Practical Forward Secure Signatures using Minimal Security Assumptions

Andreas Hülsing, Erik Dahmen, Johannes Buchmann

Digital Signatures are Important!

CASED

Software updates

amazon

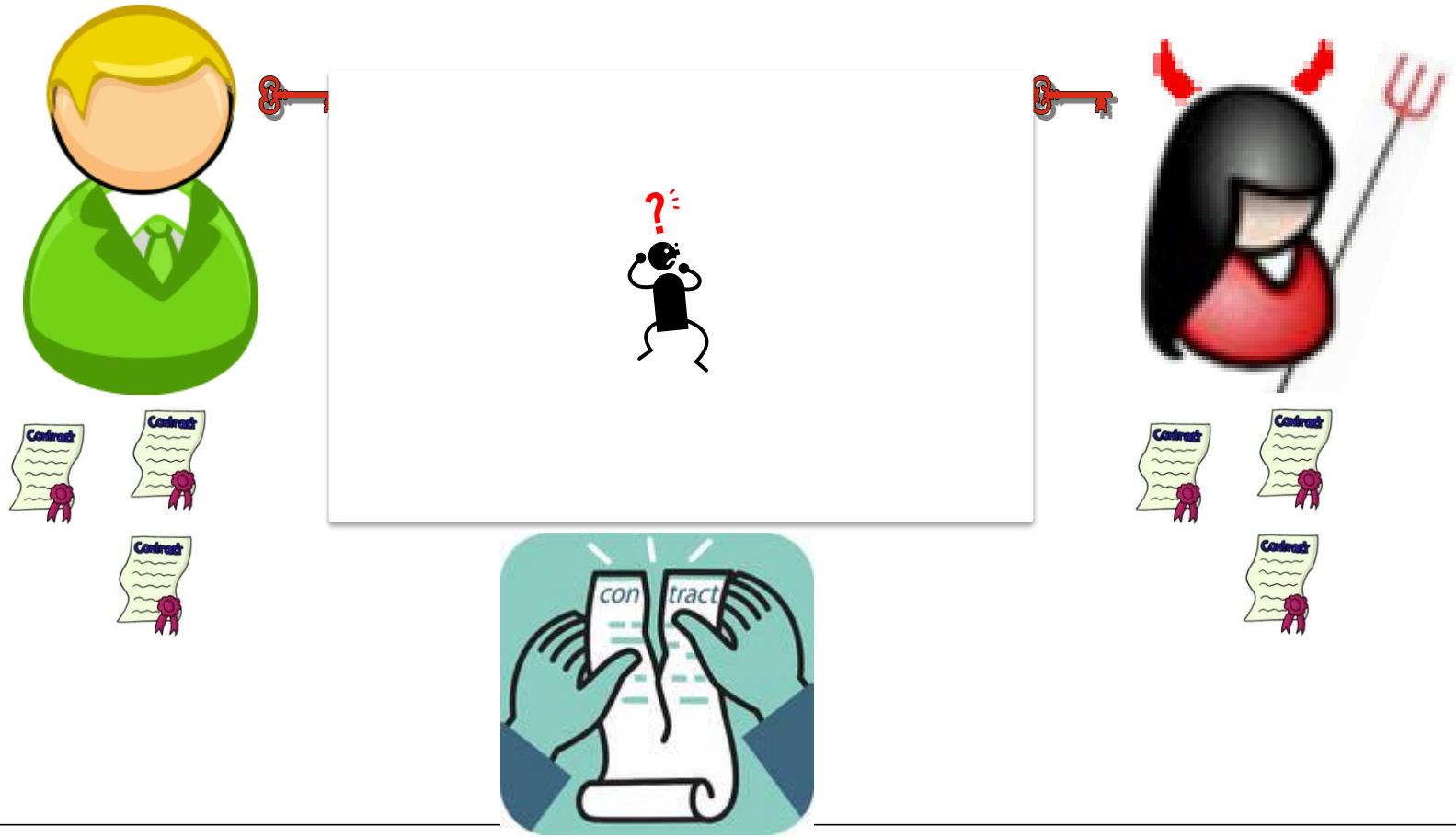
E-Commerce

ebay™

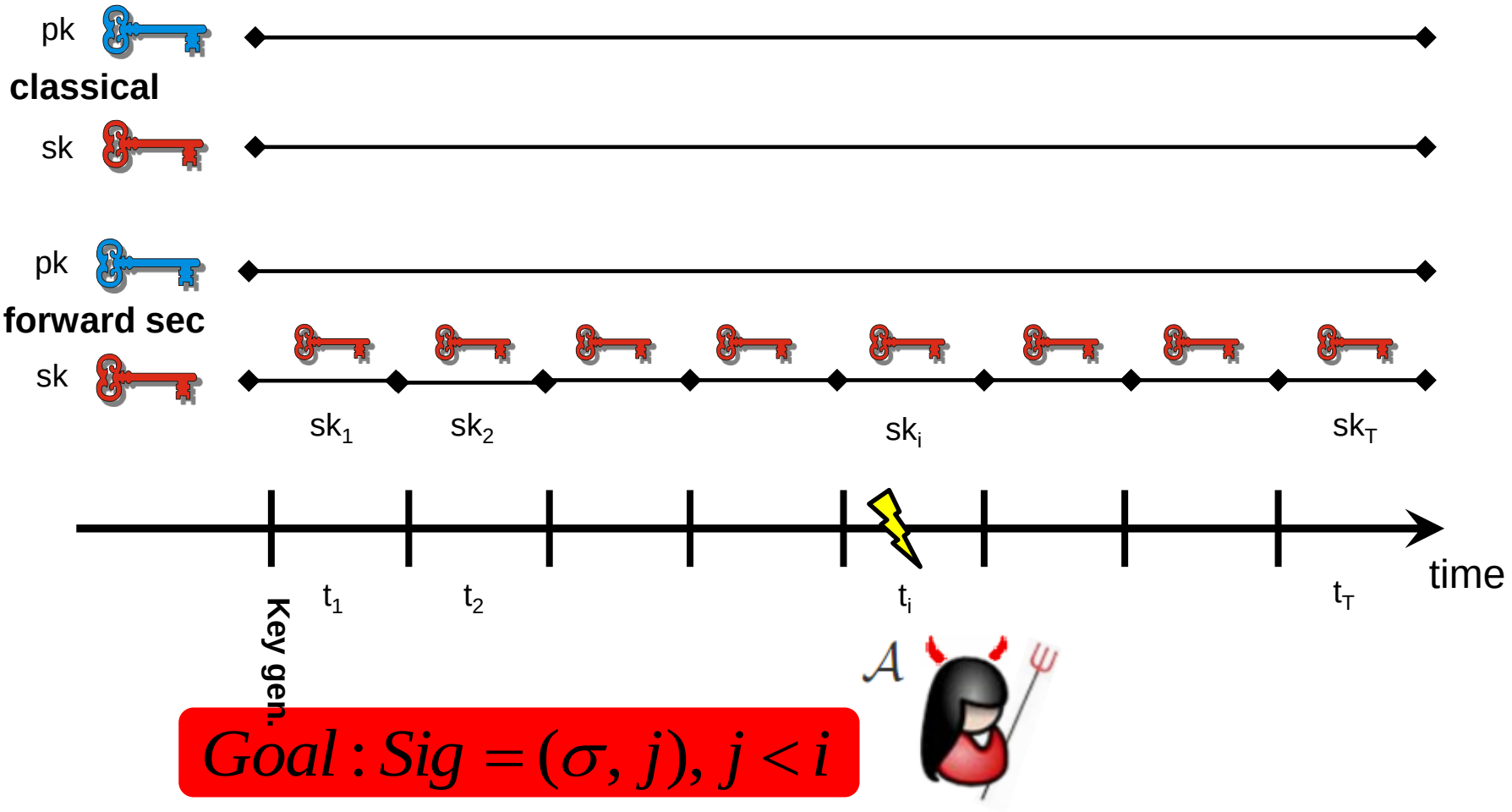
… and many others

Forward Secure Signatures
[And97]

CASED

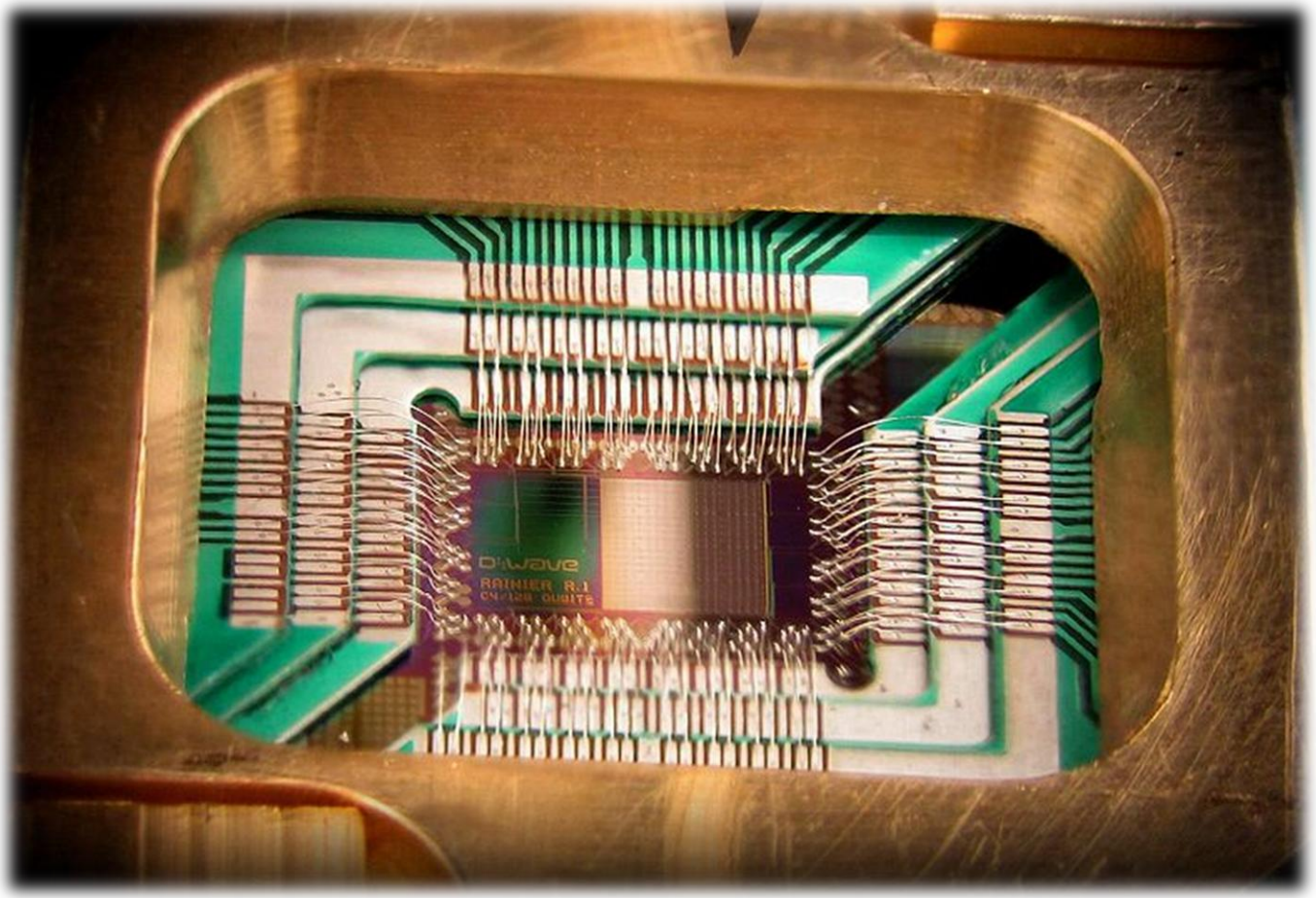


Forward Secure Signatures



What if…

CASED

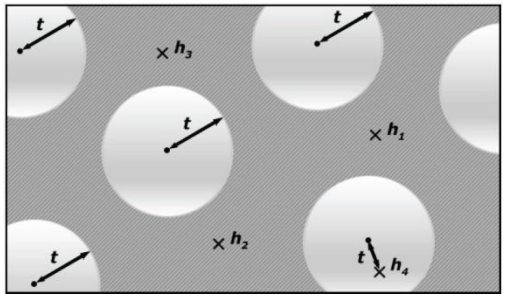


Post-Quantum Signatures

CASED

Lattice, MQ, Coding

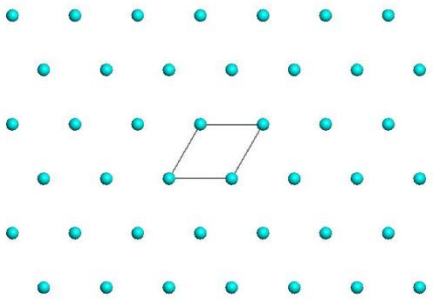
- Signature and/or key sizes
- Runtimes
- Secure parameters
- no forward secure signatures



$$y_1 = x_1^2 + x_1x_2 + x_1x_4 + x_3$$

$$y_2 = x_3^2 + x_2x_3 + x_2x_4 + x_1 + 1$$

$$y_3 = \ldots$$



Hash-based Signature Schemes
[Mer89]

CASED

- Post quantum
- Only secure hash function
- Security well understood
- Fast
- Forward secure (inefficient)

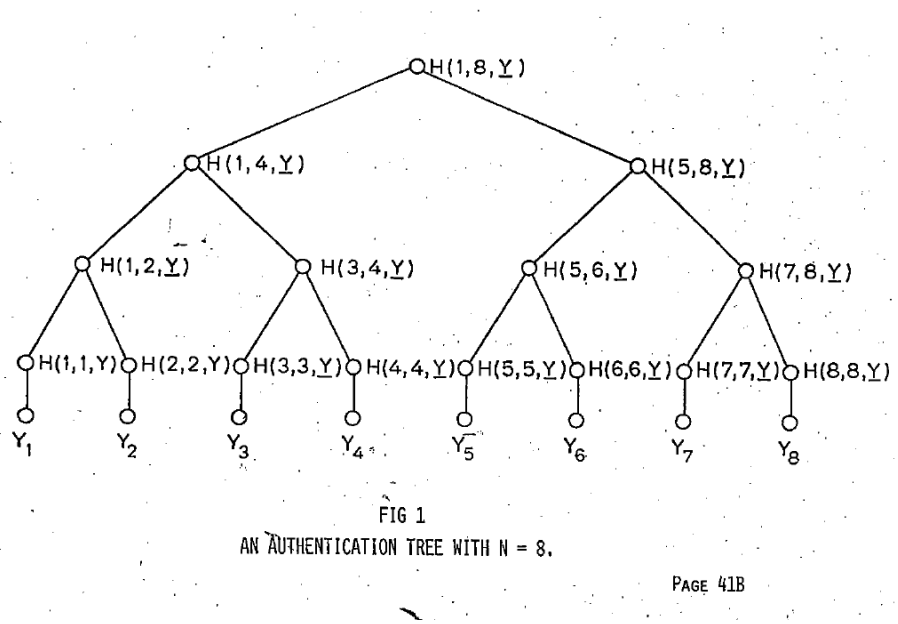


FIG 1
AN AUTHENTICATION TREE WITH N = 8.

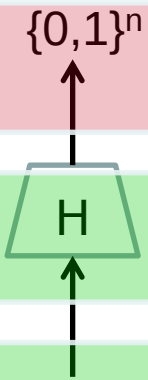
PAGE 41B

Cryptographic Hash Functions

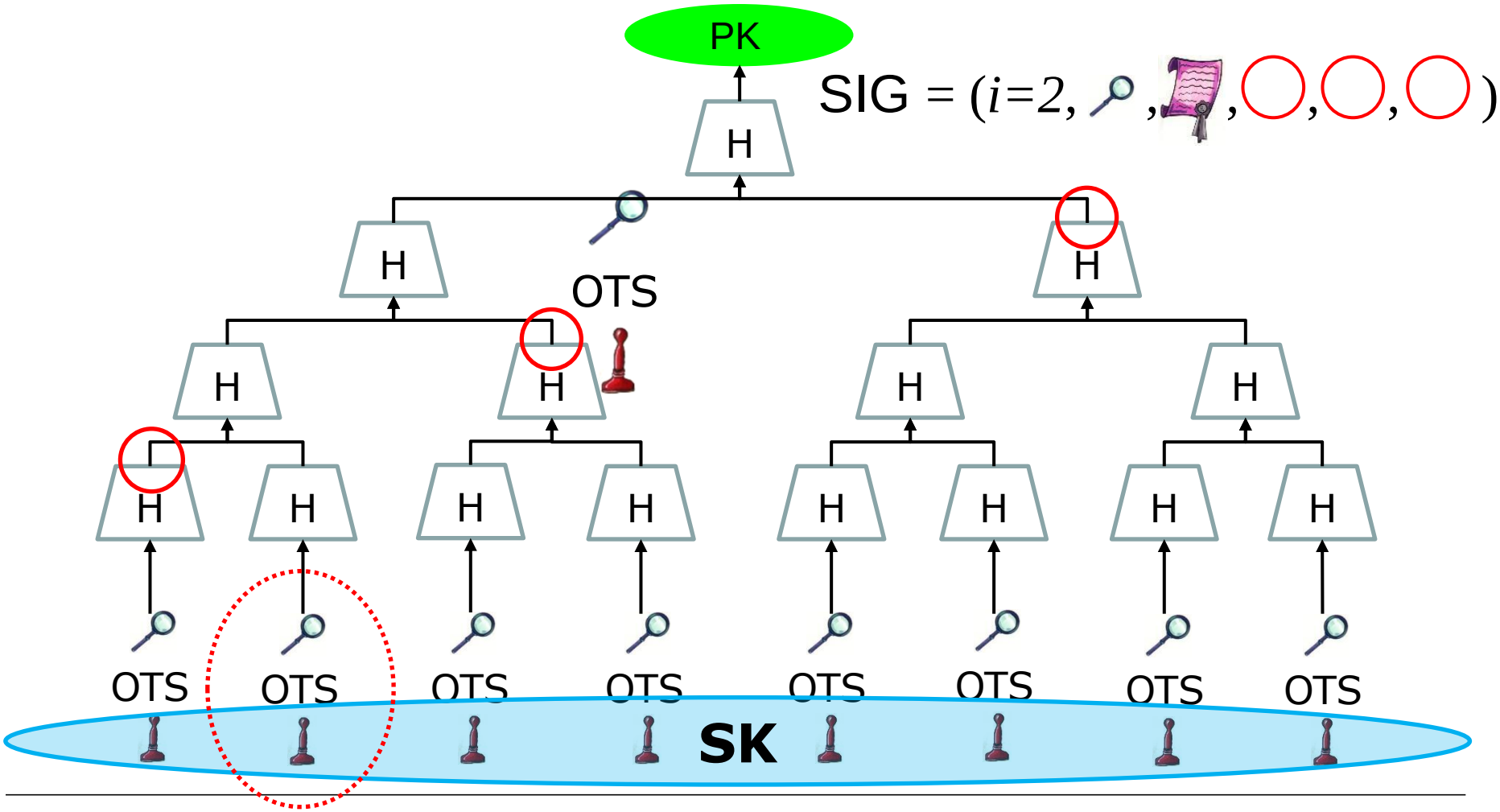
CASED

$$\mathcal{H}_n = \{H_K : \{0,1\}^m \rightarrow \{0,1\}^n \mid K \in \{0,1\}^{n'}\}$$

Collision Resistance (CR)	• Cryptomania • AC $\mathcal{O}(2^{n/2})$
Second-preimage Resistance (SPR)	• Minicrypt • AC $\mathcal{O}(2^n)$
One-wayness	• Minicrypt • AC $\mathcal{O}(2^n)$
Undetectability (UD)	• Minicrypt • AC $\mathcal{O}(2^n)$
Pseudorandomness (PRF)	• Minicrypt • AC $\mathcal{O}(2^n)$



Hash-based Signatures



Challenges & Achievements

CASED

Minimal security assumptions XOR Efficient

Forward secure XOR Efficient

Large signatures

No full smartcard implementation

Efficient

Minimal security assumptions

„Small signatures"

Forward secure

Full smartcard implementation

CASED

New Variants of the Winternitz One Time Signature Scheme

OTS

Winternitz OTS (WOTS)
[Mer89; EGM96]

CASED

SIG = (i,  ,  , ◯, ◯, ◯)

|  | = |  | = m * | ◯ |

1.  = f()

2. Trade-off between runtime and signature size

|  | ~ m/log w * | ◯ |

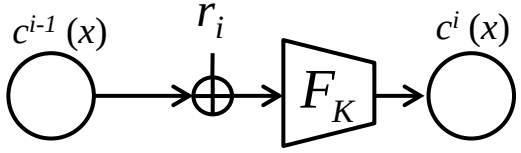
WOTS
Function Chain

Function family: $\mathcal{F}_n = \{F_K : \{0,1\}^n \rightarrow \{0,1\}^n \mid K \in \{0,1\}^{n'}\}$

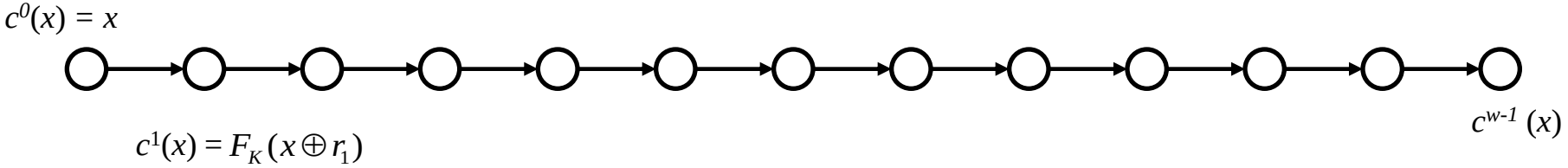
Formerly: $c^i(x) = F_K(c^{i-1}(x)) = \underbrace{F_K \circ F_K \circ \ldots \circ F_K(x)}_{i-times}, K \in \{0,1\}^{n'}$

WOTS+

For $w \geq 2$ select $\mathcal{R} = (r_1, \ldots, r_{w-1}) \in \{0,1\}^{n \times w-1}, K \in \{0,1\}^{n'}$



$$c^i(x) = F_K(c^{i-1}(x) \oplus r_i)$$

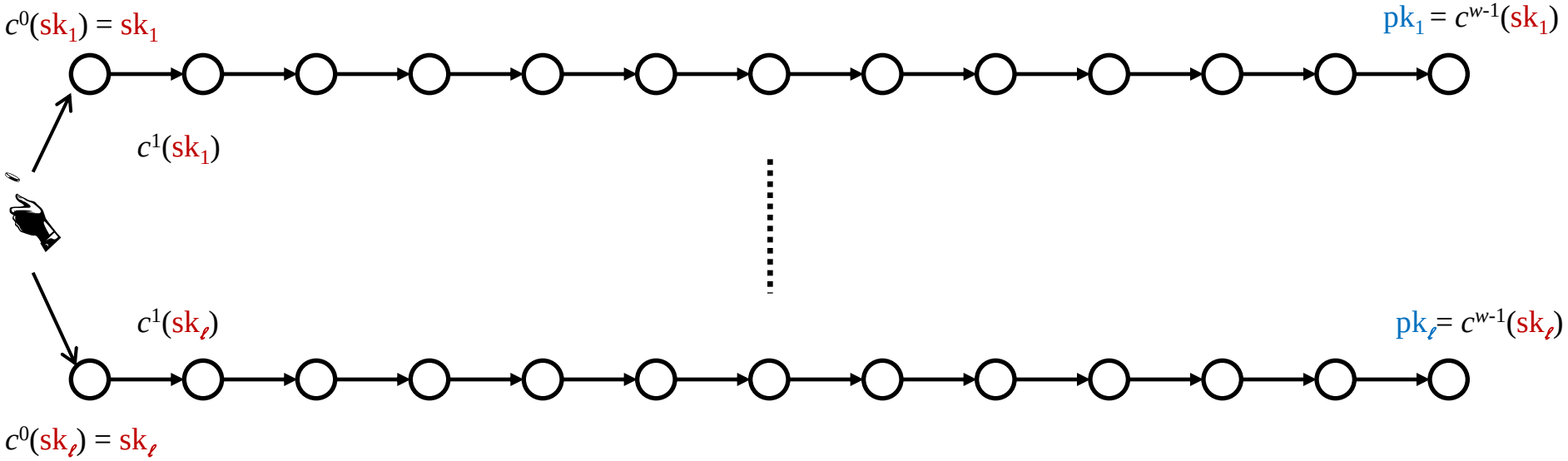


WOTS+
[Hül13]

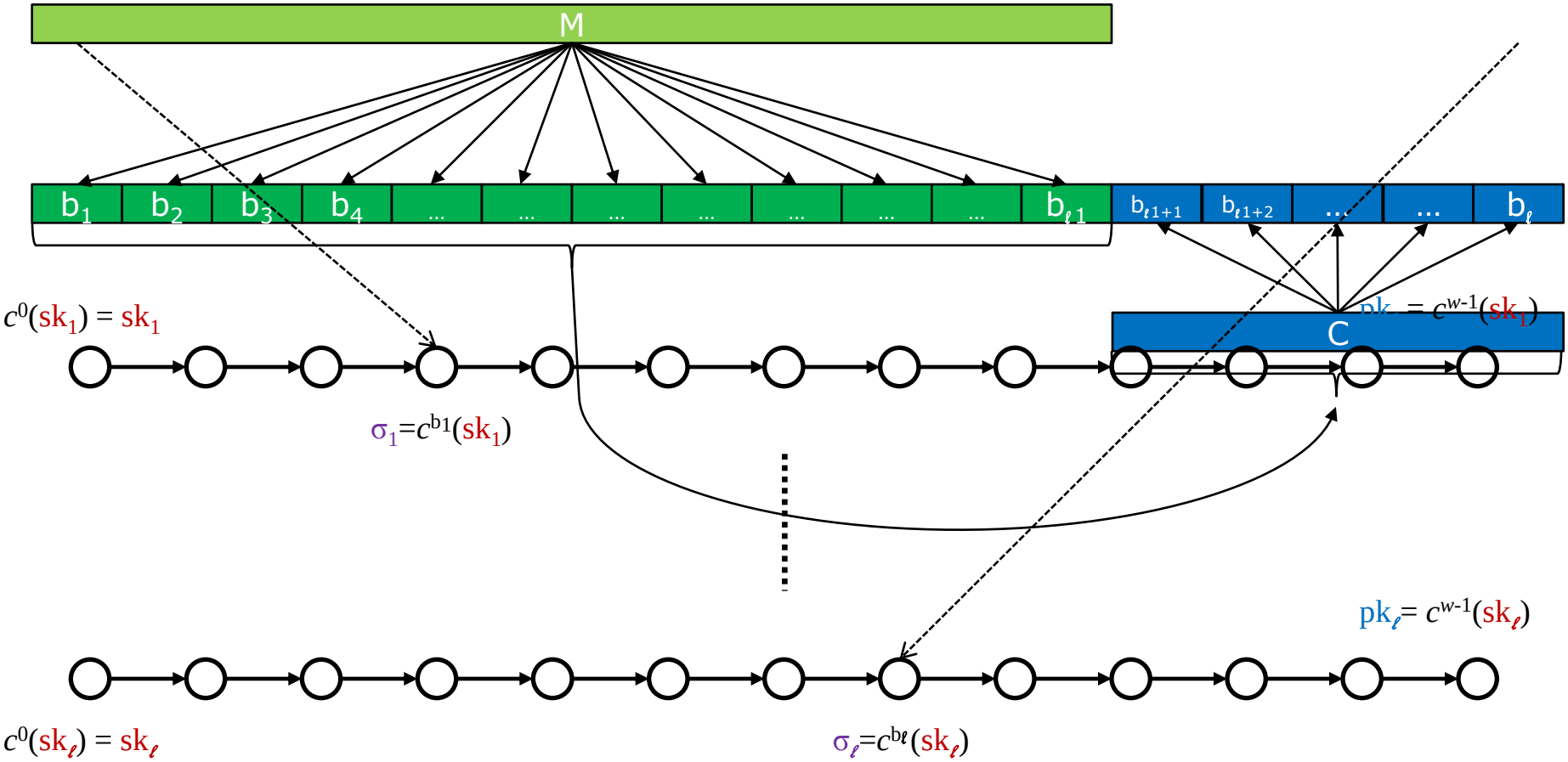
CASED

Winternitz parameter w, security parameter n, message length m, function family $\mathcal{F}_n = \{F_K : \{0,1\}^n \to \{0,1\}^n \mid K \in \{0,1\}^{n'}\}$

Key Generation: Compute ℓ, sample K, sample $\mathcal{R}$



WOTS+ Signature generation



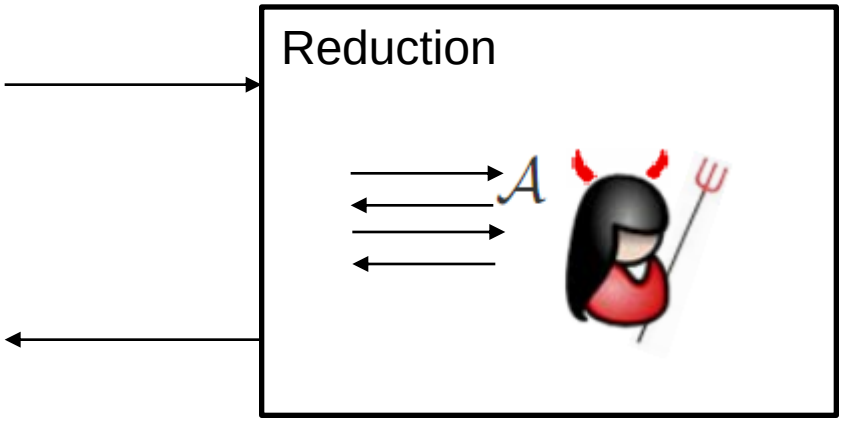
Main result

CASED

Theorem 3.9 (informally):

W-OTS$^+$ is strongly unforgeable under chosen message attacks if $\mathcal{F}$ is a 2nd-preimage resistant, undetectable one-way function family

Security Proof



Intuition

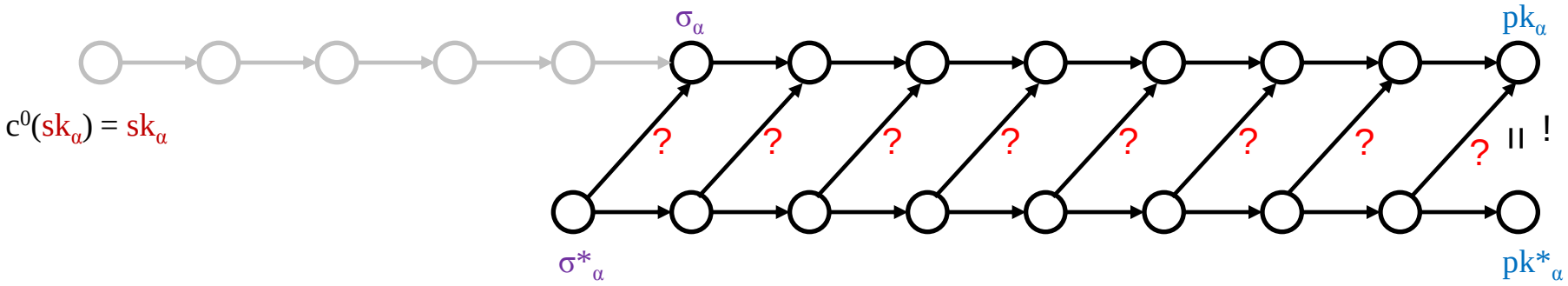
Oracle Response: $(\sigma, M);$ $M \rightarrow (b_1, \ldots, b_\ell)$

Forgery: $(\sigma^*, M^*);$ $M^* \rightarrow (b_1^*, \ldots, b_\ell^*)$

Observations:

1. Checksum: $\exists \alpha \in \{1,..,l\}\, s.th.\ b_\alpha^* < b_\alpha$
2. Verification $c^{w-1-b\alpha^*}(\sigma^*_\alpha) = pk_\alpha = c^{w-1-b\alpha}(\sigma_\alpha)$

➡ "quasi-inversion"



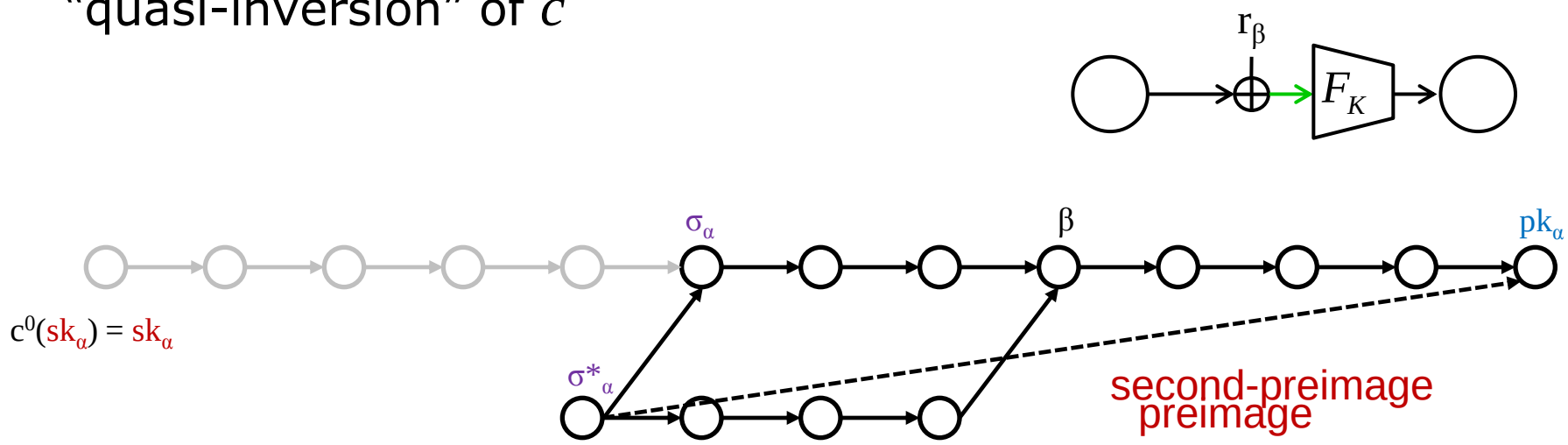
Intuition, cont'd

Oracle Response: (σ, M); $M \rightarrow (b_1, \ldots, b_\ell)$

Forgery: (σ^*, M^*); $M^* \rightarrow (b_1^*, \ldots, b_\ell^*)$

Given:

"quasi-inversion" of c



Result

CASED

Old [DSS05]	WOTS$ [BDEHR11]	WOTS+ [Hül13]
CR, UD, OW $\mathcal{F}_n$	PRF $\mathcal{F}_n$	SPR, UD, OW $\mathcal{F}_n$
Cryptomania	Minicrypt	Conj. Minicrypt
|Sig| = ℓ *2b	|Sig| = ℓ *(b+w)	|Sig| = ℓ *(b+log w)

CASED

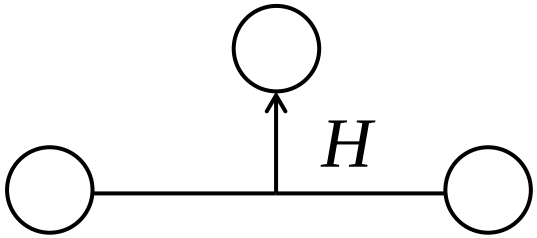
XMSS

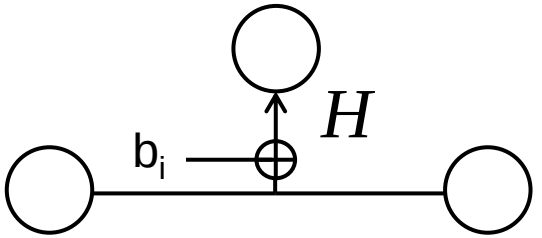
XMSS
[BDH11]

CASED

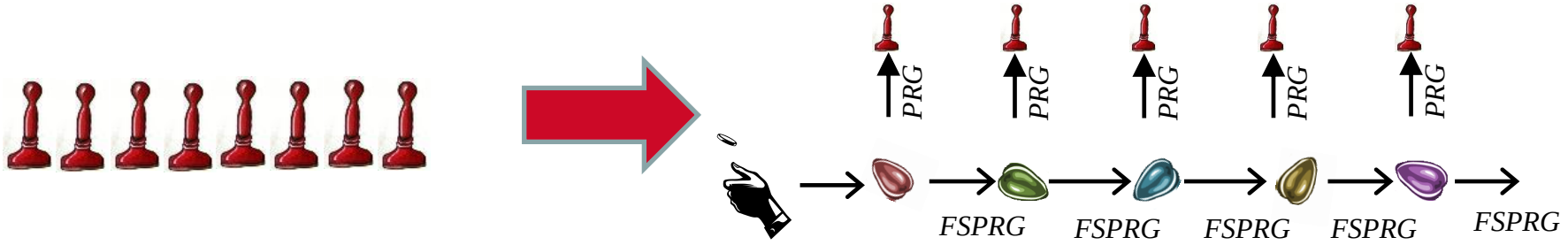
Lamport-Diffie / WOTS → WOTS$^+$ / WOTS$^{\$}$

Tree construction
[DOTV08]





Pseudorandom key generation



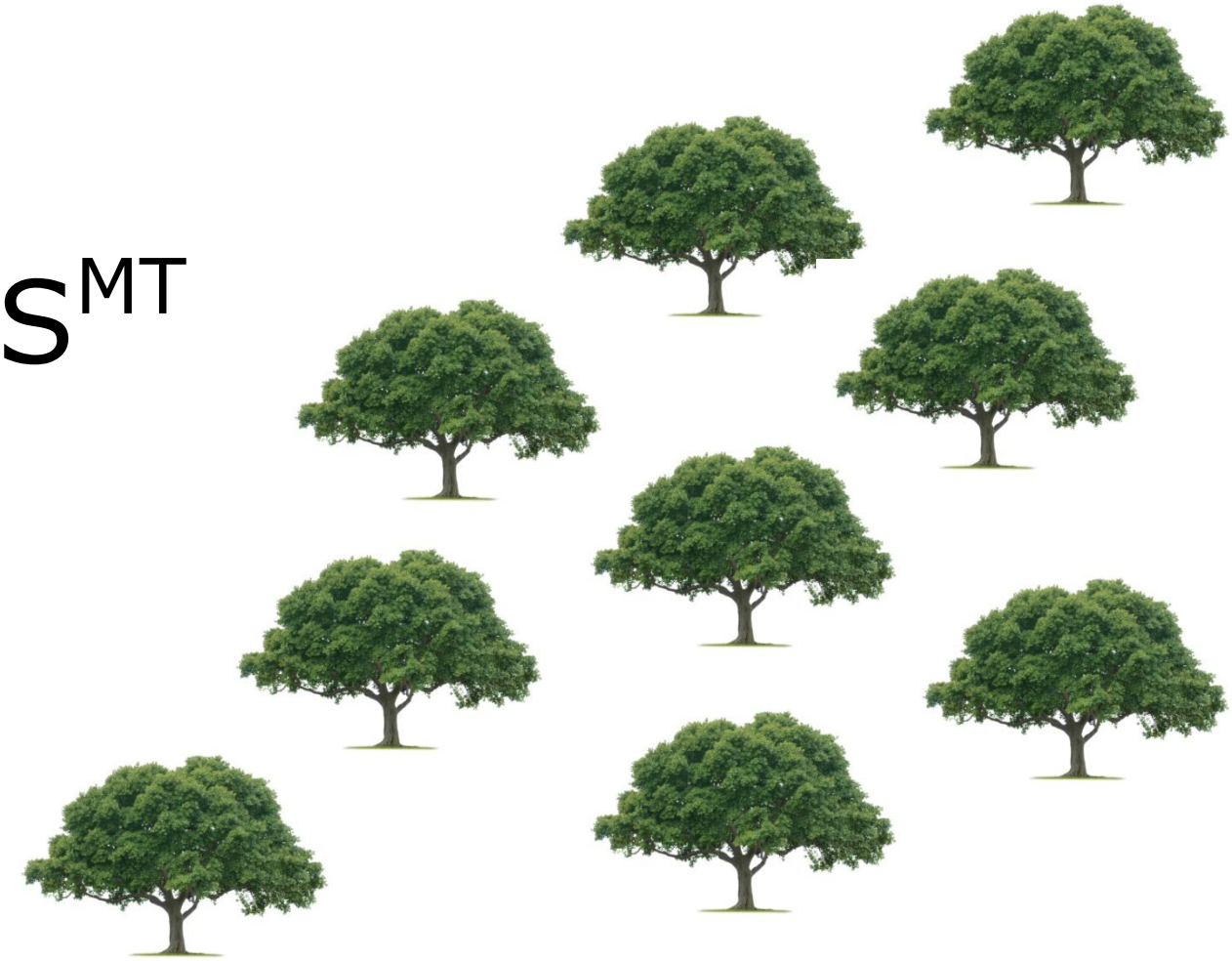
Result

CASED

SPR-MSS [DOTV08]	GMSS (Single Tree) [BDK+07]	XMSS [BDH11]
Minicrypt	Cryptomania	Minicrypt
FSS	Not FSS	FSS
$\|SK\| = 2^{h+1}bm + TTA$	$\|SK\| = b + TTA$	$\|SK\| = b + TTA$
$\|SIG\| \sim 2bm + hb$	$\|SIG\| \sim 2b(m/\log w) + h2b$	$\|SIG\| \sim b(m/\log w) + hb$

CASED

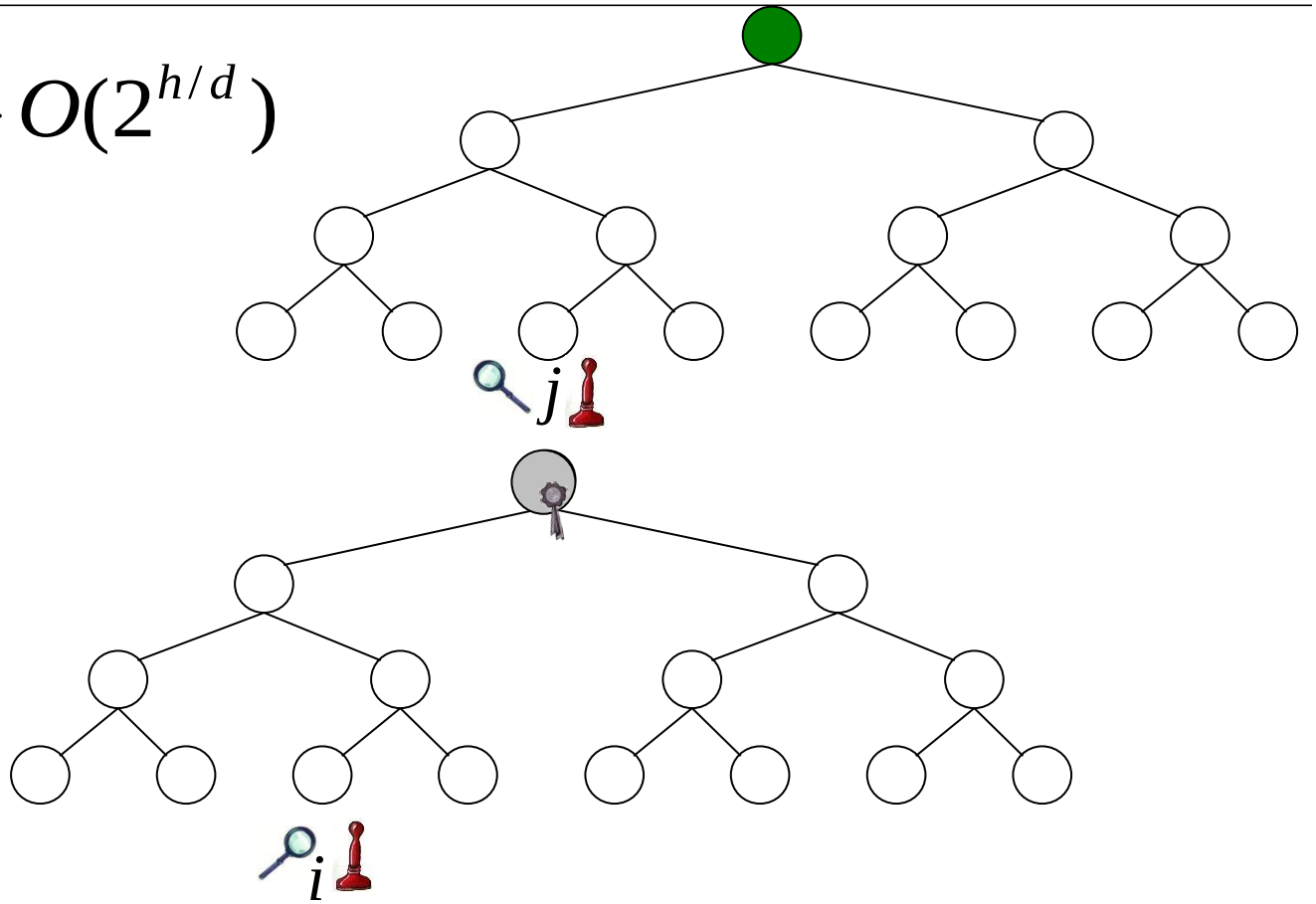
Chapter 5
XMSS^{MT}



Tree Chaining
[BGD+06,BDK+07]

CASED

$$t_{KG} : O(2^h) \rightarrow O(2^{h/d})$$



Improved distributed signature generation [HBB12,HRB13]

Result

CASED

GMSS
[BDK+07]

Cryptomania

Not FSS

$t_{SIG} = h/2 = \Sigma\, h_i/2$

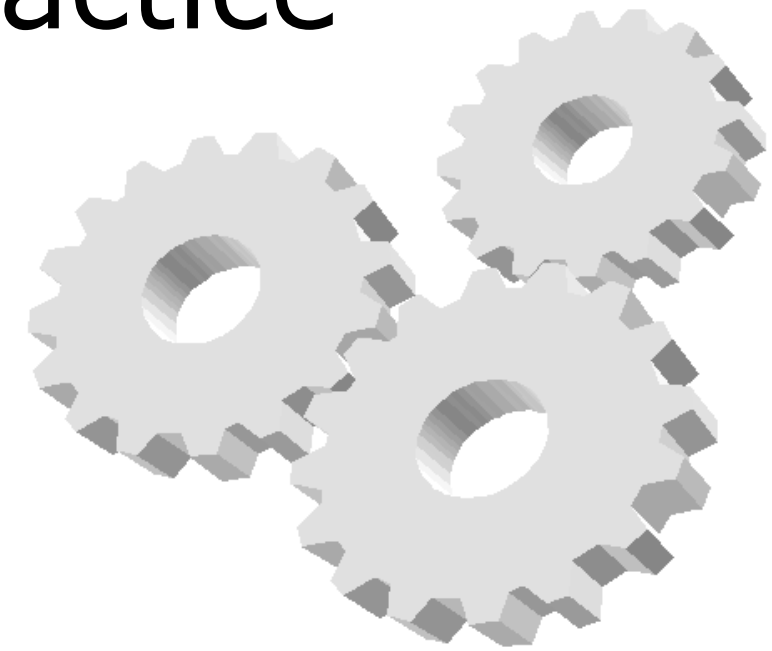
XMSS^{MT}
[HBB12,HRB13]

Minicrypt

FSS

$t_{SIG} = h_0/2$

XMSS* in Practice



XMSS Implementations
C Implementation

CASED

C Implementation, using OpenSSL [BDH2011]

	Sign (ms)	Verify (ms)	Signature (bit)	Public Key (bit)	Secret Key (byte)	Bit Security	Comment
XMSS-SHA-2	35.60	1.98	**16,672**	13,600	3,364	157	h = 20, w = 64,
XMSS-AES-NI	**0.52**	**0.07**	19,616	7,328	1,684	84	h = 20, w = 4
XMSS-AES	1.06	0.11	19,616	7,328	1,684	84	h = 20, w = 4
RSA 2048	**3.08**	**0.09**	≤ 2,048	≤ 4,096	≤ 512	87	

Intel(R) Core(TM) i5-2520M CPU @ 2.50GHz with Intel AES-NI

XMSS Implementations
Smartcard Implementation

CASED

	Sign (ms)	Verify (ms)	Keygen (ms)	Signature (byte)	Public Key (byte)	Secret Key (byte)	Bit Sec.	Comment
XMSS	**134**	23	**925,400**	2,388	800	2,448	92	H = 16, w = 4
XMSS$^+$	**106**	25	**5,600**	3,476	544	3,760	94	H = 16, w = 4
RSA 2048	190	7	11,000	≤ 256	≤ 512	≤ 512	87	

Infineon SLE78 16Bit-CPU@33MHz, 8KB RAM, TRNG, sym. & asym. co-processor

NVM: Card 16.5 million write cycles/ sector,
XMSS$^+$ < 5 million write cycles (h=20)

[HBB12]

Conclusion

Conclusion

CASED

- Efficient
- Minimal security assumptions
- „Small signatures"
- Forward secure
- Full smartcard implementation

CASED

Main Drawback: State

Easy Migration?

- Interfaces
- Key Management

CASED

Thank you!
Questions?