

CSCI 1515 Applied Cryptography

This Lecture:

- Message Authentication Code (MAC)
- RSA & DSA Signature Schemes
- Authenticated Encryption
- Collision-Resistant Hash Function (CRHF)
- Attacks & Applications of Hash Functions

ANNOUNCEMENT: Please enroll in So2 (online section) if you are taking the course remotely.

Message Authentication Code (MAC)

Alice



(message)

m

k

Authenticate



t

(tag)



(m, t)



(m', t')

Eve



Bob



(m, t)

k

Verify



0/1



Digital Signature

Alice



(message)

m

sk

Authenticate



σ

(signature)



(secret)

(m, σ)

(m', σ')

Eve



Bob



(m, σ)

vk

Verify



0/1



(public)

Syntax

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

$$k \leftarrow \text{Gen}(1^\lambda)$$

$$t \leftarrow \text{Mac}_k(m)$$

$$0/1 := \text{Vrfy}_k(m, t)$$

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

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

$$\sigma \leftarrow \text{Sign}_{sk}(m)$$

$$0/1 := \text{Vrfy}_{vk}(m, \sigma)$$

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

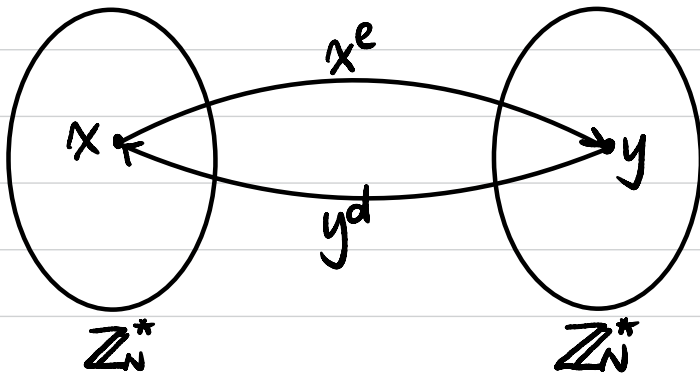
Generate two n -bit primes p, q

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?

Chosen-Message Attack (CMA)

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

Alice



$$s_0 \leftarrow \text{Sign}_{sk}(m_0)$$

$$s_1 \leftarrow \text{Sign}_{sk}(m_1)$$

$$s_2 \leftarrow \text{Sign}_{sk}(m_2)$$

⋮

Bob



$$\text{Vrfy}_{vk}(m_0, s_0)$$

$$\text{Vrfy}_{vk}(m_1, s_1)$$

$$\text{Vrfy}_{vk}(m_2, s_2)$$

⋮

$(m_0, s_0), (m_1, s_1), (m_2, s_2), \dots$



$(m^*, s^*) ?$

"unforgeable"

DSA Signature

- Gen(1^λ):

Generate an n -bit prime p & m -bit prime q st. $q | (p-1)$

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

$x \leftarrow \mathbb{Z}_q^*$, Compute $h = g^x \bmod p$ ($g^q \equiv \alpha^{p-1} \equiv 1 \bmod p$)

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

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

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

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

$\sigma = (r, s)$

$g^y \equiv g^{s^{-1} \cdot H(m)} \cdot g^{s^{-1} \cdot x \cdot r} \bmod p$
 $\equiv g^{u_1} \cdot h^{u_2} \bmod p$

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

What happens in practice

Alice



k

Diffie-Hellman Key Exchange

Bob



k

Symmetric-Key Encryption



Message Secrecy & Integrity?

Authenticated Encryption (AE)

$$k \leftarrow \text{Gen}(1^\lambda)$$

Alice



$$c_0 \leftarrow \text{Enc}_k(m_0)$$

$$c_1 \leftarrow \text{Enc}_k(m_1)$$

$$c_2 \leftarrow \text{Enc}_k(m_2)$$

$\vdots$

m_0, m_1

$$b \leftarrow \{0, 1\}$$

$$c \leftarrow \text{Enc}_k(m_b)$$

Bob



$$m_0 := \text{Dec}_k(c_0)$$

$$m_1 := \text{Dec}_k(c_1)$$

$$m_2 := \text{Dec}_k(c_2)$$

$\vdots$

$c_0, c_1, c_2, \dots$

① CPA Security:

$$b' = ? \quad \Pr[b' = b] \approx \frac{1}{2}$$

② Unforgeability:

cannot generate c^* s.t.

$$\text{Dec}_k(c^*) \neq m_i \quad \forall i$$



Encrypt-and-MAC?

Given a CPA-secure SKE scheme $\Pi_1 = (\text{Gen}_1, \text{Enc}_1, \text{Dec}_1)$
& a CMA-secure MAC scheme $\Pi_2 = (\text{Gen}_2, \text{Mac}_2, \text{Vrfy}_2)$

Construct an AE scheme $\Pi = (\text{Gen}, \text{Enc}, \text{Dec})$:

Gen(1^λ):

$k_1 \leftarrow \text{Gen}_1(1^\lambda)$

$k_2 \leftarrow \text{Gen}_2(1^\lambda)$

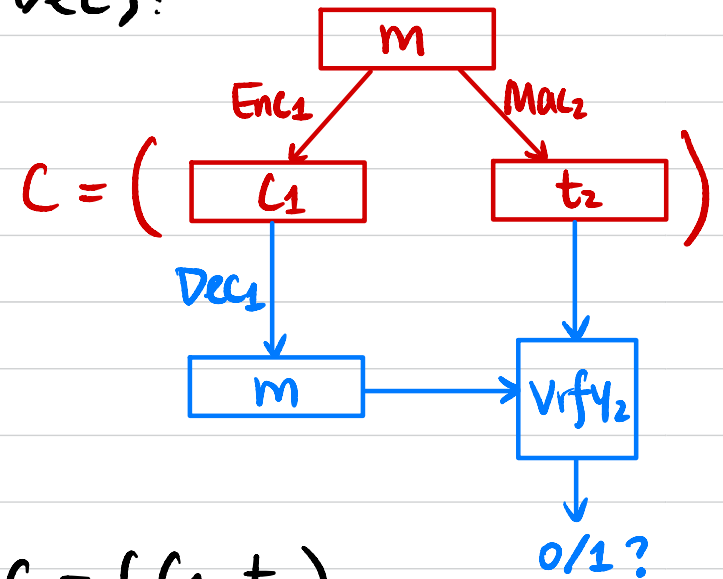
Output $k = (k_1, k_2)$

Enc_k(m):

$c_1 \leftarrow \text{Enc}_1(k_1, m)$

$t_2 \leftarrow \text{Mac}_2(k_2, m)$

output $C = (c_1, t_2)$



Dec_k(C): $C = (c_1, t_2)$

$m := \text{Dec}_1(k_1, c_1)$

$b := \text{Vrfy}_2(k_2, (m, t_2))$

If $b=1$, output m
Otherwise output $\perp$

Encrypt-then-MAC

Gen(1^λ):

$k_1 \leftarrow \text{Gen}_1(1^\lambda)$

$k_2 \leftarrow \text{Gen}_2(1^\lambda)$

Output $k = (k_1, k_2)$

Enc $_k(m)$:

$c_1 \leftarrow \text{Enc}_1(k_1, m)$

$t_2 \leftarrow \text{Mac}_2(k_2, c_1)$

output $C = (c_1, t_2)$

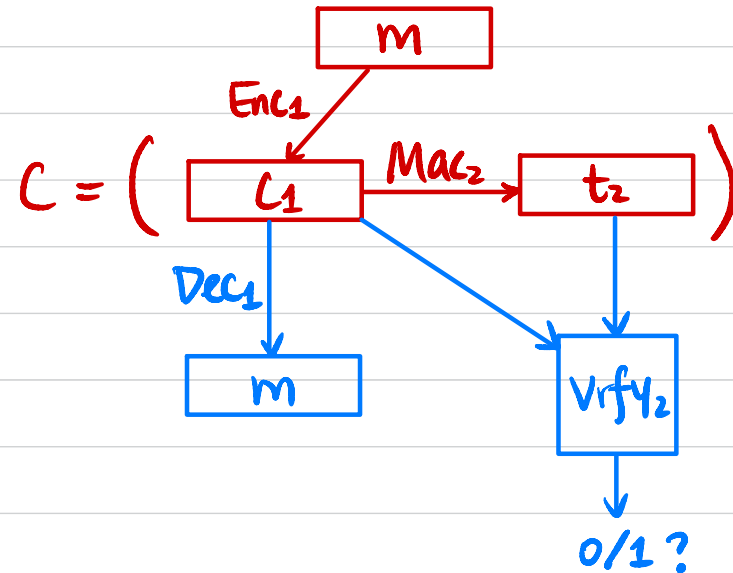
Dec $_k(C)$: $C = (c_1, t_2)$

$m := \text{Dec}_1(k_1, c_1)$

$b := \text{Vrfy}_2(k_2, (c_1, t_2))$

If $b=1$, output m

Otherwise output $\perp$



MAC-then-Encrypt

Gen(1^λ):

$$k_1 \leftarrow \text{Gen}_1(1^\lambda)$$

$$k_2 \leftarrow \text{Gen}_2(1^\lambda)$$

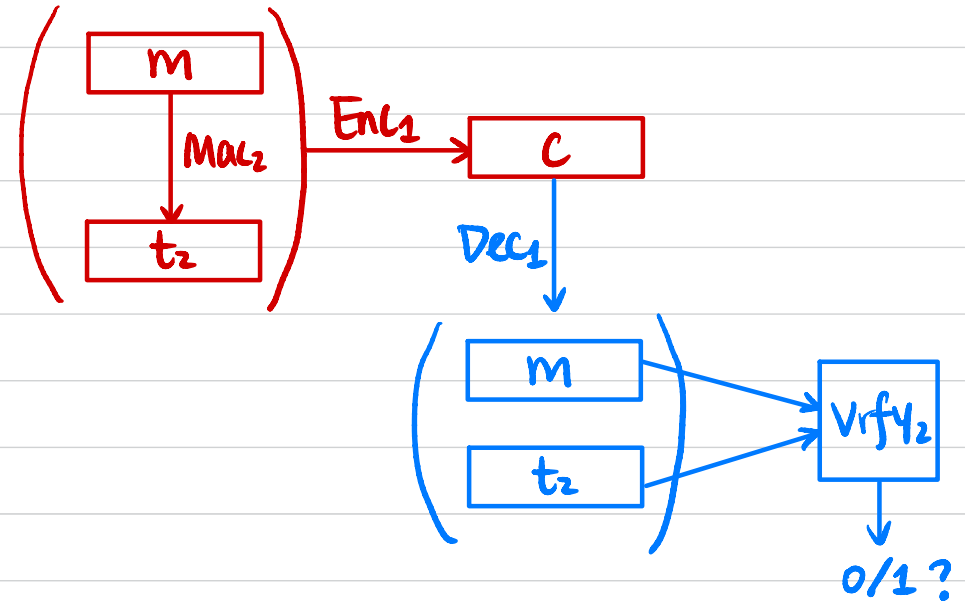
Output $k = (k_1, k_2)$

Enc $_k$ (m):

$$t_2 \leftarrow \text{Mac}_2(k_2, m)$$

$$c_1 \leftarrow \text{Enc}_1(k_1, m \parallel t_2)$$

output $c = c_1$



Dec $_k$ (c): $c = c_1$

$$m \parallel t_2 := \text{Dec}_1(k_1, c_1)$$

$$b := \text{Vrfy}_2(k_2, (m, t_2))$$

If $b=1$, output m

Otherwise output $\perp$

Chosen Ciphertext Attack (CCA)

$$k \leftarrow \text{Gen}(1^\lambda)$$

Alice



$$c_0 \leftarrow \text{Enc}_k(m_0)$$

$$c_1 \leftarrow \text{Enc}_k(m_1)$$

$$c_2 \leftarrow \text{Enc}_k(m_2)$$

⋮

m_0, m_1

$$b \leftarrow \{0, 1\}$$

$$c \leftarrow \text{Enc}_k(m_b)$$



$b' = ?$

$c'_0, c'_1, c'_2, \dots (\neq c)$

Bob



$$m_0 := \text{Dec}_k(c_0)$$

$$m_1 := \text{Dec}_k(c_1)$$

$$m_2 := \text{Dec}_k(c_2)$$

⋮

$$m'_0 := \text{Dec}_k(c'_0)$$

$$m'_1 := \text{Dec}_k(c'_1)$$

$$m'_2 := \text{Dec}_k(c'_2)$$

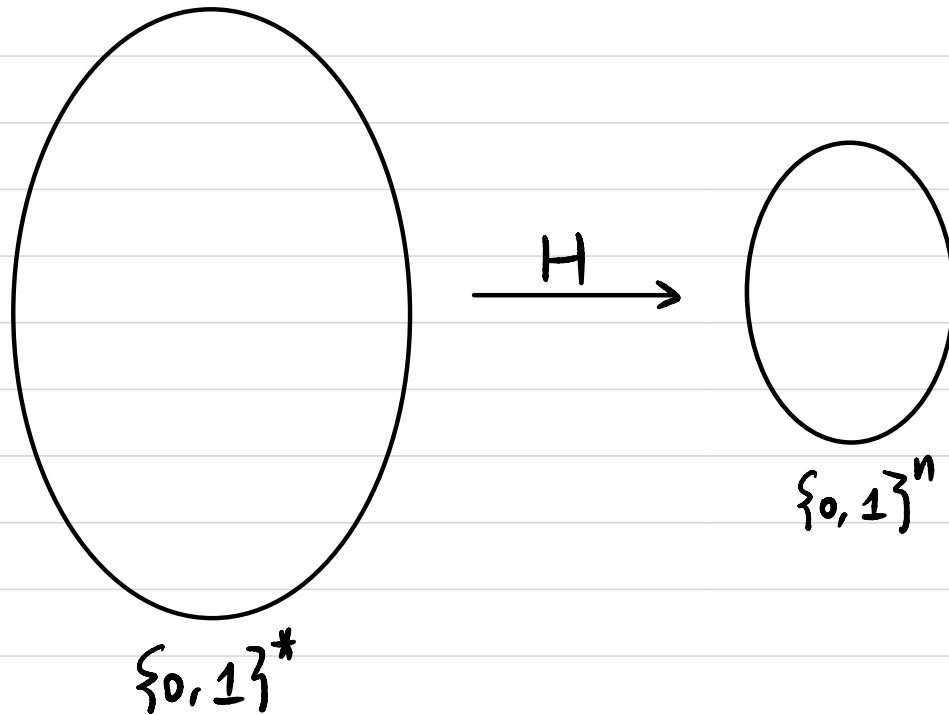
⋮

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
Secrecy & Integrity	Primitive: AE Construction: Encrypt-then-MAC	
Key Exchange		Construction: Diffie-Hellman
Important Tool	Primitive: Hash function Construction: SHA	

Hash Function

$$H: \{0,1\}^* \rightarrow \{0,1\}^n \quad \theta(n)$$



Collision-Resistant Hash Function (CRHF):

It's computationally hard to find $x, y \in \{0,1\}^*$ s.t.

$$x \neq y, \quad H(x) = H(y) \quad (\text{collision})$$

How to find a collision?

$$H: \{0,1\}^* \rightarrow \{0,1\}^n$$

Try $H(x_1), H(x_2), \dots, H(x_q)$

If $H(x_i)$ outputs a random value,
what's the probability of finding a collision?

$$\text{If } q = 2^n + 1 \Rightarrow \text{prob.} = 1$$

$$\text{If } q = 2 \Rightarrow \text{prob.} = \frac{1}{2^n}$$

$$\text{If } q = ? \Rightarrow \text{prob. is "reasonable"}$$

If security parameter $\lambda = 128$, $n = ?$

Birthday Problem / Paradox

There are q students in a class.

Assume each student's birthday is a random $y_i \in [365]$

What's the probability of a collision?

$$q = 366 \Rightarrow \text{prob.} = 1$$

$$q = 23 \Rightarrow \text{prob.} \approx 50\%$$

$$q = 70 \Rightarrow \text{prob.} \approx 99.9\%$$

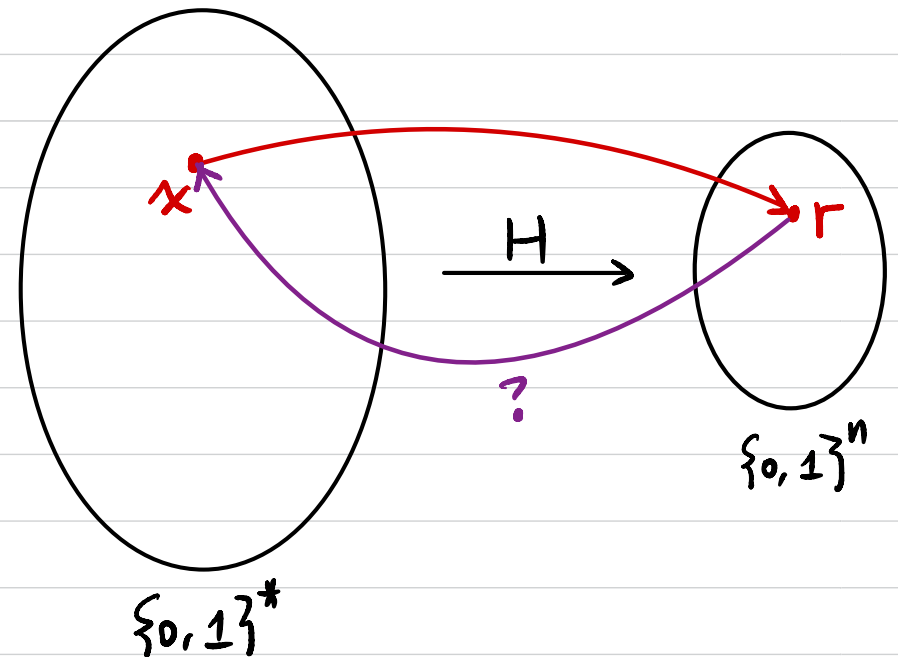
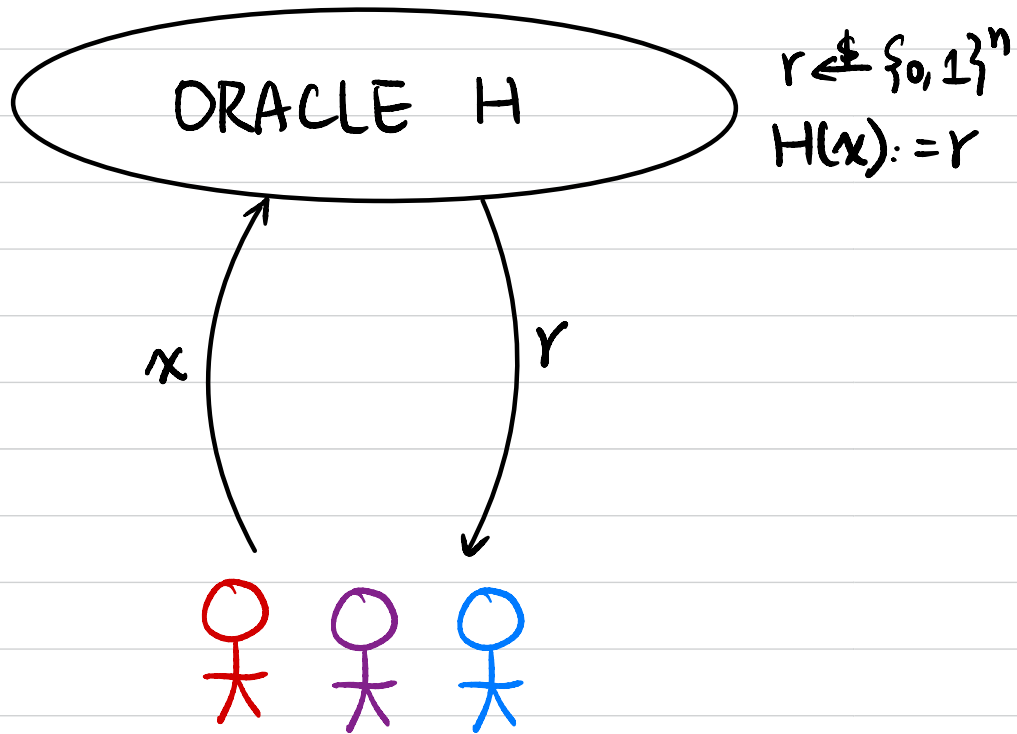
$$y_i \in [N]$$

$$q = N + 1 \Rightarrow \text{prob.} = 1$$

$$q = \sqrt{N} \Rightarrow \text{prob.} \approx 50\%$$

Random Oracle Model

$$H: \{0,1\}^* \rightarrow \{0,1\}^n$$



Constructions for Hash Function

MD5: output length 128-bit
best known attack 2^{16}
Collision found in 2004

Secure Hash Functions (SHA): Standardized by NIST.

- SHA-0: Standardized in 1993
output length 160-bit
best known attack 2^{39}
- SHA-1: Standardized in 1995
output length 160-bit
best known attack 2^{63}
Collision found in 2017

Constructions for Hash Function

Secure Hash Functions (SHA): Standardized by NIST.

- SHA-2: Standardized in 2001
output length 224, 256, 384, 512-bit
- SHA-3: Competition 2007-2012
released in 2015
output length 224, 256, 384, 512-bit

Applications of Hash Functions

- HMAC
- Hash-and-Sign : $\text{Sign}_{sk}(H(m))$
- HKDF (Key Derivation Function):

$$H(g^{ab} || 0), H(g^{ab} || 1), \dots$$

Pseudorandom Generator (PRG)

- Password-based Authentication
- Fast Membership Proof (Merkle Tree) $\rightarrow$ blockchain

Applications of Hash Functions

- Deduplication

$$\begin{aligned} H(D_1) &\rightarrow h_1 \\ H(D_2) &\rightarrow h_2 \end{aligned}$$

unique identifier



$$\text{If } h_1 \neq h_2 \Rightarrow D_1 \neq D_2$$

$$\text{If } h_1 = h_2 \Rightarrow D_1 = D_2 \text{ (why?)}$$

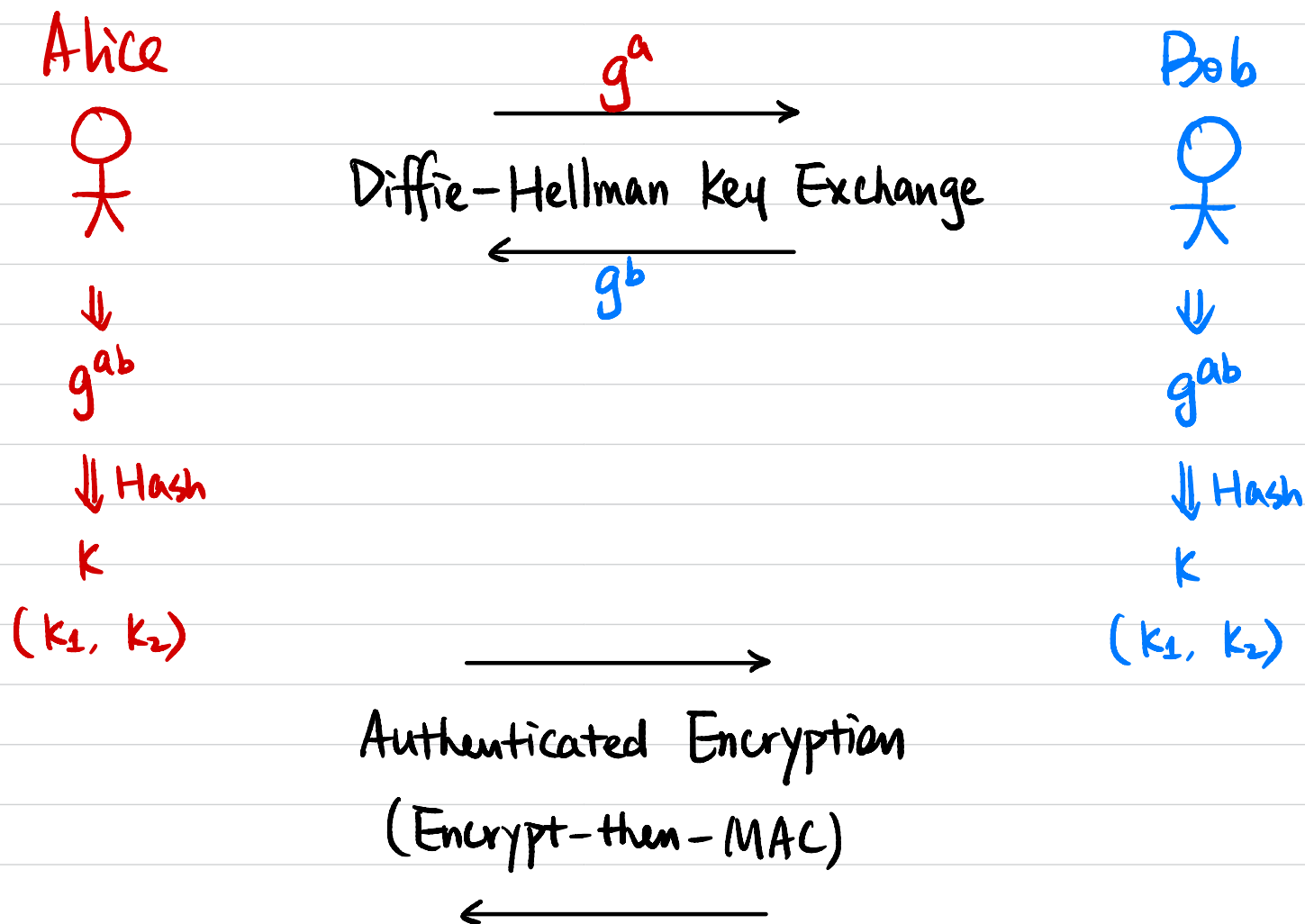
Virus Scan

$$H(F) \stackrel{?}{=} H(F^*)$$

Video Deduplication

$$H(V_1) \stackrel{?}{=} H(V_2)$$

Putting it Together: Secure Communication



Any security issue?

Signal: Diffie-Hellman Ratchet

Alice



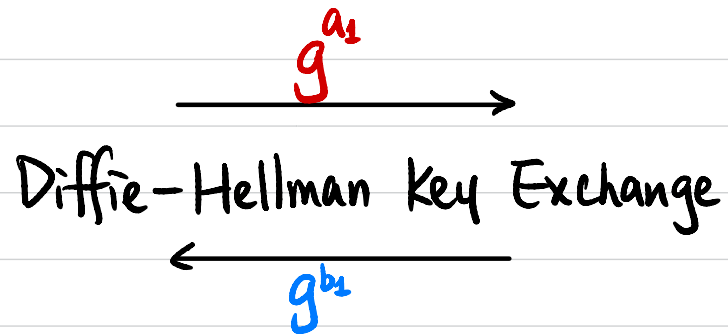
$$g^{a_1 b_1}$$



$$g^{a_2 b_1}$$



$$g^{a_2 b_2}$$



Bob



$$g^{a_1 b_1}$$



$$g^{a_2 b_1}$$



$$g^{a_2 b_2}$$

