CSCI 1515 Applied Cryptography

This Lecture:

- RSA Encryption (Continued)
- ElGamal Encryption
- Diffie-Hellman Key Exchange
- Message Integrity
- Message Authentication Code (MAC)
- Digital Signature
- RSA & DSA Signature Schemes

RSA Encryption

- $\text{Gen}(1^\lambda)$: $n = \Theta(\lambda)$ $n = 1024$, key length 2048

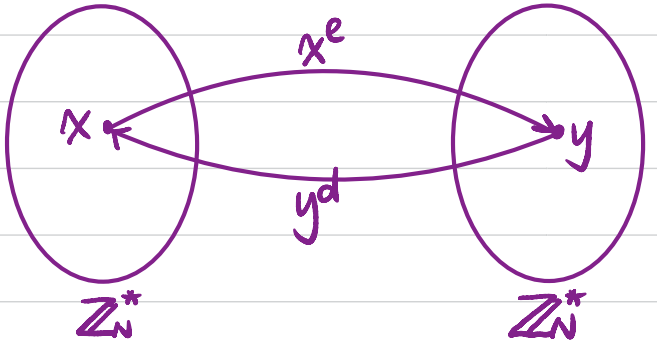
  Generate two n-bit primes p, q ← How?

  Compute $N = p \cdot q$, $\phi(N) = (p-1)(q-1)$

  Choose e s.t. $\gcd(e, \phi(N)) = 1$ ← How?

  Compute $d = e^{-1} \bmod \phi(N)$ ← How?

  $pk = (N, e)$ $\quad sk = d$.



- $\text{Enc}_{pk}(m)$: $c = m^e \bmod N$ ← How (efficiently)?
- $\text{Enc}_{sk}(c)$: $m = c^d \bmod N$ ← How (efficiently)?

Correctness: $(m^e)^d = m^{ed} \equiv m \bmod N$

$e \cdot d \equiv 1 \bmod \phi(N)$

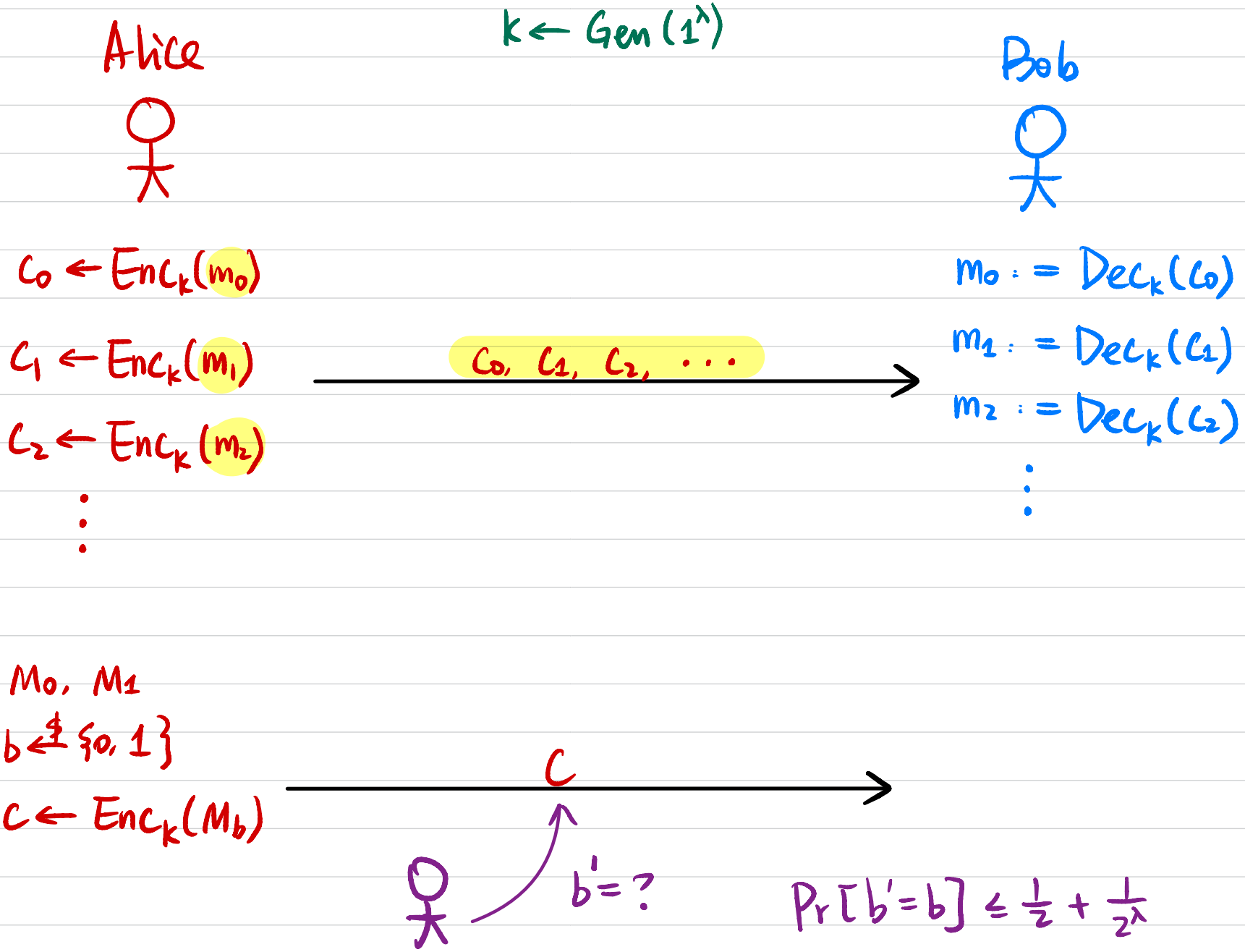
$m^{\phi(N)} \equiv 1 \bmod N$

$e \cdot d = \phi(N) \cdot k + 1$

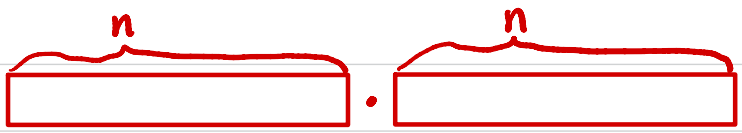
$m^{ed} = m^{\phi(N) \cdot k + 1} \equiv 1^k \cdot m \equiv m \pmod N$

Any security issue?

Chosen-Plaintext Attack (CPA) Security



RSA Assumption



- **Factoring Assumption:**

  Generate two n-bit primes p, q

  Compute $N = p \cdot q$

  Given N, it's computationally hard to find p & q (classically).

- **RSA Assumption:**

  Generate two n-bit primes p, q

  Compute $N = p \cdot q$, $\phi(N) = (p-1)(q-1)$

  Choose e s.t. $\gcd(e, \phi(N)) = 1$

  Compute $d = e^{-1} \bmod \phi(N)$

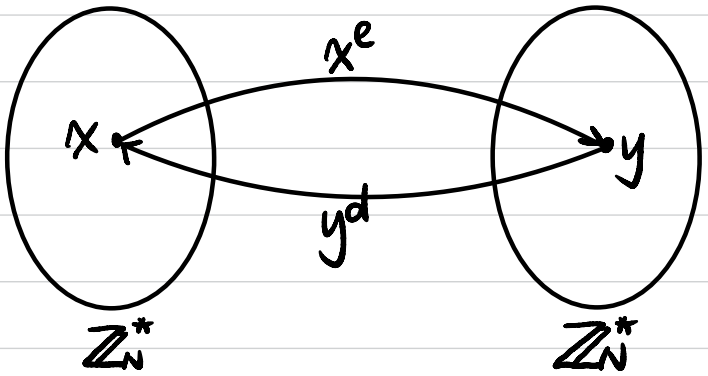
  Given N & a random $y \xleftarrow{\$} \mathbb{Z}_N^*$, it's computationally hard to find x s.t.

  $$x^e \equiv y \pmod N$$

If one can solve factoring

⇓

Can break RSA



$m \in \mathbb{Z}_N^*$?

What if $p \mid m$ (or $q \mid m$) ?

Correctness still holds.

Security: $p \mid (m^e \bmod N)$

$\Rightarrow \gcd(c, N) = p \Rightarrow$ break factoring!

$m \xleftarrow{\$} [1, N-1]$:

$$\Pr[p \mid m] \approx \frac{q}{N} = \frac{1}{p} = \frac{1}{2^{\theta(\lambda)}} = \text{negligible}(\lambda)$$

(p: $\theta(\lambda)$-bit)

If sending k messages:

$$\Pr[p \mid m \text{ for any } m] \leq \frac{k}{p}$$

($k \leftarrow \text{poly}(\lambda)$, $p \leftarrow 2^{\theta(\lambda)}$)

Group Theory

Def A group is a set G along with a binary operation $\circ$ with properties:

① **Closure**: $\forall g, h \in G, \quad g \circ h \in G$

② **Existence of an identity**: $\exists e \in G$ s.t. $\forall g \in G, \quad e \circ g = g \circ e = g$.

③ **Existence of inverse**: $\forall g \in G, \exists h \in G$ s.t. $g \circ h = h \circ g = e$

Inverse of g denoted as g^{-1}.

④ **Associativity**: $\forall g_1, g_2, g_3 \in G, \quad (g_1 \circ g_2) \circ g_3 = g_1 \circ (g_2 \circ g_3)$

We say a group is **abelian** if it satisfies:

⑤ **Commutativity**: $\forall g, h \in G, \quad g \circ h = h \circ g$

For a finite group, we use $|G|$ to denote its **order** (# of elements)

Group Theory

Ex. $(\mathbb{Z}, +)$ is an abelian group

$(\mathbb{Z}, \cdot)$ is not a group

$(\mathbb{Z}_N^*, \cdot)$ is an abelian group ($\cdot$ denotes multiplication mod N)

Def Let G be a group of order m.

Denote $\langle g \rangle = \{g^0, g^1, g^2, \cdots, g^{m-1}\}$

G is a cyclic group if $\exists g \in G$ s.t. $\langle g \rangle = G$.

g is a generator of G.

Ex. $\mathbb{Z}_p^*$ (for a prime p) is a cyclic group of order $p-1$.

$\mathbb{Z}_7^* = \{3^0 = 1, 3^1, 3^2 = 2, 3^3 = 6, 3^4 = 4, 3^5 = 5\}$

How to find a generator?

Diffie-Hellman Assumptions

Integer group key 2048-bit
Elliptic Curve group key 256-bit

$(G, q, g) \leftarrow G(1^\lambda)$ ($\Theta(\lambda)$-bit integer)

cyclic group G of order q, with generator g.

- **Discrete Logarithm (DLOG) Assumption:**

  $x \xleftarrow{\$} \mathbb{Z}_q$, compute $h = g^x$ $\qquad g^x \overset{?}{\Rightarrow} x$

  Given (G, q, g, h), it's computationally hard to find x (classically).

- **Computational Diffie-Hellman (CDH) Assumption:**

  $x, y \xleftarrow{\$} \mathbb{Z}_q$, compute $h_1 = g^x$, $h_2 = g^y$ $\qquad (g^x, g^y) \overset{?}{\Rightarrow} g^{xy}$

  Given (G, q, g, h_1, h_2), it's computationally hard to find g^{xy}.

- **Dicisional Diffie-Hellman (DDH) Assumption:**

  $x, y, z \xleftarrow{\$} \mathbb{Z}_q$, compute $h_1 = g^x$, $h_2 = g^y$

  Given (G, q, g, h_1, h_2), it's computationally hard to distinguish between

  $(g^x, g^y, g^{xy}) \overset{c}{\simeq} (g^x, g^y, g^z)$ $\qquad g^{xy}$ and g^z.

ElGamal Encryption

- $Gen(1^\lambda)$:

  $(G, q, g) \leftarrow \mathcal{G}(1^\lambda)$ ← can be re-used

  $x \xleftarrow{\$} \mathbb{Z}_q$, compute $h = g^x$

  $pk = (G, q, g, h)$ $\quad sk = x$

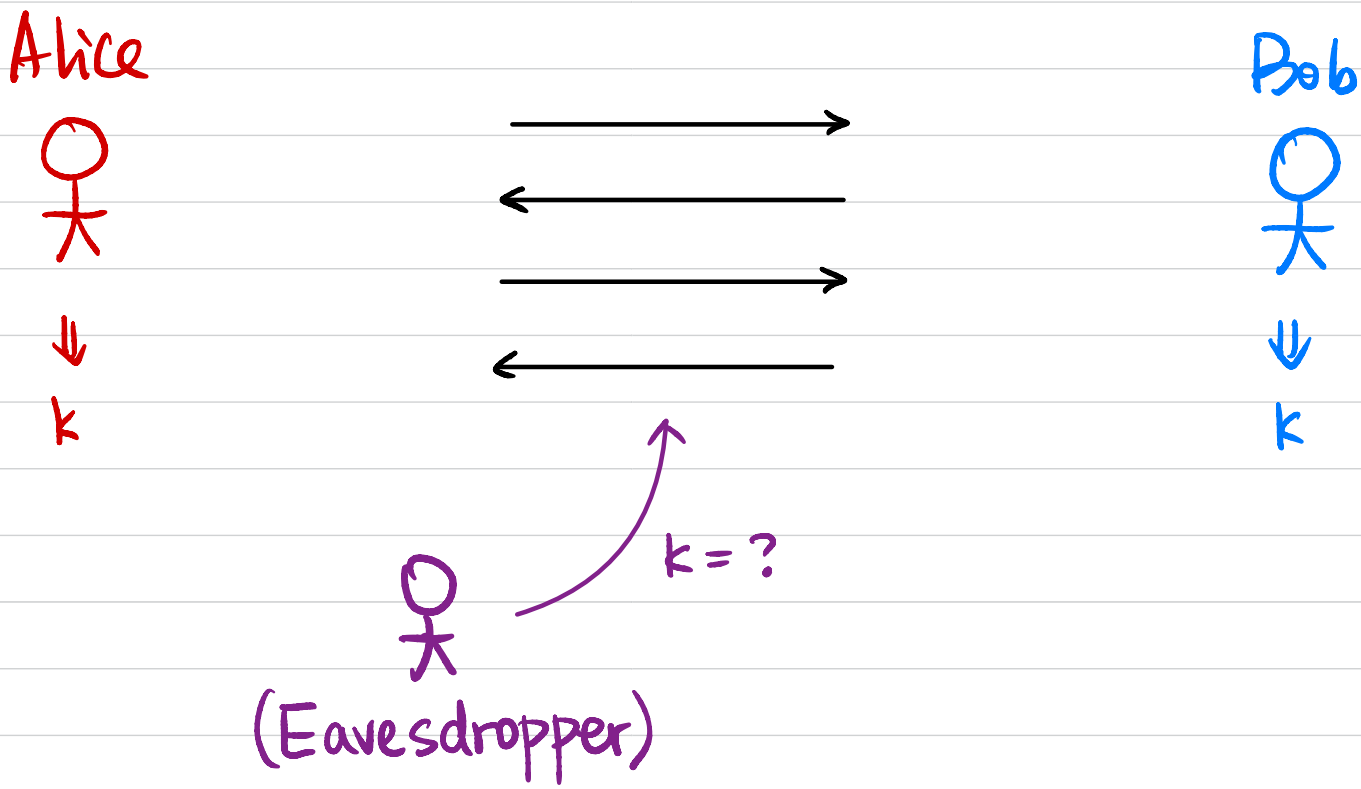
- $Enc_{pk}(m)$: $m \in G$

  $y \xleftarrow{\$} \mathbb{Z}_q$

  $c = \langle g^y, h^y \cdot m \rangle$

- $Enc_{sk}(c)$:

Secure Key Exchange



Thm (Informal): It's impossible to construct secure key exchang from SKE in a black-box way.

Key Exchange from PKE?

Diffie-Hellman Key Exchange

$(G, q, g) \leftarrow G(1^\lambda)$

$x \xleftarrow{\$} \mathbb{Z}_q$, compute $h_A = g^x$

(G, q, g, h_A) →

$y \xleftarrow{\$} \mathbb{Z}_q$, compute $h_B = g^y$

← h_B

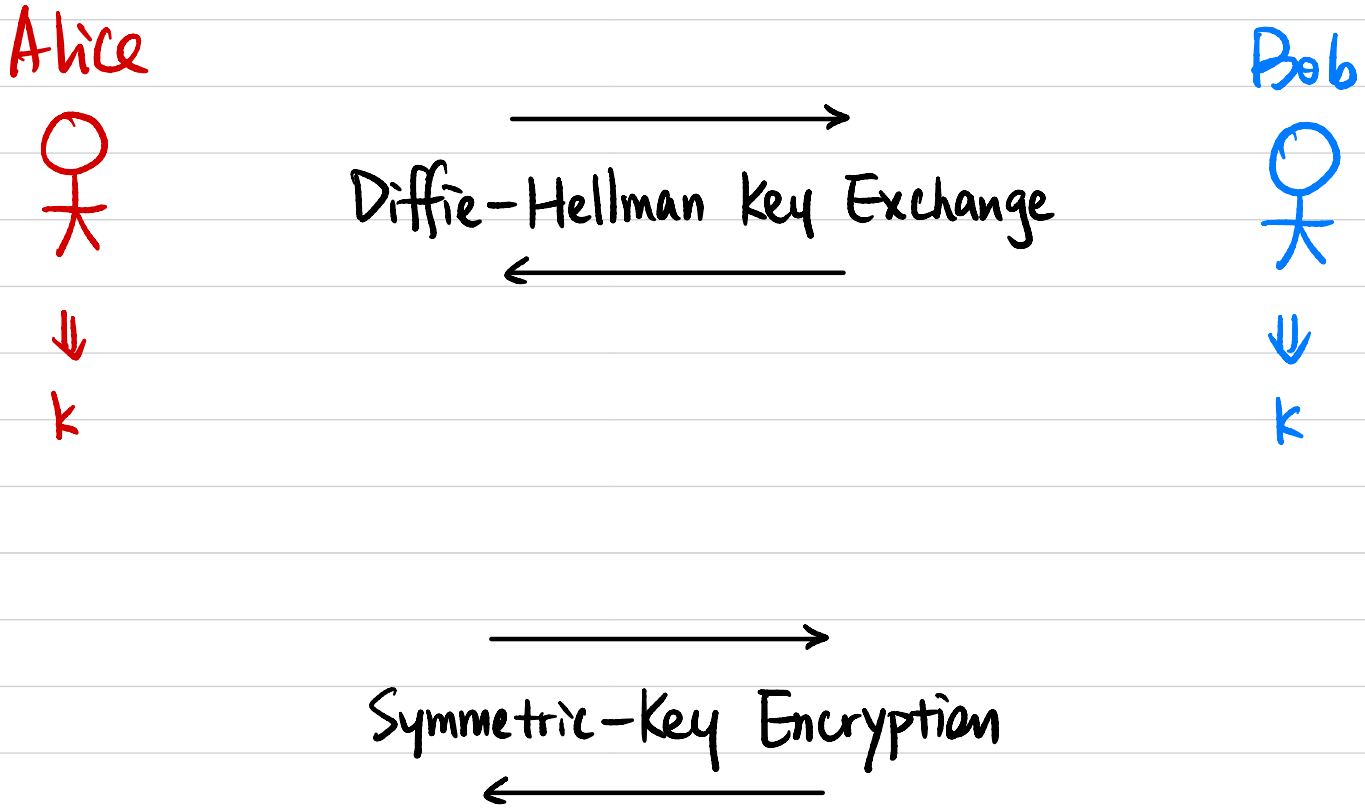
$k = h_B^x$

$k = h_A^y$

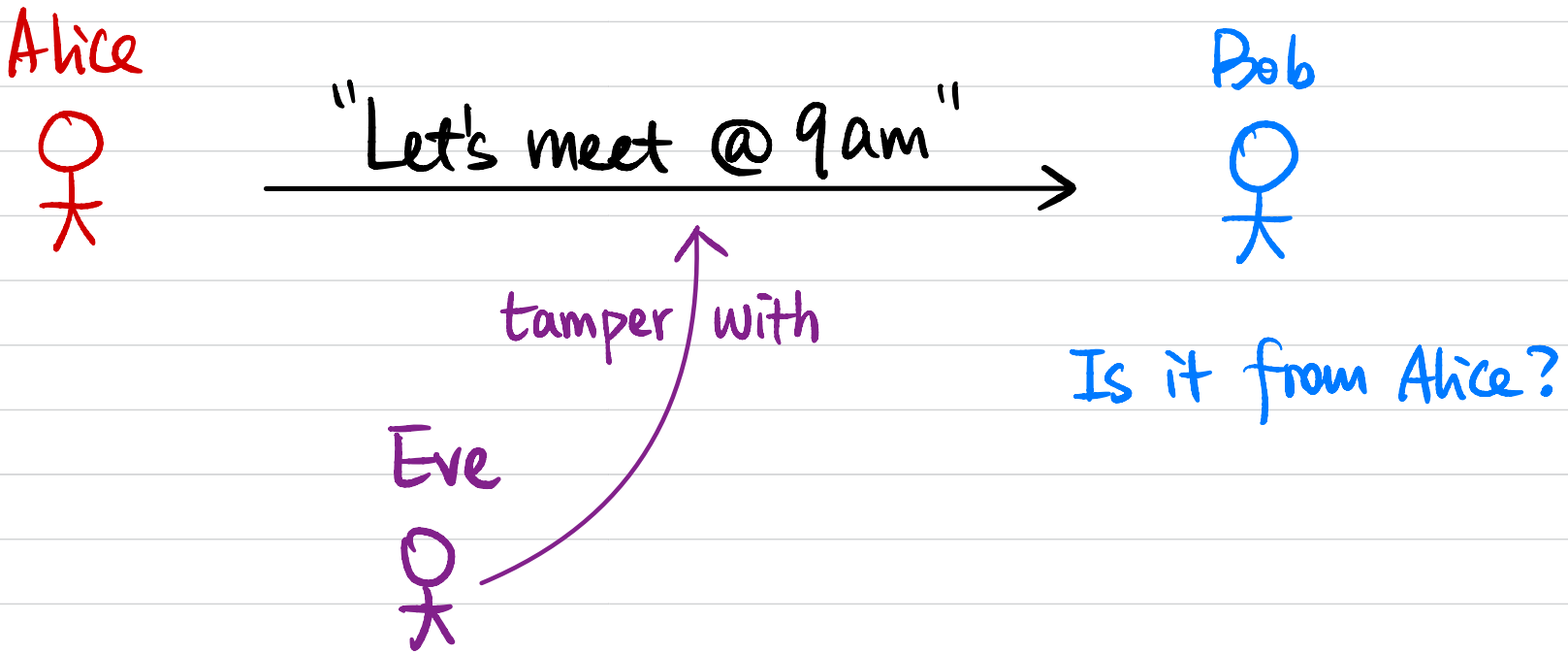
$k = ?$

(Eavesdropper)

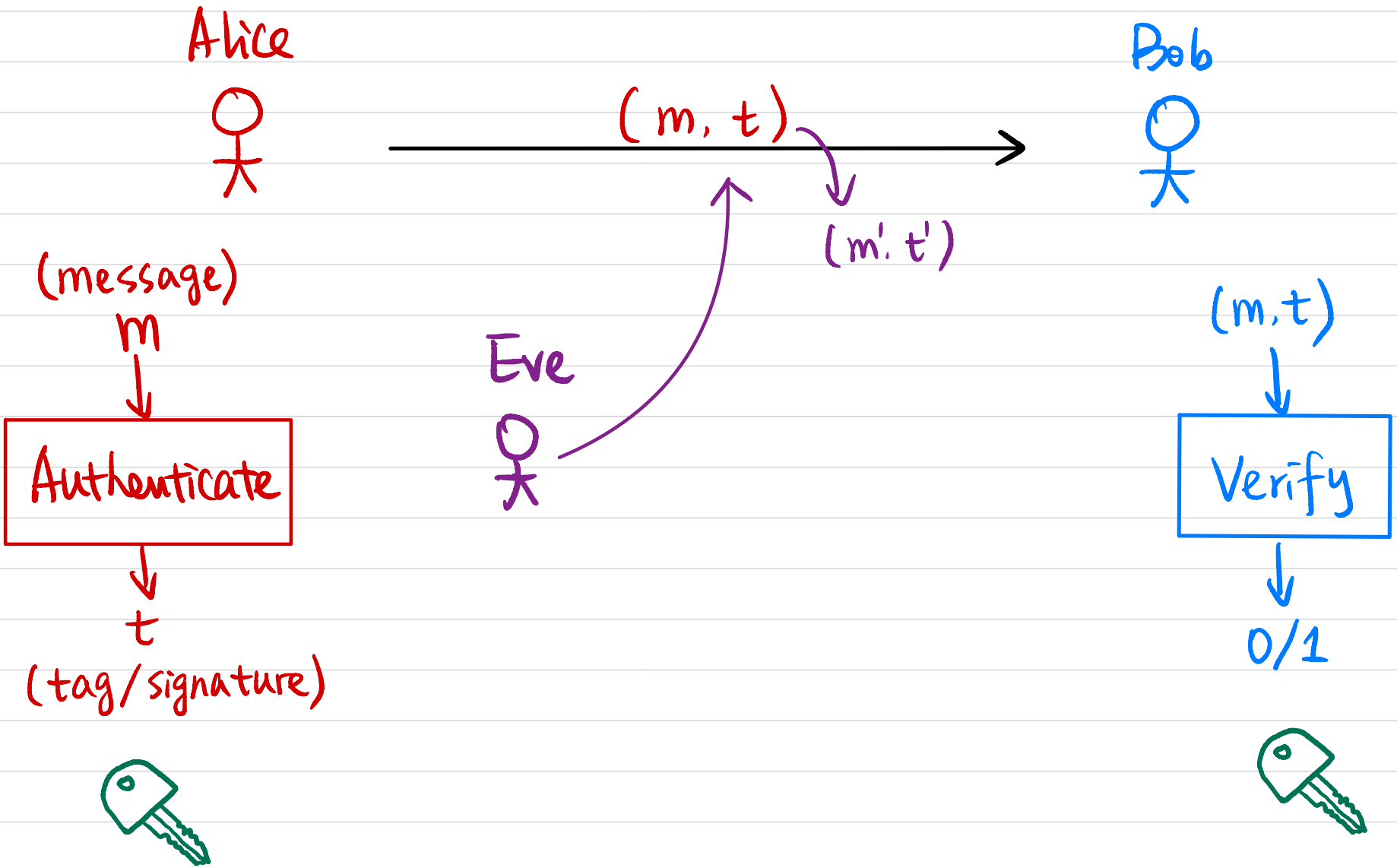
What happens in practice



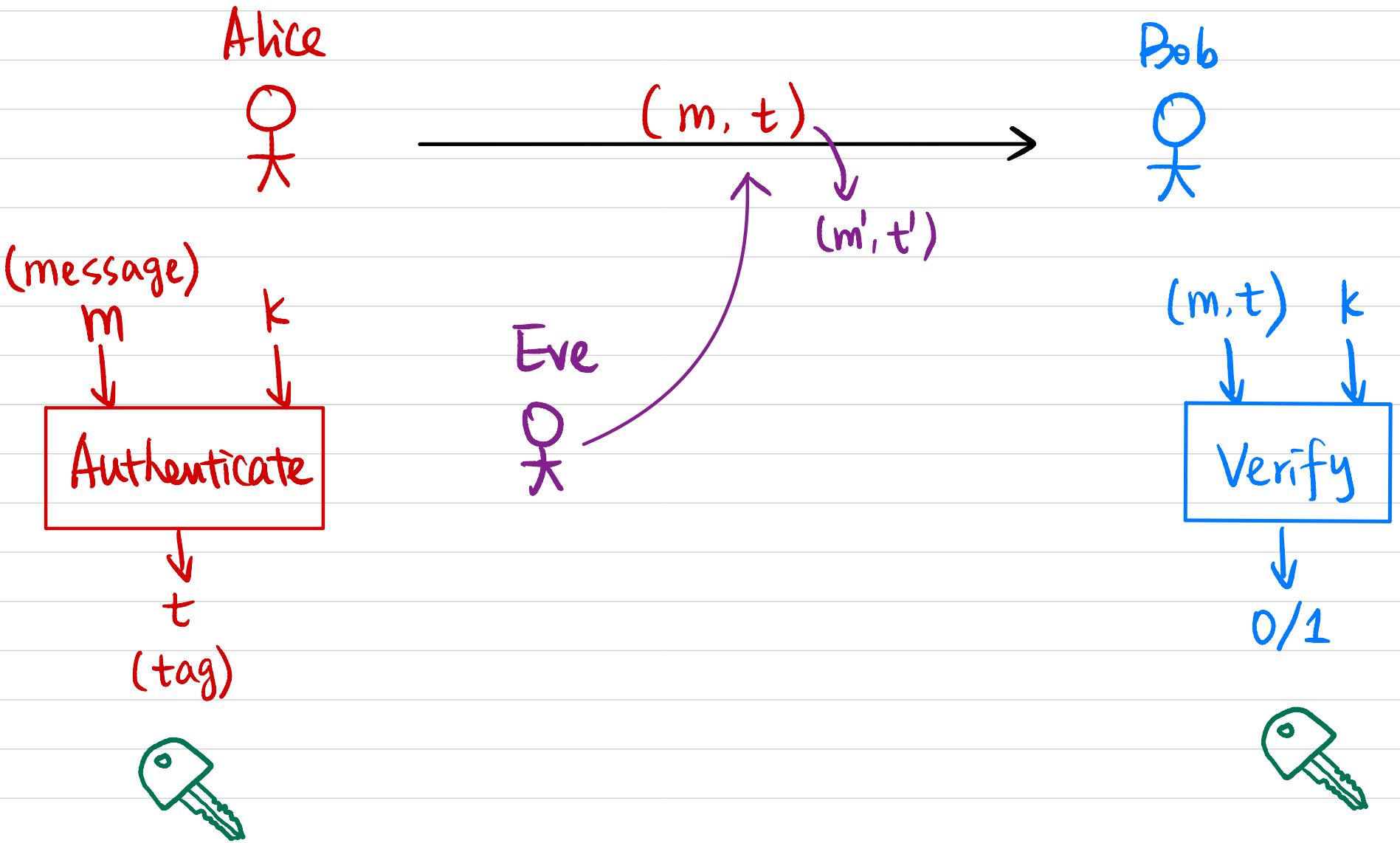
Message Integrity



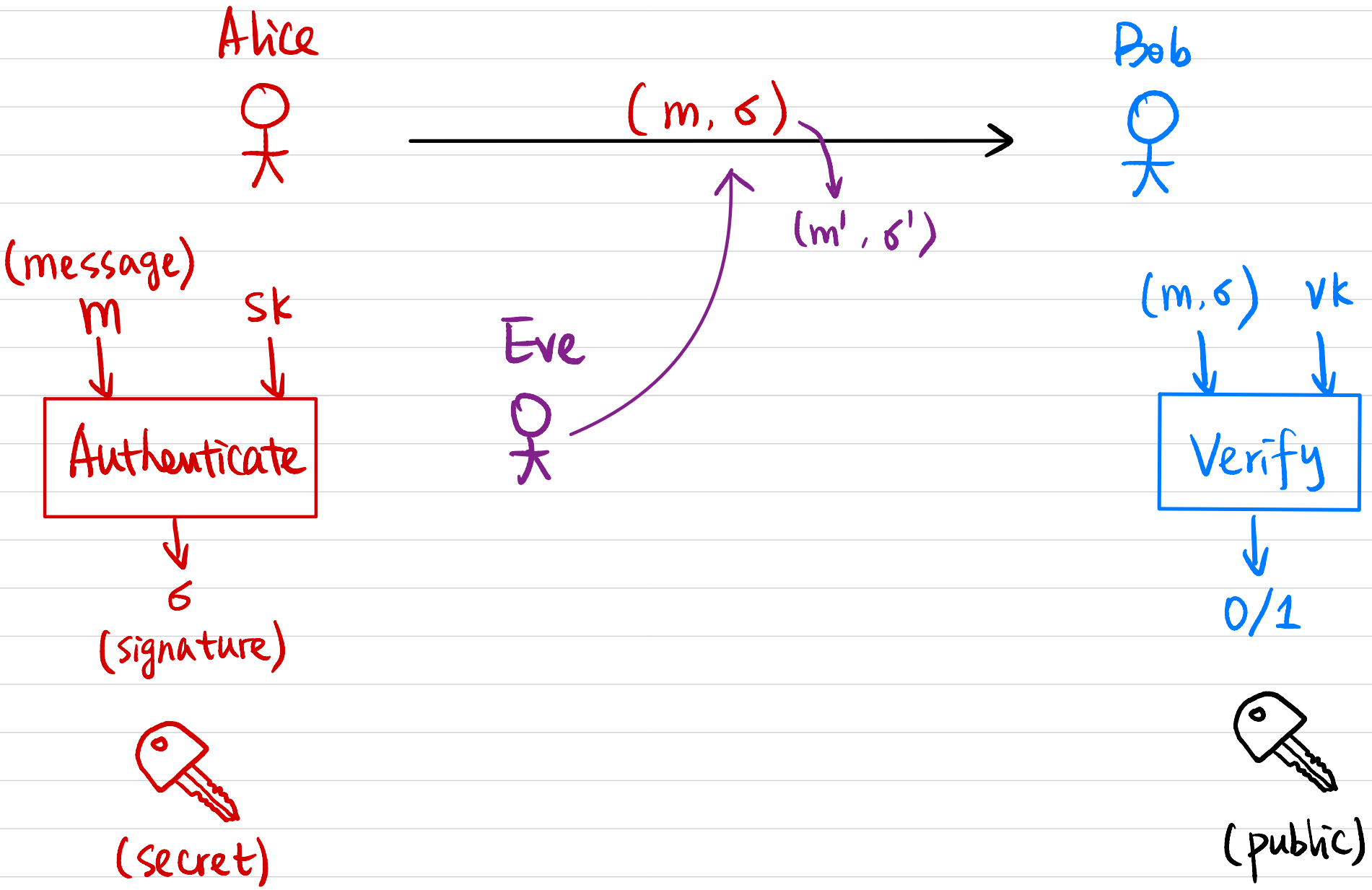
Message Integrity



Message Authentication Code (MAC)



Digital Signature



Syntax

Message Authentication Code (MAC) Scheme $\Pi = (Gen, Mac, Vrfy)$

$k \leftarrow Gen(1^\lambda)$

$t \leftarrow Mac_k(m)$

$0/1 := Vrfy_k(m, t)$

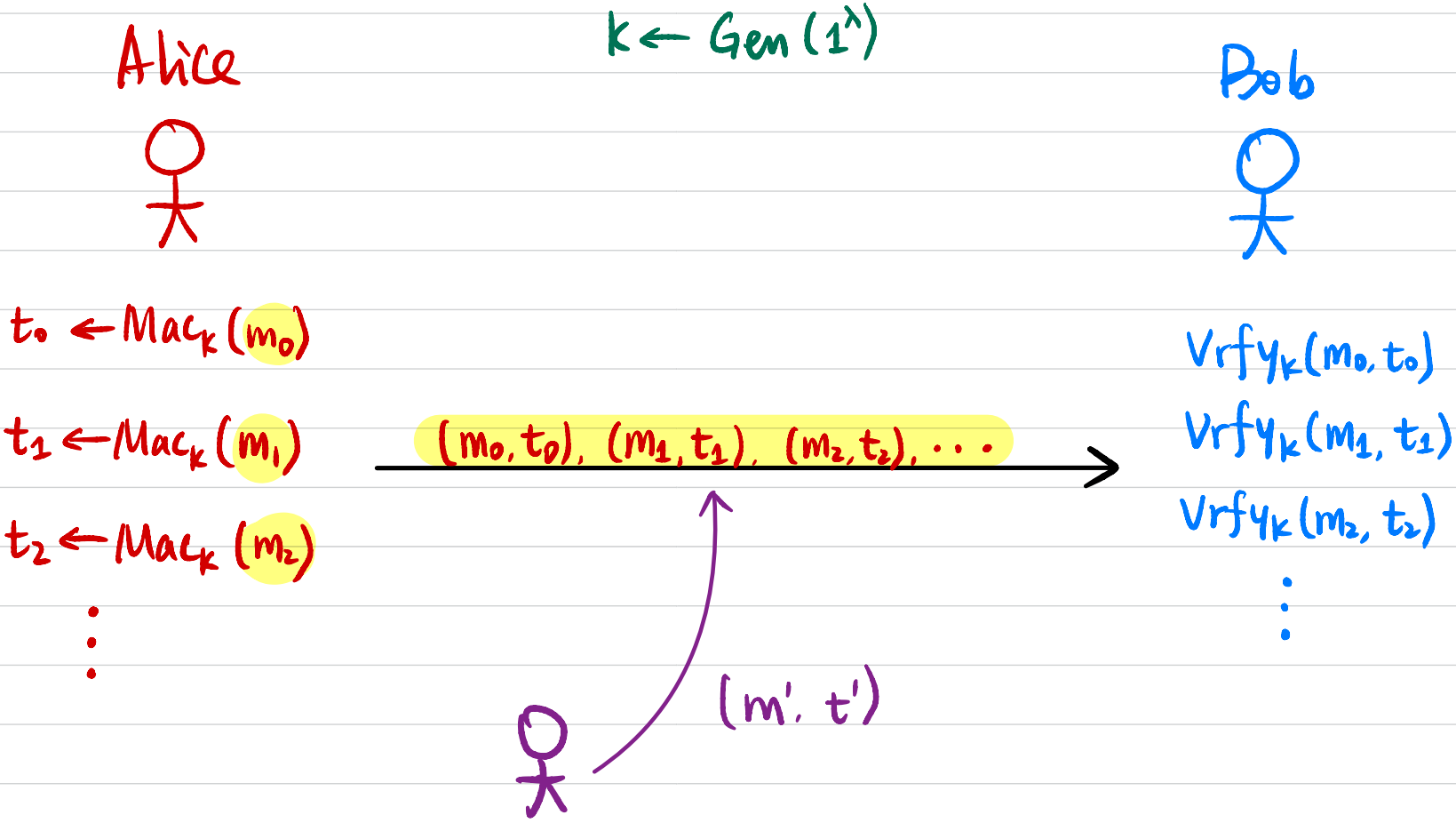
Digital Signature Scheme $\Pi = (Gen, Sign, Vrfy)$

$(sk, vk) \leftarrow Gen(1^\lambda)$

$\sigma \leftarrow Sign_{sk}(m)$

$0/1 := Vrfy_{vk}(m, \sigma)$

Chosen-Message Attack (CMA)



Constructions for MAC

From block cipher: CBC-MAC

From hash function: HMAC

Computational Assumption: "The construction is secure"

Constructions for Digital Signature

RSA Signature : RSA Assumption

DSA Signature : Discrete Logarithm Assumption

Lattice-Based Encryption Schemes (Post-Quantum Security)

RSA Signature

Generate two n-bit primes p, q

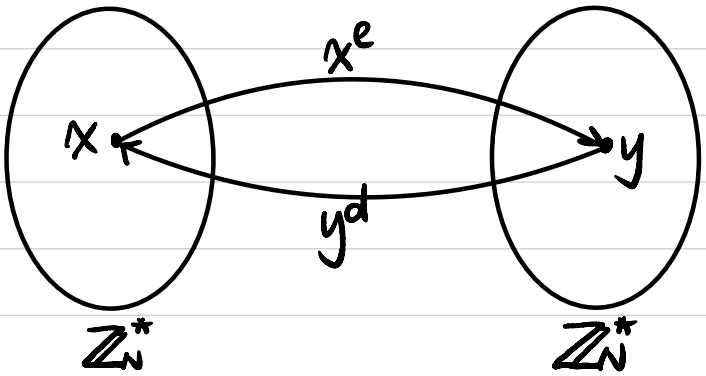
Compute $N = p \cdot q$, $\phi(N) = (p-1)(q-1)$

Choose e s.t. $\gcd(e, \phi(N)) = 1$

Compute $d = e^{-1} \bmod \phi(N)$

Given N & a random $y \xleftarrow{\$} \mathbb{Z}_N^*$, it's computationally hard to find x s.t.

$$x^e \equiv y \pmod N$$



$sk = d$ $\quad$ $vk = (N, e)$

$Sign_{sk}(m) = m^d \bmod N$

$Vrfy_{vk}(m, \sigma)$: $\sigma^e \stackrel{?}{\equiv} m \pmod N$

Any security issue ?

DSA Signature

- $\text{Gen}(1^\lambda)$:

  Generate an n-bit prime p & m-bit prime q s.t. $q \mid (p-1)$ (2048, 256)

  Pick a random $\alpha \in \mathbb{Z}_p^*$ s.t. $g = \alpha^{\frac{p-1}{q}} \bmod p \neq 1$ $\Rightarrow$ $\langle g \rangle$ has order q.

  $x \xleftarrow{\$} \mathbb{Z}_q^*$, Compute $h = g^x \bmod p$

  $vk = (p, q, g, h)$, $sk = x$

- $\text{Sign}_{sk}(m)$:

  $y \xleftarrow{\$} \mathbb{Z}_q^*$, Compute $r = (g^y \bmod p) \bmod q$

  Compute $s = y^{-1} \cdot (H(m) + x \cdot r) \bmod q$ $\Rightarrow$ $y \equiv s^{-1} \cdot (H(m) + x \cdot r) \bmod q$

  $\sigma = (r, s)$

  $g^y \equiv g^{s^{-1} \cdot H(m)} \cdot g^{s^{-1} \cdot x \cdot r} \bmod p$

  $\equiv g^{u_1} \cdot h^{u_2} \bmod p$

- $\text{Vrfy}_{vk}(m, \sigma)$:

  Compute $w = s^{-1} \bmod q$, $u_1 = H(m) \cdot w \bmod q$

  Compute $u_2 = r \cdot w \bmod q$

  Verify if $r = (g^{u_1} \cdot h^{u_2} \bmod p) \bmod q$

Summary

	Symmetric-Key	Public-Key
Message Secrecy	Primitive: SKE Construction: block Cipher	Primitive: PKE Constructions: RSA / ElGamal
Message Integrity	Primitive: MAC Constructions: CBC-MAC / HMAC	Primitive: Signature Constructions: RSA / DSA
Key Exchange		Construction: Diffie-Hellman
Important Tool	Primitive: Hash function Construction: SHA	

What happens in practice