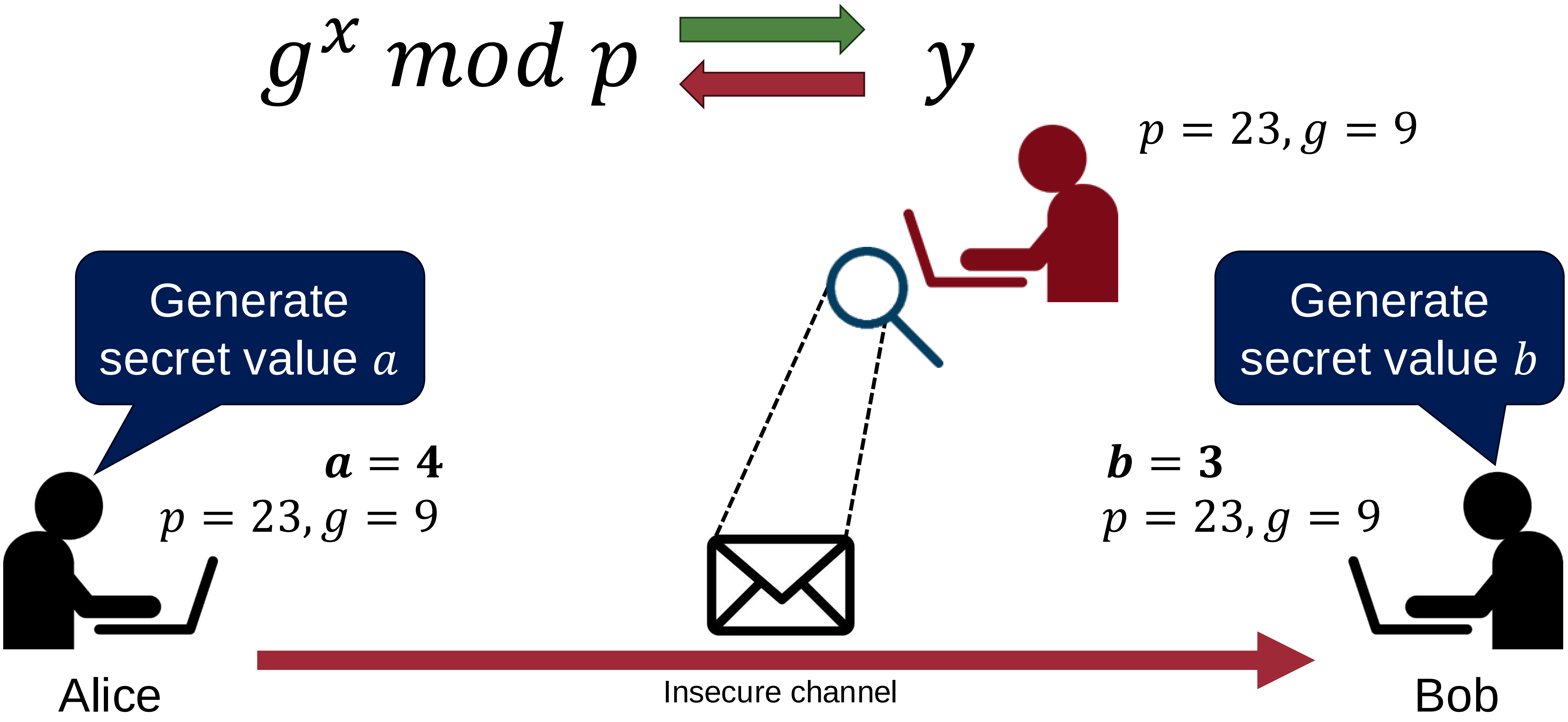


Diffie-Hellman Key Exchange (2)

$$g^x \bmod p \rightleftarrows y$$



Diffie-Hellman Key Exchange (3)



Diffie-Hellman Key Exchange (4)

$$g^x \bmod p \rightleftarrows y$$

$p = 23, g = 9$

Send
$A = g^a \bmod p$
to Bob

$\boldsymbol{a = 4}$
$p = 23, g = 9$
$A = (g^a \bmod p) = 6$

$\boldsymbol{b = 3}$
$p = 23, g = 9$
$A = (g^a \bmod p) = 6$

Alice

Insecure channel

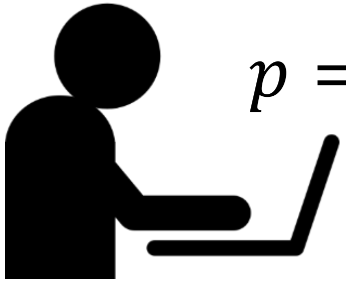
Bob

Diffie-Hellman Key Exchange (4)

$$g^x \bmod p \rightleftarrows y$$

$p = 23, g = 9$

$A = (g^a \bmod p) = 6$



$\boldsymbol{a = 4}$

$p = 23, g = 9$

$A = (g^a \bmod p) = 6$

Alice

$\boldsymbol{b = 3}$

$p = 23, g = 9$

$A = (g^a \bmod p) = 6$

Bob

Insecure channel

Diffie-Hellman Key Exchange (4)

$$g^x \; mod \; p \longleftarrow y$$

$p = 23, g = 9$

$\mathrm{A} = (g^a \; mod \; p) = 6$

Given $\underline{g, p, y}$, it is very difficult to compute a

$a = 4$

$p = 23, g = 9$

$\mathrm{A} = (g^a \; mod \; p) = 6$

$p = 23, g = 9$

$\mathrm{A} = (g^a \; mod \; p) = 6$

Alice

Insecure channel

Bob

Discrete Logarithm Problem

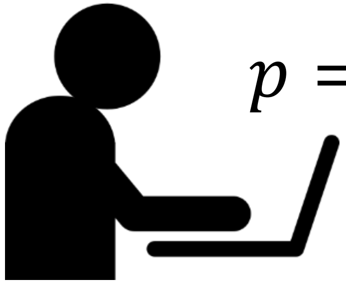
Diffie-Hellman Key Exchange (4)

$$g^x \; mod \; p \rightleftarrows y$$

$p = 23, g = 9$

$A = (g^a \; mod \; p) = 6$

Send
$B = g^b \; mod \; p$
to Alice



$\boldsymbol{a = 4}$

$p = 23, g = 9$

$A = (g^a \; mod \; p) = 6$

$B = (g^b \; mod \; p) = 16$

Alice

$\boldsymbol{b = 3}$

$p = 23, g = 9$

$A = (g^a \; mod \; p) = 6$

$B = (g^b \; mod \; p) = 16$

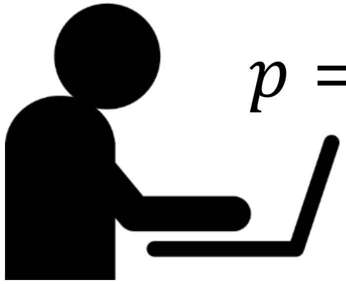
Bob

Insecure channel

Diffie-Hellman Key Exchange (4)

$$g^x \ mod \ p \rightleftarrows y$$

$p = 23, g = 9$
$A = (g^a \ mod \ p) = 6$
$B = (g^b \ mod \ p) = 16$



$\boldsymbol{a = 4}$
$p = 23, g = 9$
$A = (g^a \ mod \ p) = 6$
$B = (g^b \ mod \ p) = 16$

Alice

$\boldsymbol{b = 3}$
$p = 23, g = 9$
$A = (g^a \ mod \ p) = 6$
$B = (g^b \ mod \ p) = 16$

Bob

Insecure channel

Diffie-Hellman Key Exchange (5)

Symmetric key:

$$K = g^{ab} \bmod p$$

$p = 23, g = 9$

$A = (g^a \bmod p) = 6$

$B = (g^b \bmod p) = 16$

$\boldsymbol{a = 4}$

$p = 23, g = 9$

$A = (g^a \bmod p) = 6$

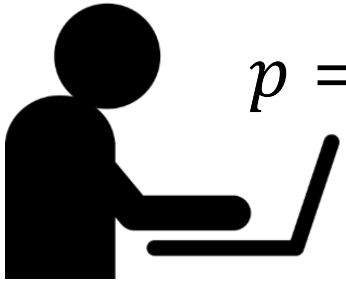
$B = (g^b \bmod p) = 16$

$\boldsymbol{b = 3}$

$p = 23, g = 9$

$A = (g^a \bmod p) = 6$

$B = (g^b \bmod p) = 16$



Alice

Insecure channel

Bob

Diffie-Hellman Key Exchange (5)

Symmetric key:

$$K = g^{ab} \; mod \; p$$

$$p = 23, g = 9$$
$$A = (g^a \; mod \; p) = 6$$
$$B = (g^b \; mod \; p) = 16$$

$$K = (B^{a} \; mod \; p) = (g^{ab} mod \; p)$$
$$= (16^4 \; mod \; 23) = 9$$

Theorem:

$$((X \; mod \; p)^k \; mod \; p) = (X^k \; mod \; p)$$

$$\boldsymbol{a = 4}$$
$$p = 23, g = 9$$
$$A = (g^a \; mod \; p) = 6$$
$$B = (g^b \; mod \; p) = 16$$

$$\boldsymbol{b = 3}$$
$$p = 23, g = 9$$
$$A = (g^a \; mod \; p) = 6$$
$$B = (g^b \; mod \; p) = 16$$

Alice

Insecure channel

Bob

Diffie-Hellman Key Exchange (5)

Symmetric key:

$$K = g^{ab} \; mod \; p$$

$p = 23, g = 9$

$A = (g^a \; mod \; p) = 6$

$B = (g^b \; mod \; p) = 16$

$K = (B^{a} \; mod \; p) = \left(g^{ab} mod \; p\right)$

$= (16^4 \; mod \; 23) = 9$

$=$

$K = \left(A^{b} \; mod \; p\right) = \left(g^{ab} mod \; p\right)$

$= (6^3 \; mod \; 23) = 9$

$\boldsymbol{a = 4}$

$p = 23, g = 9$

$A = (g^a \; mod \; p) = 6$

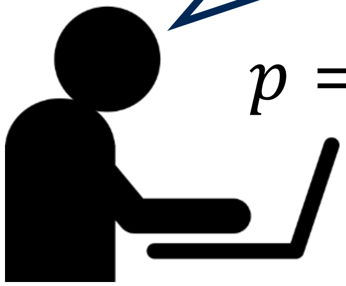
$B = (g^b \; mod \; p) = 16$

$\boldsymbol{b = 3}$

$p = 23, g = 9$

$A = (g^a \; mod \; p) = 6$

$B = (g^b \; mod \; p) = 16$



Alice

Insecure channel

Bob

Diffi… ge (5)

The attacker cannot efficiently compute $(g^{ab} \; mod \; p)$
without knowing a and b

Sym…

$K = g^{ab} \; mod \; p$

$p = 23, g = 9$
$A = (g^a \; mod \; p) = 6$
$B = (g^b \; mod \; p) = 16$

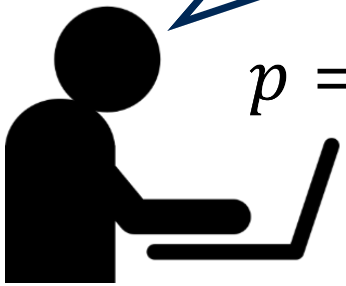
$K = (B^a \; mod \; p) = \left(g^{ab} mod \; p\right)$
$= (16^4 \; mod \; 23) = 9$

$=$

$K = \left(A^b \; mod \; p\right) = \left(g^{ab} mod \; p\right)$
$= (6^3 \; mod \; 23) = 9$

$\boldsymbol{a = 4}$
$p = 23, g = 9$
$A = (g^a \; mod \; p) = 6$
$B = (g^b \; mod \; p) = 16$

$\boldsymbol{b = 3}$
$p = 23, g = 9$
$A = (g^a \; mod \; p) = 6$
$B = (g^b \; mod \; p) = 16$



Alice

Insecure channel

Bob

Why should p be Prime?

Symmetric key:

$$K = g^{ab} \bmod p$$

$g = 2$

Too simple key pattern that can be inferred

$p = 11$	$p = 12$
$2^0 \bmod 11 = 1$	$2^0 \bmod 12 = 1$
$2^1 \bmod 11 = 2$	$2^1 \bmod 12 = 2$
$2^2 \bmod 11 = 4$	$2^2 \bmod 12 = 4$
$2^3 \bmod 11 = 8$	$2^3 \bmod 12 = 8$
$2^4 \bmod 11 = 5$	$2^4 \bmod 12 = 4$
$2^5 \bmod 11 = 10$	$2^5 \bmod 12 = 8$
$2^6 \bmod 11 = 9$	$2^6 \bmod 12 = 4$
$2^7 \bmod 11 = 7$	$2^7 \bmod 12 = 8$
$2^8 \bmod 11 = 3$	$2^8 \bmod 12 = 4$
$2^9 \bmod 11 = 6$	$2^9 \bmod 12 = 8$
$2^{10} \bmod 11 = 1$	$2^{10} \bmod 12 = 4$

Diffie-Hellman Key Exchange

Symmetric key:

$$K = g^{ab} \bmod p$$

Problems?

$$K = (B^{a} \bmod p) = \left(g^{ab} \bmod p\right) = (16^4 \bmod 23) = 9$$

=

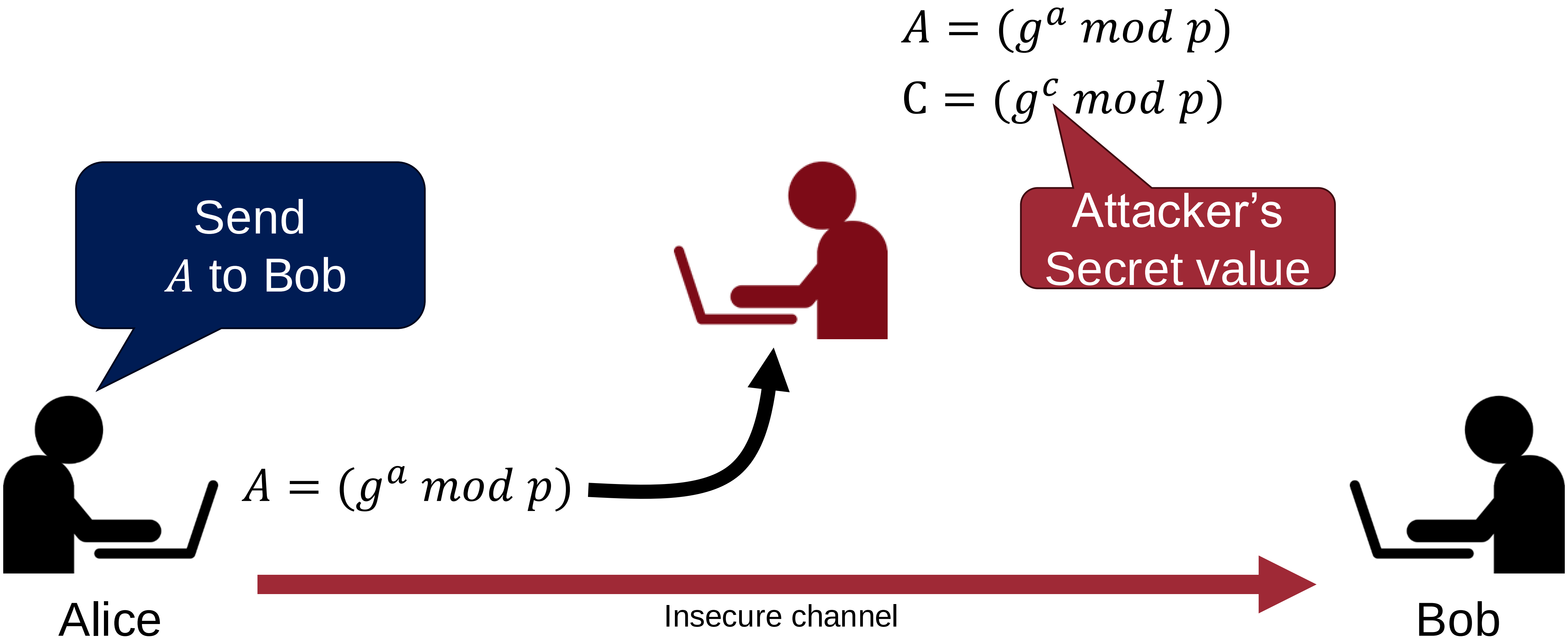
$$K = \left(A^{b} \bmod p\right) = \left(g^{ab} \bmod p\right) = (6^3 \bmod 23) = 9$$

Alice

Insecure channel

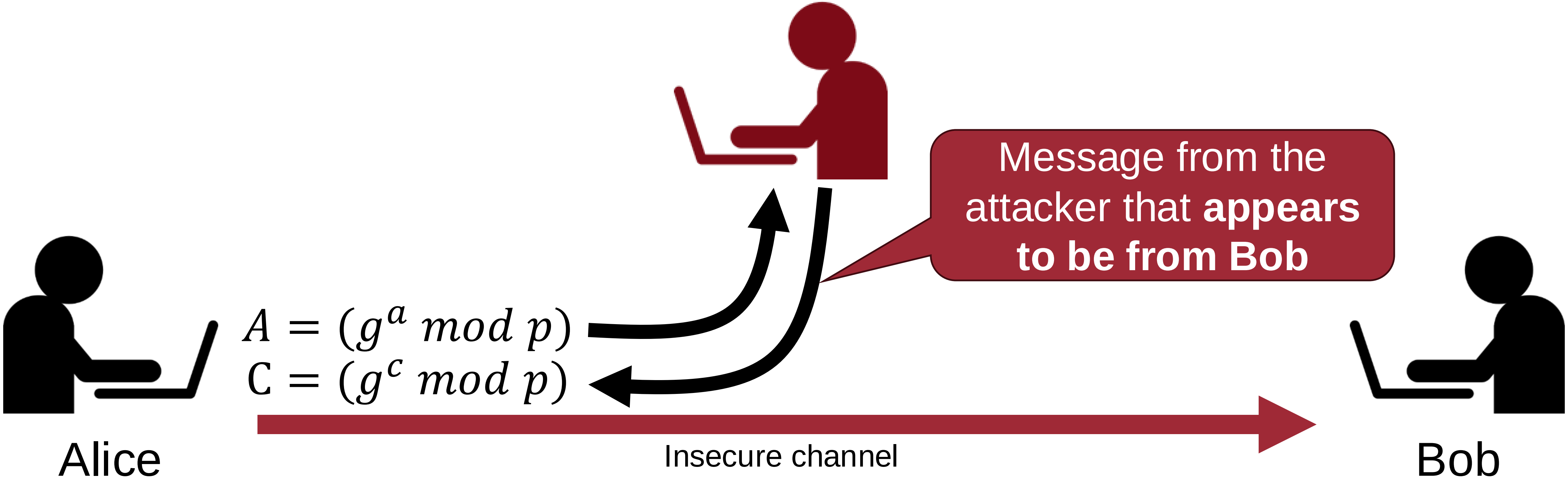
Bob

Problem (1): Man-in-the-Middle Attack

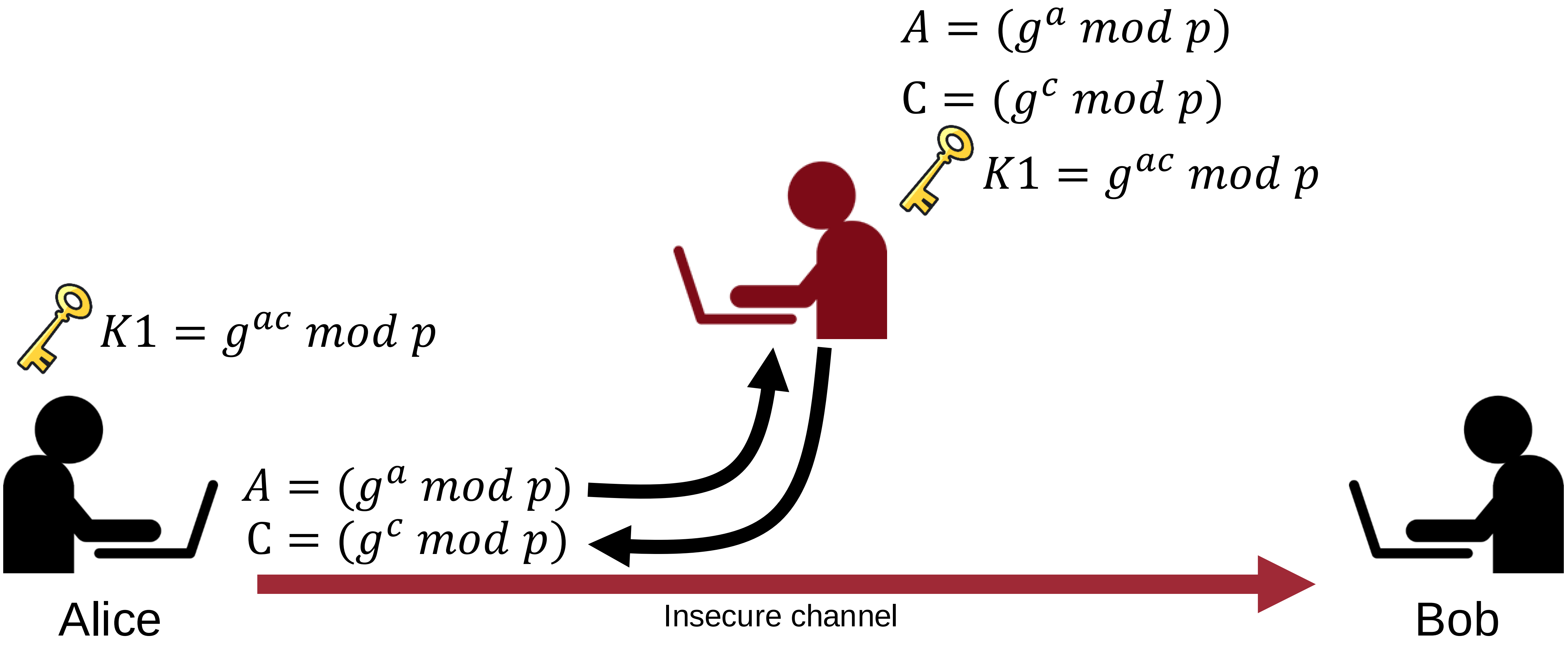


Problem (1): Man-in-the-Middle Attack

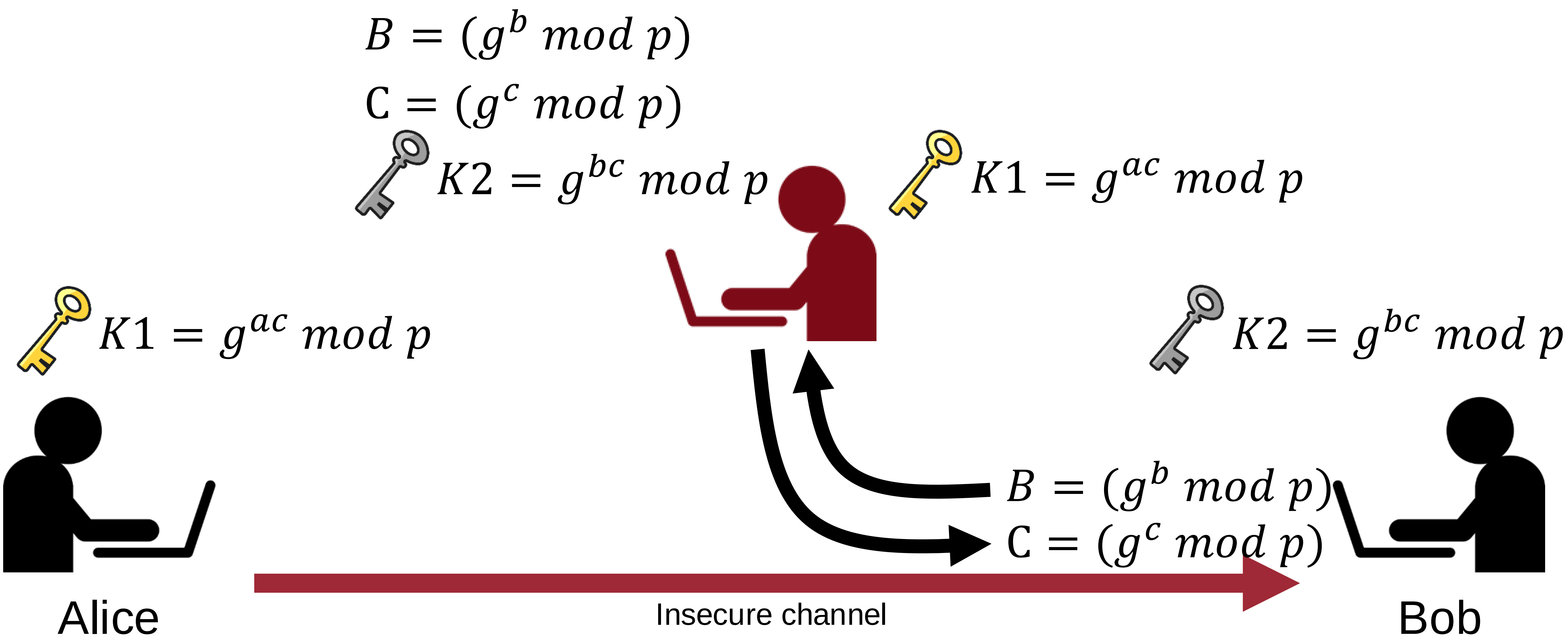
$$A = (g^a \, mod \, p)$$
$$C = (g^c \, mod \, p)$$



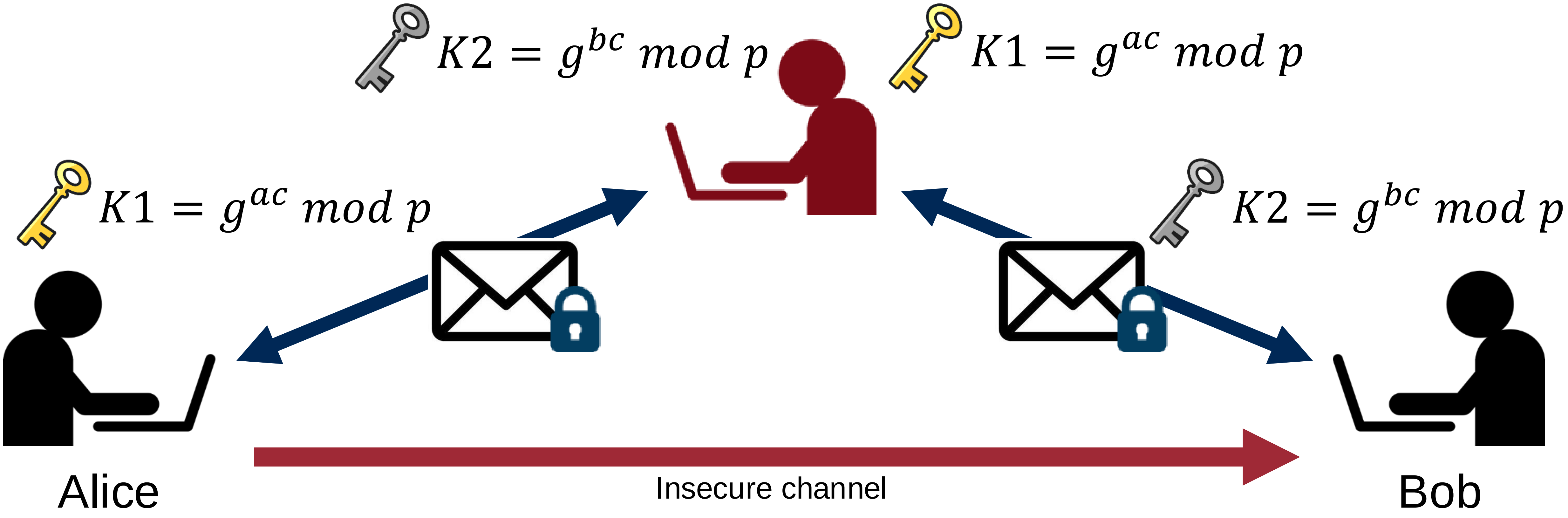
Problem (1): Man-in-the-Middle Attack



Problem (1): Man-in-the-Middle Attack



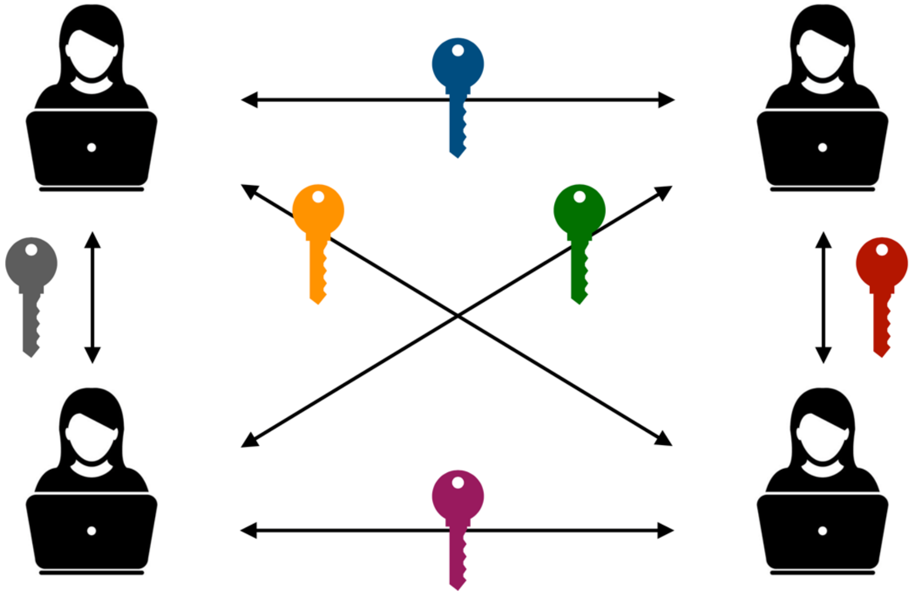
Problem (1): Man-in-the-Middle Attack



Problem (2): Maintenance Problems

- Recap: the same key shared between two parties
- What happens if there are many users?
  - n users: $\binom{n}{2} = n(n-1)/2$
  - Example: 100 users → 4,950 keys
- Key distribution and maintenance problem

How to solve this issue?



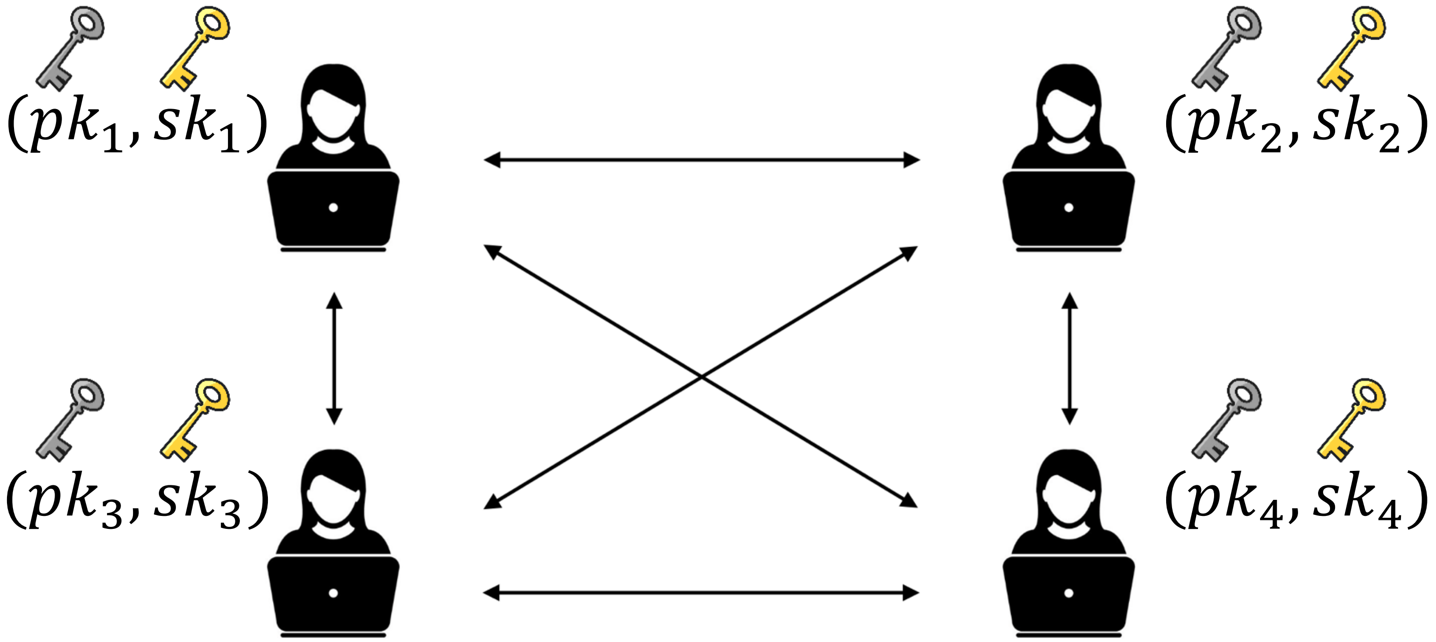
Asymmetric-key Cryptography

Asymmetric-key Cryptography

- Each party has two distinct keys: public key and private key
  - Also known as public-key algorithm
- Invented in 1976 by Diffie and Hellman (ACM Turing Award 2015)

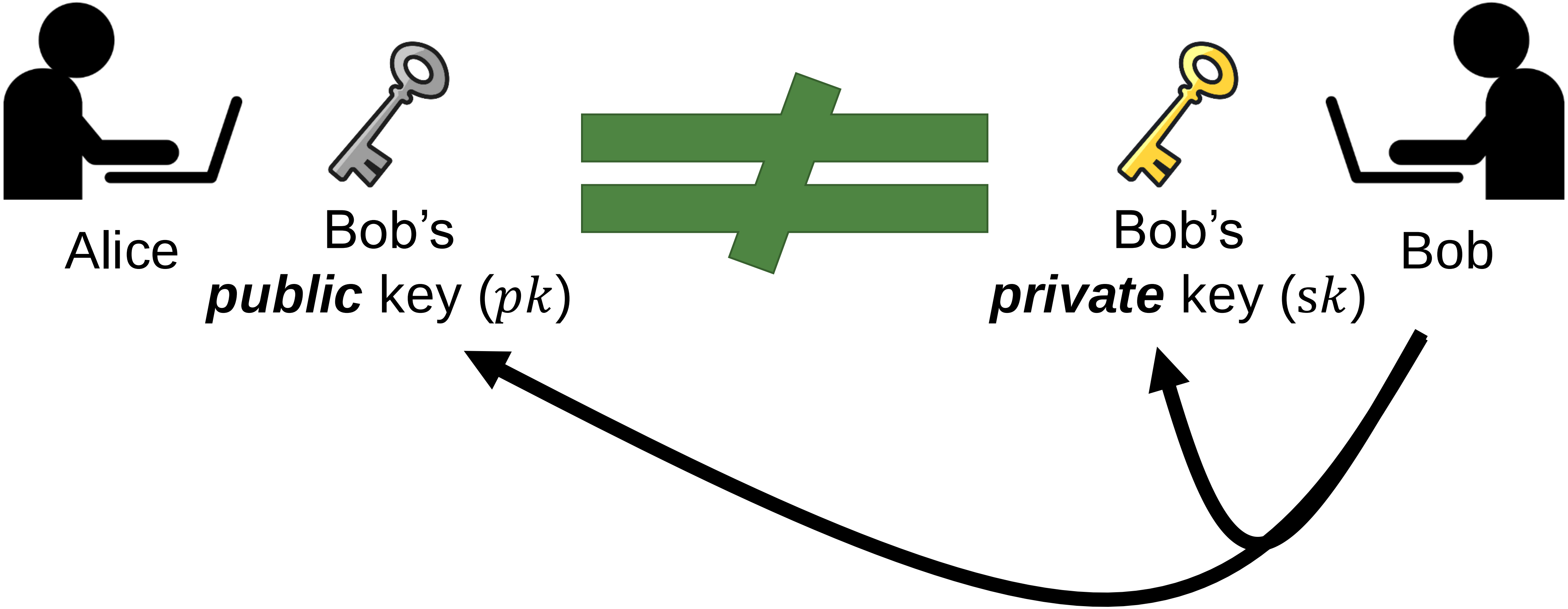
Asymmetric-key Cryptography

- pk: public key, widely disseminated, used for encryption
- sk: private key kept secretly, used for decryption
- **More robust against man-in-the-middle attack**
- **Good maintenance**: n users $\rightarrow 2n$ keys



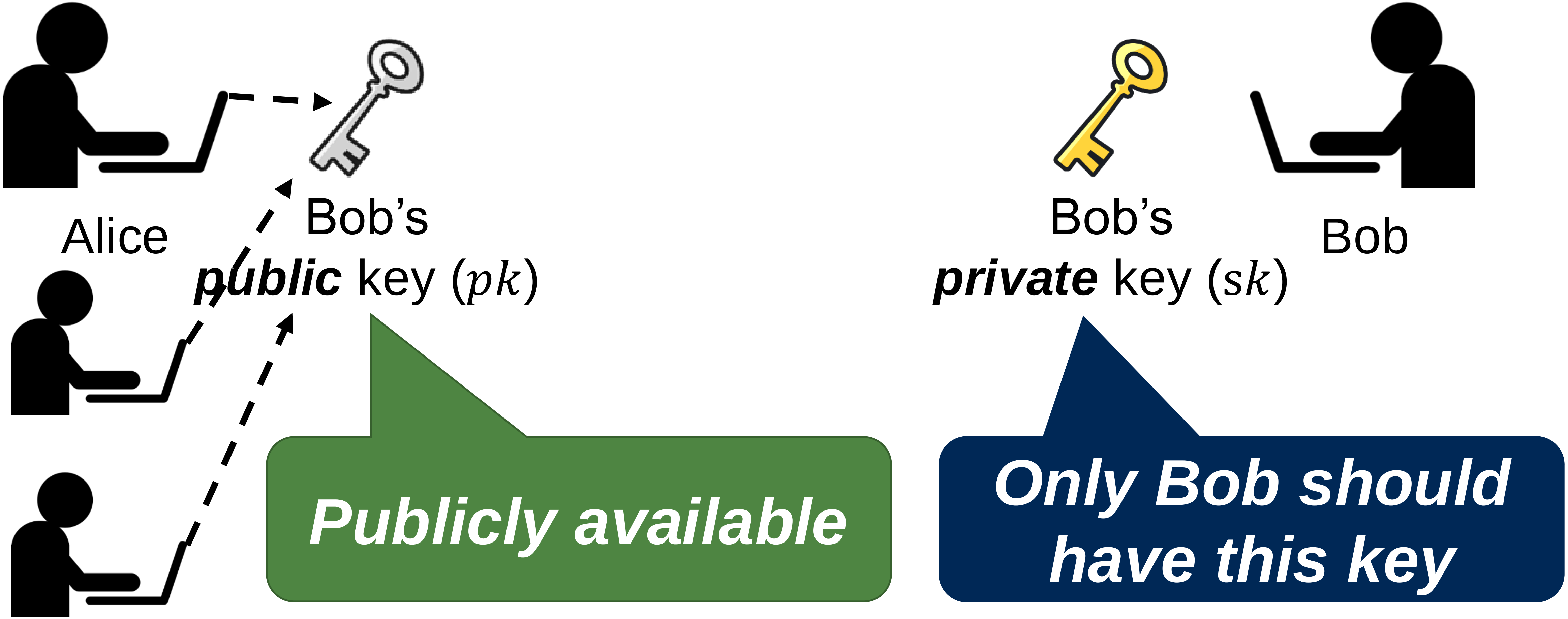
Asymmetric-key Cryptography

- pk: public key, widely disseminated, used for encryption
- sk: private key kept secretly, used for decryption



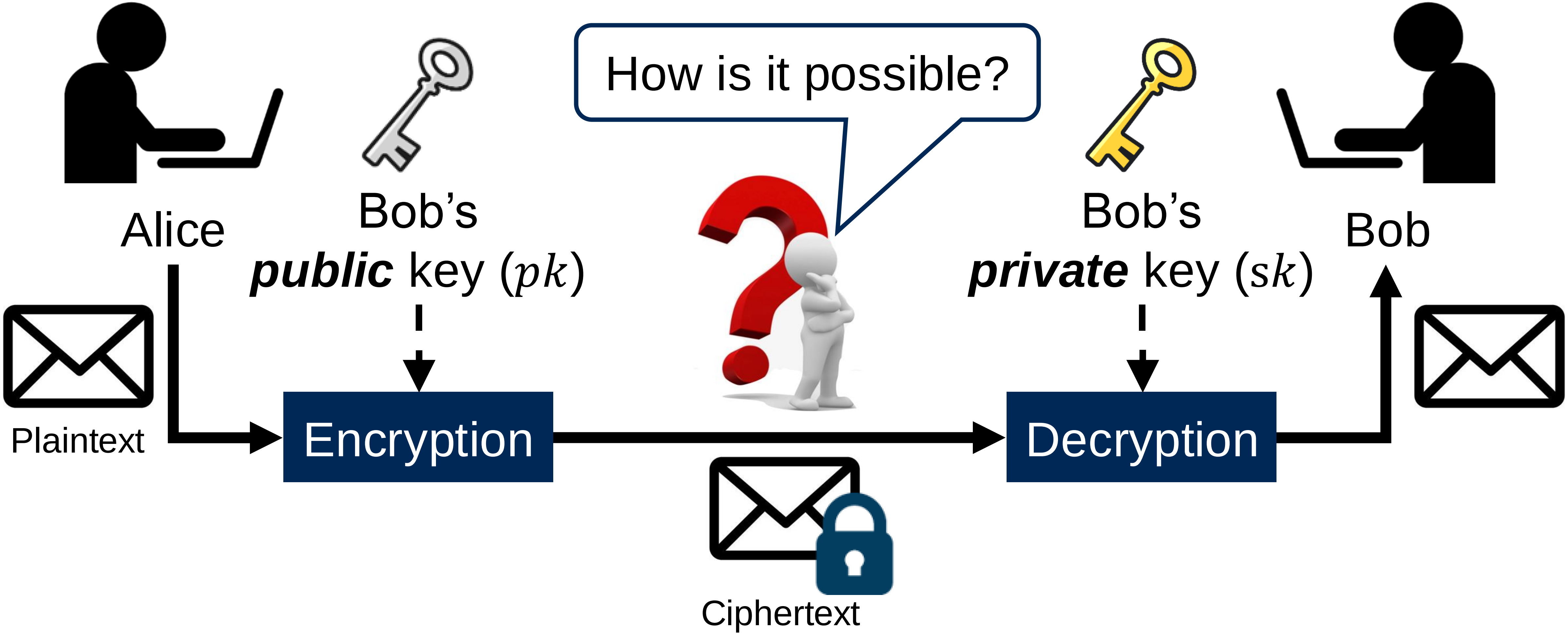
Asymmetric-key Cryptography

- pk: public key, widely disseminated, used for encryption
- sk: private key kept secretly, used for decryption



Asymmetric-key Cryptography

- pk: public key, widely disseminated, used for encryption
- sk: private key kept secretly, used for decryption



RSA Cryptosystem

RSA Cryptosystem

- Invented by Rivest, Shamir, and Adleman (MIT) in 1977
  - ACM Turing award in 2002

- Rely on the practical ***difficulty of factoring the product of two large prime numbers***
  - Security based on *Prime Factorization Problem*

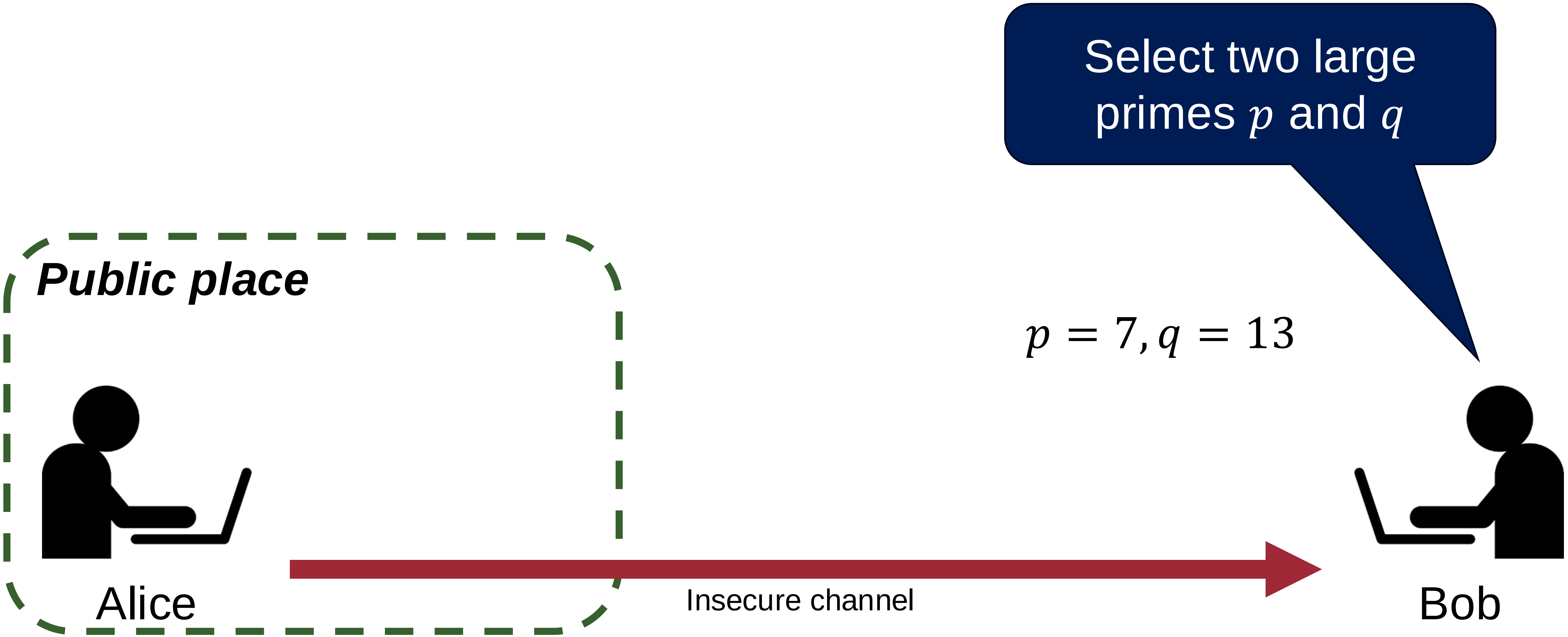
Prime Factorization Problem

Given large prime p and q, it is easy to calculate n

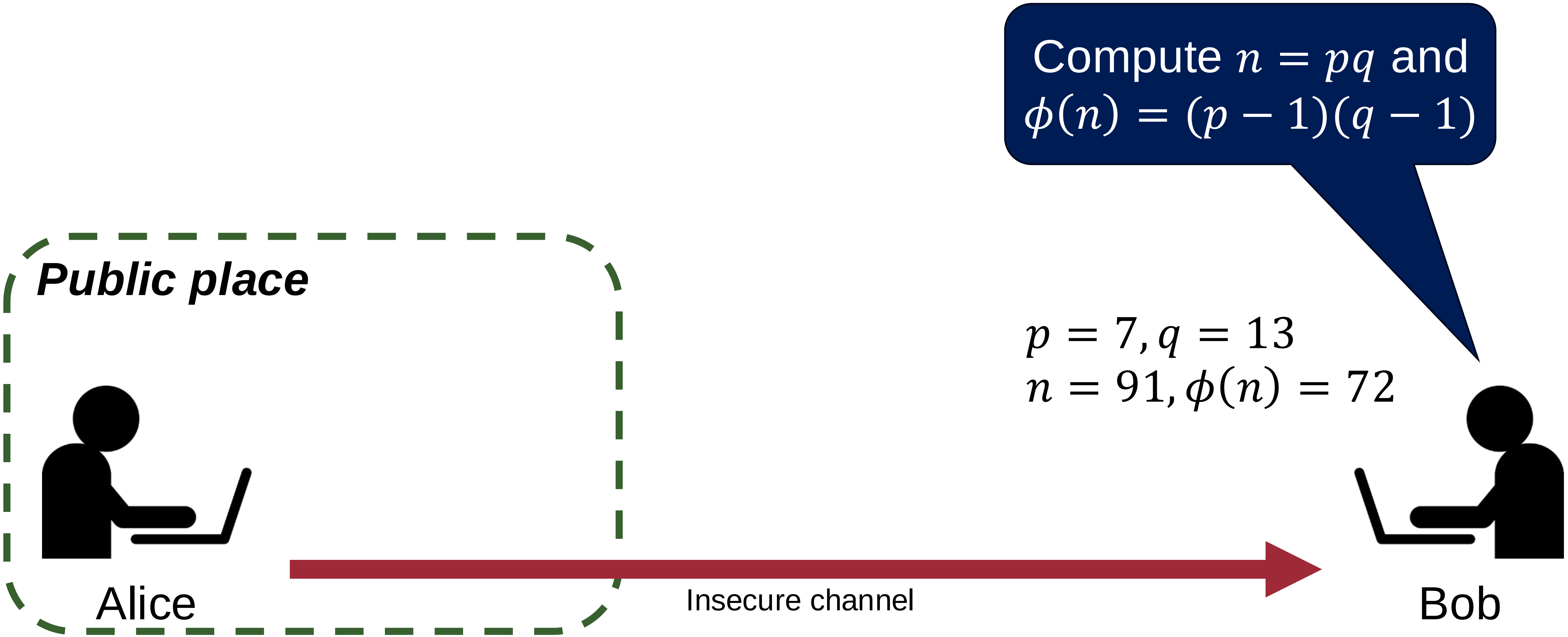
$$p \times q \rightleftarrows n$$

Given $\underline{n}$, it is very difficult to compute p and q

RSA Algorithm (1): Key Generation



RSA Algorithm (1): Key Generation



RSA Algorithm (1): Key Generation

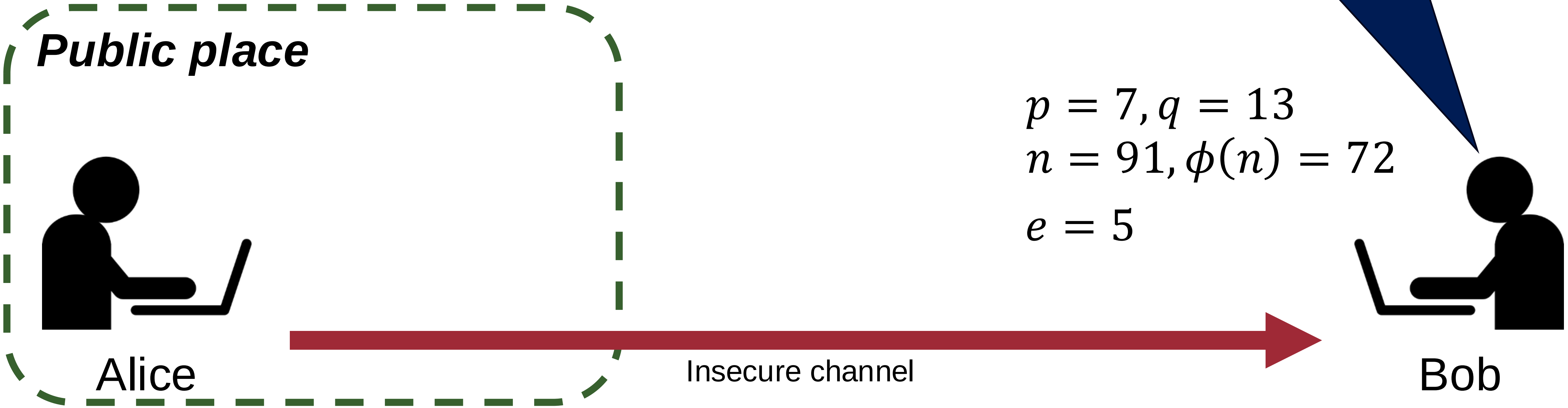
Choose e s.t.

- $1 < e < \phi(n)$ **and**
- $\gcd(\phi(n), e) = 1$

$p = 7, q = 13$

$n = 91, \phi(n) = 72$

$e = 5$



RSA Algorithm (1): Key Generation

How to find d?
→ Extended Euclidean Algorithm!

Choose d s.t.

- $1 < d < \phi(n)$ **and**
- $(ed \bmod \phi(n)) = 1$

Public place

$p = 7, q = 13$
$n = 91, \phi(n) = 72$

$e = 5$
$d = 29$

Alice

Bob

Insecure channel

Euclidean Algorithm

Goal: Finding Greatest Common Divisor (GCD)

Fact 1: $\mathrm{gcd}(a, 0) = a$

Fact 2: $\mathrm{gcd}(a, b) = \mathrm{gcd}(b, r)$, where r is the remainder of dividing a by b ($a > b$)

Example

$\mathrm{gcd}(72, 5)$

Euclidean Algorithm

Goal: Finding Greatest Common Divisor (GCD)

Fact 1: $\gcd(a, 0) = a$
Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b $(a > b)$

Example

$\gcd(72, 5)$ $\quad 72 = (5 * 14) + 2$

Euclidean Algorithm

Goal: Finding Greatest Common Divisor (GCD)

Fact 1: $\gcd(a, 0) = a$
Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b $(a > b)$

Example

b *r*

$\gcd(72, 5)$ $72 = (5 * 14) + 2$

$\gcd(5, 2)$ $5 = (2 * 2) + 1$

Euclidean Algorithm

Goal: Finding Greatest Common Divisor (GCD)

Fact 1: $\gcd(a, 0) = a$
Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b $(a > b)$

Example

$\gcd(72, 5) \qquad 72 = (5 * 14) + 2$

$\gcd(5, 2) \qquad 5 = (2 * 2) + 1$

$\gcd(2, 1) \qquad 2 = (2 * 1) + 0$

Euclidean Algorithm

Goal: Finding Greatest Common Divisor (GCD)

Fact 1: $\gcd(a, 0) = a$
Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b $(a > b)$

Example

$\gcd(72, 5)$ $\quad 72 = (5 * 14) + 2$

$\gcd(5, 2)$ $\quad 5 = (2 * 2) + 1$

$\gcd(2, 1)$ $\quad 2 = (2 * 1) + 0$

$\gcd(1, 0)$

Euclidean Algorithm

Goal: Finding Greatest Common Divisor (GCD)

Fact 1: $\gcd(a, 0) = a$
Fact 2: $\gcd(a, b) = \gcd(b, r)$, where r is the remainder of dividing a by b $(a > b)$

Example

$\gcd(72, 5) \qquad 72 = (5 * 14) + 2$

$\gcd(5, 2) \qquad 5 = (2 * 2) + 1$

$\gcd(2, 1) \qquad 2 = (2 * 1) + 0$

$\gcd(1, 0) = 1$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

Choose e s.t.
- $1 < e < \phi(n)$ **and**
- $\gcd(\phi(n), e) = 1$

Choose d s.t.
- $1 < d < \phi(n)$ **and**
- $(ed \bmod \phi(n)) = 1$

$p = 7, q = 13$
$n = 91, \phi(n) = 72$

$e = 5$
$d = 29$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

Choose e s.t.
- $1 < e < \phi(n)$ **and**
- $\gcd(\phi(n), e) = 1$

Choose d s.t.
- $1 < d < \phi(n)$ **and**
- $(ed \; mod \; \phi(n)) = 1$

$p = 7, q = 13$
$n = 91, \phi(n) = 72$

$e = 5$
$d = 29$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

We can find the value d!

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

Choose e s.t.
- $1 < e < \phi(n)$ **and**
- $\gcd(\phi(n), e) = 1$

Choose d s.t.
- $1 < d < \phi(n)$ **and**
- $(ed \ mod \ \phi(n)) = 1$

$p = 7, q = 13$
$n = 91, \phi(n) = 72$

$e = 5$
$d = 29$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

$(e = 5, \phi(n) = 72)$

Example

$\gcd(72, 5)$ $\quad 72 = (5 * 14) + 2$

$\gcd(5, 2)$ $\quad 5 = (2 * 2) + 1$

$\gcd(2, 1)$ $\quad 2 = (2 * 1) + 0$

$\gcd(1, 0) = 1$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

$$(e = 5, \phi(n) = 72)$$

Example

$\gcd(72, 5)$ $\quad 72 = (5 * 14) + 2$

$\gcd(5, 2)$ $\quad 5 = (2 * 2) + 1 \Rightarrow 5 - (2 * 2) = 1$

$\gcd(2, 1)$ $\quad 2 = (2 * 1) + 0$

$\gcd(1, 0) = 1$

$x = 1$
$y = -2$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

$$(e = 5, \phi(n) = 72)$$

Example

$$2 = 72 - (5 * 14)$$

$\gcd(72, 5)$ $\quad 72 = (5 * 14) + 2$

$\gcd(5, 2)$ $\quad 5 = (2 * 2) + 1 \Rightarrow 5 - (2 * 2) = 1$

$\gcd(2, 1)$ $\quad 2 = (2 * 1) + 0$

$\gcd(1, 0) = 1$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

$$(e = 5, \phi(n) = 72)$$

Example

$\gcd(72, 5)$ $72 = (5 * 14) + 2$ ➡ $5 - ((72 - 5 * 14) * 2) = 1$

$\gcd(5, 2)$ $5 = (2 * 2) + 1$ ➡ $5 - (2 * 2) = 1$

$\gcd(2, 1)$ $2 = (2 * 1) + 0$

$\gcd(1, 0) = 1$

Extended Euclidean Algorithm

- **Goal:** Computing integers x and y s.t.

$$ax + by = \gcd(a, b)$$

$$ed + \phi(n)(-k) = \gcd(\phi(n), e) = 1$$

$$(e = 5, \phi(n) = 72)$$

Example

$\gcd(72, 5)$ $\quad 72 = (5 * 14) + 2 \;\rightarrow\; 5 * 29 + 72(-2) = 1$

$\gcd(5, 2)$ $\quad 5 = (2 * 2) + 1 \;\rightarrow\; 5 - (2 * 2) = 1$

$\gcd(2, 1)$ $\quad 2 = (2 * 1) + 0$

$\gcd(1, 0) = 1$

$x = \mathrm{d} = 29$
$\mathrm{y} = -\mathrm{k} = -2$

Extended Euclidean Algorithm: Logic Flow

$r_1 \leftarrow a; \quad r_2 \leftarrow b;$ (Initialization)

while $(r_2 > 0)$

{

$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$

$r_1 \leftarrow r_2; \quad r_2 \leftarrow r;$

}

$\gcd(a, b) \leftarrow r_1$

Euclidean Algorithm

$r_1 \leftarrow a; \quad r_2 \leftarrow b;$
$s_1 \leftarrow 1; \quad s_2 \leftarrow 0;$ (Initialization)
$t_1 \leftarrow 0; \quad t_2 \leftarrow 1;$

while $(r_2 > 0)$

{

$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$
$r_1 \leftarrow r_2; \quad r_2 \leftarrow r;$ (Updating r's)

$s \leftarrow s_1 - q \times s_2;$
$s_1 \leftarrow s_2; \quad s_2 \leftarrow s;$ (Updating s's)

$t \leftarrow t_1 - q \times t_2;$
$t_1 \leftarrow t_2; \quad t_2 \leftarrow$ t; (Updating t's)

}

$\gcd(a, b) \leftarrow r_1; \quad s \leftarrow s_1; \quad t \leftarrow t_1$

Extended Euclidean Algorithm

RSA Algorithm (1): Key Generation

How to find d?
→ Extended Euclidean Algorithm!

Choose d s.t.

- $1 < d < \phi(n)$ **and**
- $(ed \bmod \phi(n)) = 1$

Public place

$p = 7, q = 13$
$n = 91, \phi(n) = 72$

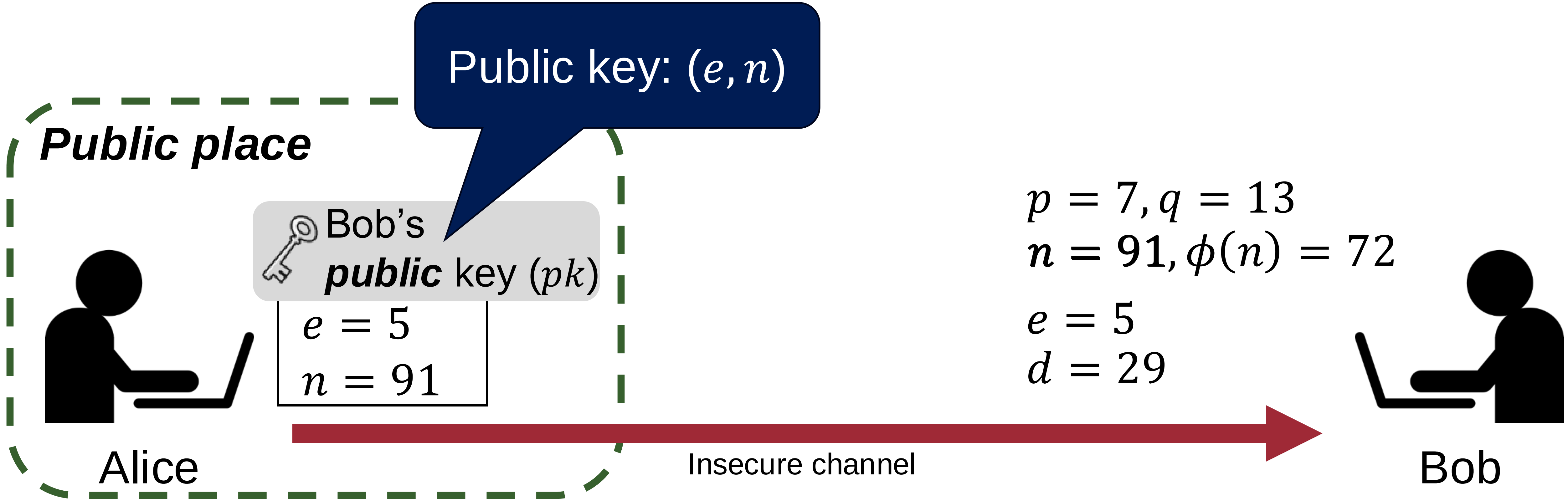
$e = 5$
$d = 29$

Alice

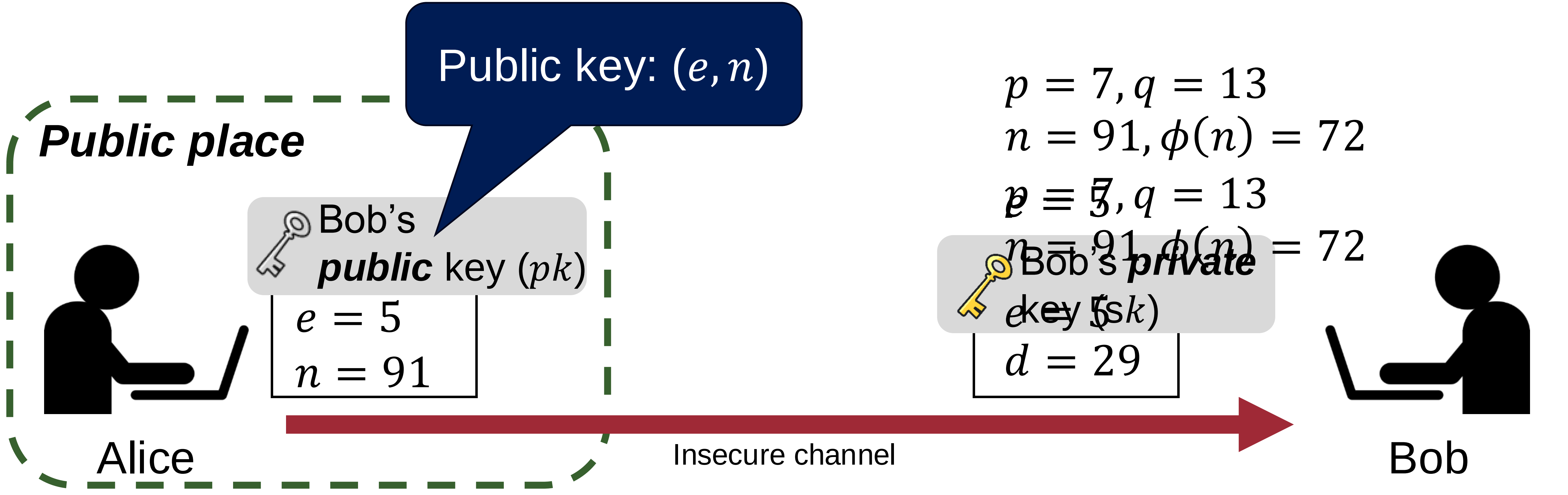
Insecure channel

Bob

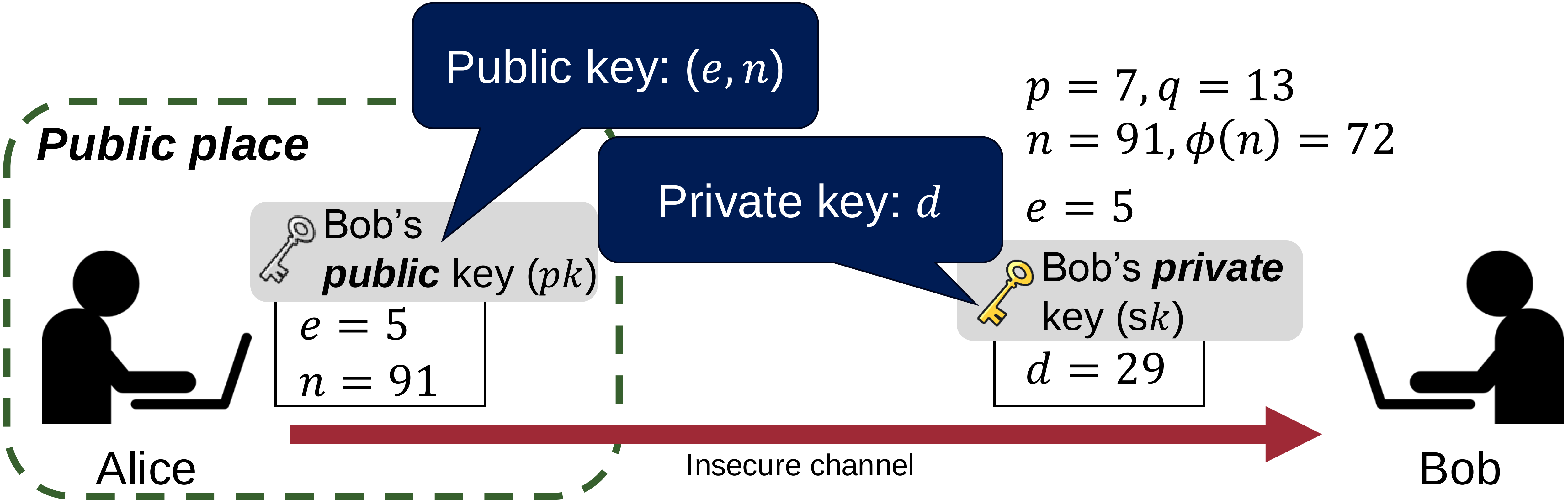
RSA Algorithm (1): Key Generation



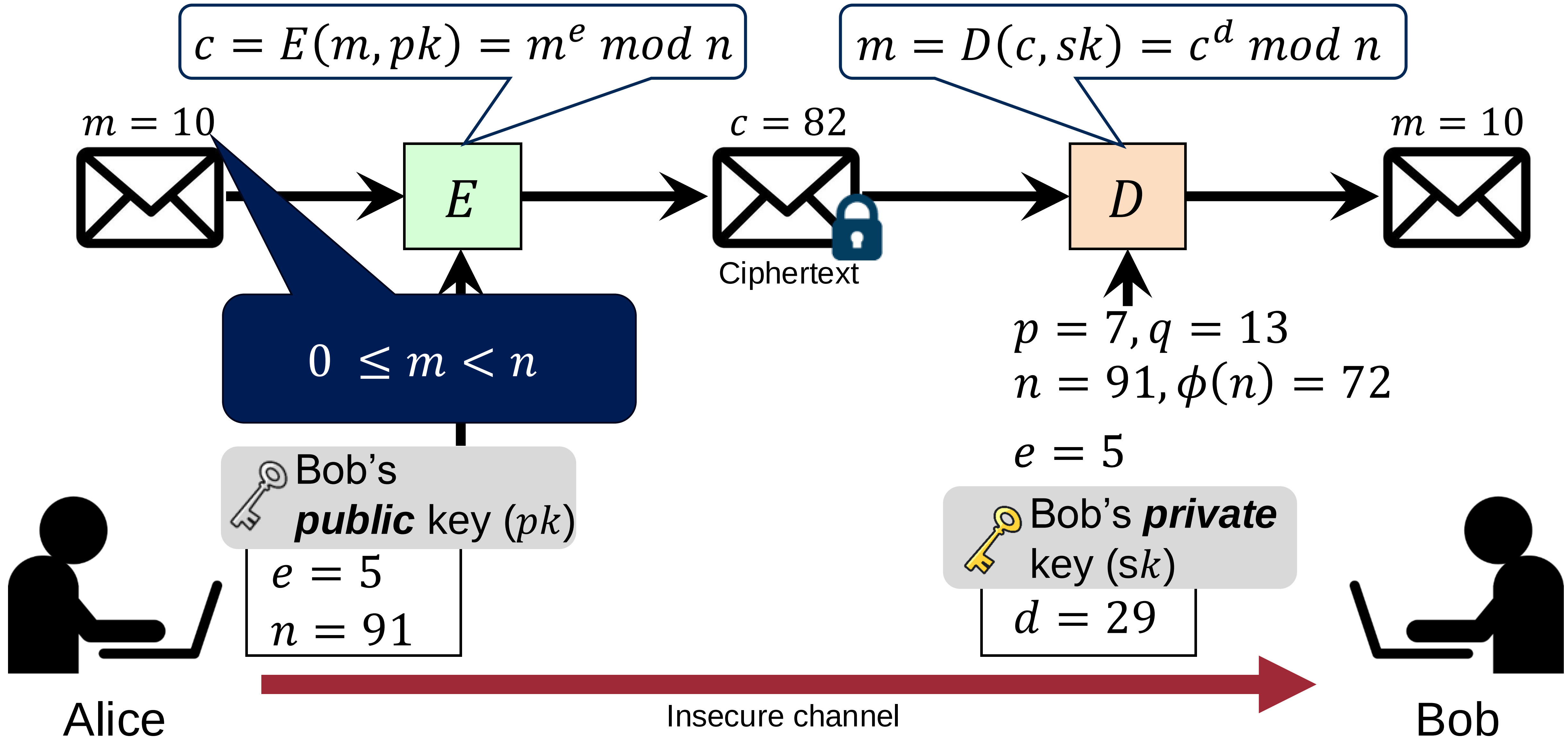
RSA Algorithm (1): Key Generation



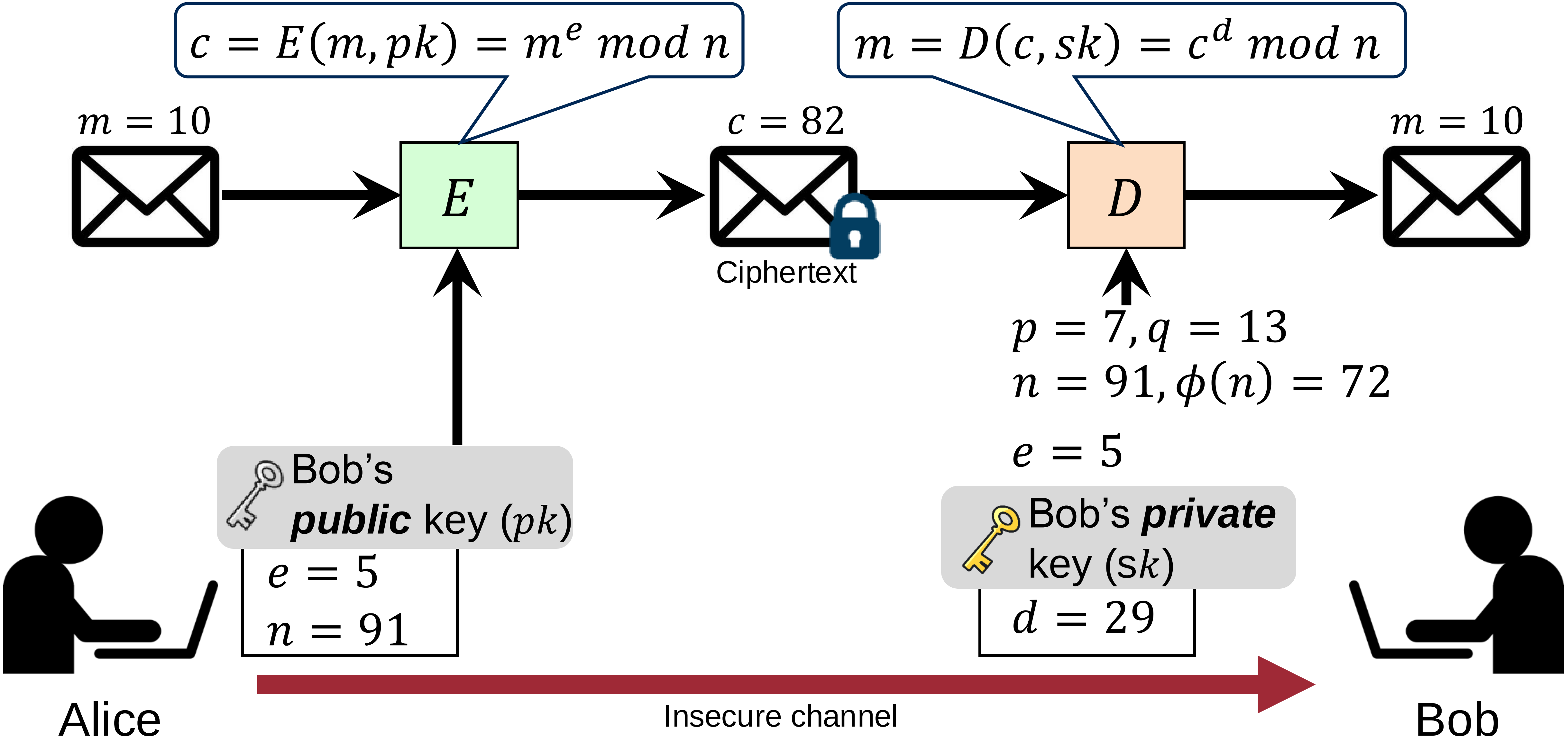
RSA Algorithm (1): Key Generation



RSA Algorithm (2): Encryption and Decryption



RSA Algorithm (2): Encryption and Decryption



Correctness of the RSA Algorithm

$c = E(m, pk) = m^e \bmod n$

$m = D(c, sk) = c^d \bmod n$

Correctness: $m = (m^e \bmod n)^d \bmod n$

$= m^{ed} \bmod n$

Theorem:

$((X \bmod p)^k \bmod p) = (X^k \bmod p)$

Correctness of the RSA Algorithm

$\boxed{c = E(m, pk) = m^e \bmod n}$ $\boxed{m = D(c, sk) = c^d \bmod n}$

Theorem:

$((X \bmod p)^k \bmod p) = (X^k \bmod p)$

Correctness: $m = (m^e \bmod n)^d \bmod n$

$= m^{ed} \bmod n$

$= m^{1+k\cdot\phi(n)} \bmod n$

We choose d s.t. $(ed \bmod \phi(n)) = 1$

Correctness of the RSA Algorithm

$$c = E(m, pk) = m^e \mod n$$

$$m = D(c, sk) = c^d \mod n$$

Correctness: $m = (m^e \mod n)^d \mod n$

$= m^{ed} \mod n$

$= m^{1+k\cdot\phi(n)} \mod n$

$= m \cdot \left(m^{\phi(n)}\right)^k \mod n$

$= m \mod n$

$= m$

Theorem:

$((X \mod p)^k \mod p) = (X^k \mod p)$

We choose d s.t. $(ed \mod \phi(n)) = 1$

Euler's Theorem:

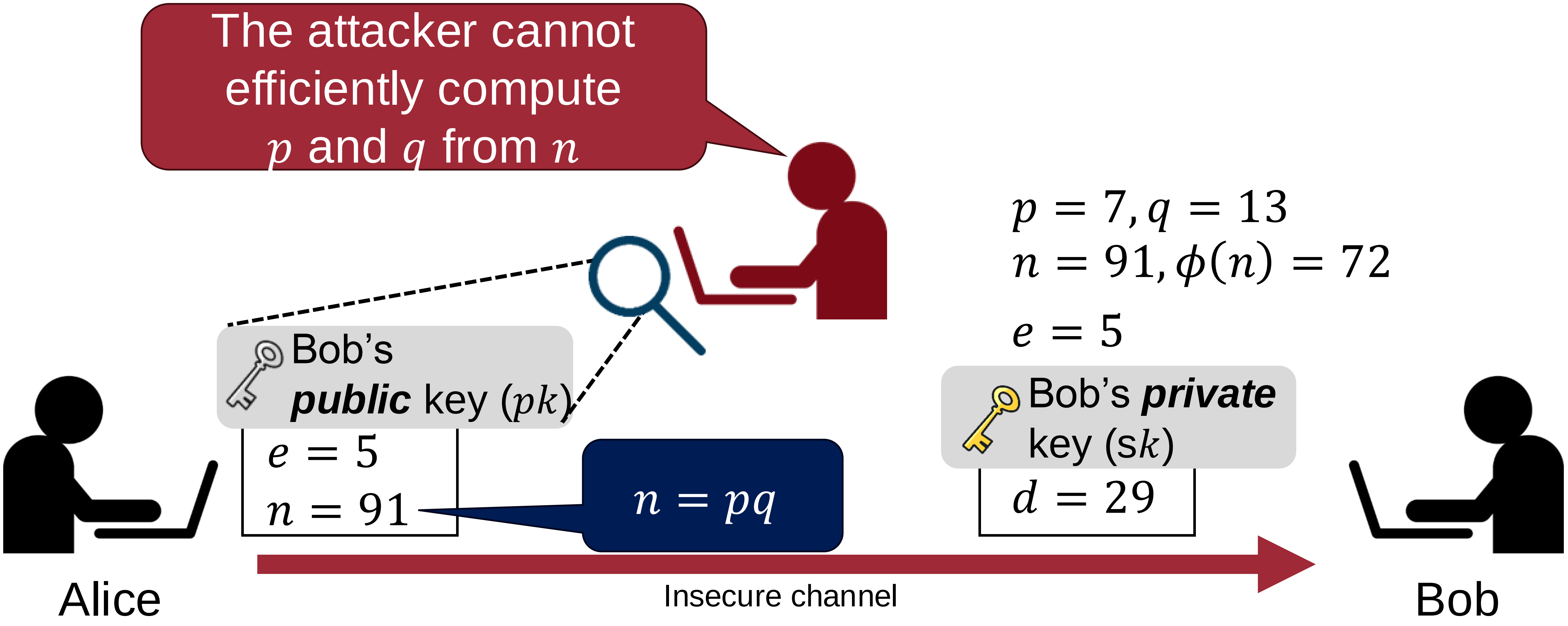
$\left(X^{\phi(n)} \mod n\right) = 1$ where $\gcd(X, n) = 1$

Also, refer to *Fermat's little theorem* ☺

Security of the RSA Algorithm

$$c = E(m, pk) = m^e \bmod n$$

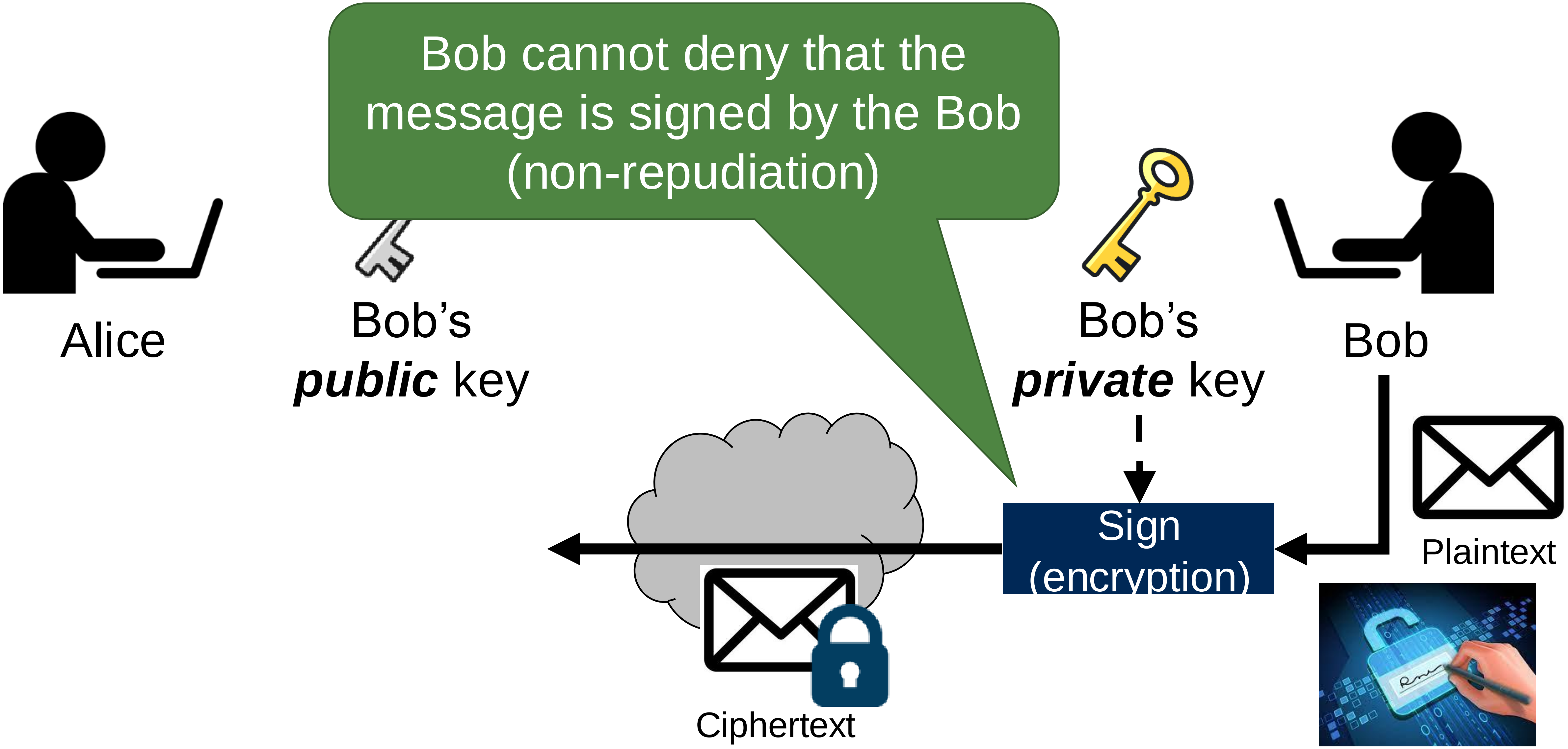
$$m = D(c, sk) = c^d \bmod n$$



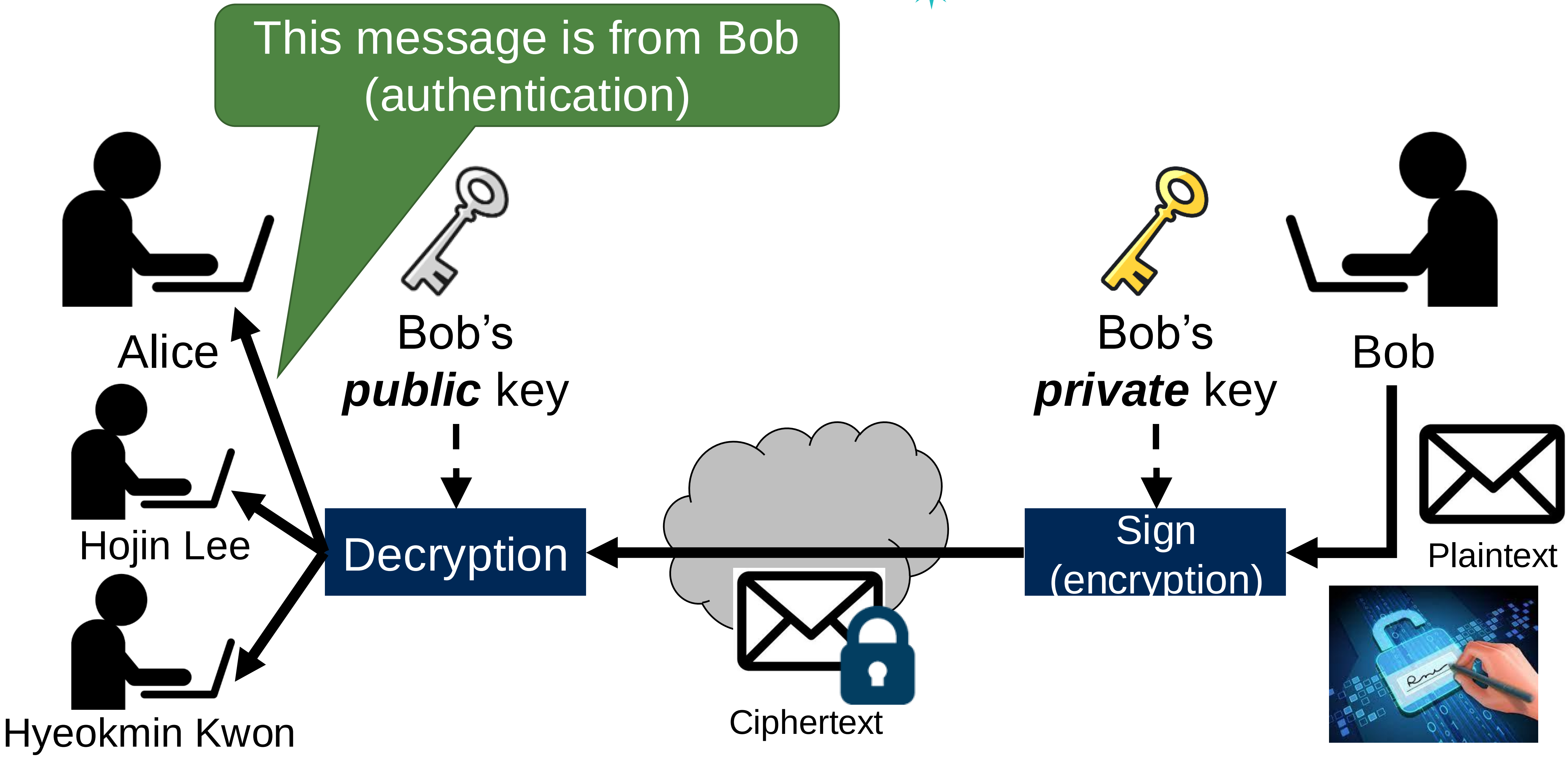
Comparison with Symmetric-Key Cryptography

- Pros
  - No need to share a secret
  - Enable multiple senders to communicate privately with a single receiver
  - More applications: Digital sign

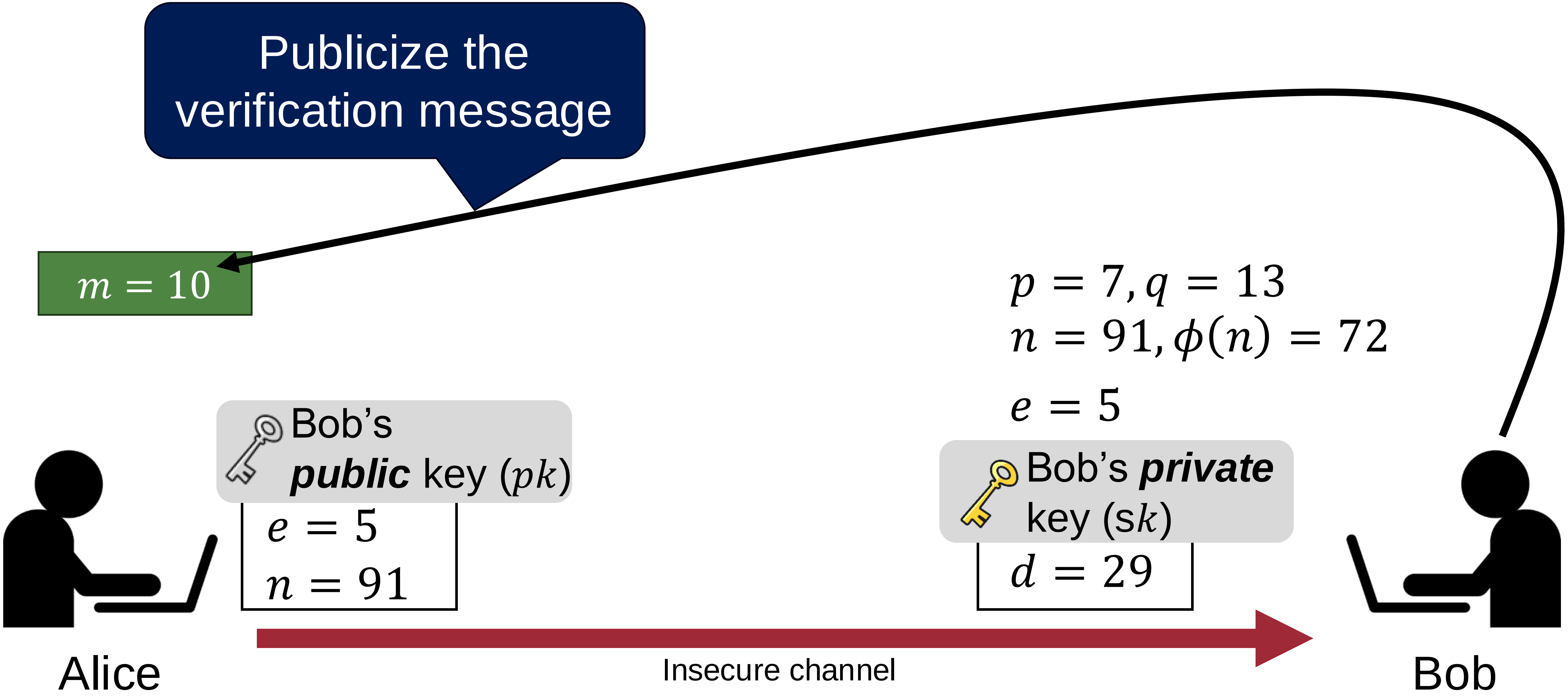
Digital Signature



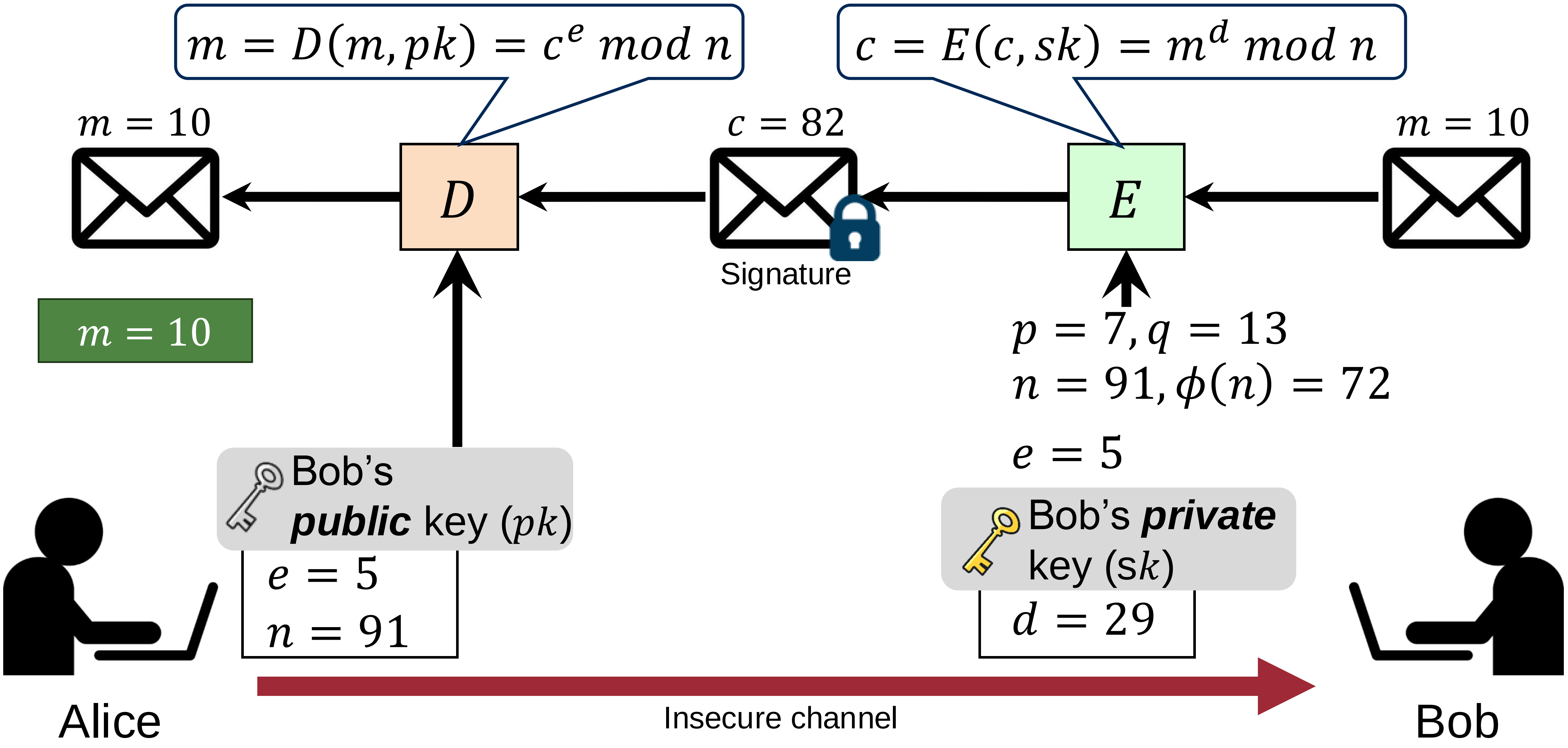
Digital Signature



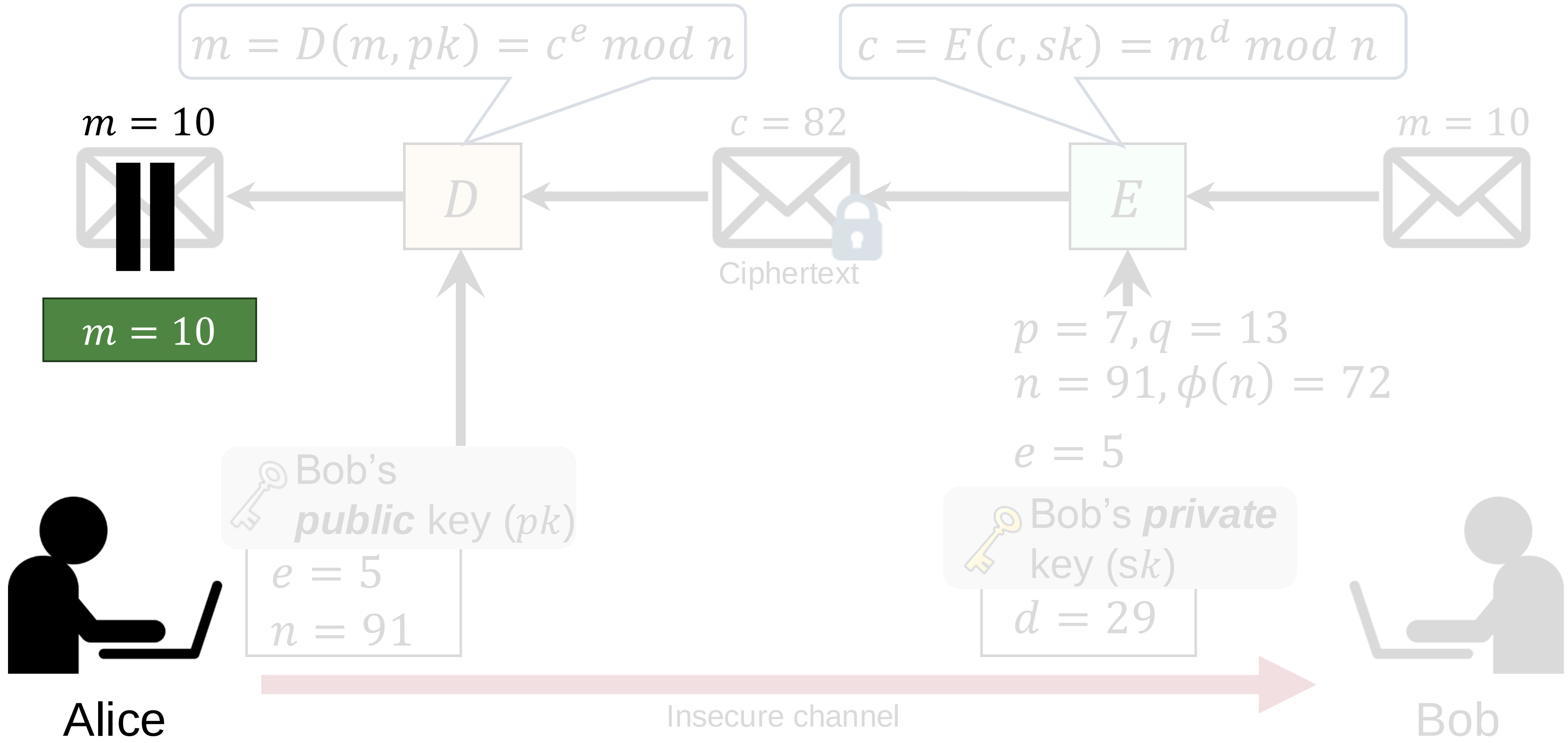
Digital Signature in Detail (1)



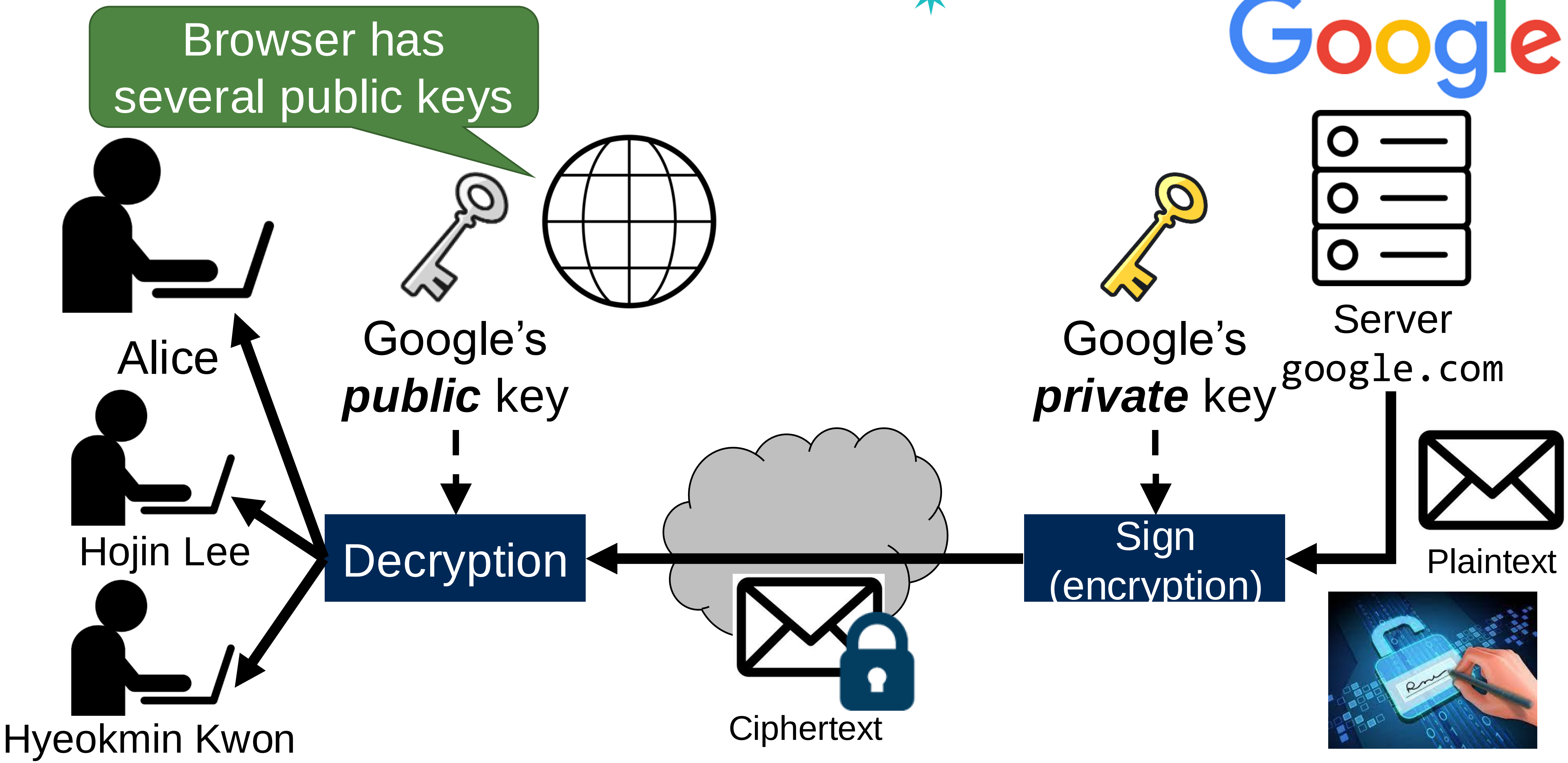
Digital Signature in Detail (2)



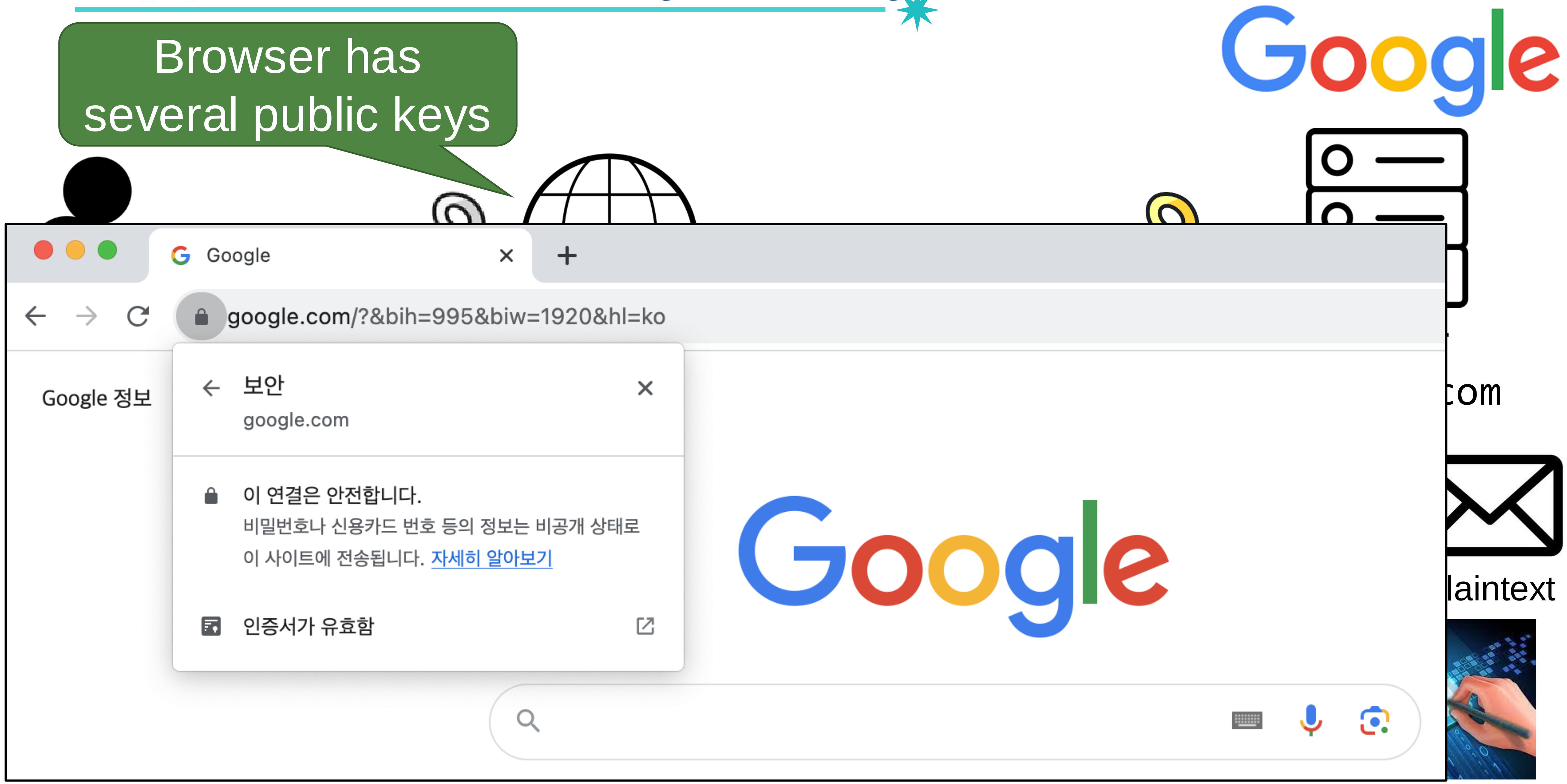
Digital Signature in Detail (3)



Application of Digital Signature in HTTPs



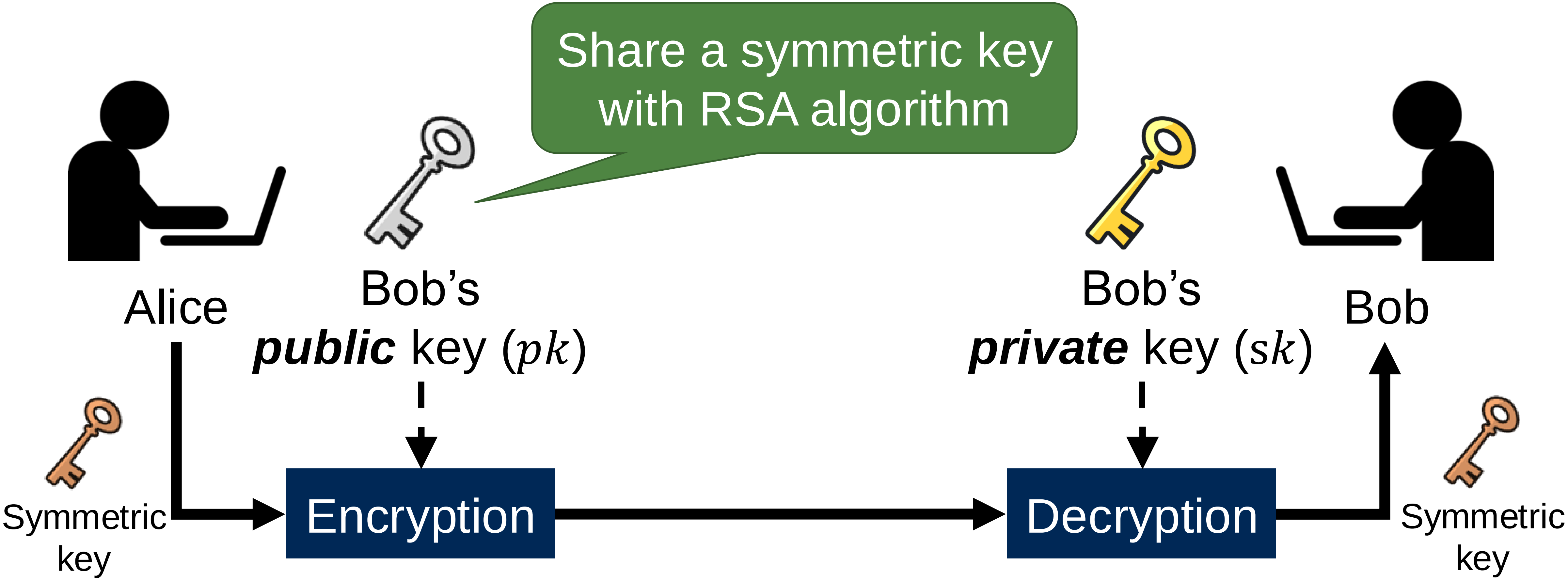
Application of Digital Signature in HTTPs



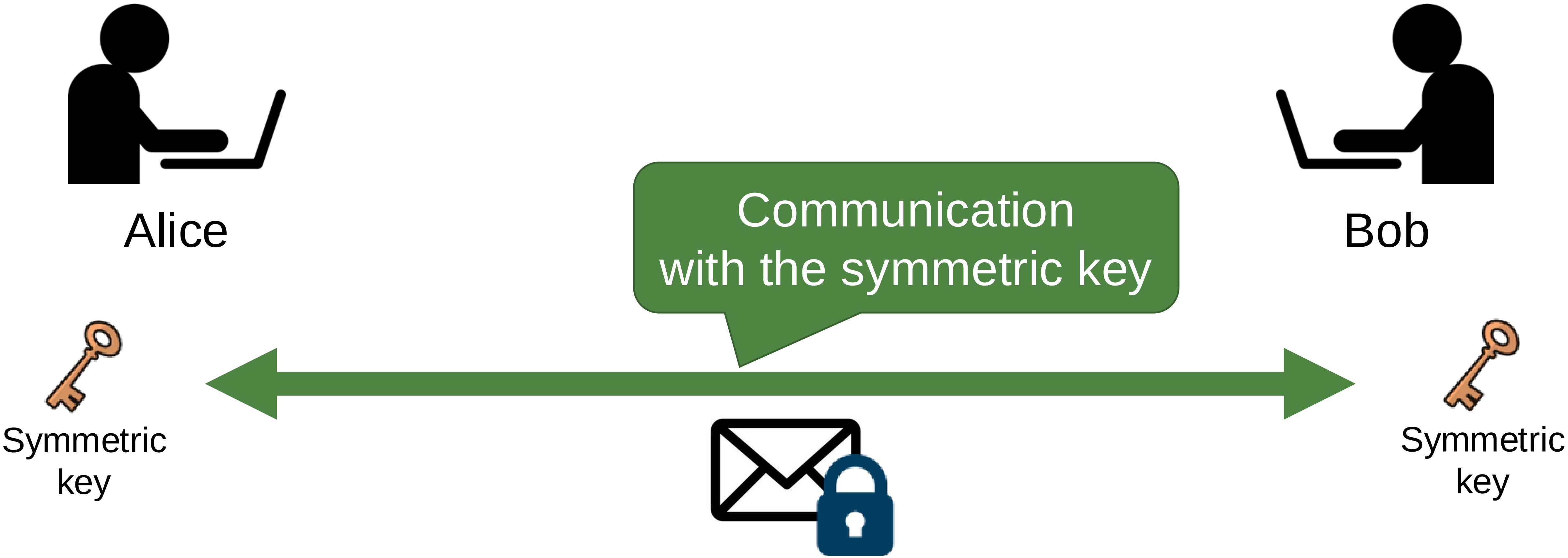
Comparison with Symmetric-Key Cryptography

- Pros
  - No need to share a secret
  - Enable multiple senders to communicate privately with a single receiver
  - More applications: Digital sign
- Cons
  - Slower in general: due to the larger key
    - Roughly 2-3 orders of magnitude slower

In Practice: Combination of Two Schemes



In Practice: Combination of Two Schemes



Summary

- Public-key revolution: solve key distribution and maintenance problem
  - Diffie-Hellman key exchange
  - Public-key encryption
  - Digital signature

- (Next lecture) Public key infrastructure, hash, MAC, and homomorphic encryption

Question?