One Time Signatures

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY SRI CITY

CHITTOOR, INDIA

Lamport one-time signature

- Key Gen
- Sign msg
- Ver: Sign

Key Gen:

→ cryptographic Hash function

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

→ $m \in (0,1)^k$

→ choose $2k$ random numbers

$$x_{ij}, \quad 1 \le i \le k \;\&\; j = \{0,1\}$$

→ for each $i \;\&\; j$, compute

$$y_{ij} = H(x_{ij})$$

private key

$$x = (x_{ij})$$

public key

$$y = (y_{ij})$$

Signing on message:

Suppose $M = m_1 m_2 \cdots m_k$, $m_i \in \{0,1\}$

$$Sig_i = \begin{cases} X_{i0}, & \text{if } m_i = 0 \\ X_{i1}, & \text{if } m_i = 1 \end{cases}$$

$$Sig = (Sig_1 \| Sig_2 \| \cdots \| Sig_k)$$

Signature Verification: Given (M, Sig) and (Y_{ij})

$$H(Sig_i) = \begin{cases} Y_{i0}, & \text{if } m_i = 0 \\ Y_{i1}, & \text{if } m_i = 1 \end{cases}$$

if above is true, the signature is valid.

Complexity:-

→ 2K hashes.

Note:- for $O(2^{80})$ security, hash value must atleast 160 bits.

→ $|n| = k = 160$

→ $|x| = |y| = 160 * 2K$

$= 320K$ bits

$= 51200$ bits

$= 6400$ bytes.

* Equivalent 1024-bit RSA public key size 50 times less.

* Signature size

$sig = (sig_1 || sig_2 || \cdots || sig_k)$

$|sig| = 160K$

$= 25600$ bits

$= 3200$ bytes

⇒ 25 times larger than RSA sign.

Winternitz one-time signature

* Reduced the signature size.

private key $X = (X_1 \| X_2 \| \cdots \| X_L)$

public key $Y = H(Y_1 \| Y_2 \| \cdots \| Y_L)$

Key Gen :

→ Cryptographic hash function

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

→ choose a parameter $w \in \mathbb{N}$

→ compute $L = \lceil n/w \rceil + \lfloor (L \log \lceil n/w \rceil + 1 + w)/w \rfloor$

→ choose L random numbers $x_1, x_2, \cdots x_L \in \{0,1\}^n$

→ Compute $Y_i = H^{2^w - 1}(x_i)$ for $i = 1, \ldots, L$

Signature generation:

Message $M = m_1 m_2 \cdots m_s$, $m_i \in \{0, 1\}$

private key $x_1, x_2, \cdots x_t$

parameters w, t

Note: If necessary, the message is padded with zeros from left.

→ M split up into $\lceil s/w \rceil$ blocks $b_1, b_2, \cdots, b_{\lceil s/w \rceil}$ of length w

→ Assume that b_i is integer encoded by respective block.

→ Compute checksum $C = \sum_{i=1}^{\lceil s/w \rceil} 2^w - b_i$

→ Split binary representation of C into $\lceil (L \log_2 \lceil s/w \rceil + 1 + w)/w \rceil$ blocks, $b_{\lceil s/w \rceil + 1}, \cdots, b_t$ of length w

→ b_i as an integer encoded by the block b_i

→ Compute $sig_i = H^{b_i}(x_i)$ for $i = 1, \ldots, t$, $H^0(x_i) = x_i$

$sig = (sig_1 \| sig_2 \| \cdots \| sig_t)$

$|sig| = t * s$

$\approx s/w$

Signature Verification: $m = \{0,1\}^s$

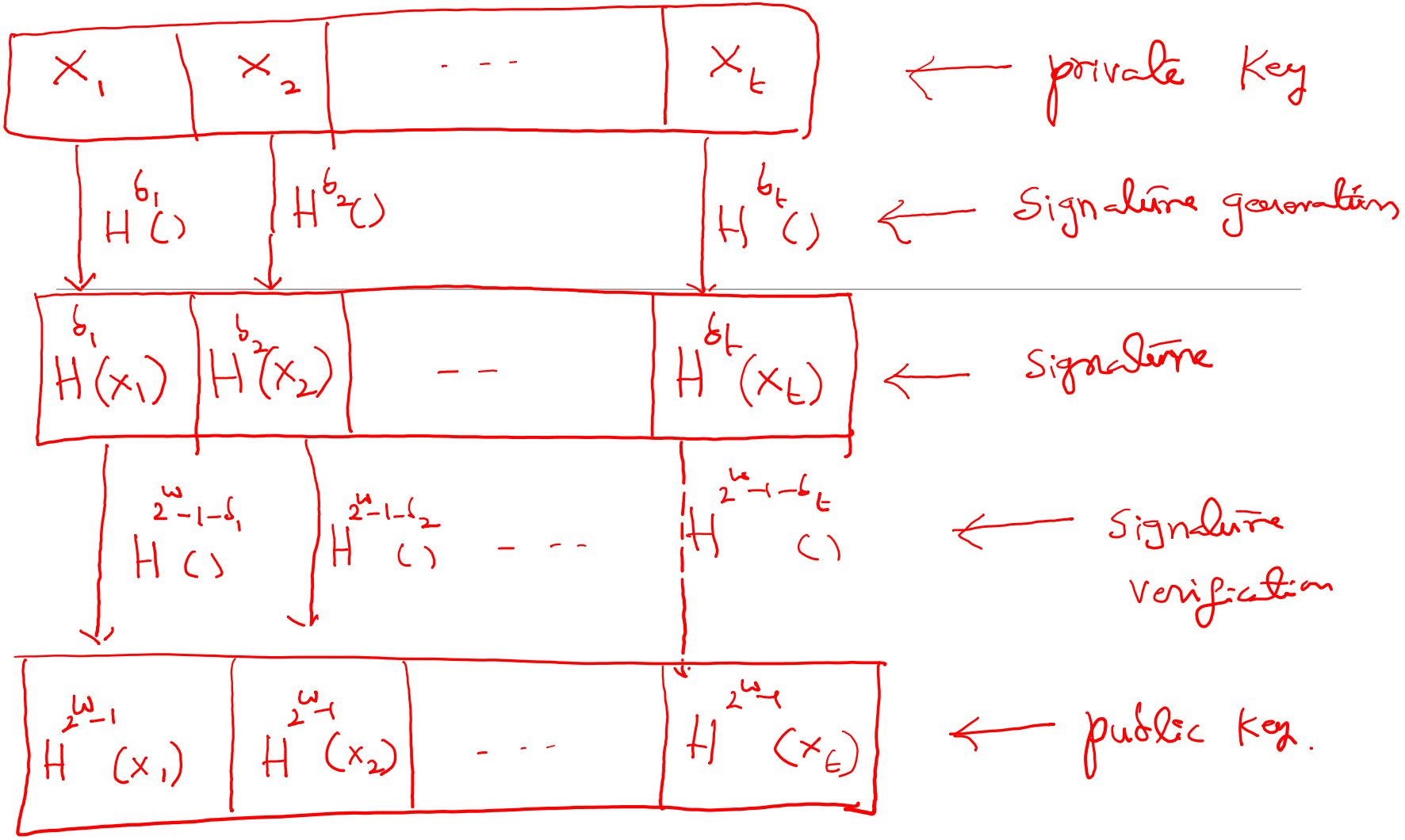
parameters $b_1, b_2, \ldots, b_t$ compute as in the signature.

$sig = (sig_1 \| sig_2 \| \cdots \| sig_t)$

→ Compute $sig_i' = H^{2^w - 1 - b_i}(sig_i) = H^{2^w - 1 - b_i}(H^{b_i}(x_i))$

$= H^{2^w - 1}(x_i) = y_i$

→ If $y' = H(sig_1' \| sig_2' \| \cdots \| sig_t') = y$, then valid.


x_1
x_2
- - -
x_t
← private key
$H^{b_1}()$
$H^{b_2}()$
$H^{b_t}()$
← Signature generation
$H^{b_1}(x_1)$
$H^{b_2}(x_2)$
- -
$H^{b_t}(x_t)$
← Signature
$H^{2^w-1-b_1}()$
$H^{2^w-1-b_2}()$ - - -
$H^{2^w-1-b_t}()$
← Signature verification
$H^{2^w-1}(x_1)$
$H^{2^w-1}(x_2)$
- - -
$H^{2^w-1}(x_t)$
← public key.

Drawbacks:

→ We can not use public key for more signatures.

→ size of keys and signature is large.

Summary: Computational, storage, communication overhead

Next class: Merkle-Signature scheme