Vietnam National University Ho Chi Minh City, University of Science
Department of Information Technology

Topic 7:

Digital signature

Assoc. Prof. Trần Minh Triết
PhD. Trương Toàn Thịnh

TRUONG DAI HOC KHOA HOC TU NHIEN
KHTN
TP. HO CHI MINH
cdio 4.0
fit@hcmus
KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

fit@hcmus

Contents

Introduction

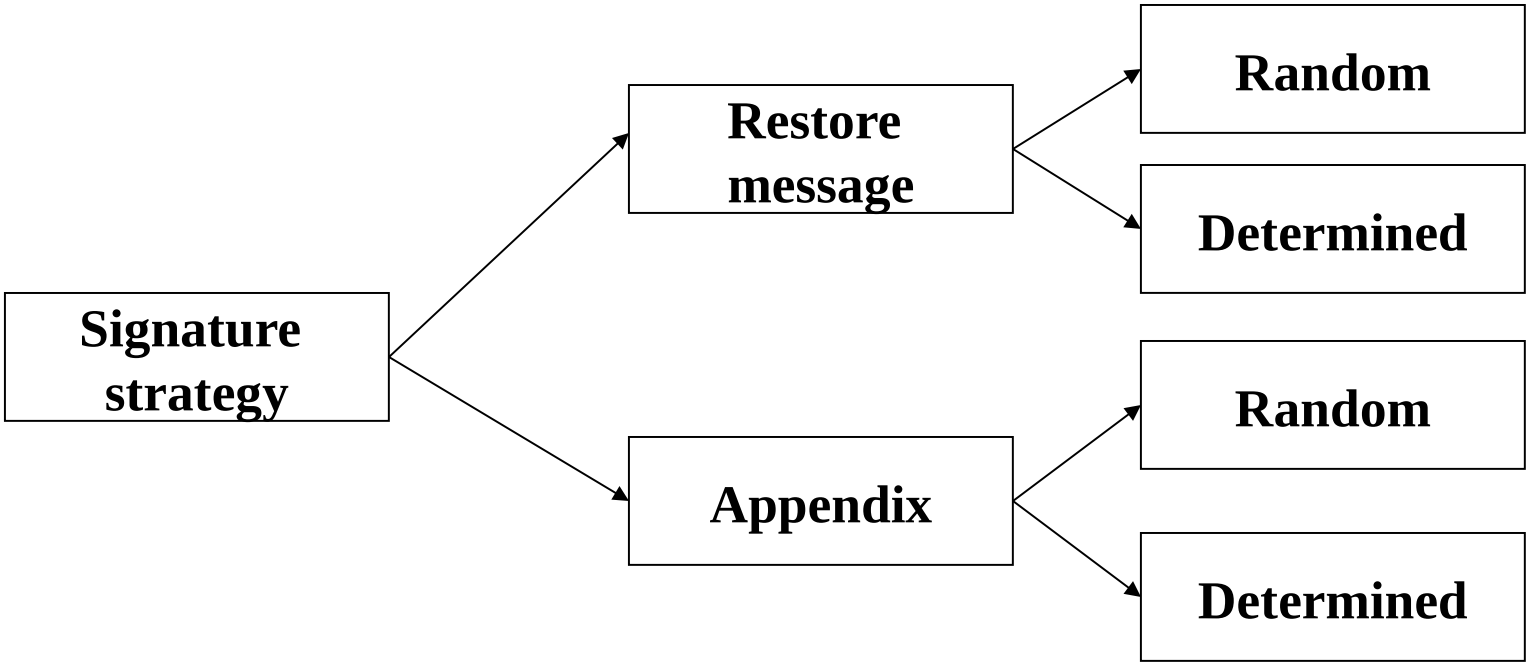
- Goals of digital signature:
  - User authentication
  - Data integrity
  - Non-repudiation
- Basic concepts:
  - Digital signature: data string that allows to identify the origin/entity that generated a message.
  - Algorithm for generating digital signatures: methods for generating digital signatures
  - Digital signature strategy includes ***algorithm generating digital signature*** and ***algorithm verifying digital signature***.
  - **Digital signature scheme = Digital signature generation algorithm + Digital signature verification algorithm**

Introduction

- Notations:
  - M Message-space
  - M_S Signed-message space
  - S Signature-space
  - R Map 1-1 from M to M_S (redundancy function)
  - M_R Image of R
  - R^{-1} Inverse function of R
  - h One-way function with a source set M
  - M_h Hash-value space ($h: M \rightarrow M_h$)

Introduction

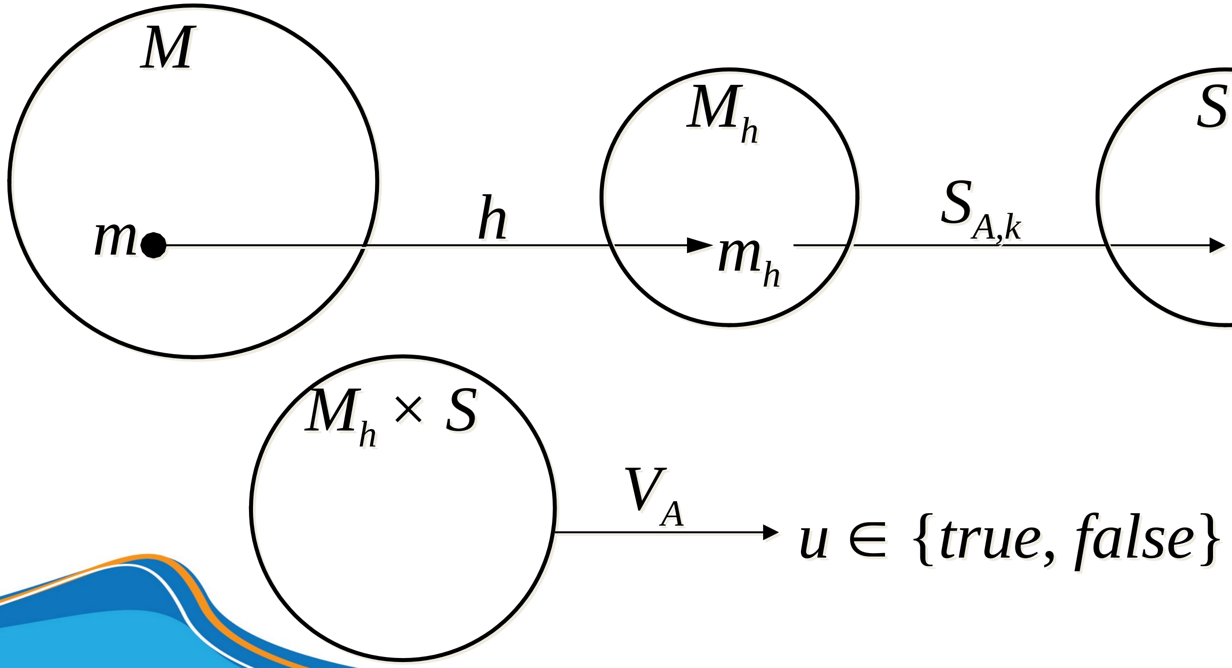
- Classification of digital signatures



Introduction

$$s^* = S_{A,k}(m_h)$$
$$u = V_A(m_h, s^*)$$

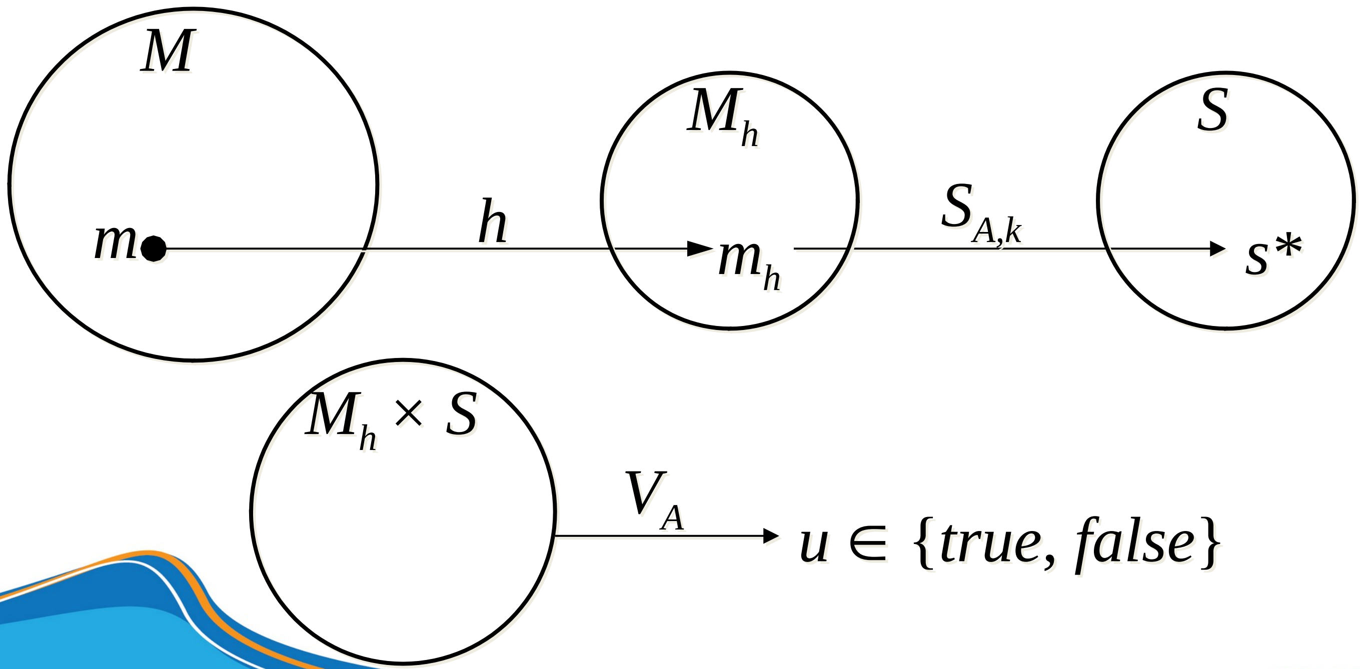
- Signature strategies with appendix
  - Signature accompanies original message (Ex: DSA, ElGamal, Schnorr…)
  - Message (original) required for digital signature verification
  - Use a cryptographic hash function (instead of a redundancy function)



Introduction

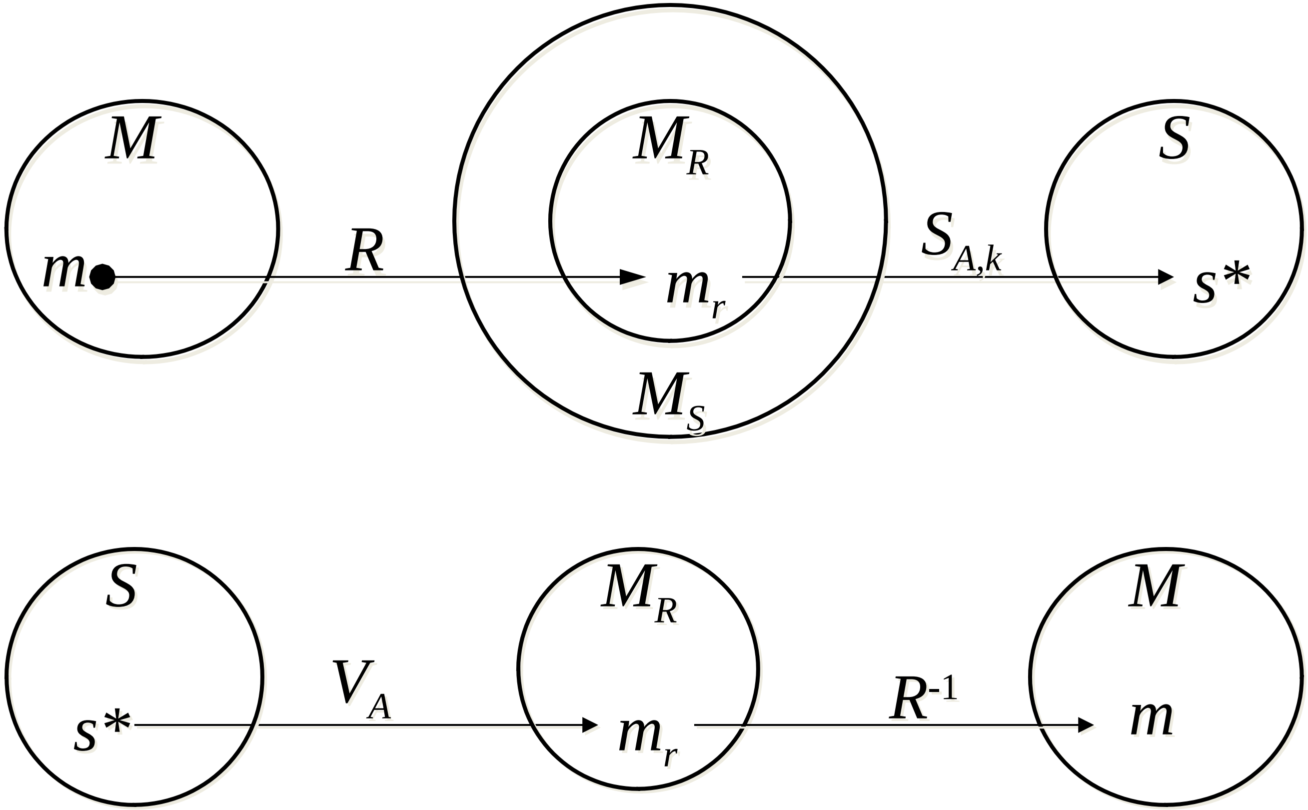
$s^* = S_{A,k}(m_h)$
$u = V_A(m_h, s^*)$

- Requirements:
  - For each $k \in R$, easy to compute $S_{A,k}$
  - Easy to compute V_A
  - Hard for those are not *signer* to find $m \in M$ and $s^* \in S$ such that $V_A(m', s^*) = true$, for $m' = h(m)$



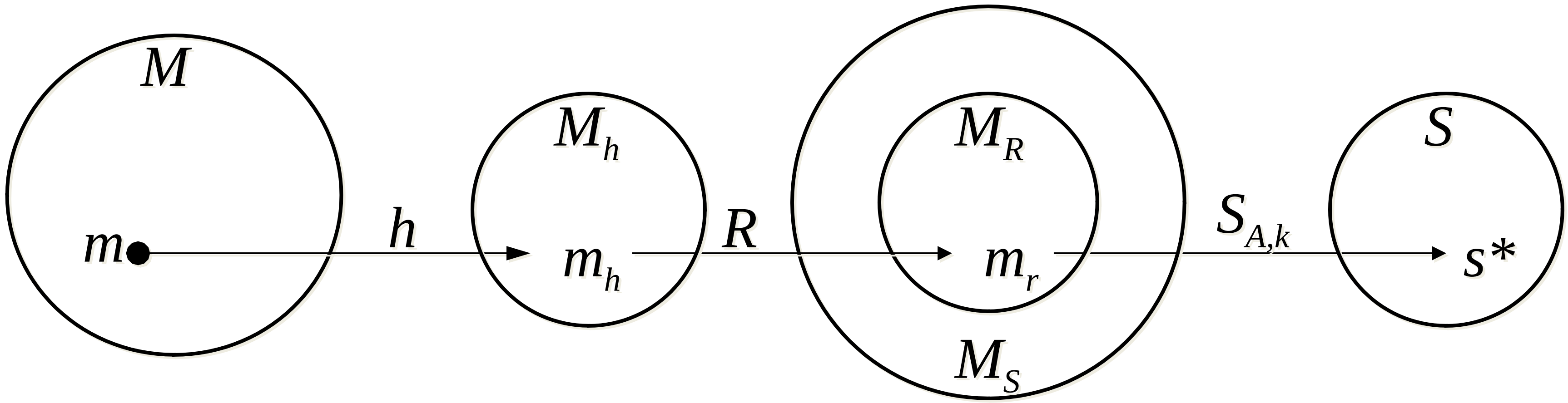
Introduction

- A digital signature can allow the message to be recovered



Introduction

- Requirements:
  - For each $k \in$ R, easy to compute $S_{A,k}$
  - easy to compute V_A
  - Hard for (computationally infeasible) those are not A to find $s^* \in S$ such that $V_A(s^*) \in M_R$

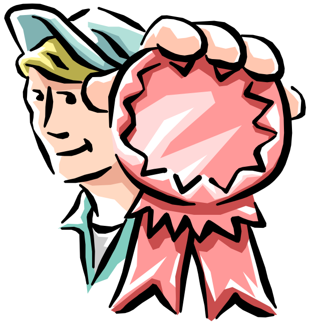


Introduction

Severity

- Levels of "Breaking" the e-signature strategy:
  - ***Total Break***: find an efficient way to "fake" a valid signature.
    - Know private key?
    - Do not know private-key but find an efficient way to create a valid signature.
  - ***Selective forgery***: Given a message, attacker *can* create a valid signature for this message.
  - ***Existential forgery***: can find and point to a message (maybe nonsense) but easy for attacker to create valid signature for this message.
- Classification of attacks
  - ***Key-only***: attacker **only know public-key**
  - Message attack
    - ***Known-message attack***: attacker **has the signatures of a set of messages**. Attacker **knows contents of these messages** but **is not allowed to pre-choose** these messages.
    - ***Chosen-message attack***: attacker **has valid signatures** of **a set of selective messages** (non-adaptive)
    - ***Adaptive chosen-message attack***: attacker can use signer as a "**oracle**"

RSA signature



- Generate key n, p, q, e, d
- Create signature
  - Compute $m_r = R(m)$
  - Compute $s = m_r^{d} \bmod n$
  - Signature of m is s
- Verify signature
  - Receive public-key (n, e)
  - Compute $m_r = s^e \bmod n$
  - Verify $m_r \in M_r$
  - Recover $m = R^{-1}(m_r)$

RSA signature

- Attack
  - Factorize a large integer
  - Possibility of multiple key pairs yielding the same signature
  - Homomorphic property:

    $E(x_1) \times E(x_2) = x_1^e \times x_2^e \bmod n = (x_1 \times x_2)^e \bmod n = E(x_1 \times x_2 \bmod n)$
- Re-blocking problem
- Importance of redundancy function: ISO/IEC 9796
- Efficiency (p, q are k-bit prime-numbers): cost of generating signature $O(k^3)$ and verifying signature $O(k^2)$
- Bandwidth
  - Bandwidth depends on R.
  - Example: ISO/IEC 9796 maps a k-bit message to a $2k$-bit string in M_S with $2k$-bit signature.

DSA signature

- Generating key:
  - Choose 1 prime number q 160 bits
  - Choose $0 \leq t \leq 8$, choose $2^{511 + 64t} < p < 2^{512 + 64t}$ for $q \mid p - 1$
  - Choose g in Z_p^*, and $\alpha = g^{(p - 1) / q} \bmod p$, $\alpha \neq 1$ (α is a primitive of sub-group with order q of Z_p^*)
  - Choose $1 \leq a \leq q - 1$, compute $y = \alpha^a \bmod p$
  - Public-key (p, q, α, y) and private-key a
- Create signature:
  - Randomly choose an integer k, $0 < k < q$
  - Compute $r = (\alpha^k \bmod p) \bmod q$ and $k^{-1} \bmod q$
  - Compute $s = k^{-1} \times (h(m) + a \times r) \bmod q$
  - Signature $= (r, s)$

DSA signature

- Verifying signature
  - Check $0 < r < q$ and $0 < s < q$, if this doesn't hold, signature isn't valid
  - Compute $w = s^{-1} \bmod q$ and $h(m)$
  - Compute $u_1 = w \times h(m) \bmod q$, $u_2 = r \times w \bmod q$
  - Compute $v = ((\alpha^{u_1} \times y^{u_2}) \bmod p) \bmod q$
  - Signature is valid $\Leftrightarrow v = r$
  - Explain:

$$h(m) \equiv -a \times r + k \times s \pmod q$$

$$\Leftrightarrow w \times h(m) + a \times r \times w \equiv k \pmod q$$

$$\Leftrightarrow u_1 + a \times u_2 \equiv k \pmod q$$

$$\Leftrightarrow (\alpha^{u_1} \times y^{u_2} \bmod p) \pmod q \equiv (\alpha^k \bmod p) \pmod q$$

DSA signature

- Safety issues of DSA: discrete logarithm in Z_p^* and cyclic subgroup with order q
- Parameters: $q \approx 160$ bits, $p \approx 768$ bits ~ 1Kb
- Failure probability: During verification, necessary to calculate the inverse of s. If $s \equiv 0$ there doesn't exist inverse $\mathbf{Pr}[s = 0] = (\frac{1}{2})^{160}$
- Efficiency
  - Create signature
    - A modulo exponentiation operation and several operations 160-bit (if p ~768 bit)
    - The exponentiation can be calculated in advance
    - ***Faster than RSA***
  - Check signature
    - Two operations of exponentiation modulo
    - ***Slower than RSA***

ElGamal signature

- Generate key: $p, q, \alpha, a, y = \alpha^a \bmod p$
  - α is a primitive of Z^*_p
  - Public-key (p, α), private-key (a)
- Create signature
  - Randomly choose k, $1 \le k \le p - 1$, $gcd(k, p - 1) = 1$
  - Compute $r = \alpha^k \bmod p$, $k^{-1} \bmod (p-1)$ and $s = k^{-1} \times (h(m) - a \times r) \bmod (p-1)$
  - Signature (r, s)
- Check signature
  - Check $1 \le r \le p - 1$
  - Compute $v_1 = y^r \times r^s \bmod p$, $h(m)$ and $v_2 = \alpha^{h(m)} \bmod p$
  - Signature is valid $\Leftrightarrow$ $v_1 = v_2$

Explain

$$s \cong k^{-1} \times (h(m) - a \times r) \pmod{p-1}$$
$$\Leftrightarrow k \times s \cong h(m) - a \times r \pmod{p-1}$$
$$\Leftrightarrow \alpha^{h(m)} \cong \alpha^{a \times r + k \times s} \cong (\alpha^a)^r \times r^s \bmod p$$

ElGamal signature

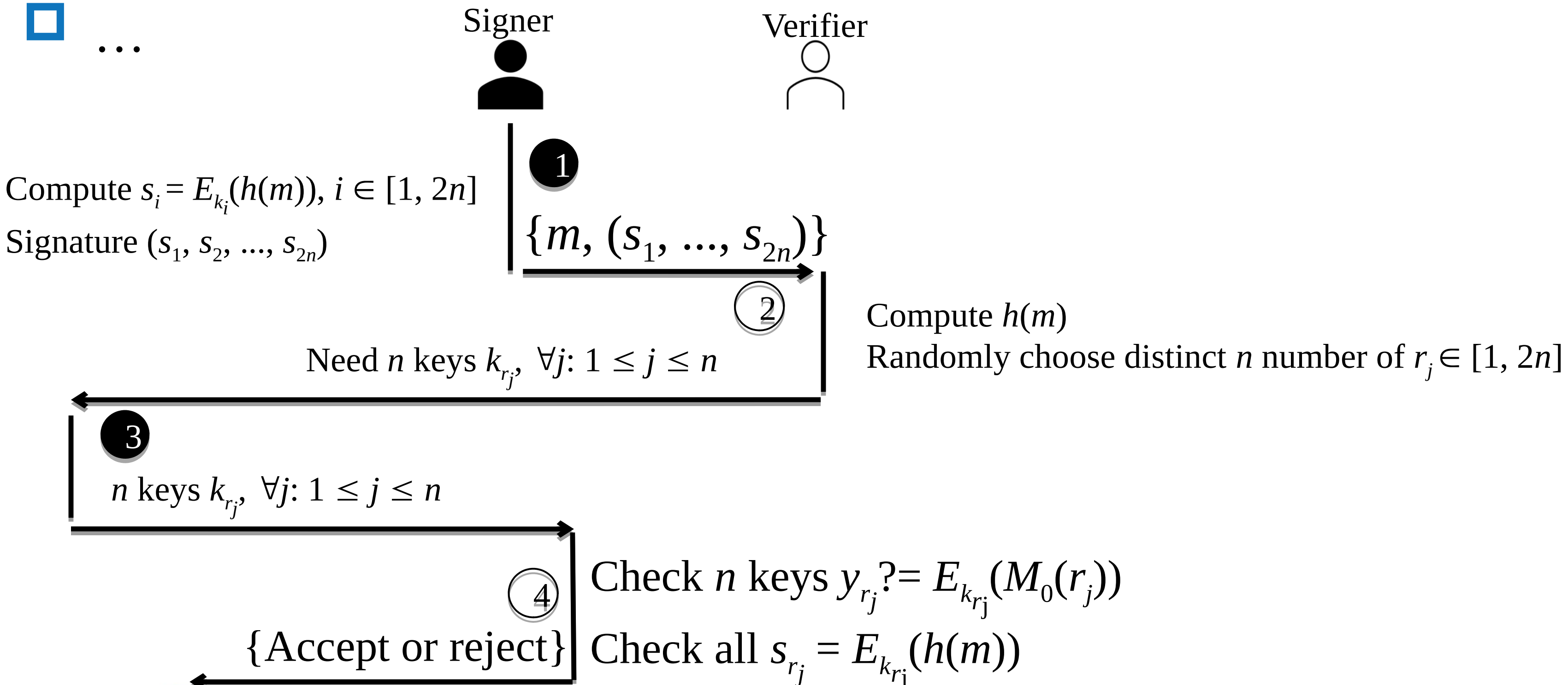
- Problems
  - Value k must be distinguished for each signed message
    - $(s_1 - s_2) \times k = (h(m_1) - h(m_2)) \bmod (p - 1)$
    - If $gcd((s_1 - s_2), p - 1) = 1$ then can determine the value k, and have private-key a
  - If you don't use the hash function, you may get an existential forgery
- Efficiency
  - Create signature
    - An exponentiation operation modulo
    - An operation that uses the Euclidean algorithm to compute the inverse
    - Two multiplication operations modulo
  - Check signature: three exponentiation operations modulo
- Read more: Generalized ElGamal Signature

One-time signature (Rabin method)

- Definition: the signature strategy is used to **sign up to one message**
  - If reused, can be hacked to create fake signatures.
  - A new public key is required for each message to be signed
- Most one-time signature strategies are characterized by very fast signature generation and verification
- Create key: choose symmetric encryption algorithm E (ex: DES)
  - Generating $2n$ random string $k_1, k_2 \ldots, k_{2n} \in K$, each has length of l
  - Compute $y_i = E_{k_i}(M_0(i))$, $i \in [1, 2n]$)
    - $M_0(i)$ is the binary representation of i filled with more bits 0 at the beginning to get the sequence l-bit
  - Public key is $(y_1, y_2, \ldots y_{2n})$ and private key is $(k_1, k_2, \ldots k_{2n})$

One-time signature (Rabin method)

- If there is a conflict, there is a need Trusted Third Party (*TTP*)
  - Signer may deliberately deny having created the signature
  - Verifier may deliberately not accept the signature
  - …



fit@hcmus

One-time signature (Rabin method)

- Context: Signer refuse to generate signature provided by Verifier