Blockchain Technology and Applications

CS 731

Cryptographic Techniques for Blockchain

Digital Signatures

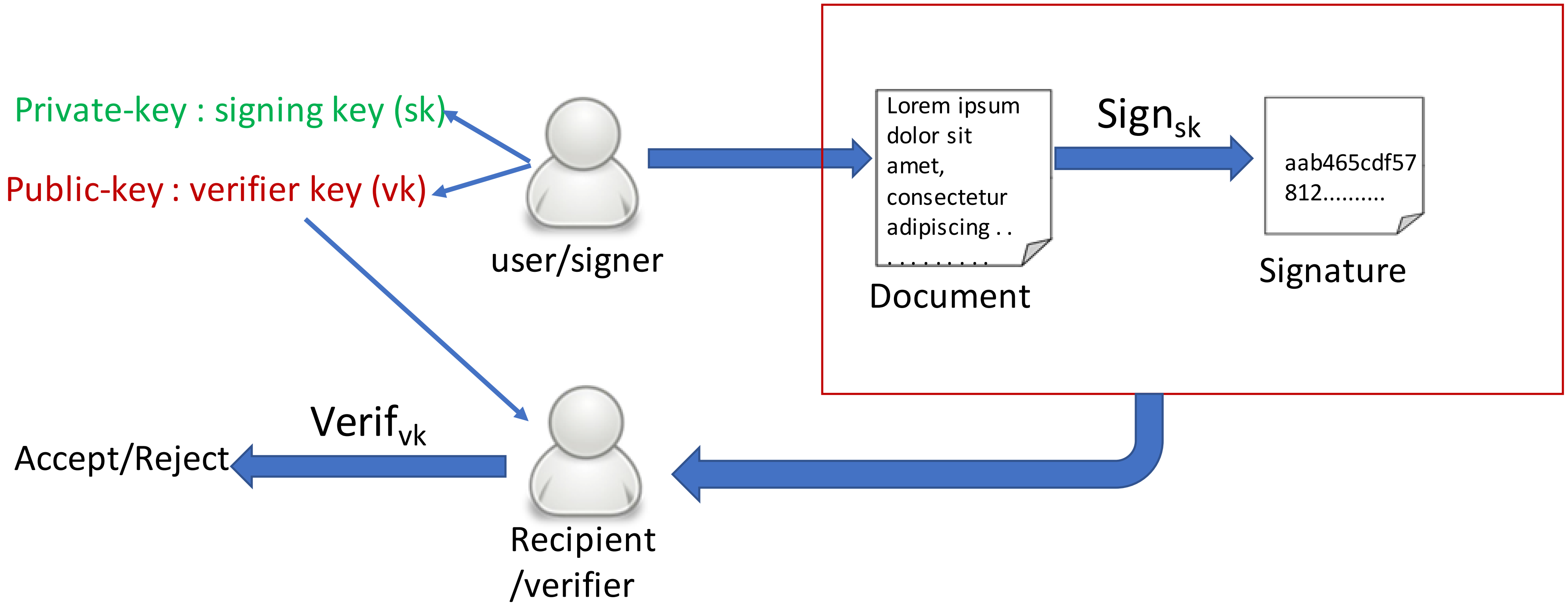
Dr. Ir. Angshuman Karmakar

IIT Kanpur

Teaching assistants

- **Sumit Lahiri** (sumitl@cse.iitk.ac.in)
- **Chavan Sujeet** (sujeetc@cse.iitk.ac.in)
- **Indranil Thakur** (indra@cse.iitk.ac.in)

Digital signatures



- *Emulates* ink-paper signatures

Digital signatures

- Each vk uniquely associates with each signing key sk
- Resistance against forgery
  - Sign(sk, m) ≠ Sign(sk, m') if m ≠ m'
  - Sign(sk, m) ≠ Sign(sk', m) if sk ≠ sk'
- Correctness
  - Verif (vk, m, Sign(sk,m))=1 except with negligible probability

- Applications
  - Authenticity : assures the identity of the signee
  - Non-repudiation : signee cannot deny ownership later
  - Integrity : Difficult to create same signature for two different messages
    - Although we use hash for this

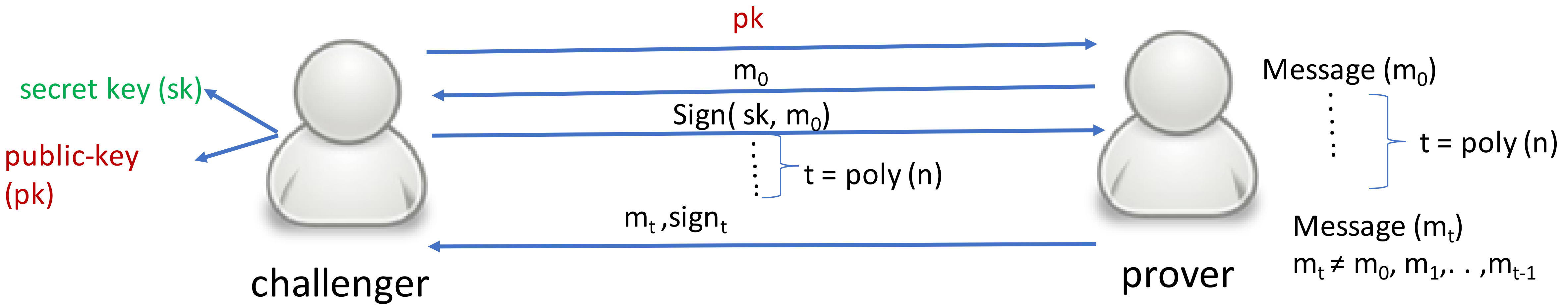
Digital signatures

Notions of security

- Game between *challenger (user)* and *prover (attacker)*
  - Captures different real-life scenarios
  - Stronger security notions gives more freedom to attackers
- For signatures one strong security notion is EUF-CMA
  - **E**xistential **U**nforgeability under **C**hosen **M**essage **A**ttack
- Imagine a passive attacker listens to all the communication
  - It knows all message and signature pair
  - Attacker *should not* be able to produce a valid message-signature pair based on this

Digital signatures

Notions of security



- Prover wins if verif(pk, m_t ,$sign_t$)=1 with non-negligible propoerty
- Not secure under EUF-CMA

Digital signatures

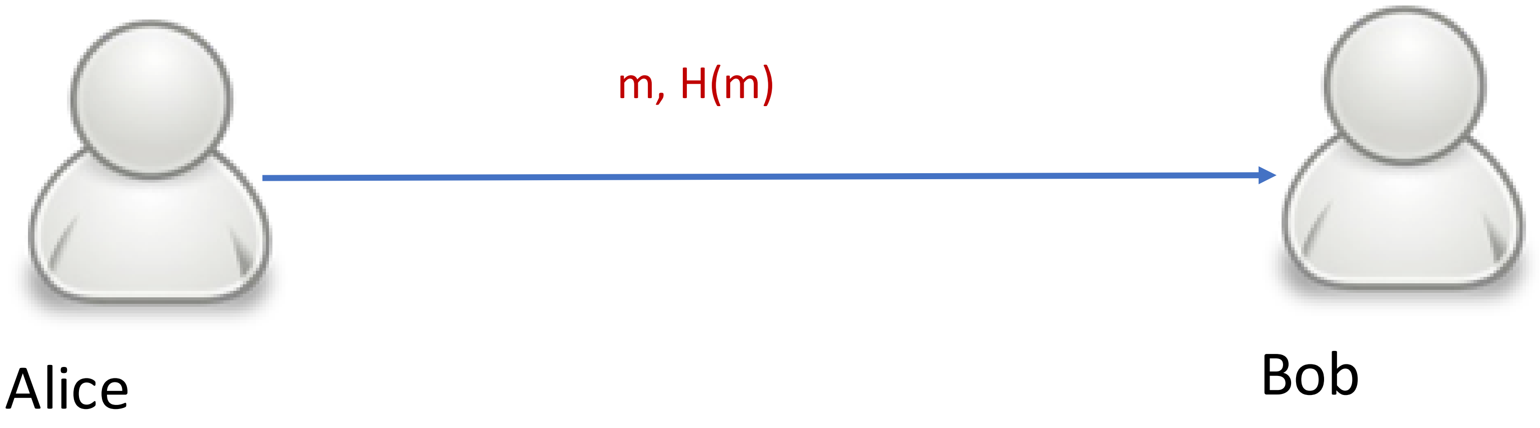
Notions of security

- SUF-CMA
  - **S**trong Existential **U**nf**o**rgeability under **C**hosen **M**essage **A**ttack
  - Stronger notion than EUF-CMA
- To prevent *malleable* signature schemes
  - For a same message, the adversary can *tweak* the signature to produce another valid signature
  - For example, if an attacker can re-randomize a valid signature such that the signature remains valid then it is not SUF-CMA secure
- Bitcoin uses **E**lliptic **C**urve **D**iscrete **S**ignature **A**lgorithm
  - Earlier versions was malleable
- Improvements proposed in BIP 0146, BIP 0340[1]

[1] https://github.com/bitcoin/bips/blob/master/bip-0146.mediawiki https://en.bitcoin.it/wiki/BIP_0340

Digital signatures

Hashing and signature



- Alice sends the message and its hash to Bob
- Eve can intercept the message and replace with m’ and H(m’)
- Bob has no way to realize that this happened

Hash and sign

- Or Sign and hash
  - Digital signatures have small payload
  - Signatures have small payload $m \dashrightarrow m_1 || m_2 || \ldots || m_t$
    - $Sign(m_1) || Sign(m_2) || \ldots || Sign(m_t)$
    - $h(Sign(m_1)) || h(Sign(m_2)) || \ldots || h(Sign(m_t))$
    - Adversary can create $m' \dashrightarrow m_i || m_j || \ldots || m_p$
    - Valid signature $h(Sign(m_i)) || h(Sign(m_j)) || \ldots || h(Sign(m_p))$
  - More bandwidth
  - More operations
  - Less secure

Hash and sign

- m--> h(m) $\underset{\text{256 bits}}{\text{-->}}$ sign(h(m))
- Less bandwidth
- Less signatures
- Removes some existential forgery attacks e.g. textbook RSA
- Also, assures the integrity of the message
- Extra reading : Email encryption using GNU PGP

SOME DEFINITIONS

$\mathbb{Z}$: The set of integers $=\{\ldots,-2,-1,0,1,2,\ldots\}$

$\mathbb{Z}_n$: $\{0,1,2\ldots,n-1\}$

$\mathbb{Z}_n^*$: $\{a \in \mathbb{Z} : 0 < a < n \;\texttt{and}\; gcd(a,n)=1\}$

$\phi(n)$: The number of elements in $\mathbb{Z}_n^*$

$a \equiv b \mod n$: $(a-b)$ is divisible by n

Euler's theorem: If $gcd(a,n)=1$, then $a^{\phi(n)} \equiv 1 \mod n$.

Fermat's Little theorem: If n is a prime number, then for any integer a, $a^n \equiv a \mod n$.

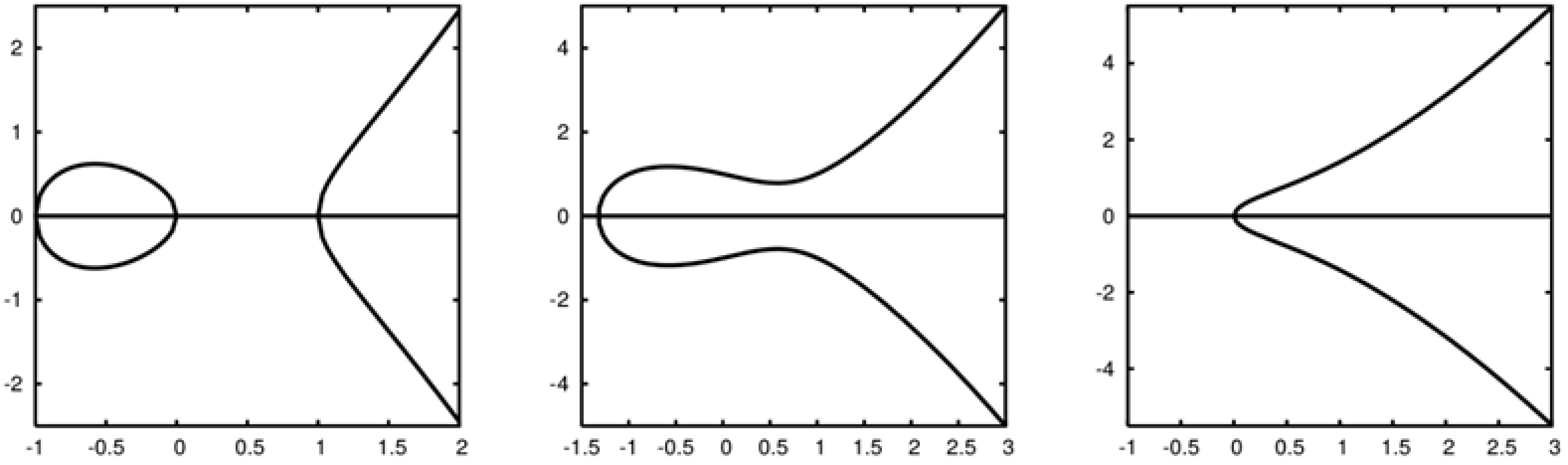
If n is a prime number, then there exists an element g in $\mathbb{Z}_n^*$ such that $\mathbb{Z}_n^* = \{g^i : i = 0,1,2\ldots\}$.
We call this element g as generator of $\mathbb{Z}_n^*$.

ELLIPTIC CURVE

- Let $\mathbf{F_p}$ be a field with characteristic other than 2 and 3.
- We define a **elliptic curve E** over $\mathbf{F_p}$ as follows

$$E = \{(x, y) : y^2 = x^3 + ax + b\} \cup \mathcal{O}, \text{ where } a, b \text{ are constants taken from } F_p \text{ such that } 4a^3 + 27b^2 \neq 0$$

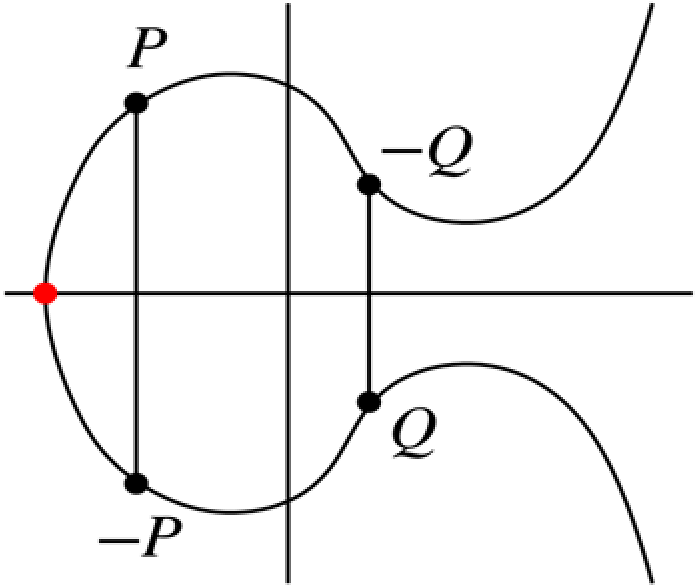
- We call **O** as a point of infinity and the others points are called finite points.

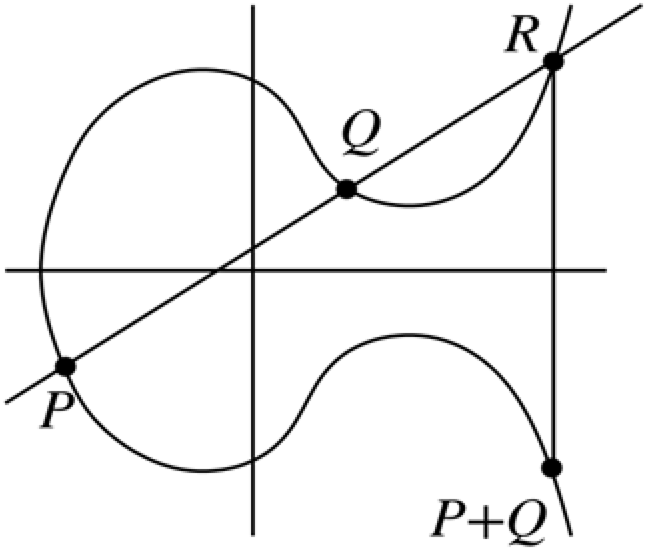


(a) $Y^2 = X^3 - X$ (b) $Y^2 = X^3 - X + 1$ (c) $Y^2 = X^3 + X$

GROUP OPERATION

- We define a group operation addition ("+") on the elliptic curve **E** as follows
- A + O = O + A = A for all A in **E. (Identity property)**
- For each A = (x, y) (≠ O) in **E**, the point –A = (x, -y) also lie in **E** and A + (-A) = (-A) + A = O. (**Inverse property**)





GROUP OPERATION

- For any two distinct points A = (x_1, y_1) and B = (x_2, y_2) ($\neq$ -A) in **E**, A + B = C = (x_3, y_3) where

$$x_3 = (\frac{y_2 - y_1}{x_2 - x_1})^2 - x_1 - x_2$$

$$y_3 = (\frac{y_2 - y_1}{x_2 - x_1})(x_1 - x_3) - y_1$$

- For a point A = (x_1, y_1) (A $\neq$ -A) in **E**, 2A = A+A = (x_3, y_3) where

$$x_3 = (\frac{3x_1^2 + a}{2y_1})^2 - 2x_1$$

$$y_3 = (\frac{3x_1^2 + a}{2y_1})(x_1 - x_3) - y_1$$

- For any points A, B ,C in E, A + (B + C) = (A + B) + C holds. (**Associativity**)
- For any integer n, we define nA by nA = A + A +…+ A (n times), where A is in **E**.

ECDSA signature

ECDSA Keygen

1. G <-- Generator of the elliptic-curve group
2. n <-- Order of the group *i.e.* n*G--> 0 (identity)
3. q <-- private key
4. R=q*G <-- public-key
5. m <-- message to send

ECDSA Signature

1. z <-- Hash(m) take z = m for simplicity
2. k <-- random [1,n]
3. r <-- k*G
4. s <-- k^-1 (z+ r*q)
5. (x_1, y_1)= (r,s) is a signature of m

ECDSA Verification

1. u_1<-- z*s^-1
2. v_1 <-- r*s^-1
3. x <-- u_1 * G + u_2 * R
4. Accept if r equals x

ELLIPTIC CURVE DIGITAL SIGNATURE ALGORITHM

Correctness :

$$\begin{aligned} S &= u_1 G + u_2 R \\ &= u_1 G + u_2 \cdot qG \\ &= (u_1 + u_2 \cdot q)G \\ &= (z \cdot s^{-1} + r \cdot s^{-1} \cdot q)G \\ &= (z + r \cdot q) \cdot s^{-1} G \\ &= (z + r \cdot q) \cdot (k \cdot (z + r \cdot q)^{-1})G \\ &= kG \end{aligned}$$

If, $S = (x_1, y_1)$, r should be x_1.

Examples

- Sony's PS3 hack
  - https://medium.com/asecuritysite-when-bob-met-alice/not-playing-randomly-the-sony-ps3-and-bitcoin-crypto-hacks-c1fe92bea9bc
- K is not only secret but chosen randomly for each signature
- Keeping k constant leaks the long-term signing-key
- Should be chosen from a good-quality RNG
- After seeing some signatures secret-key can be leaked
- Bitcoin wallets on Android OS
  - https://bitcoin.org/en/alert/2013-08-11-android,
  - https://par.nsf.gov/servlets/purl/10174436
- Deterministic ECDSA: k is derived from the message and private-key

Digital signatures from hash

- Lamport one time signature scheme
- KeyGen :

$$sk = \begin{pmatrix} x_1^0, x_2^0, x_3^0, \ldots\ldots\ldots\ldots\ldots, x_n^0 \\ x_1^1, x_2^1, x_3^1, \ldots\ldots\ldots\ldots\ldots, x_n^1 \end{pmatrix} \quad x_i^j \in_{\$} \{0,1\}^n$$

$$pk = \begin{pmatrix} y_1^0, y_2^0, y_3^0, \ldots\ldots\ldots\ldots\ldots, y_n^0 \\ y_1^1, y_2^1, y_3^1, \ldots\ldots\ldots\ldots\ldots, y_n^1 \end{pmatrix} \quad y_i^j = h(x_i^j),\ j = \{0,1\}, i = [1,n]$$

- Sizes :
  - sk : 256 x 256 x 2 = 128 Kbits
  - pk : 256 x 256 x 2 = 128 Kbits

Digital signatures from hash

- Signing :
  - Hash message $h(m)=b_1b_2b_3. \dots .b_n$
  - Output signature as : $x_1^{b1}, x_2^{b2}, x_3^{b3}, \dots, x_n^{bn}$
  - Shows part of the secret key
  - Example for h(m) = 0100……1

$$\text{signature} = \begin{pmatrix} \boxed{x_1^0}, x_2^0, \boxed{x_3^0}, \dots\dots\dots\dots\dots, x_n^0 \\ x_1^1, \boxed{x_2^1}, x_3^1, \dots\dots\dots\dots\dots, \boxed{x_n^1} \end{pmatrix}$$

- Sizes :
  - signature : 256 x 256 = 64 Kbits

Digital signatures from hash

- Verification
  - Hash message $h(m)=b_1b_2b_3. \ . \ . \ . \ .b_n$
  - For i= 1 to n check
    - $h(\text{Signature}_{bi}) \stackrel{?}{=} y_i^{bi}$

- Security is guaranteed by the security of the hash function
- Forgery or secret-key recovery both requires inverting the hash function

Digital signatures from hash

- Huge key-sizes
- Can be used only once
- Post-quantum secure
  - RSA and ECDSA are not
- Improvement (extra reading) :
  - Merkle-Lamport signatures (Merkle, Ralph (1979). Secrecy, authentication and public key systems)
  - SPHINCS signature (https://sphincs.org/resources.html)

Further reading

1. Cryptography : An Introduction. Nigel P Smart
2. Cryptography : Theory and practice. Douglas R Stinson

The end !!