EXPLORING CRYPTOGRAPHY PROTOCOLS

WITH LIMITED EMPHASIS ON MATHEMATICS ☺

Presented By
@7eRoM

ATTENTION

THESE SLIDES HAVE BEEN CRAFTED USING THE FOUNDATION OF MY MSC COURSE IN CRYPTOGRAPHY PROTOCOLS AT THE UNIVERSITY OF ISFAHAN.

I'VE MADE ADJUSTMENTS TO THE CONTENT TO ALIGN WITH THE SPECIFIC OBJECTIVES OF THIS PRESENTATION.

ALSO, MY INTENTION HAS BEEN TO MINIMIZE THE USE OF MATHEMATICAL CONCEPTS, WHICH MAY RESULT IN SOME CONCEPTS BEING SIMPLIFIED OR LESS PRECISE.

Presented By
@7eRoM

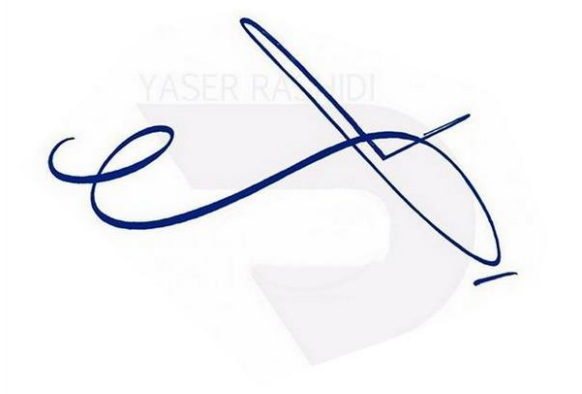
Agenda

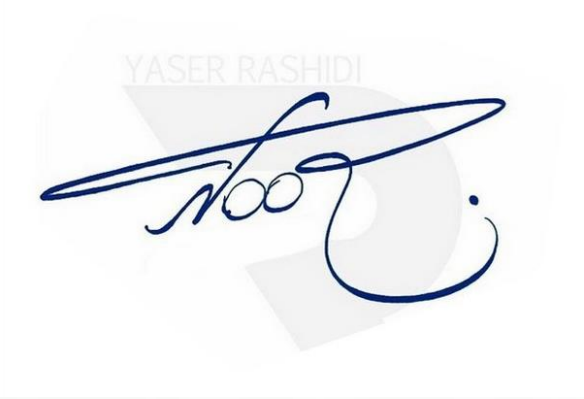
1. Identification and Entity Authentications Protocols
2. Zero Knowledge Protocols
3. Key Establishment Protocols
4. Threshold Cryptography and Secret Sharing Protocols
5. Special Purpose Protocols (like simultaneous contract signing, mental poker, fair exchange)
6. Identity Based Cryptography
7. **Types of Digital Signatures**
8. Secure Multiparty Computations

Types of Digital Signatures

Warm Up!

Some Beautiful Signatures





Types of Digital Signatures

Warm Up!

Some Beautiful Signatures

Types of Digital Signatures

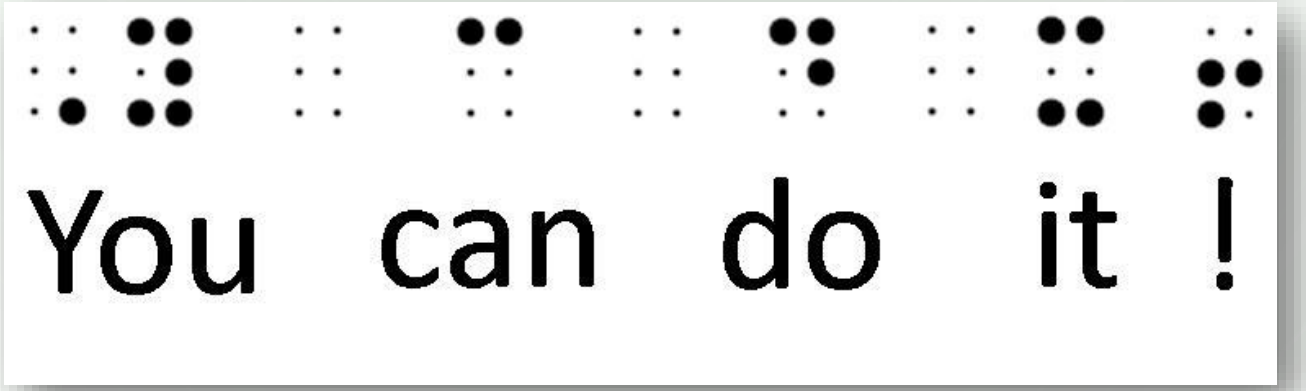
Content

- Blind Signature
- Fail-Stop Signature
- Undeniable Signature
- Ring Signature (RSA and ECC based)

Types of Digital Signatures

Types of Digital Signatures

Blind Signature



Types of Digital Signatures

Blind Signature

- The content of what is being signed is not disclosed to the signer.
- 3 Steps:
  - **Blinding**: The requester blinds the original message and sends it to the signer
  - **Signing**: The signer signs the blinded message in a manner that is equivalent to signing the original message.
  - **Unblinding**: The requester extracts the signed original message from the blinded message.
- Abstract Example:
  - **Blinding**: The requester places the message inside a carbon paper envelope.
  - **Signing**: The signer signs the envelope in a way that the message would be signed.
  - **Unblinding**: The requester extracts the signed message from the envelope.

Types of Digital Signatures

Blind Signature – based on RSA

Objective:
Getting m^d

Requester

$r \in Z^*_n$

blinding

$X = mr^e \bmod n$

signing

$Y = X^d \bmod n$

Signer

public e
private d
$ed \bmod \varphi(n) = 1$

unblinding

$Yr^{-1} \bmod n$

$= m^d \bmod n$

$$Yr^{-1} = X^d r^{-1}$$
$$= m^d r^{ed} r^{-1}$$
$$= m^d$$

When an attacker, posing as Alice, sends a previously encrypted message from Bob to him and successfully decrypts it

The usage of each key must be determined.

Types of Digital Signatures

Blind Signature - Usage

- Use in e-cash
- Objective: Preserve the anonymity of the customer
- 3 roles:
  - **Customer** or **Buyer**
  - **Salesman** or **Merchant**
  - **Bank:** Checks the banknote been spent before?

Types of Digital Signatures

Blind Signature – e-cash

Only Bank can calculate
$1/v \bmod \varphi(n)$

Bank

private random large prime p,q
public $n=pq$
$ed \bmod \varphi(n) = 1$
Database of spent SNs DBx

SN of banknote x
value of banknote v
random r
hash function $h()$

$D = h(x)^{1/v} r \bmod n$

$c = D/r = h(x)^{1/v}$

Money (x,c)

Please deduct v from my balance
$B = h(x) r^v \bmod n$

$D = B^{1/v} \bmod n$

Has x been spent before?

Yes / No

v , (x,c)

Checking to ensure the bank's signature

$c^v \bmod n == h(x) \bmod n$

Types of Digital Signatures

Fail-Stop

Types of Digital Signatures

Fail-Stop

- If Bob can forge Alice's signature on message m, Alice will be able to prove the forgery.
- After the first forgery, all other participants in the system and the system operator know that the signature scheme has been broken, and the system will be stopped.

- 3 Algorithms:
  - **Signature Generation**
  - **Signature Verification**
  - **Forgery Proof**: Proving the forgery of a signature

Types of Digital Signatures

Fail-Stop – Heyst-Pedersen Scheme

Bank

private $s \in Z^*_q$

Alice's public key $h = g^s \bmod p$

$q \mid p$

private $\{x_1, x_2, y_1, y_2\}$

public key $p_1 = g^{x1} * h^{x2} \bmod p$

public key $p_2 = g^{y1} * h^{y2} \bmod p$

$\sigma_1 = x_1 + y_1 m \bmod q$

$\sigma_2 = x_2 + y_2 m \bmod q$

$sig(m) = (\sigma_1, \sigma_2)$

$P_1 P_2^m == g^{\sigma 1} h^{\sigma 2} \bmod p$

$g^{\sigma 1} h^{\sigma 2} = g^{x1+y1m} \; h^{x2+y2m}$

$= g^{x1} \; h^{x2} \; g^{y1m} \; h^{y2m}$

$= P_1 P_2^m \bmod p$

Types of Digital Signatures

Fail-Stop – Heyst-Pedersen Scheme

Imagine the forged signature (σ''_1, σ''_2) on message m' that passes the verification

Alice tries to calculate Bank's private s to prove her innocence

For given forged signature: $P_1 P_2^{m'} = g^{\sigma''_1} h^{\sigma''_2} \bmod p$

Alice calculates his signature on m': $P_1 P_2^{m'} = g^{\sigma'_1} h^{\sigma'_2} \bmod p$

$g^{\sigma''_1} h^{\sigma''_2} = g^{\sigma'_1} h^{\sigma'_2} \bmod p$

$g^{\sigma''_1 + s\sigma''_2} = g^{\sigma'_1 + s\sigma'_2} \bmod p$

$\sigma''_1 + s\sigma''_2 = \sigma'_1 + s\sigma'_2 \bmod q$

$s = (\sigma'_1 - \sigma''_1)/(\sigma''_2 - \sigma'_2) \bmod q$

private $s \in Z^*_q$

Alice's public key $h = g^s \bmod p$

private $\{x_1, x_2, y_1, y_2\}$

public key $p_1 = g^{x1} * h^{x2} \bmod p$

public key $p_2 = g^{y1} * h^{y2} \bmod p$

$\sigma_1 = x_1 + y_1 m \bmod q$

$\sigma_2 = x_2 + y_2 m \bmod q$

$P_1 P_2^m = g^{\sigma 1} h^{\sigma 2} \bmod p$

Types of Digital Signatures

Undeniable Signature

1. The verification of the signature requires interaction with the signer.
2. The signer can not deny his signature.

- Usage Example
  - Avoid copying the signature by the other party (like software license)

Types of Digital Signatures

Undeniable Signature – Chaum–Antwerpen Protocol

prime r
prime $p=2r+1$
public generator $g \in Z^*_p$
private key x from range $[1,r-1]$
public key $y=g^x \bmod p$

Signer

$z = x^{-1} \bmod r$

$S = sig(m) = m^x \bmod p$

$C = S^{e1} * y^{e2}$

$d = C^z \bmod p$

Verifier

random e_1, e_2 from range $[1,r-1]$

Verify

$d == m^{e1} g^{e2}$

$m^{e1} g^{e2} = (m^{xe1} g^{xe2})^{x^{-1}}$

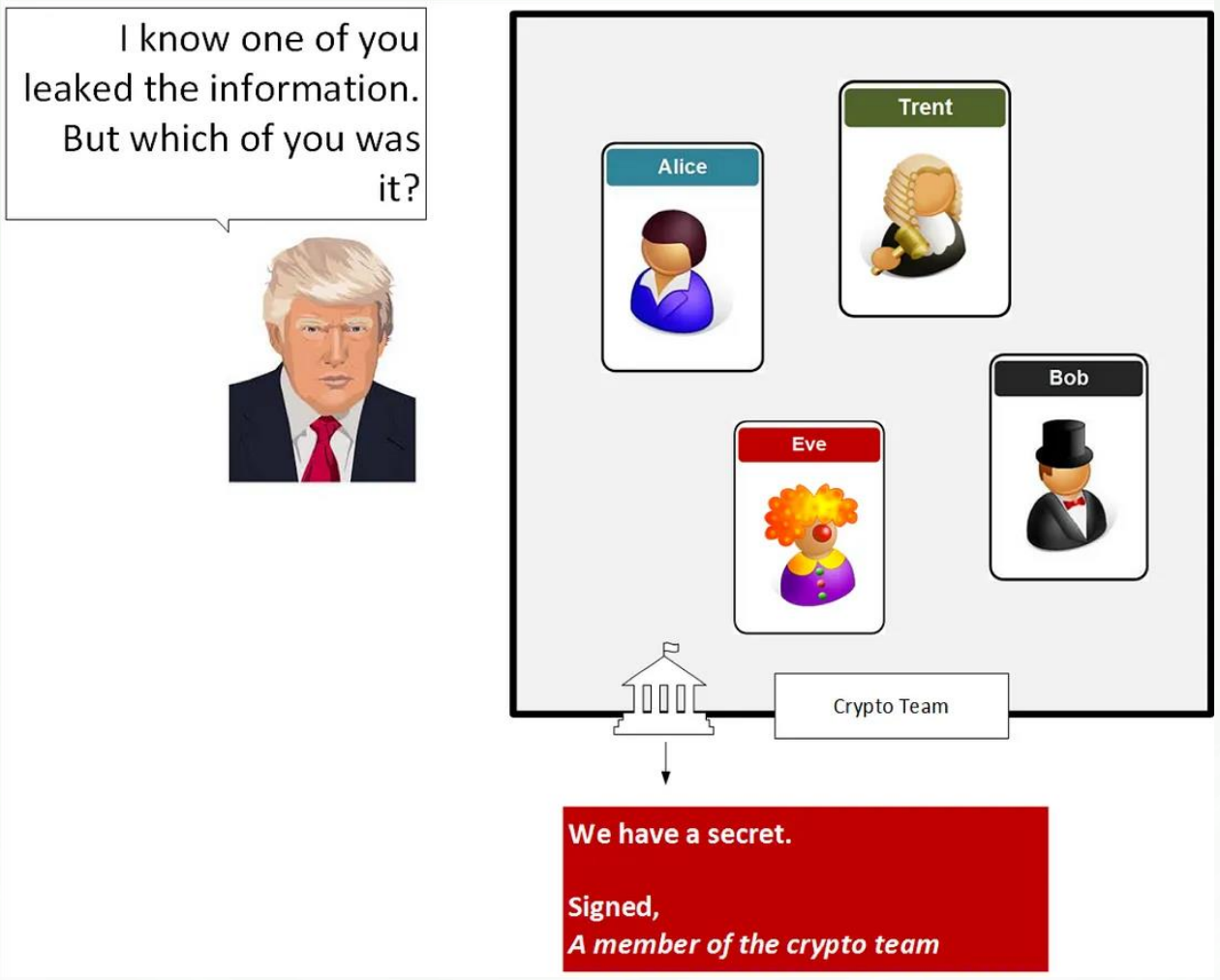
$= (S^{e1} * y^{e2})^{x^{-1}}$

$= C^z$

Types of Digital Signatures

Ring Signature

How to leak a secret !?



Types of Digital Signatures

Ring Signature

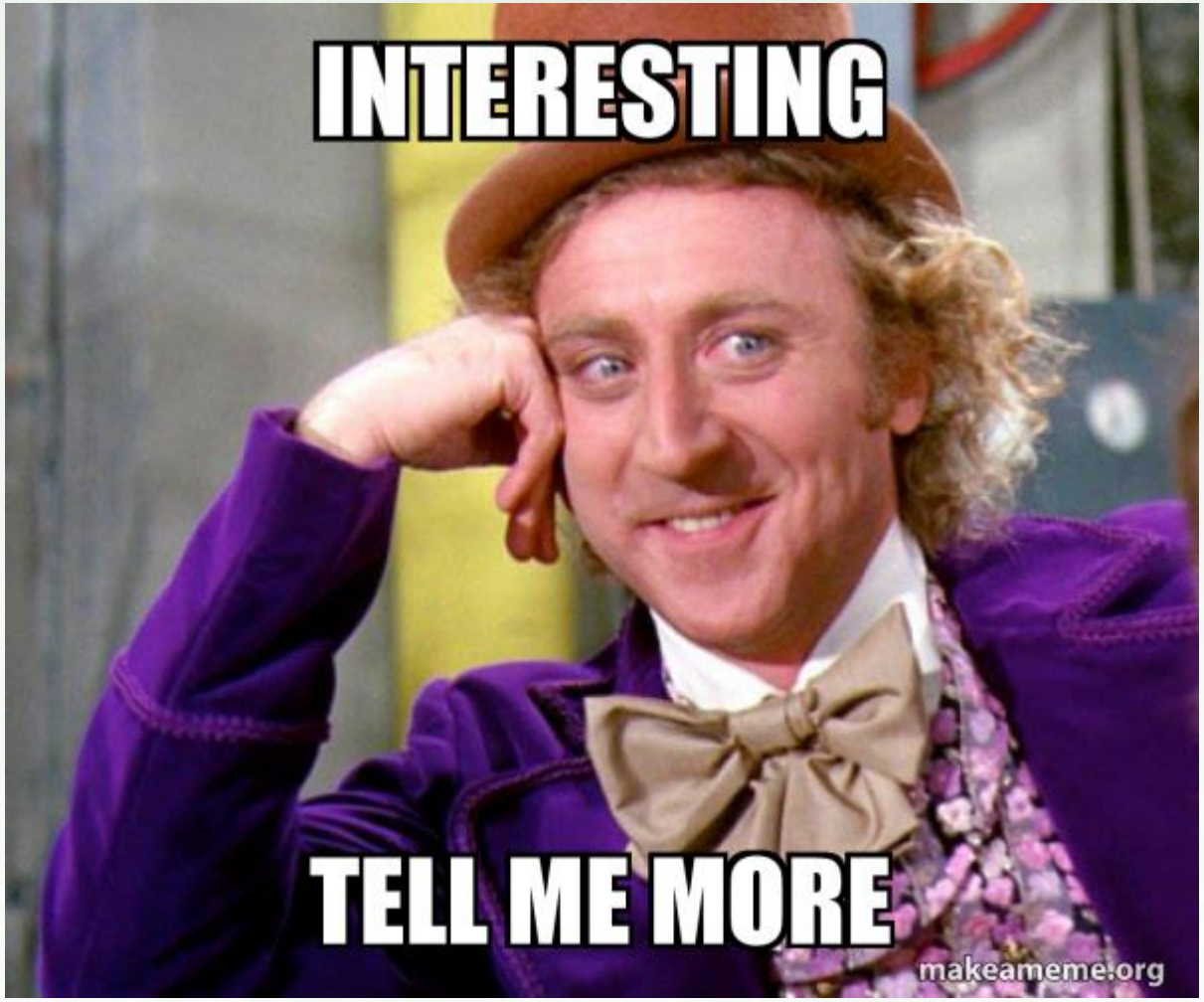
- Suppose Allice knows a secret and wishes to leak it to journalist
- BUT Alice wants to remain anonymous!

- A standard group signature scheme does not solve the problem, since it requires the prior cooperation of the other group members to set up.

- The CryptoNote technology uses ring signatures. It was first implemented by Bytecoin.
  - **ShadowCash**: uses traceable ring signature to anonymize the sender of a transaction
  - **Monero**: uses ring signatures to obfuscate the true spend in a transaction.

Types of Digital Signatures

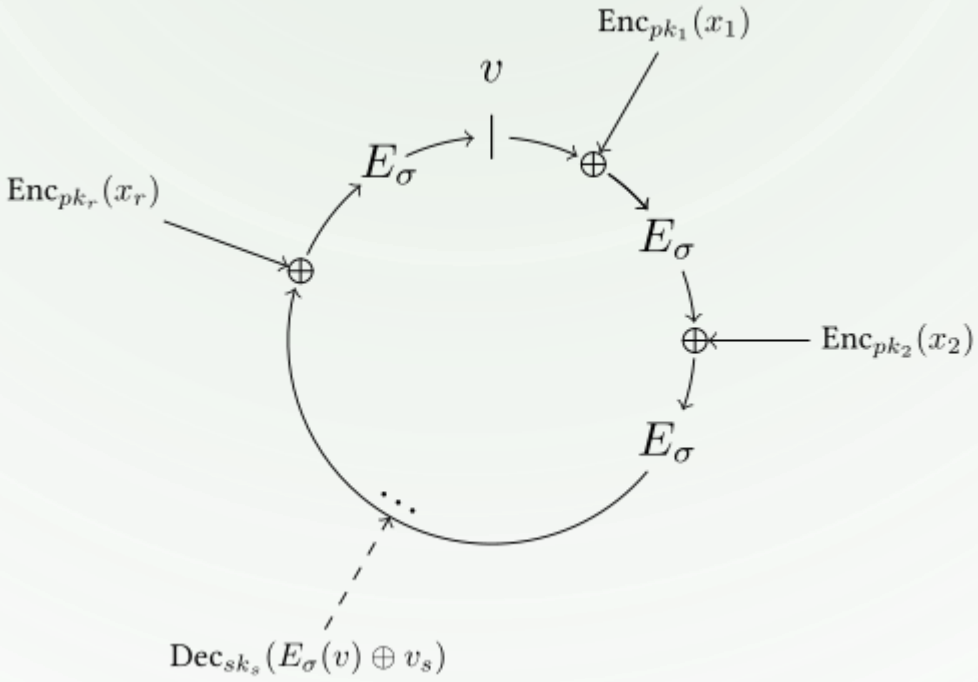
Ring Signature



Types of Digital Signatures

Ring Signature – RSA based

Rivest, Shamir, and Tauman (2001)



Types of Digital Signatures

Ring Signature – RSA based

Trapdoor functions

- a function that is easy to compute in one direction, yet difficult to compute in the opposite direction (finding its inverse) without special information

- In mathematical terms, if f is a trapdoor function, then there exists some secret information t, such that given $f(x)$ and t, it is easy to compute x.

Types of Digital Signatures

Ring Signature – RSA based

Requirements

1. Each user has public and private key pairs, $(P_1, S_1), (P_2, S_2), \ldots, (P_n, S_n)$
2. A keyed "combining function" $C_{k,v}(y_1,y_2,\ldots,y_n)$ which takes a key k, an initialization value v, and a list of arbitrary value $y_1,y_2,\ldots,y_n$
3. y_i is defined as $g_i(x_i)$ where g_i is a trap-door function
4. The function $C_{k,v}(y_1,y_2,\ldots,y_n)$ is called the ring equation, and is defined below

   $$C_{k,v}(y_1,y_2,\ldots,y_n) = E_k(y_n \oplus E_k(y_{n-1} \oplus E_k(y_{n-2} \oplus E_k(\ldots \oplus E_k(y_2 \oplus E_k(y_1 \oplus v))\ldots))))$$
5. Calculate using the signer's private key $x_s = g_s^{-1}(y_s)$

Types of Digital Signatures

Ring Signature – RSA based

Signature Generation

- Calculate the key $k = H(m)$ using a cryptographic hash function. m is the message, H is a hash function and k will be used as key for E_k
- Pick a random glue value v
- Pick random x_i for all ring members but yourself (x_s will be calculated using the signer's private key), and calculate corresponding $y_i = g_i(x_i)$
- Solve the ring equation

$$C_{k,v}(y_1, y_2, \ldots, y_n) = E_k(y_n \oplus E_k(y_{n-1} \oplus E_k(y_{n-2} \oplus E_k(\ldots \oplus E_k(y_2 \oplus E_k(y_1 \oplus v))\ldots)))) = v$$

- Calculate using the signer's private key $x_s = g_s^{-1}(y_s)$
- The ring signature now is the $(2n+1)$-tuple $(P_1, P_2, \ldots, P_n; v; x_1, x_2, \ldots, x_n)$

Types of Digital Signatures

Ring Signature – RSA based

Signature Verification

1. Apply the public key trap door on all $y_i = g_i(x_i)$
2. Calculate the symmetric key $k = H(m)$
3. Verify that the ring equation holds $C_{k,v}(y_1, y_2, \ldots, y_n) = v$

Types of Digital Signatures

Ring Signature – ECC based

Setup (Carry out by TA)

1. Let P be a generator of G
2. Choose a random number $s \in Z^*_q$ as the master key of TA
3. Set $P_{pub} = sP$
4. Define two cryptographic hash functions $H: \{0,1\}^* \rightarrow Z^*_q$ and $H_1 : \{0,1\}^* \rightarrow G^*$
5. The system parameters are $params = \{G, q, P, P_{pub}, H, H_1\}$
6. For given an identity , the corresponding public key is $Q_{ID} = H_1(ID)$ and the private key is $S_{ID} = sQ_{ID}$
7. Let $L = \{ID_i\}$

Types of Digital Signatures

Ring Signature – ECC based

Signature Generation

- (Initialization): Choose randomly an element $A \in G$, compute $c_{k+1} = H(L \| m \| e(A,P))$
- (Generate forward ring sequence): For $i=k+1, \cdots, n-1, 0, 1, \cdots, k-1$, choose randomly $T_i \in G$ and compute $c_{i+1} = H(L \| m \| e(T_i, P)\, e(c_i H_1(ID_i), P_{pub}))$
- (Forming the ring): Compute $T_k = A - c_k S_{IDk}$
- (Output the ring signature): Select 0 (i.e., n) as the glue value, the resulting signature for m and L is the $(n+1)$–tuple: $(c_0, T_0, T_1, \cdots, T_{n-1})$

Types of Digital Signatures

Ring Signature – ECC based

Signature Verification

- Compute $c_{i+1} = H(L \| m \| e(T_i, P)\, e(c_i H_1(ID_i), P_{pub}))$ for $i=0, 1, \cdots, n-1$
- Accept if $c_n = c_0$, and reject otherwise

Types of Digital Signatures

Ring Signature – ECC based

Signature Verification (Proof)

- From the procedure of ring signature generation, we have:
  - $c_{k+1} = H(L \| m \| e(A, P))$
  - $c_{k+2} = H(L \| m \| e(T_{k+1},P)\, e(c_{k+1}H_1(ID_{k+1}), P_{pub}))$
  - ...
  - $c_n = H(L \| m \| e(T_{n-1},P)\, e(c_{n-1}H_1(ID_{n-1}), P_{pub})) = c_0$
  - $c_1 = H(L \| m \| e(T_0,P)\, e(c_0H_1(ID_0), P_{pub}))$
  - $c_2 = H(L \| m \| e(T_1,P)\, e(c_1H_1(ID_1), P_{pub}))$
  - ...
  - $c_k = H(L \| m \| e(T_{k-1},P)\, e(c_{k-1}H_1(ID_{k-1}), P_{pub}))$

Types of Digital Signatures

Ring Signature – ECC based

Signature Verification (Proof)

- Since $T_k = A - c_k S_{IDk}$, in the procedure of ring signature verification, we have:
  - $c_{k+1} = H(L \| m \| e(T_k,P)\, e(c_k H_1(ID_k), P_{pub}))$
  - $= H(L \| m \| e(A - c_k S_{IDk},P)\, e(c_k H_1(ID_k), P_{pub}))$
  - $= H(L \| m \| e(A,P)\, e(-c_k S_{IDk},P)\, e(c_k H_1(ID_k), P_{pub}))$
  - $= H(L \| m \| e(A,P)\, e(-c_k H_1(ID_k),sP)\, e(c_k H_1(ID_k), P_{pub}))$
  - $= H(L \| m \| e(A,P)\, e(-c_k H_1(ID_k) + c_k H_1(ID_k), P_{pub}))$
  - $= H(L \| m \| e(A, P))$
- The sequence $\{c_i\}\ (i = 0, 1, \cdots, n-1)$ in the ring signature verification procedure is the same as the ring signature generation procedure, so we have $c_n = c_0$

Ref

1. Cryptography Protocols Course, Dr. Hamid Mala, University of Isfahan
2. “How to Make Efficient Fail-stop-Signatures”, Eugkne van Heyst and Torben Pryds Pedersen
3. “ID-Based Blind Signature and Ring Signature from Pairings”, Fangguo Zhang and Kwangjo Kim
4. https://medium.com/asecuritysite-when-bob-met-alice/ring-signatures-and-anonymisation-c9640f08a193
5. https://en.wikipedia.org/wiki/Ring_signature
6. https://en.wikipedia.org/wiki/Trapdoor_function
7. https://delbaraneh.com/
8. https://www.iconfinder.com/UsersInsights
9. https://www.iconfinder.com/Chanut-is
10. https://www.iconfinder.com/iconsets/softwaredemo