Hash-Based Signatures

Johannes Buchmann, Andreas Hülsung
Supported by DFG and DAAD

Part IV: XMSS

CASED

XMSS:
A practical signature scheme with minimal security assumptions

J.B., Carlos Coronado, Erik Dahmen, Andreas Hülsing

XMSS (2006-2013)

CASED

CMSS – An Improved Merkle Signature Scheme

Johannes Buchmann[1], Luis Carlos Coronado García[2], Erik Dahmen[1], Martin Döring[1,⋆], and Elena Klintsevich[1]

Merkle tree traversal revisited

Johannes Buchmann, Erik Dahmen, and Michael Schneider

Merkle Signatures with Virtually Unlimited Signature Capacity

Johannes Buchmann[1], Erik Dahmen[1], Elena Klintsevich[1], Katsuyuki Okeya[2], and Camille Vuillaume[2]

Forward Secure Signatures on Smart Cards*

Andreas Hülsing, Christoph Busold, and Johannes Buchmann

On the Security of the Winternitz One-Time Signature Scheme

Johannes Buchmann, Erik Dahmen, Sarah Ereth, Andreas Hülsing*, and Markus Rückert**

Optimal Parameters for XMSS^{MT}

Andreas Hülsing*, Lea Rausch, and Johannes Buchmann

W-OTS^{+}– Shorter Signatures for Hash-Based Signature Schemes

Andreas Hülsing*

XMSS – A Practical Forward Secure Signature Scheme based on Minimal Security Assumptions

Johannes Buchmann, Erik Dahmen, and Andreas Hülsing*
{buchmann,dahmen,huelsing}@cdc.informatik.tu-darmstadt.de

XMSS Security

CASED

Security parameter n

Requires family of functions $\mathcal{F} : \{0,1\}^n \times \{0,1\}^n \rightarrow \{0,1\}^n$

Requires family of functions $\mathcal{K} : \{0,1\}^{2n} \times \{0,1\}^n \rightarrow \{0,1\}^n$

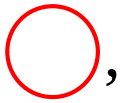
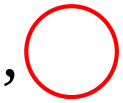
Theorem:

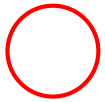
XMSS is existentially unforgeable under adaptive chosen message attacks if $\mathcal{F}$ is ***pseudorandom*** *and $\mathcal{K}$ is* ***second preimage resistant***

XMSS uses Winternitz OTS

CASED

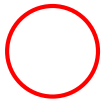
Security level b

SIG = (*i*,  ,  ,  ,  , )

|  | = |  | = m * |  | = m*b

1.  = f()

2. Trade-off between runtime and signature size

|  | ~ m/log w * |  |

XMSS Secret Key Size

CASED

Security level b, tree height h, message length m

MSS

Secret key size: $2bm*2^h$

XMSS

- OTS signature keys computed using pseudorandom function $\mathcal{F}$
- XMSS secret key only a seed: b
- Secret key size independent of message length!

Spoiler: State needs additional storage! (both cases)

XMSS Public Key Generation

CASED

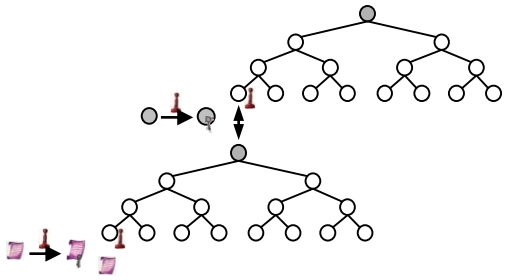
Security level b, tree height h

MSS

- Generate tree of size 2^h
- Cost ~ 2^h

XMSS

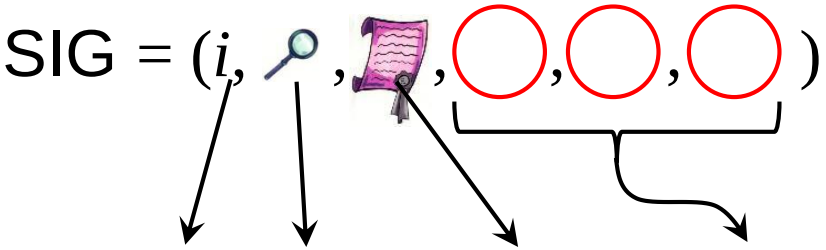
- Tree chaining
- Use t layers of trees of height h/t
- Generate t trees of height $2^{h/t}$
- Cost ~ $t*2^{h/t}$
- Example: h = 40, t = 2, costs ~ $2*2^{20} = 2^{21}$
- Slightly increased signature size



MSS Signature Size

CASED

Security level b, tree height h, message length m

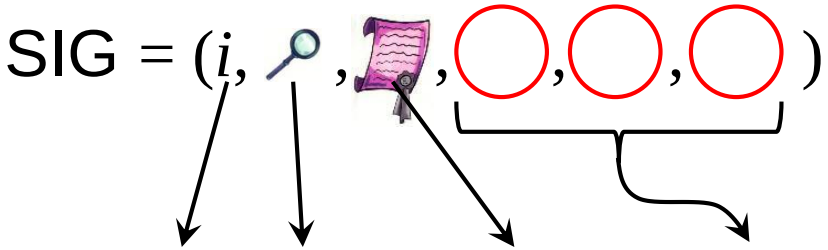


h + 2bm + bm + h*2b = h (2b+1) + 3bm

XMSS Signature Size

CASED

Security level b, tree height h, message length m

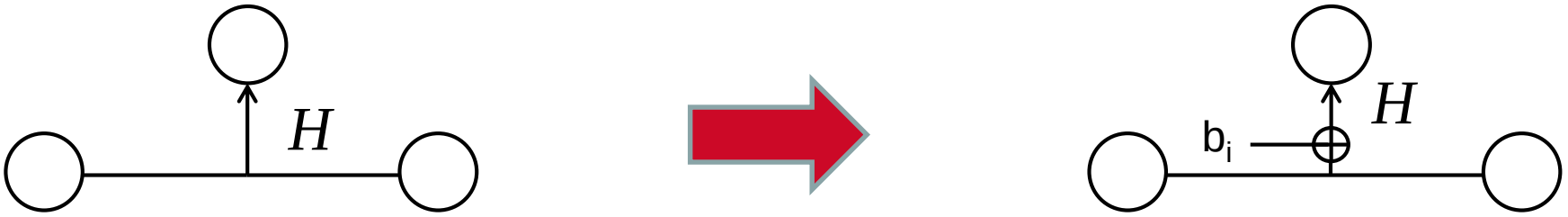


MSS: $h + 2bm + bm + h*2b = h(2b+1) + 3bm$

$h + 2bm + bm + h*b = h(b+1) + 3bm$

Modify tree construction:

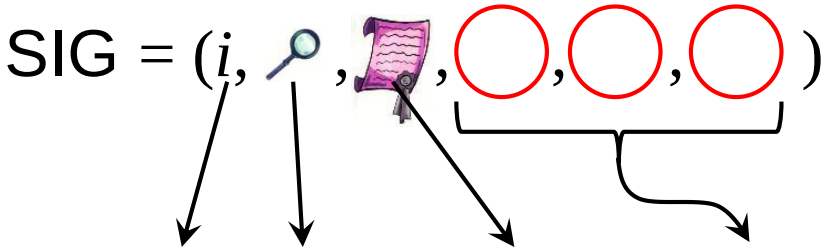
- Allows to use second preimage resistant hash function:
- Output length = node size 2n -> n



XMSS Signature Size

CASED

Security level b, tree height h, message length m



MSS: h + 2bm + bm + h*2b = h (2b+1) + 3bm

h + 2bm + bm + h*b = h (b+1) + 3bm

XMSS: h + 0 + (bm/log w) + h*b = h (b+1) + (bm/log w)

Use WOTS:

- OTS public key can be computed from signature
- Size – Runtime Trade-Off

XMSS Authentication Path Computation

CASED

Tree height h

Naïve: requires computing or storing 2^h nodes

Use BDS Algorithm: time - memory trade-off controlled by parameter k

- Time: $\sim (h-k)/2$ (per signature)
- Memory: $\sim (5.5h - 5k - 2^k)$