Introduction to Post-quantum cryptography

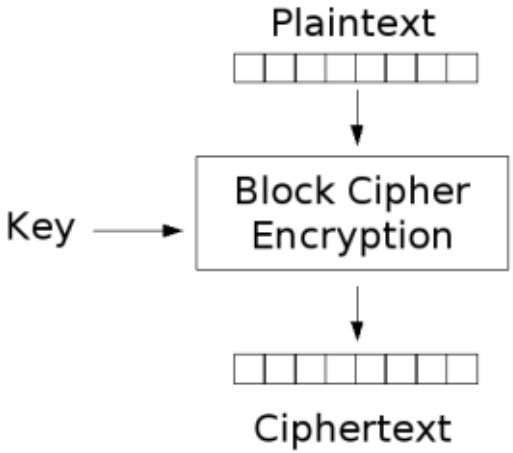
Andreas Hülsing

26.03.2019

Secret-key cryptography

Main (Secret-key) primitives

- Block- / Stream Cipher
  - Encryption of data
  - Provides Secrecy
- Massage authentication code
  - Authentication of data
  - Provides authenticity
- Hash function
  - Cryptographic checksum
  - Allows efficient comparison



Public-key cryptography

Main (public-key) primitives

- Digital signature
  - Proof of authorship
  - Provides:
    - Authentication
    - Non-repudiation

- Public-key encryption / key exchange
  - Establishment of commonly known secret key
  - Provides secrecy

Applications

- Code signing (Signatures)
  - Software updates
  - Software distribution
  - Mobile code

- Communication security (Signatures, PKE / KEX)
  - TLS, SSH, IPSec, ...
  - eCommerce, online banking, eGovernment, ...
  - Private online communication

Connection security (simplified)

Hi

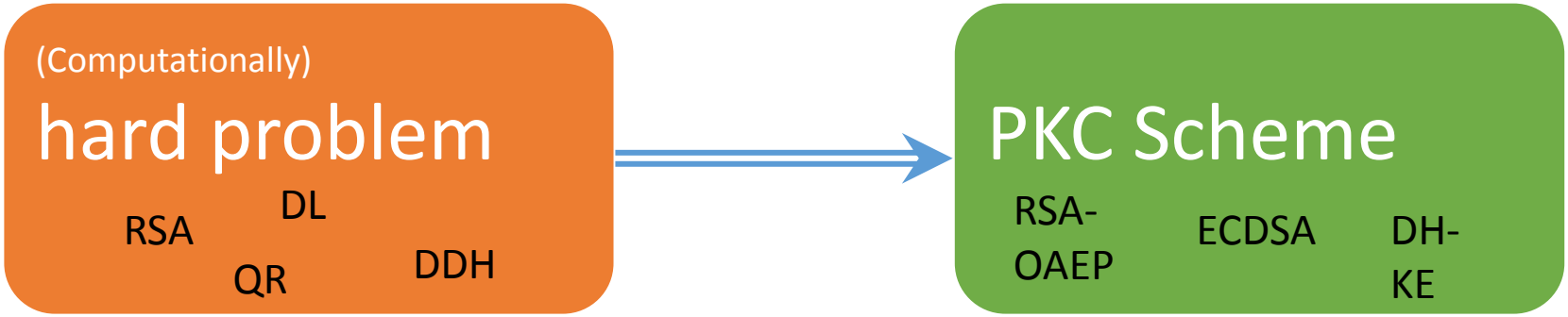
pk, Cert(pk belongs to shop)

PKC to establish shared secret sk

SKC secured communication using sk

The TEA SHOP
FINE TEAS & TISANES
PURE & BLENDED

How to build PKC



Today‘s Crypto-Eco-System

Public key cryptography:

- Deployed schemes are based on RSA- and discrete logarithm problem (incl. ECC, DH...).

Secret key / Symmetric cryptography:

- Wide range of different schemes.
- Not based on „hard problems“, rather on „design principles“

What happens if the TWO problems are solved?

- No (practical) secure communication
- No online payment
- No e-Commerce
- No Internet privacy
- No private online communication
  - with insurance company, public institutions, etc.
  - With private contacts (this includes Skype, whatsapp, etc. (although these are already questionable today..))
- Everyone in same WiFi network can listen to your connection

Quantum Computing

Quantum Computing

"Quantum computing studies theoretical computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data."

-- Wikipedia

The Quantum Threat

Quantum computing

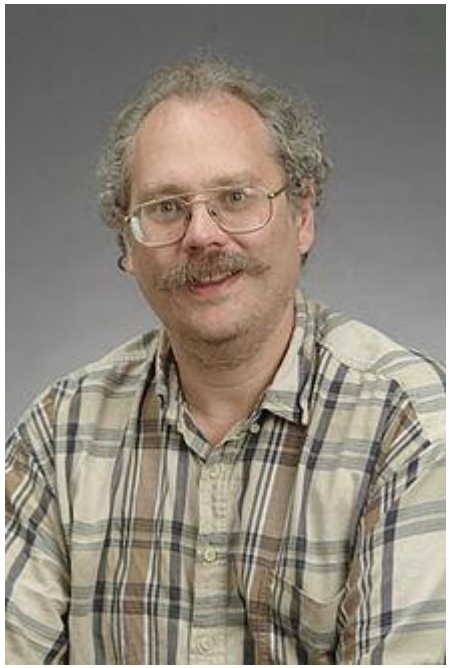
- Qubit state: $\alpha_0 |0\rangle + \alpha_1 |1\rangle$ with $\alpha_i \in \mathbb{C}$ such that $\alpha_0^2 + \alpha_1^2 = 1$
- Ket: $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
- Qubit can be in state $\frac{|0\rangle + |1\rangle}{\sqrt{2}}$
  -> like computing with 0 and 1 at the same time!
- Restriction: Only invertible computation.
- Restriction: Impossible to clone (copy) quantum state.

Quantum computing II

- Restriction: To learn outcome one has to measure.
  - Collapses qubit to basis state
  - 1 qubit leads 1 classical bit of information
  - Randomized process
- Goal: Amplify amplitude of solution vector.
  - See later in lecture.
- Many fancy things like quantum teleportation
  - Not important for us.

Shor‘s algorithm (1994)

- Quantum computers can do FFT very efficiently
- Can be used to find period of a function
- This can be exploited to factor in (quantum)-poly-time
- Shor also shows how to solve discrete log in (quantum)-poly-time



Shor's factoring algorithm – classical part - outline

1. Pick random $a < N$.
2. If $\gcd(a, N) \neq 1$ return $(\gcd(a, N), \frac{\mathrm{N}}{\gcd(a,N)})$.
3. Use quantum algorithm to find period π of
   $$f(x) = a^x \bmod N,$$
   i.e., smallest π with $f(x + \pi) = f(x)$.
4. Set $\mathrm{r} = \pi - 1$, i.e., r is the order of a $(a^r \equiv 1 \bmod N)$
5. If r is odd or $a^{r/2} \equiv -1 \bmod N$, restart.
6. Return $(\gcd(a^{r/2} + 1, N), \gcd(a^{r/2} - 1, N))$.

Shor's factoring algorithm – classical part - $a^{r/2}$ is non-trivial root of 1

- $a^r \equiv 1 \bmod N$
- If r is even and $b = a^{r/2} \not\equiv -1 \bmod N$, $a^{r/2}$ is non-trivial root of 1 ($a^{r/2} \equiv 1 \bmod N$ would imply $r/2$ is order of a, but we know r is order of a).
- Hence, $a^r - 1 = (a^{\frac{r}{2}} - 1)(a^{\frac{r}{2}} + 1) \equiv 0 \bmod N$ and $\gcd(a^{\frac{r}{2}} - 1, N)$, $\gcd(a^{\frac{r}{2}} + 1, N)$ are factors of N.

Shor's factoring algorithm – classical part - $a^{\frac{r}{2}} \pm 1$ are non-trivial factors of N

- $\gcd(a^{\frac{r}{2}} - 1, N) \neq N$, otherwise $N | a^{\frac{r}{2}} - 1 \Longrightarrow a^{\frac{r}{2}} - 1 \equiv 0 \bmod N \Longrightarrow a^{\frac{r}{2}} \equiv 1 \bmod N$ (contradicts r is order)
- $\gcd(a^{\frac{r}{2}} - 1, N) \neq 1$, otherwise $(\exists u, v)$:

$$u\left(a^{\frac{r}{2}} - 1\right) + Nv = 1 \qquad | * (a^{\frac{r}{2}} + 1)$$

$$u(a^r - 1) + N\left(a^{\frac{r}{2}} + 1\right)v = \left(a^{\frac{r}{2}} + 1\right)$$

as $N|(a^r - 1)$ this implies $\mathrm{N}|\left(a^{\frac{r}{2}} + 1\right)$ and so $a^{\frac{r}{2}} + 1 \equiv 0 \bmod N \Longrightarrow a^{\frac{r}{2}} \equiv -1 \bmod N$ (false by construction)

Example

- $N = 15, a = 7$
- $r = 4$ (check yourself)
- r is even and $a^{r/2} = 7^2 = 49 \equiv 4 \bmod 15$
- $\gcd\left(a^{\frac{r}{2}} \pm 1, N\right) = \gcd(49 \pm 1, 15)$

  $\gcd(48,15) = 3$

  $\gcd(50,15) = 5$

Grover‘s algorithm

- Finds „marked item“ in database with 2^n elements using $\Omega(2^{\frac{n}{2}})$ queries
- Can be adopted to find (second-)preimages using $\Omega(2^{\frac{n}{2}})$ and collisions using $\Omega(2^{\frac{n}{3}})$ queries for n bit (hash) function
- Nice: Grover is provably optimal! (For random function)
- So far: This is the best known attack against symmetric crypto
- Double security parameter and we are fine.

Why care today?

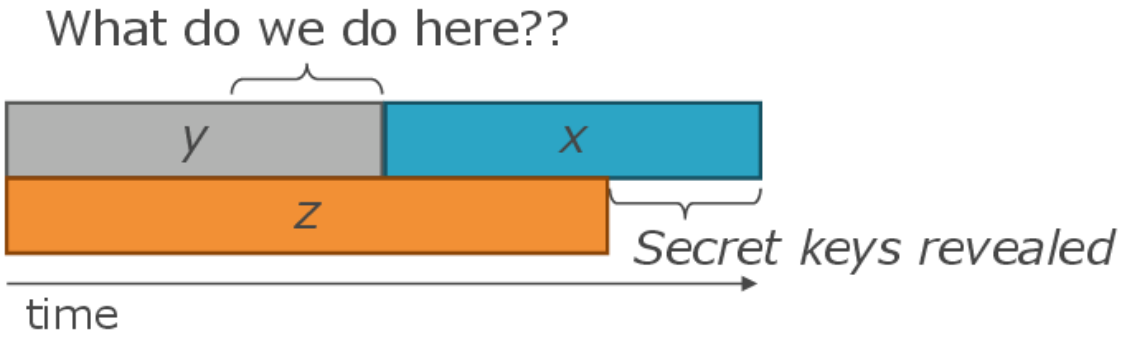
It‘s a question of risk assessment

How soon do we need to worry?

Depends on:

- How long do you need your keys to be secure? (*x* years)
- How much time will it take to re-tool the existing infrastructure with large-scale quantum-safe solution? (*y* years)
- How long will it take for a large-scale quantum computer to be built (or for any other relevant advance? (*z* years)

Theorem 1: If $x + y > z$, then worry.



Quantum Computing

"Quantum computing studies theoretical computation systems (quantum computers) that make direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data."

-- Wikipedia

Bad news

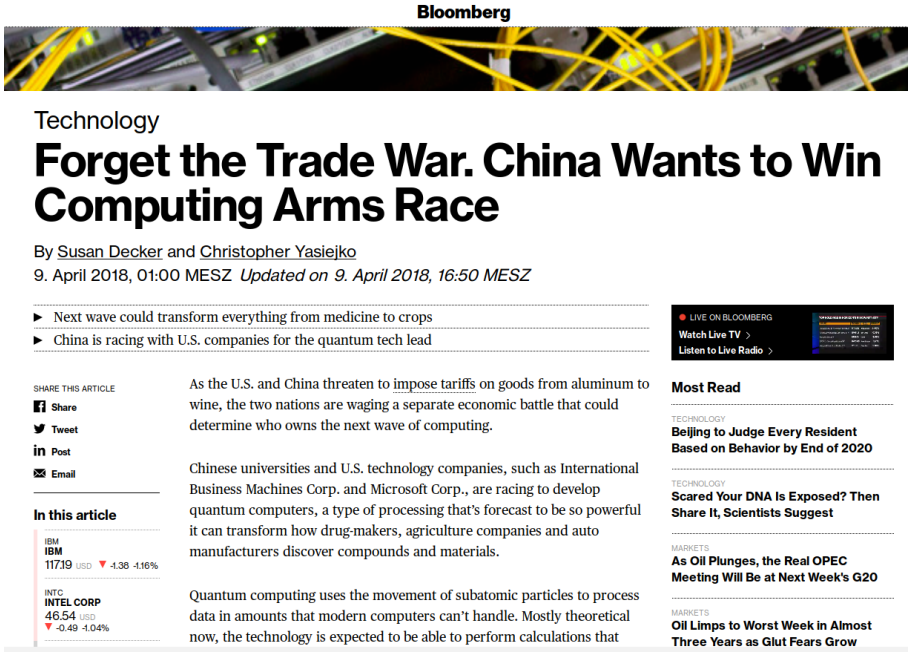
I will not tell you when a quantum computer will be built!

Quantum computers

- 1980 Theoretical concept
- 1994 Shor‘s algorithm
- 1995 First quantum gate experimentally realized
- 1996 Grover‘s algorithm
- 2014 Largest number factored: 56153

Why care today

- **EU** launched a one billion Euro project on quantum technologies
- Similar range is spent in **China**
- **US** administration passed a bill on spending $1.275 billion US dollar on quantum computing research
- **Google**, **IBM**, **Microsoft**, **Alibaba**, and others run their own research programs.



Bloomberg

Technology

Forget the Trade War. China Wants to Win Computing Arms Race

By Susan Decker and Christopher Yasiejko

9. April 2018, 01:00 MESZ *Updated on 9. April 2018, 16:50 MESZ*

- Next wave could transform everything from medicine to crops
- China is racing with U.S. companies for the quantum tech lead

SHARE THIS ARTICLE

Share
Tweet
Post
Email

In this article

IBM
IBM
117.19 USD ▼ -1.36 -1.16%

INTC
INTEL CORP
46.54 USD
▼ -0.49 -1.04%

As the U.S. and China threaten to impose tariffs on goods from aluminum to wine, the two nations are waging a separate economic battle that could determine who owns the next wave of computing.

Chinese universities and U.S. technology companies, such as International Business Machines Corp. and Microsoft Corp., are racing to develop quantum computers, a type of processing that's forecast to be so powerful it can transform how drug-makers, agriculture companies and auto manufacturers discover compounds and materials.

Quantum computing uses the movement of subatomic particles to process data in amounts that modern computers can't handle. Mostly theoretical now, the technology is expected to be able to perform calculations that

LIVE ON BLOOMBERG
Watch Live TV
Listen to Live Radio

Most Read

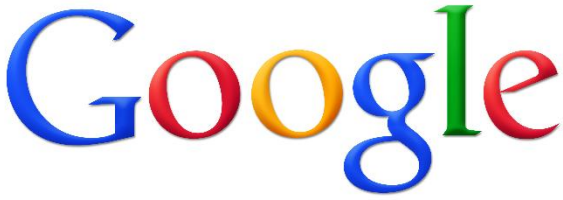
TECHNOLOGY
Beijing to Judge Every Resident Based on Behavior by End of 2020

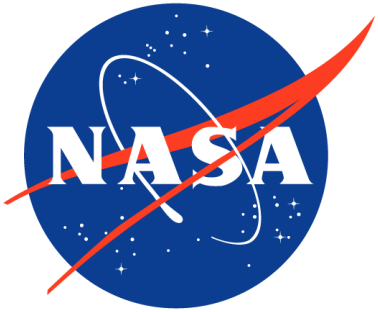
TECHNOLOGY
Scared Your DNA Is Exposed? Then Share It, Scientists Suggest

MARKETS
As Oil Plunges, the Real OPEC Meeting Will Be at Next Week's G20

MARKETS
Oil Limps to Worst Week in Almost Three Years as Glut Fears Grow

(Big) Players in quantum game

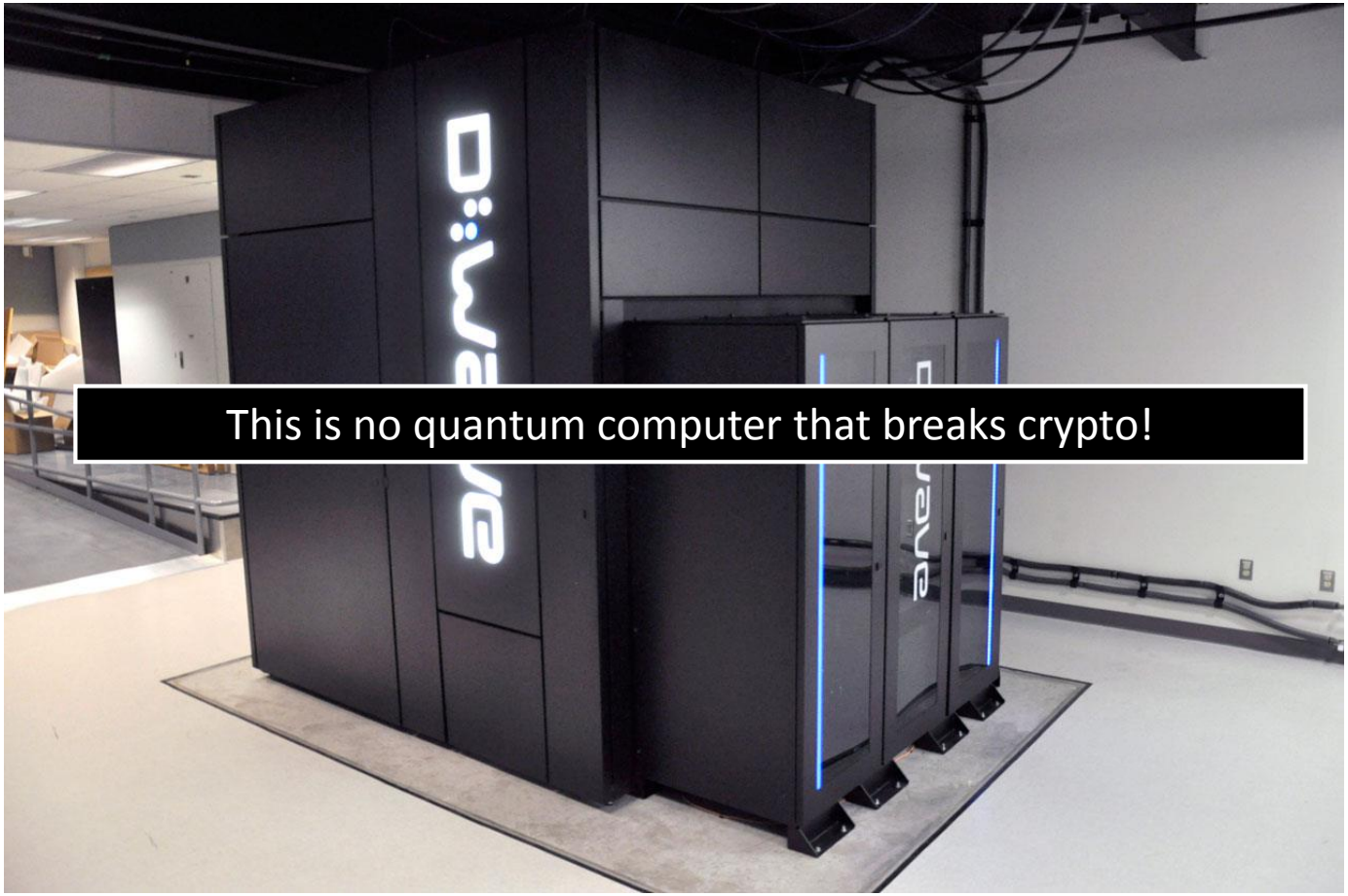




Universiteit Leiden

UNIVERSITY OF WATERLOO | IQC Institute for Quantum Computing

A comment on D-Waves quantum annealing computer



Interim conclusion

- Quantum computers are powerful but not almighty
- Can be used to break some crypto but not all crypto: Deployed asymmetric falls, symmetric survives
- Unclear when large scale QC‘s ready
- If we want to preserve privacy for more than a short time: We have to react now!

Quantum Cryptography

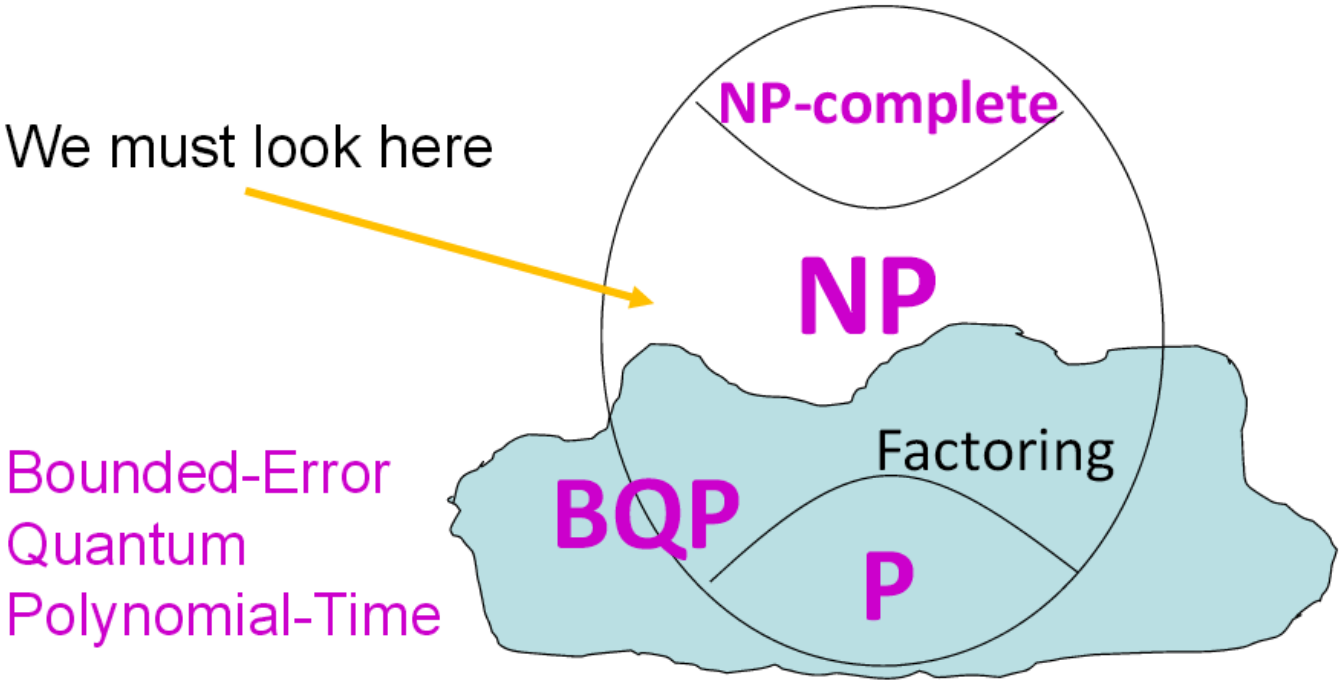
Why not beat ‘em with their own weapons?

- QKD: Quantum Key distribution.
  - Based on some nice quantum properties: entanglement & collapsing measurments
  - Information theoretic security -> Great!
  - For sale today!
- So why don‘t we use this?
- Only short distance, point-to-point connections!
  - Internet? No way!
- Longer distances require „trusted-repeaters“ ☺
  - We all know where this leads...
- More issues with actual implementations...

Post-Quantum Cryptography

Quantum-secure problems

No provably quantum resistant problems



Credits: Buchmann, Bindel 2015

Conjectured quantum-secure problems

- Solving multivariate quadratic equations (MQ-problem)
  -> Multivariate Crypto
- Bounded-distance decoding (BDD)
  -> Code-based crypto
- Short(est) and close(st) vector problem (SVP, CVP)
  -> Lattice-based crypto
- Breaking security of symmetric primitives (SHA-x-, AES-, Keccak-,... problem)
  -> Hash-based signatures / symmetric crypto

Multivariate Crypto

$$4x + x^2 + y^2z \equiv 1 \mod 13$$

$$7y^2 + 2xz^2 \equiv 12 \mod 13$$

$$x + y^2 + 12xz^2 \equiv 4 \mod 13$$

Solution: $x = 15,\ y = 29,\ z = 45$

Credits: Buchmann, Bindel 2015

MQ-Problem

Let $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{F}_q^{\,n}$ and $\mathbf{MQ}(n, m, \mathbb{F}_q)$ denote the family of vectorial functions $\boldsymbol{F}: \mathbb{F}_q^{\,n} \longrightarrow \mathbb{F}_q^{\,m}$ of degree 2 over $\mathbb{F}_q$:

$$\mathbf{MQ}\left(n, m, \mathbb{F}_q\right)$$

$$= \left\{ \boldsymbol{F}(\boldsymbol{x}) = (f_1(\boldsymbol{x}), \dots, f_m(\boldsymbol{x}) | f_s(\boldsymbol{x}) = \sum_{i,j} a_{i,j} x_i x_j + \sum_i b_i x_i\,, \qquad s \in [1, m] \right\}$$

The $\mathbf{MQ}$ Problem $\mathbf{MQ}(\boldsymbol{F}, \boldsymbol{v})$ is defined as given $\boldsymbol{v} \in \mathbb{F}_q^{\,m}$ find, if any, $\boldsymbol{s} \in \mathbb{F}_q^{\,n}$ such that $\boldsymbol{F}(\boldsymbol{s}) = \boldsymbol{v}$.

Decisional version is NP-complete [Garey, Johnson´79]

Multivariate Signatures (the traditional way)

$P: F^n \to F^m$, easily invertible non-linear

$S: F^n \to F^n$, $T: F^m \to F^m$, affine linear

Public key: $G = S \circ P \circ T$, hard to invert

Secret Key: S, P, T allows to find G^{-1}

$$G^{-1} = T^{-1} \circ P^{-1} \circ S^{-1}$$

Signing: $s = T^{-1} \circ P^{-1} \circ S^{-1}(m)$

Verifying: $G(s) =^? m$

Forging signature: Solve $G(s) - m = 0$

Fast

Large keys:
100 kBit for 100 bit security
Compared to
1776 bit
RSA modulus

- UOV , Goubin et al., 1999
- Rainbow, Ding, et al. 2005
- pFlash, Cheng, 2007
- Gui, Ding, Petzoldt, 2015

Credits: Buchmann, Bindel 2015

Multivariate Cryptography

- Breaking scheme ⇔ Solving random MQ-instance
  -> NP-complete is a worst-case notion
  (there might be – and there are for MQ -- easy instances)
  -> Not a random instance
  Many broken proposals
  -> Oil-and-Vinegar, SFLASH, MQQ-Sig, (Enhanced) TTS, Enhanced STS.
  -> Security somewhat unclear
- Only signatures
  -> (new proposal for encryption exists but too recent)
- Really large keys
- **New proposal with security reduction, small keys, but large signatures.**

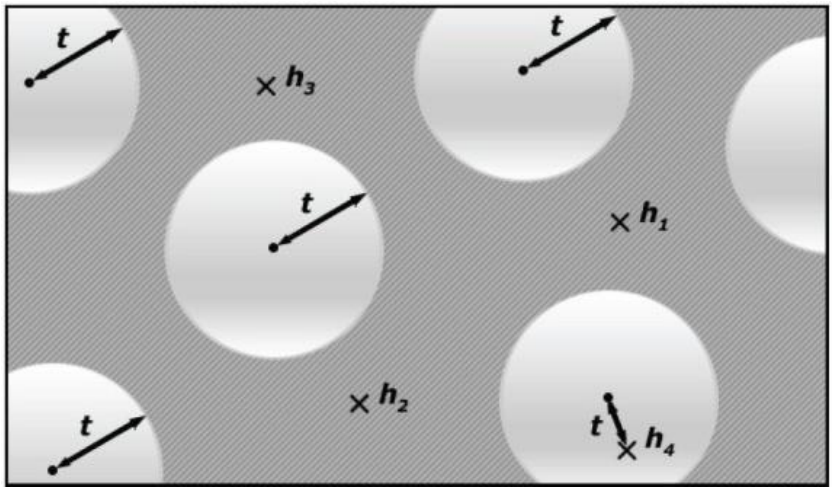
MQ-DSS / SOFIA

- New (2016 / 2018) proposal for MQ-based signatures
- Security reduction from MQ-problem in (Q)ROM
- Small keys, fast, large signatures (41 kB)

Coding-based cryptography - BDD

Given:
- Linear code $C \subseteq F_2^n$
- $y \in F_2^n$
- $t \in \mathbb{N}$

Find:
- $x \in C$: $dist(x, y) \leq t$



BDD is NP-complete (Berlekamp et al. 1978) (Decisional version)

Credits: Buchmann, Bindel 2015

McEliece PKE (1978)

S, G, P matrices over F

G generator matrix for Goppa code

Allows to solve BDD

Public key: $G' = S \circ G \circ P,\ t$

Secret Key: P, S, G

Encryption: $c = mG' + z \in F^n$

Decryption: $x = cP^{-1} = mSG + zP^{-1}$

solve BDD to get $y = mSG$

decode to obtain m

Fast

Large public keys!
500 kBits for 100 bit security
Compared to 1776 bit RSA modulus

IND-CPA secure version

Credits: Buchmann, Bindel 2015

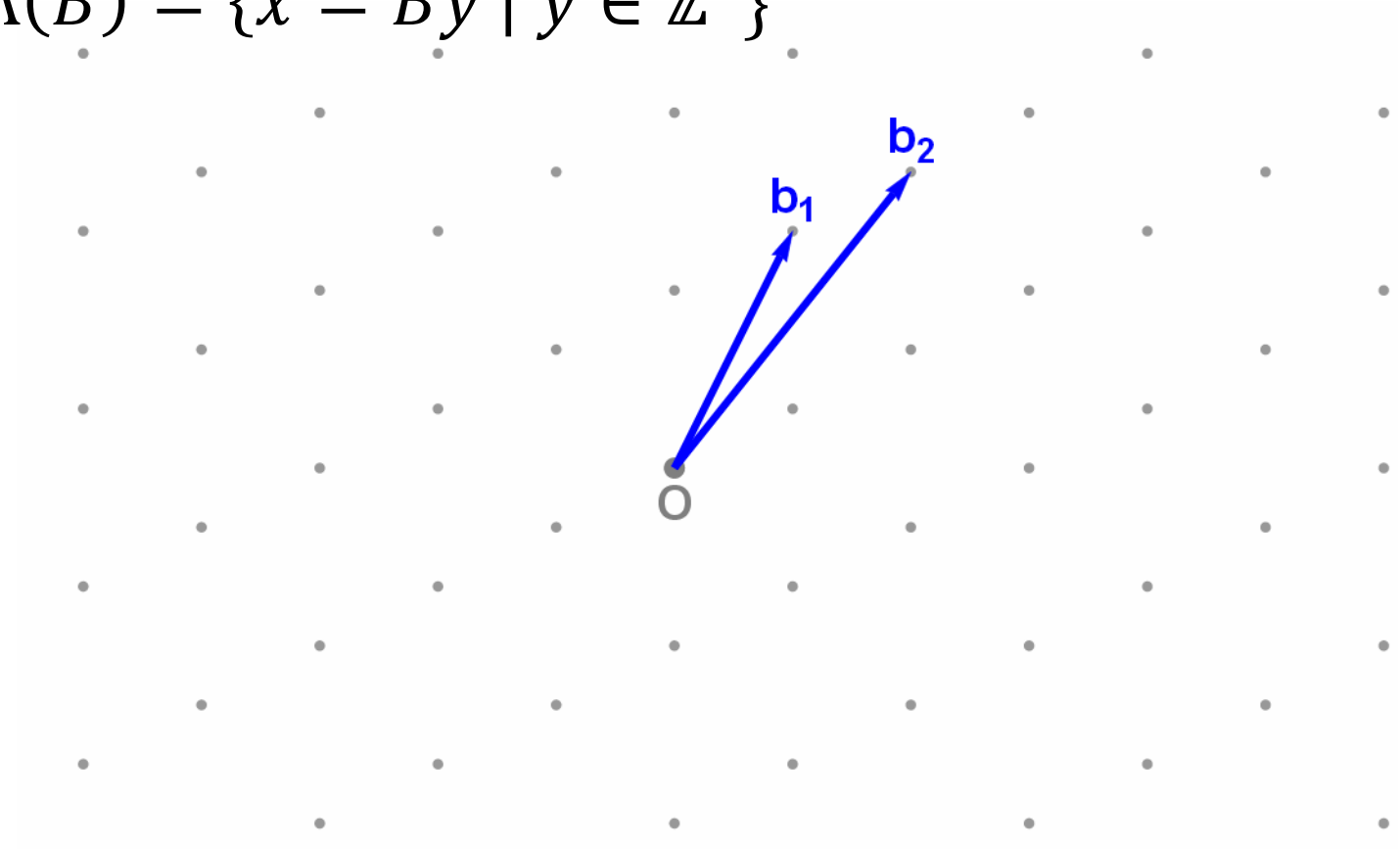
Code-based cryptography

- Breaking scheme $\not\Leftrightarrow$ Solving BDD
  -> NP-complete is a worst-case notion
  (there might be – and there are for BDD -- easy instances)
  -> Not a random instance
  However, McEliece with binary Goppa codes survived for almost 40 years (similar situation as for e.g. AES)
- Using more compact codes often leads to break
- So far, no practical signature scheme
- Really large public keys

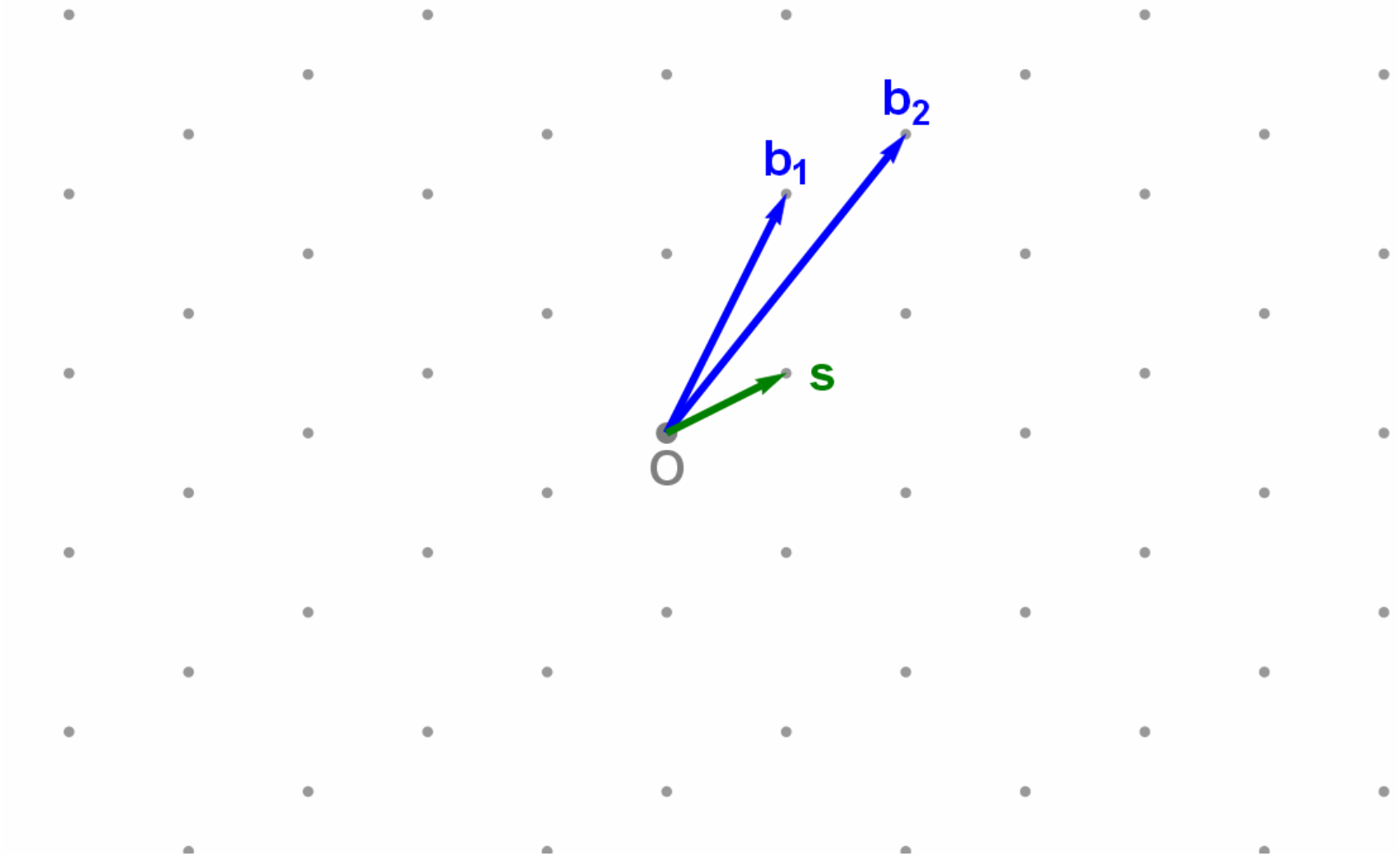
Lattice-based cryptography

Basis: $B = (b_1, b_2) \in \mathbb{Z}^{2\times 2}; b_1, b_2 \in \mathbb{Z}^2$

Lattice: $\Lambda(B) = \{x = By \mid y \in \mathbb{Z}^2\}$



Shortest vector problem (SVP)



(Worst-case) Lattice Problems

- **SVP:** Find shortest vector in lattice, given random basis. NP-hard (Ajtai'96)
- **Approximate SVP (αSVP):** Find short vector (norm < α times norm of shortest vector). Hardness depends on α (for α used in crypto not NP-hard).
- **CVP:** Given random point in underlying Vectorspace (e.g. $\mathbb{Z}^n$), find the closest lattice point. (Generalization of SVP, reduction from SVP)
- **Approximate SVP (αCVP):** Find a „close" lattice point. (Generalization of αSVP)

(Average-case) Lattice Problems Short Integer Solution (SIS)

$\mathbb{Z}_p^n =$ n-dim. vectors with entries mod p $(\approx n^3)$

Goal:
Given $\boldsymbol{A} = (\boldsymbol{a_1}, \boldsymbol{a_2}, \ldots, \boldsymbol{a_m}) \in \mathbb{Z}_p^{n \times m}$
Find „small“ $\boldsymbol{s} = (s_1, \ldots, s_m) \in \mathbb{Z}^m$ such that

$$\boldsymbol{As} = \boldsymbol{0} \bmod p$$

Reduction from worst-case αSVP.

Hash function

Set $m > n \log p$ and define $f_{\boldsymbol{A}}: \{0,1\}^m \to \mathbb{Z}_p^n$ as

$$f_A(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} \bmod p$$

Collision-resistance: Given short $\boldsymbol{x_1}, \boldsymbol{x_2}$ with $\boldsymbol{A}\boldsymbol{x_1} = \boldsymbol{A}\boldsymbol{x_2}$ we can find a short solution as

$$\boldsymbol{A}\boldsymbol{x_1} = \boldsymbol{A}\boldsymbol{x_2} \Rightarrow \boldsymbol{A}\boldsymbol{x_1} - \boldsymbol{A}\boldsymbol{x_2} = \boldsymbol{0}$$
$$\boldsymbol{A}(\boldsymbol{x_1} - \boldsymbol{x_2}) = \boldsymbol{0}$$

So, $\boldsymbol{z} = \boldsymbol{x_1} - \boldsymbol{x_2}$ is a solution and it is short as $\boldsymbol{x_1}, \boldsymbol{x_2}$ are short.

Lattice-based crypto

- SIS: Allows to construct signature schemes, hash functions, ... , basically minicrypt.
- For more advanced applications: Learning with errors (LWE)
  - Allows to build PKE, IBE, FHE,...
- Performance: Sizes can almost reach those of RSA (just small const. factor), really fast.
- BUT: Exact security not well accessed, yet. Especially, no good estimate for quantum computer aided attacks.

Hash-based Signature Schemes

[Mer89]

- Post quantum
- Only secure hash function
- Security well understood
- Fast

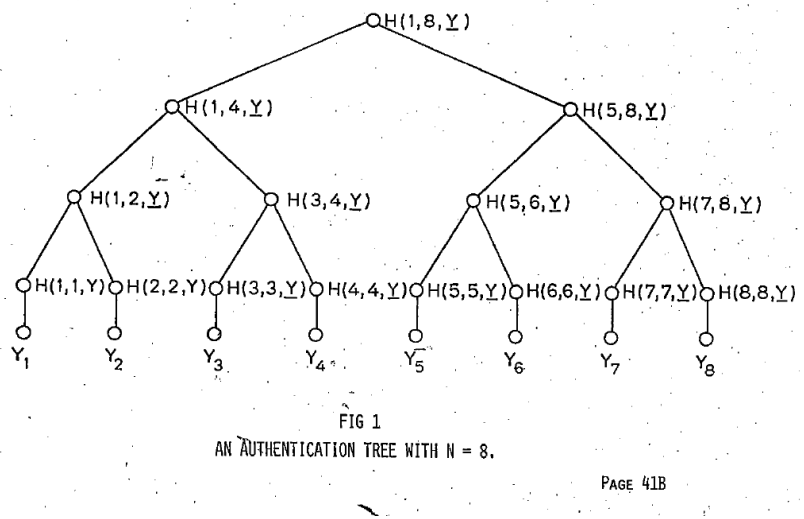
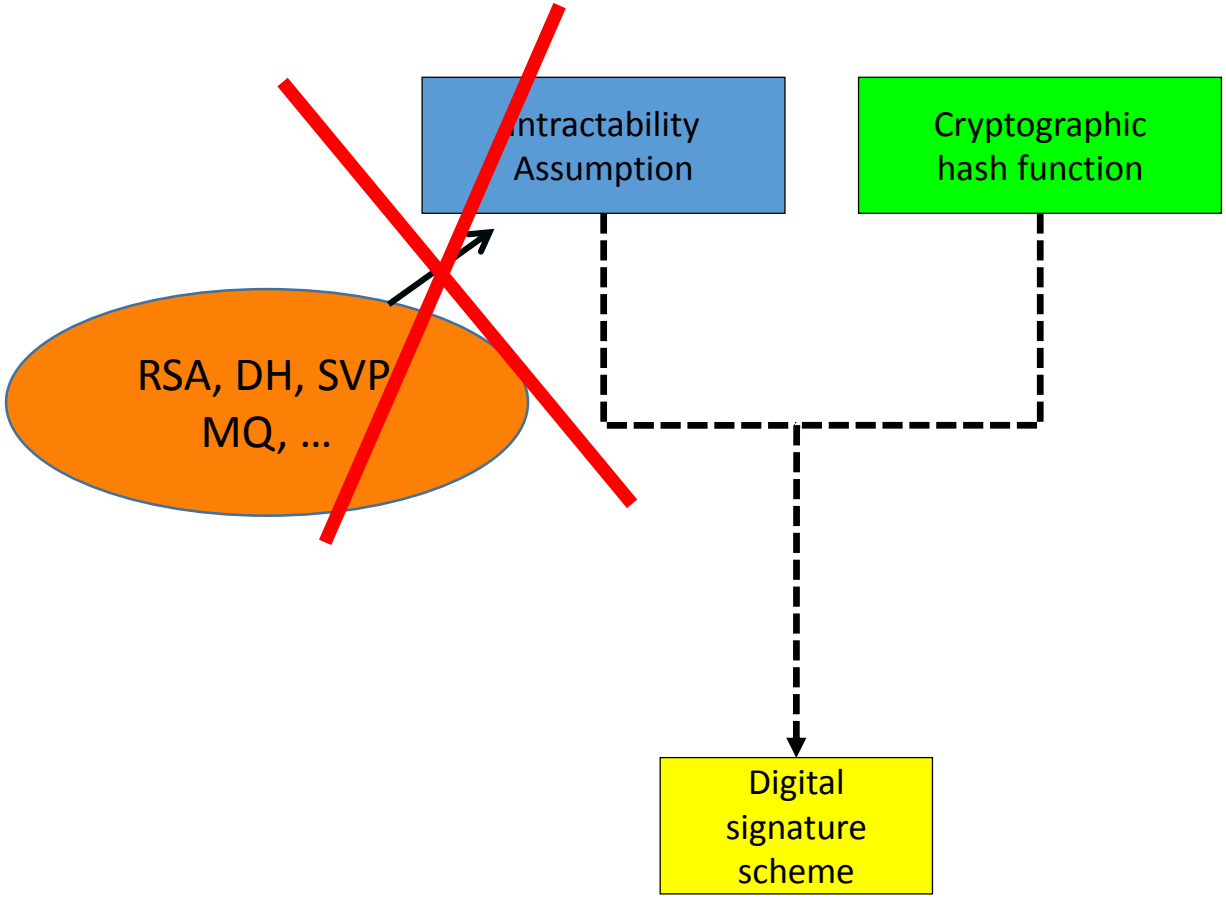


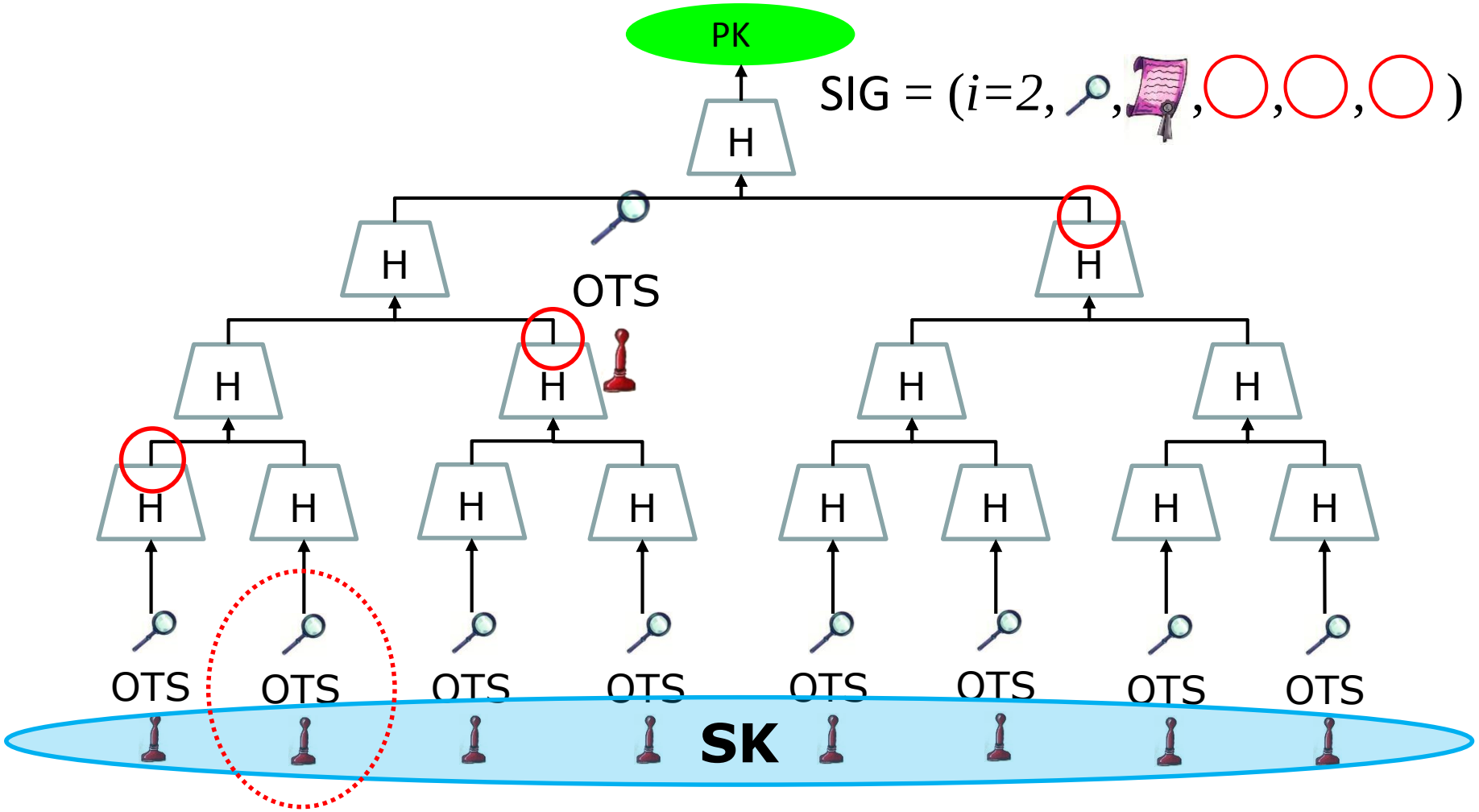
FIG 1
AN AUTHENTICATION TREE WITH N = 8.

PAGE 41B

RSA – DSA – EC-DSA…



Merkle's Hash-based Signatures



Hash-based signatures

- Only signatures
- Minimal security assumptions
- Well understood
- Fast & compact (2kB, few ms), but stateful, or
- Stateless, bigger and slower (41kB, several ms).

- Two Internet drafts (drafts for RFCs), one in „RFC Editor queue“

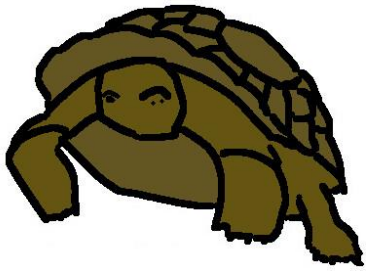
NIST Competition

Resources

- PQ Summer School: https://2017.pqcrypto.org/school/index.html
- NIST PQC Standardization Project: https://csrc.nist.gov/Projects/Post-Quantum-Cryptography
- Master Math (Selected Areas in Cryptology): https://elo.mastermath.nl/

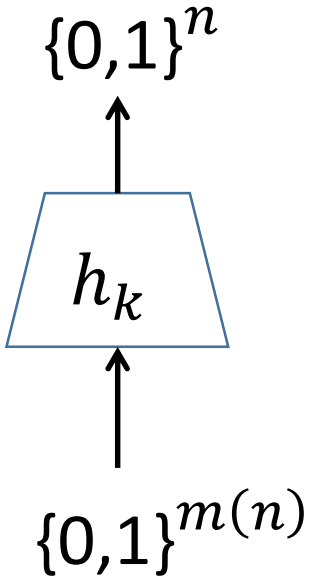
PQCrypto

PQCRYPTO
ICT-645622



(Hash) function families

- $H_n := \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$
- $m(n) \geq n$
- „efficient“



One-wayness

$H_n := \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

$x \xleftarrow{\$} \{0,1\}^{m(n)}$

$y_c \leftarrow h_k(x)$

Success if $h_k(x^*) = y_c$

Collision resistance

$H_n := \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

Success if
$h_k(x_1^*) = h_k(x_2^*)$

Second-preimage resistance

$H_n := \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

$x_c \xleftarrow{\$} \{0,1\}^{m(n)}$

Success if
$h_k(x_c) = h_k(x^*)$

Undetectability

$H_n := \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

$b \xleftarrow{\$} \{0,1\}$

If $b = 1$

$x \xleftarrow{\$} \{0,1\}^{m(n)}$

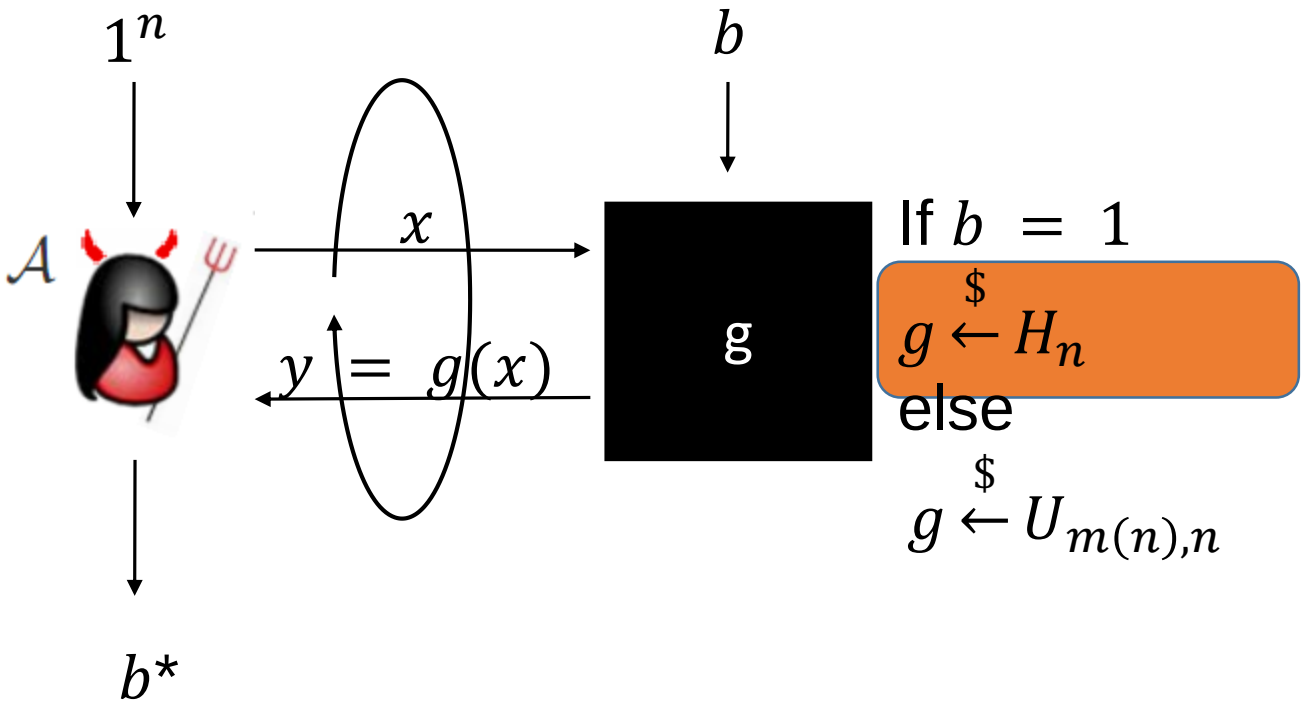
$y_c \leftarrow h_k(x)$

else

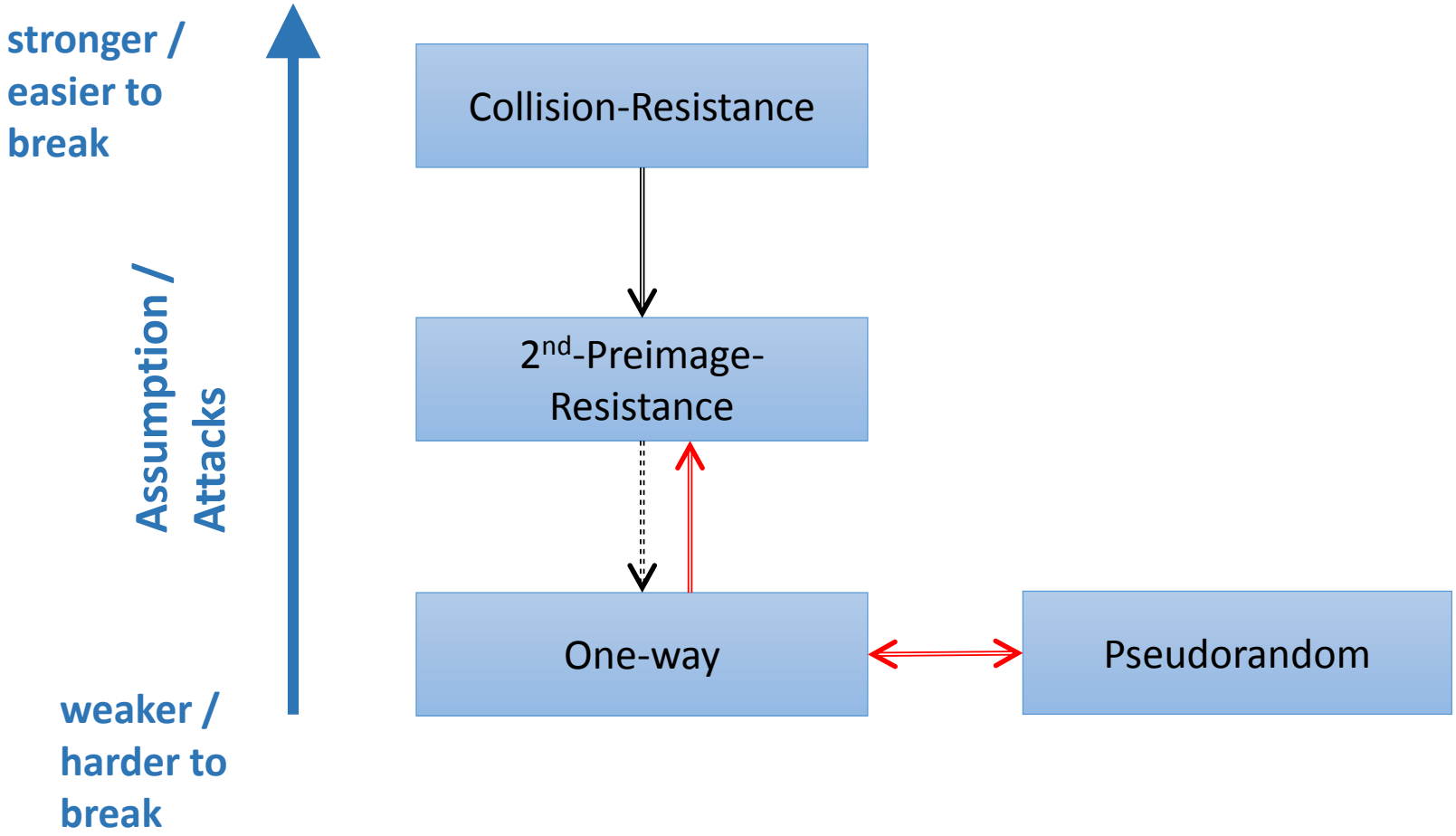
$y_c \xleftarrow{\$} \{0,1\}^n$

Pseudorandomness

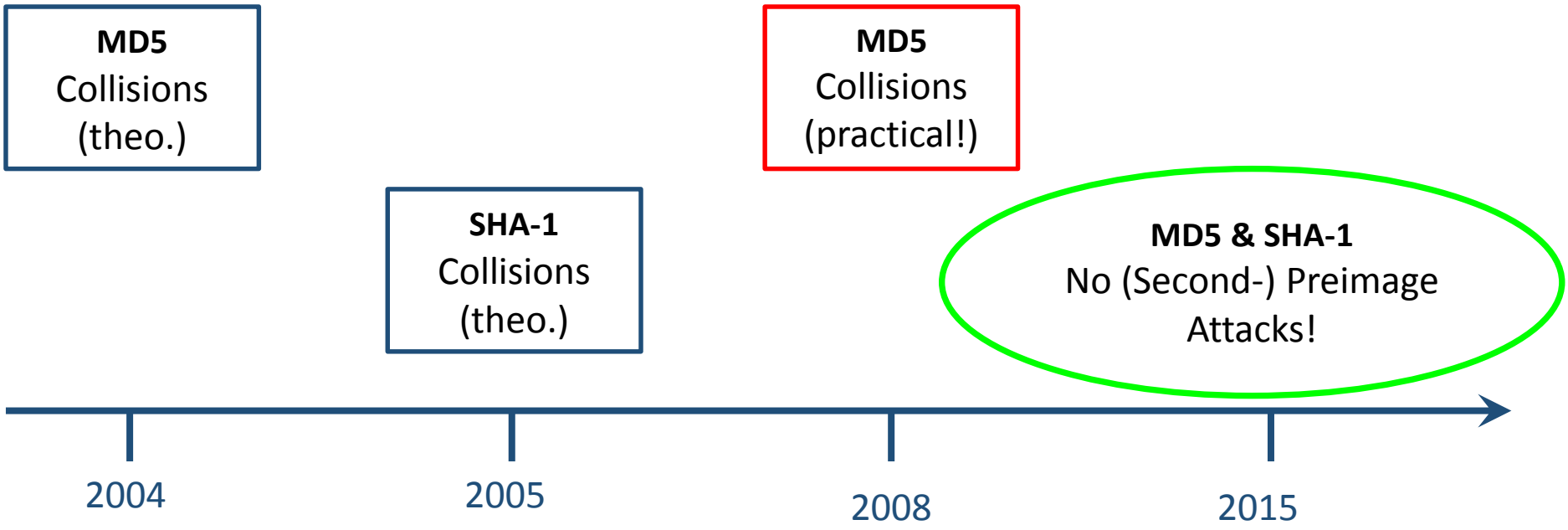
$H_n := \{h_k : \{0,1\}^{m(n)} \to \{0,1\}^n\}$



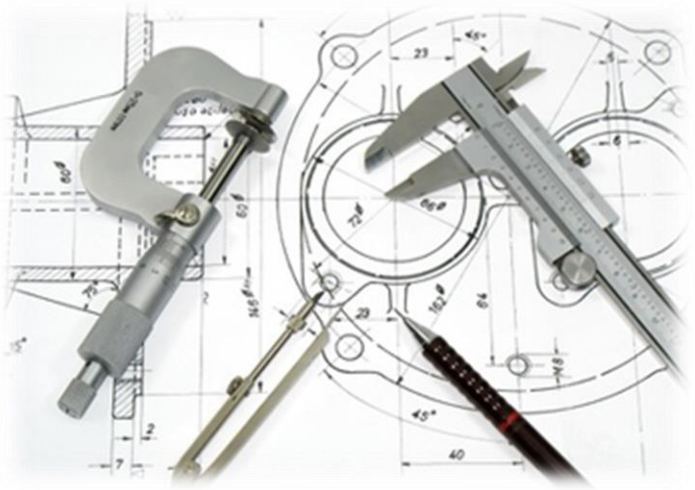
Hash-function properties



Attacks on Hash Functions

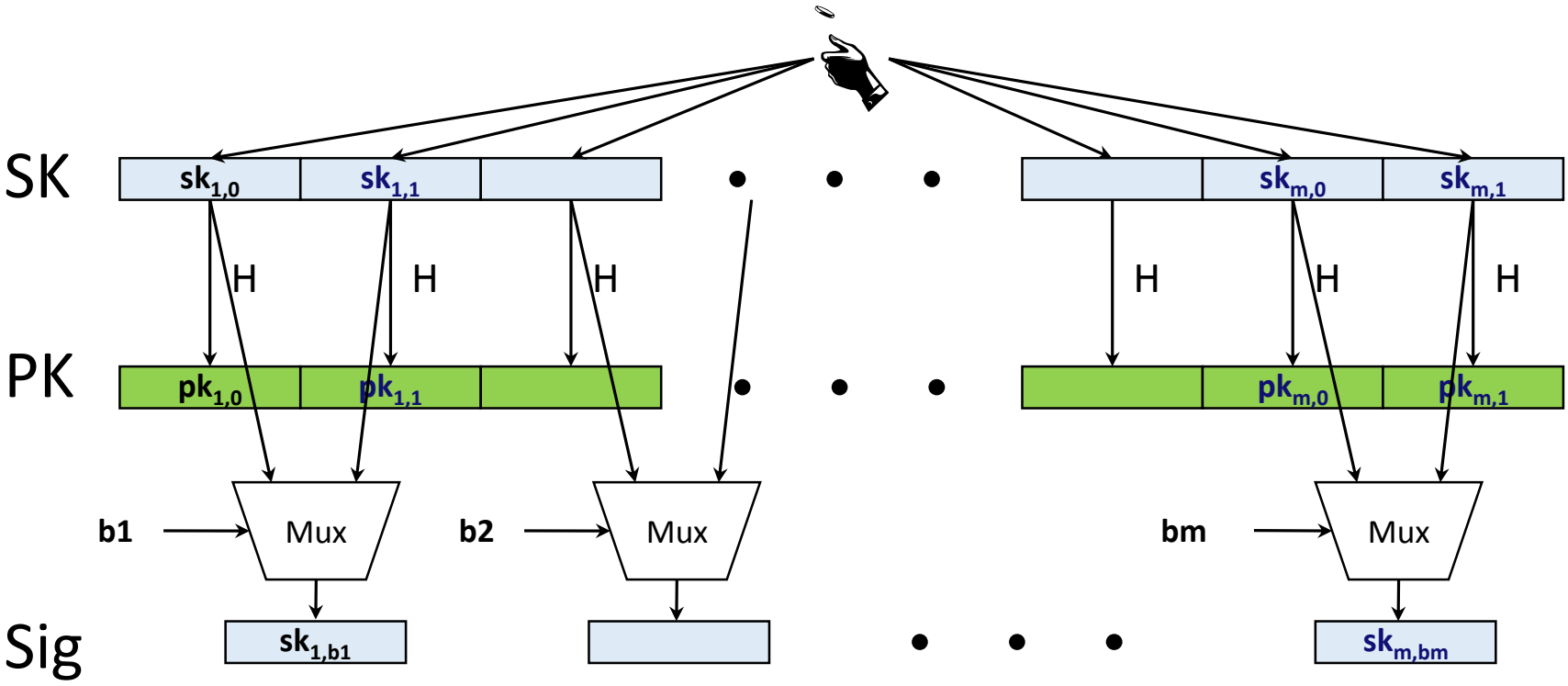


Basic Construction

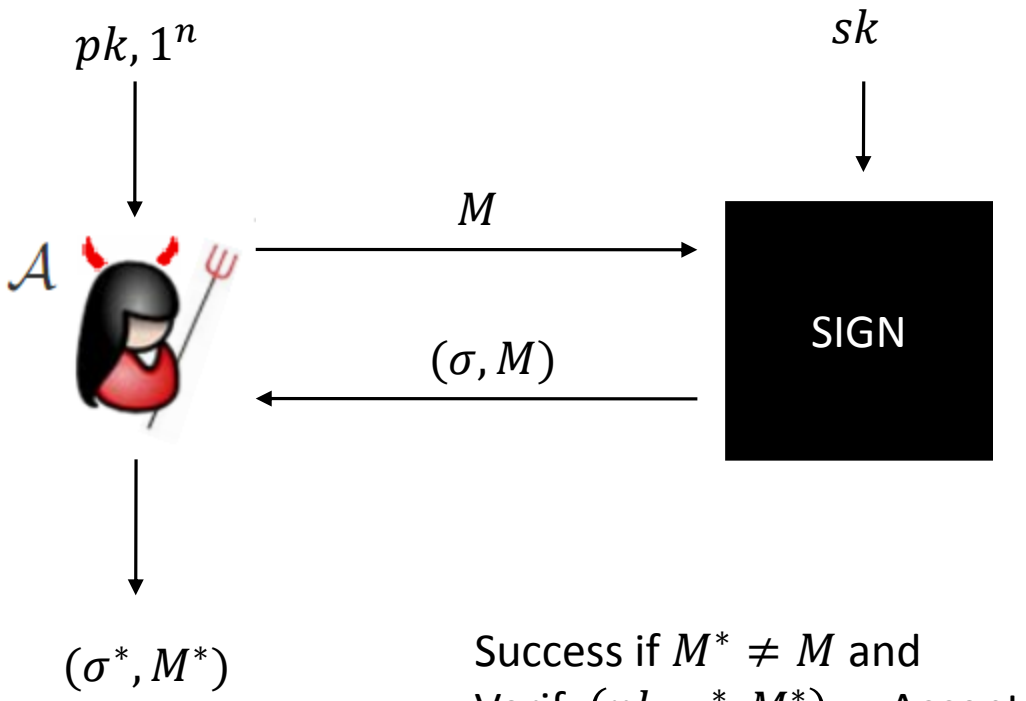


Lamport-Diffie OTS [Lam79]

Message M = b1,…,bm, OWF H * = n bit



EU-CMA for OTS



Success if $M^* \neq M$ and
Verify$(pk, \sigma^*, M^*) =$ Accept

Security

Theorem:

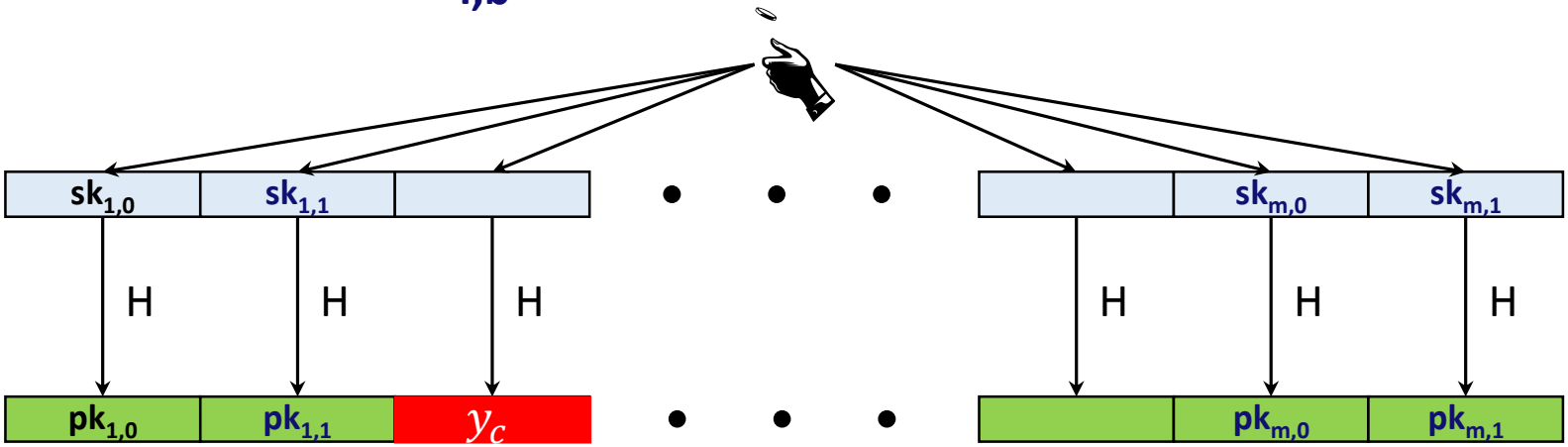
If H is one-way then LD-OTS is one-time eu-cma-secure.

Reduction

Input: y_c, k

Set $H \leftarrow h_k$

Replace random $\mathbf{pk_{i,b}}$



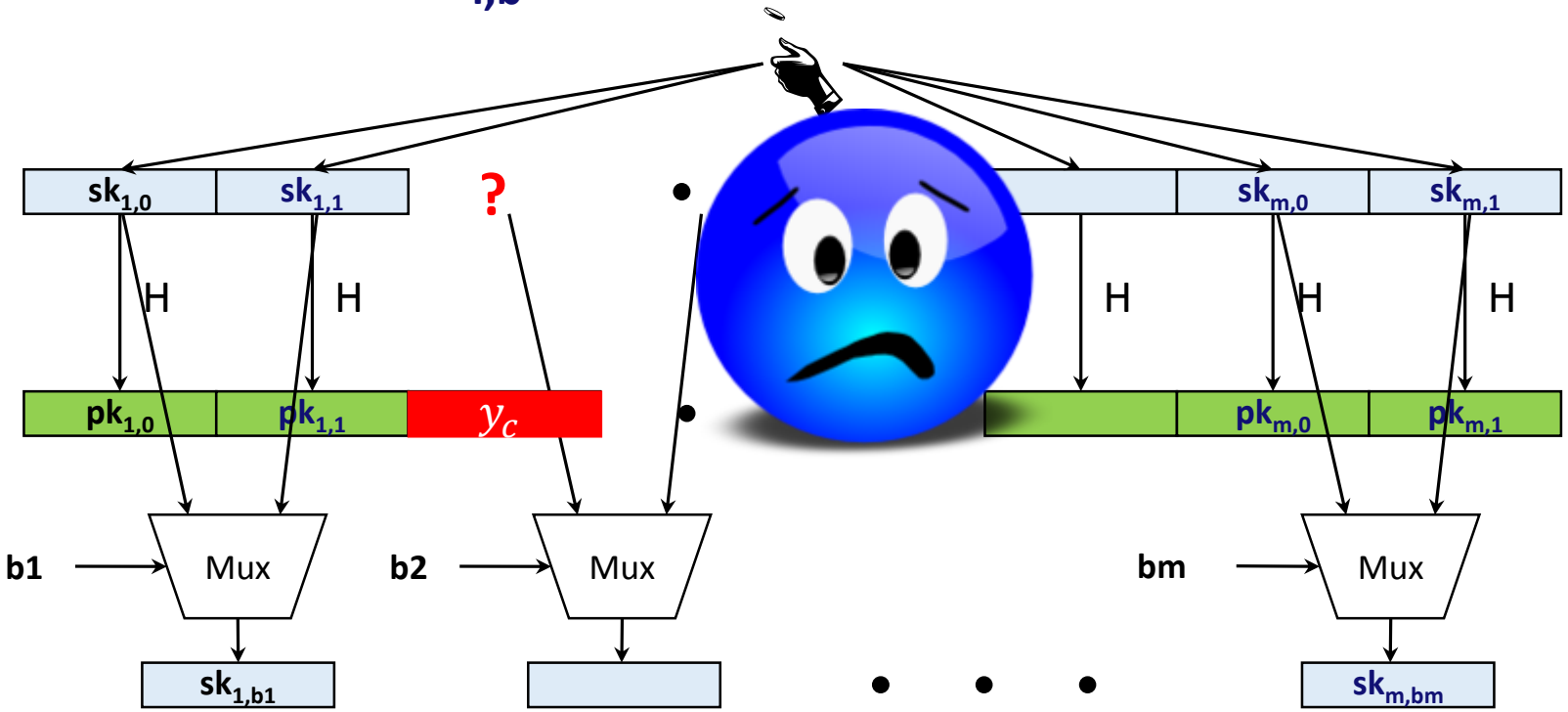
Reduction

Input: y_c, k

Set $H \leftarrow h_k$

Replace random $\mathbf{pk_{i,b}}$

Adv. Message: M = b1,…,bm
If bi = b return fail
else return Sign(M)



Reduction

Input: y_c, k

Set $H \leftarrow h_k$

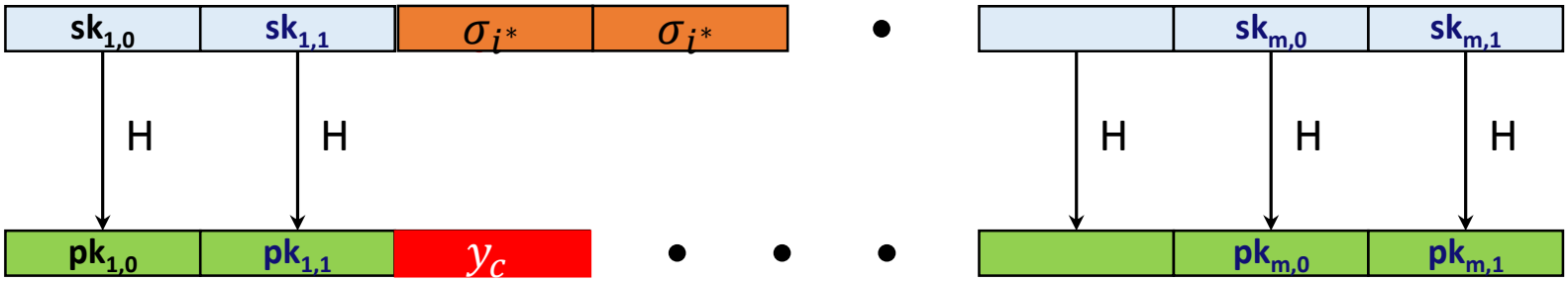
Choose random $\mathbf{pk_{i,b}}$

Forgery: M* = b1*,…,bm*,

$$\sigma = \sigma_1, \ldots, \sigma_m$$

If bi ≠ b return fail

Else return σ_{i^*}



Reduction - Analysis

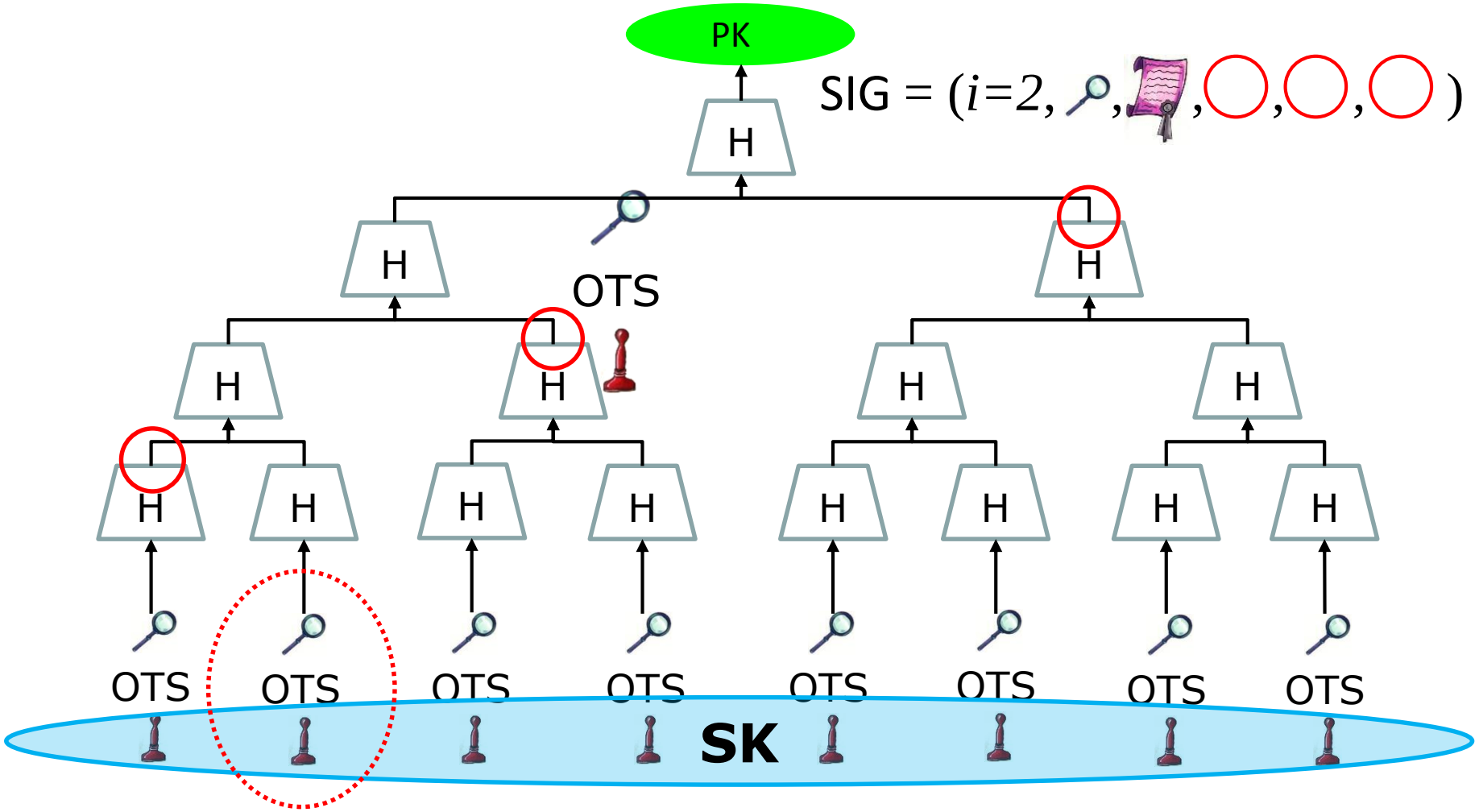
Abort in two cases:

1. $b_i = b$
   probability ½ : b is a random bit
2. $b_i \neq b$

   probability 1 - 1/m: At least one bit has to flip as $M^* \neq M$

Reduction succeeds with A‘s success probability times 1/2m.

Merkle's Hash-based Signatures



Security

Theorem:

MSS is eu-cma-secure if OTS is a one-time eu-cma secure signature scheme and H is a random element from a family of collision resistant hash functions.

Reduction

Input: k, pk_{OTS}

1. Choose random $0 \leq i < 2^h$
2. Generate key pair using pk_{OTS} as ith OTS public key and $H \leftarrow h_k$
3. Answer all signature queries using sk or sign oracle (for index i)
4. Extract OTS-forgery or collision from forgery

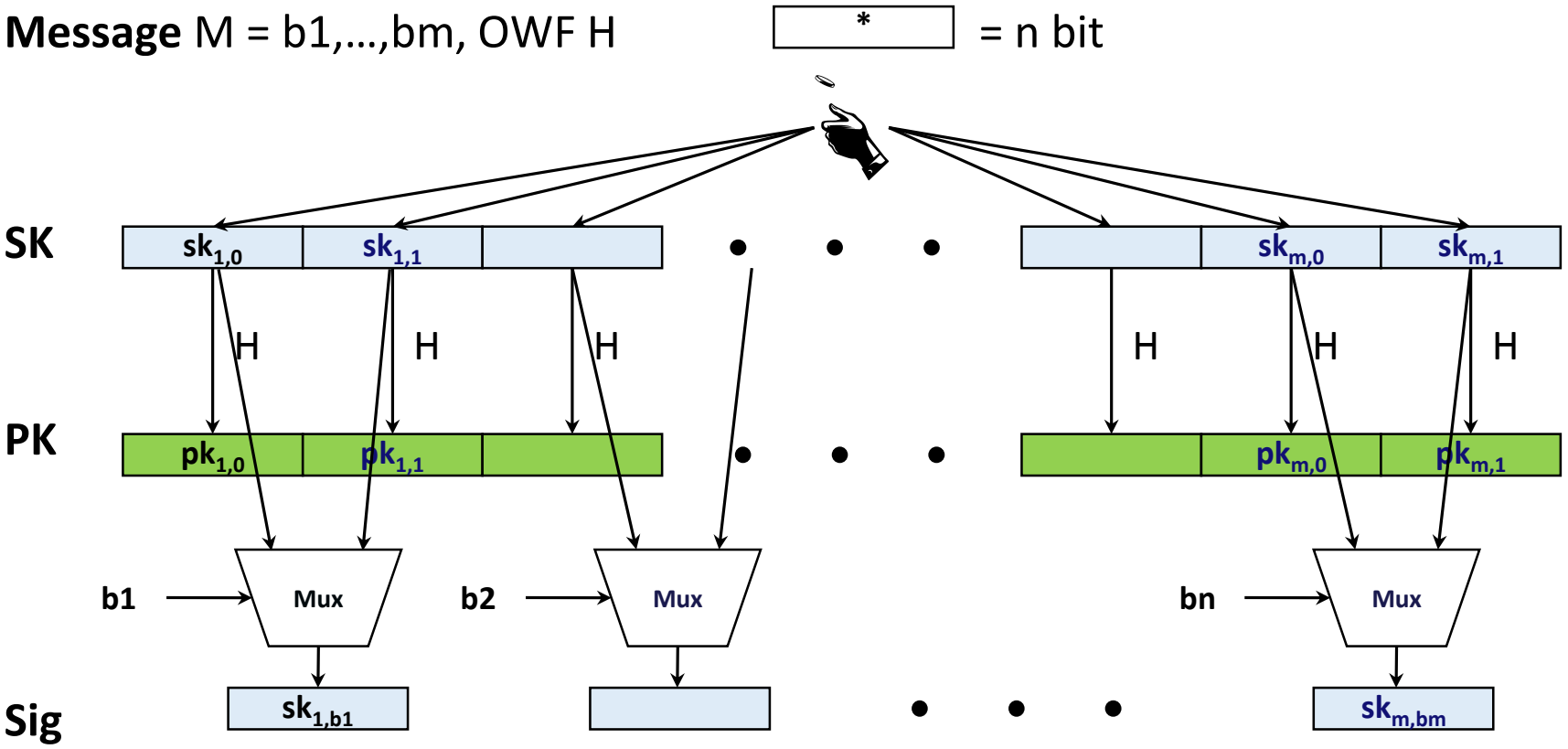
Reduction (Step 4, Extraction)

Forgery: ($i^*, \sigma^*_{OTS}, pk^*_{OTS}$, AUTH)

1. If pk^*_{OTS} equals OTS pk we used for i^* OTS, we got an OTS forgery.
   - Can only be used if $i^* = i$.
2. Else adversary used different OTS pk.
   - Hence, different leaves.
   - Still same root!
   - Pigeon-hole principle: Collision on path to root.

Winternitz-OTS

Recap LD-OTS [Lam79]



LD-OTS in MSS

SIG = (i=2,  ,  , ◯, ◯, ◯)

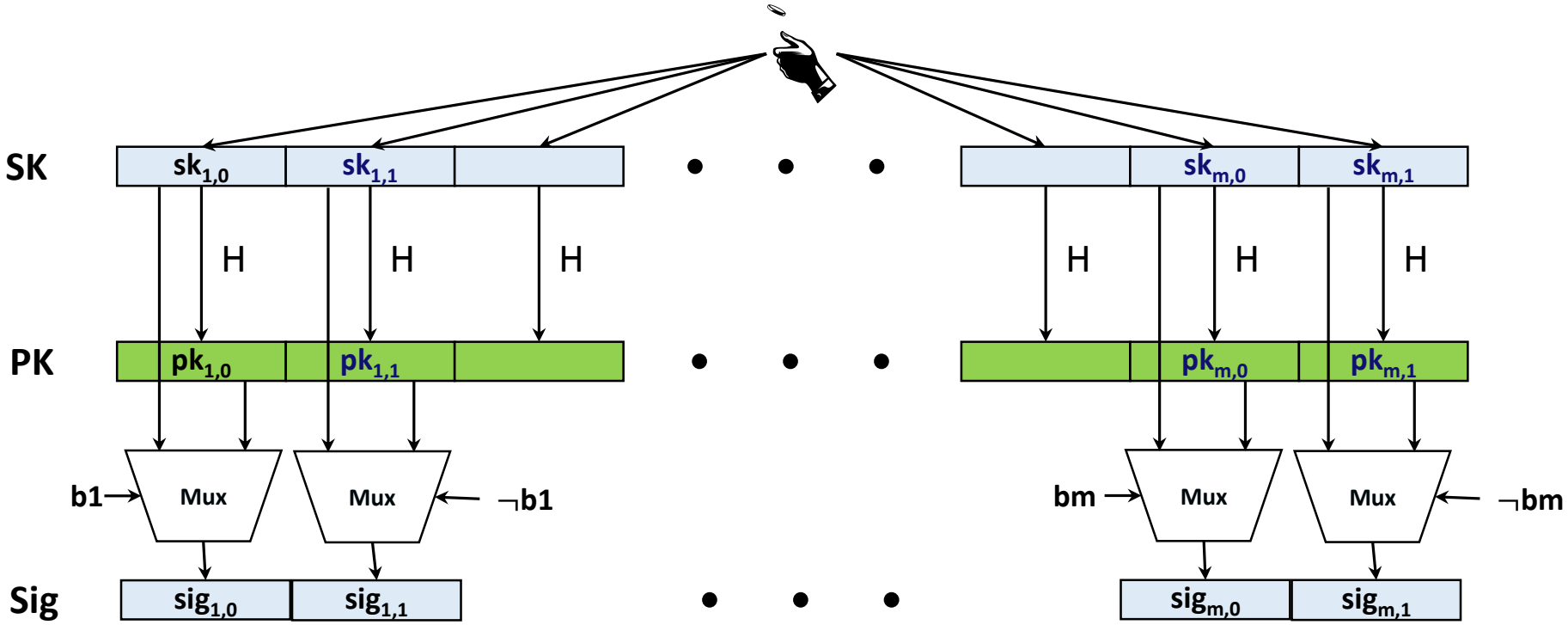
Verification:

1. Verify 
2. Verify authenticity of 

We can do better!

Trivial Optimization

Message M = b1,…,bm, OWF H * = n bit



Optimized LD-OTS in MSS

SIG = (i=2,  , ○, ○, ○)

Verification:

1. Compute  from 
2. Verify authenticity of 

Steps 1 + 2 together verify 

Germans love their „Ordnung“!

Message $M = b_1,\ldots,b_m$, OWF H

SK: $sk_1,\ldots,sk_m,sk_{m+1},\ldots,sk_{2m}$

PK: $H(sk_1),\ldots,H(sk_m),H(sk_{m+1}),\ldots,H(sk_{2m})$

Encode M: $M' = M||\neg M = b_1,\ldots,b_m,\neg b_1,\ldots,\neg b_m$

(instead of $b_1, \neg b_1,\ldots,b_m,\neg b_m$)

Sig: $sig_i =$

$$\begin{cases} sk_i & \text{, if } b_i = 1 \\ H(sk_i) & \text{, otherwise} \end{cases}$$

Checksum with bad performance!

Optimized LD-OTS

Message M = $b_1,\ldots,b_m$, OWF H

SK: $sk_1,\ldots,sk_m,sk_{m+1},\ldots,sk_{m+\log m}$

PK: $H(sk_1),\ldots,H(sk_m),H(sk_{m+1}),\ldots,H(sk_{m+\log m})$

Encode M: M‘ = $b_1,\ldots,b_m,\neg \sum_1^m b_i$

Sig: sig_i **=**

$$\begin{cases} sk_i & \text{, if } b_i = 1 \\ H(sk_i) & \text{, otherwise} \end{cases}$$

IF one b_i is flipped from 1 to 0, another b_j will flip from 0 to 1

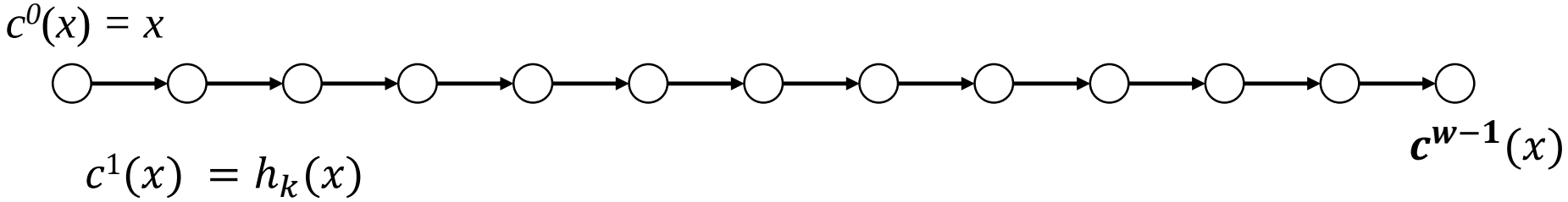
Function chains

Function family: $H_n := \{h_k : \{0,1\}^n \rightarrow \{0,1\}^n\}$

$h_k \xleftarrow{\$} H_n$

Parameter w

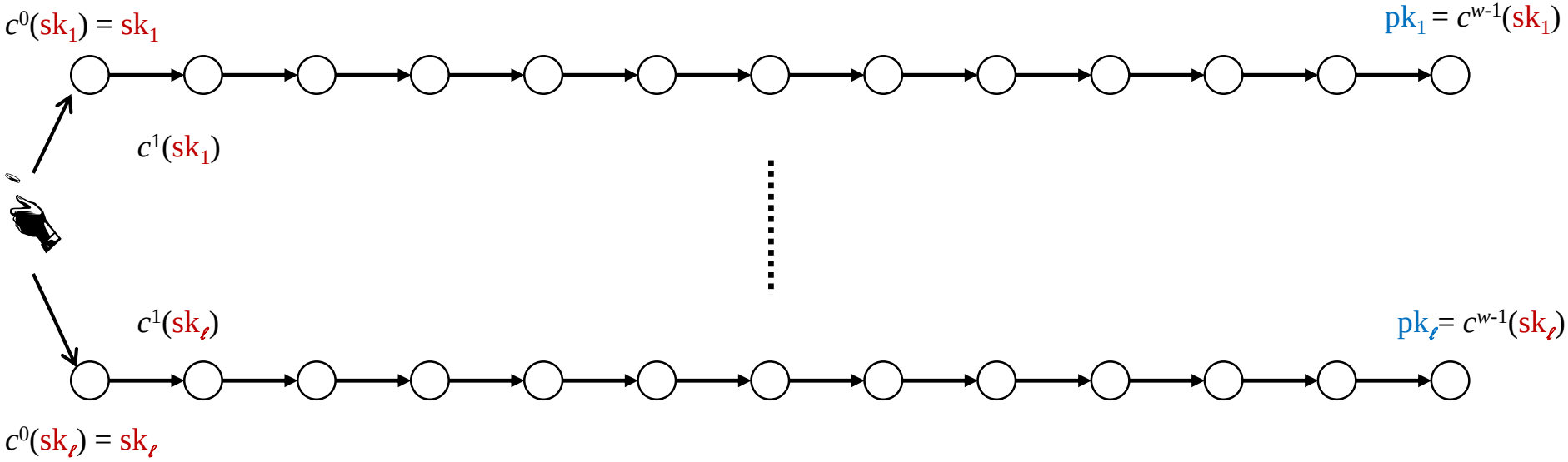
Chain: $c^i(x) = h_k(c^{i-1}(x)) = \underbrace{h_k \circ h_k \circ \ldots \circ h_k(x)}_{i-times}$



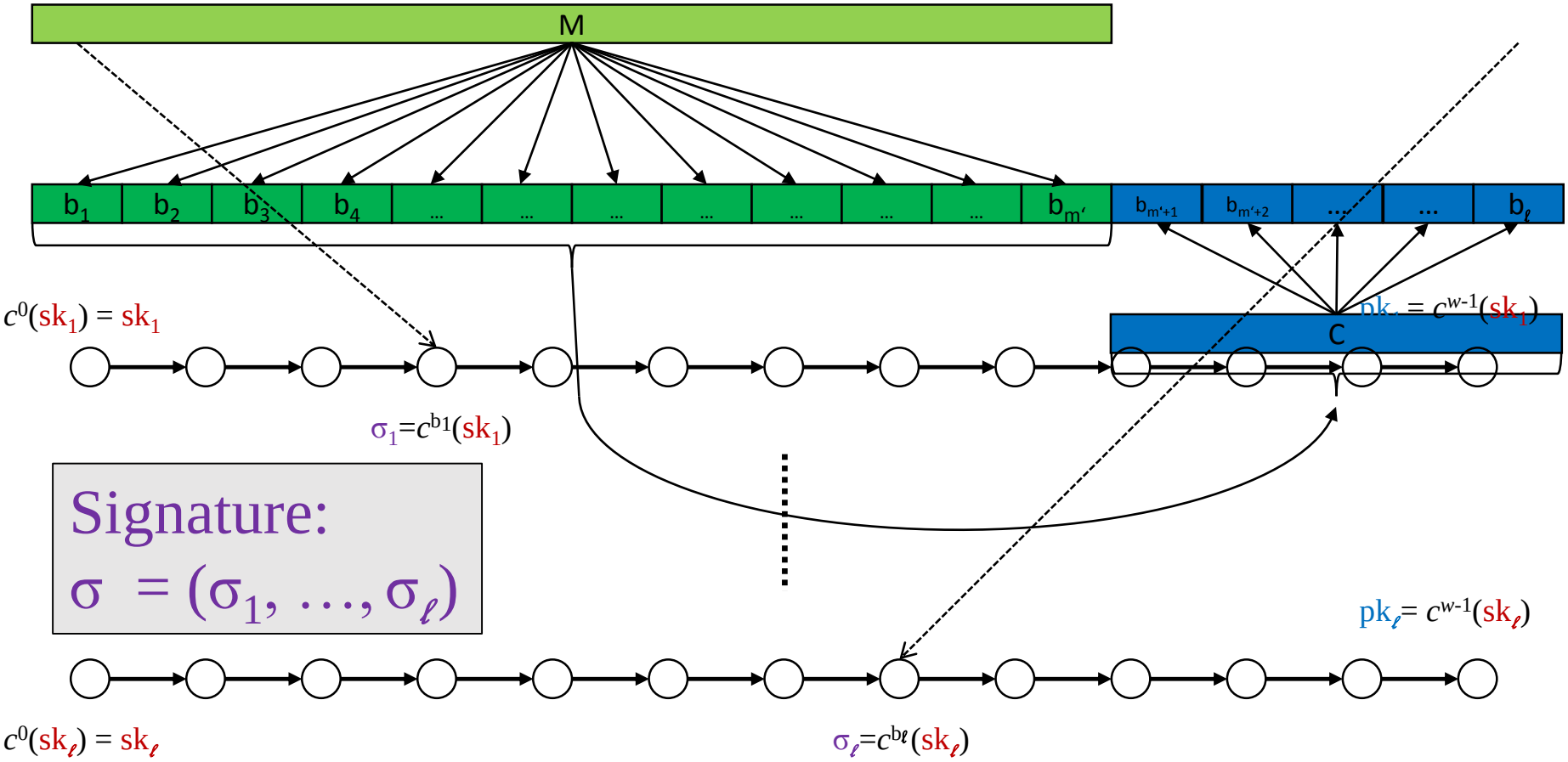
WOTS

Winternitz parameter $\boldsymbol{w}$, security parameter $\boldsymbol{n}$, message length $\boldsymbol{m}$, function family H_n

Key Generation: Compute l, sample h_k

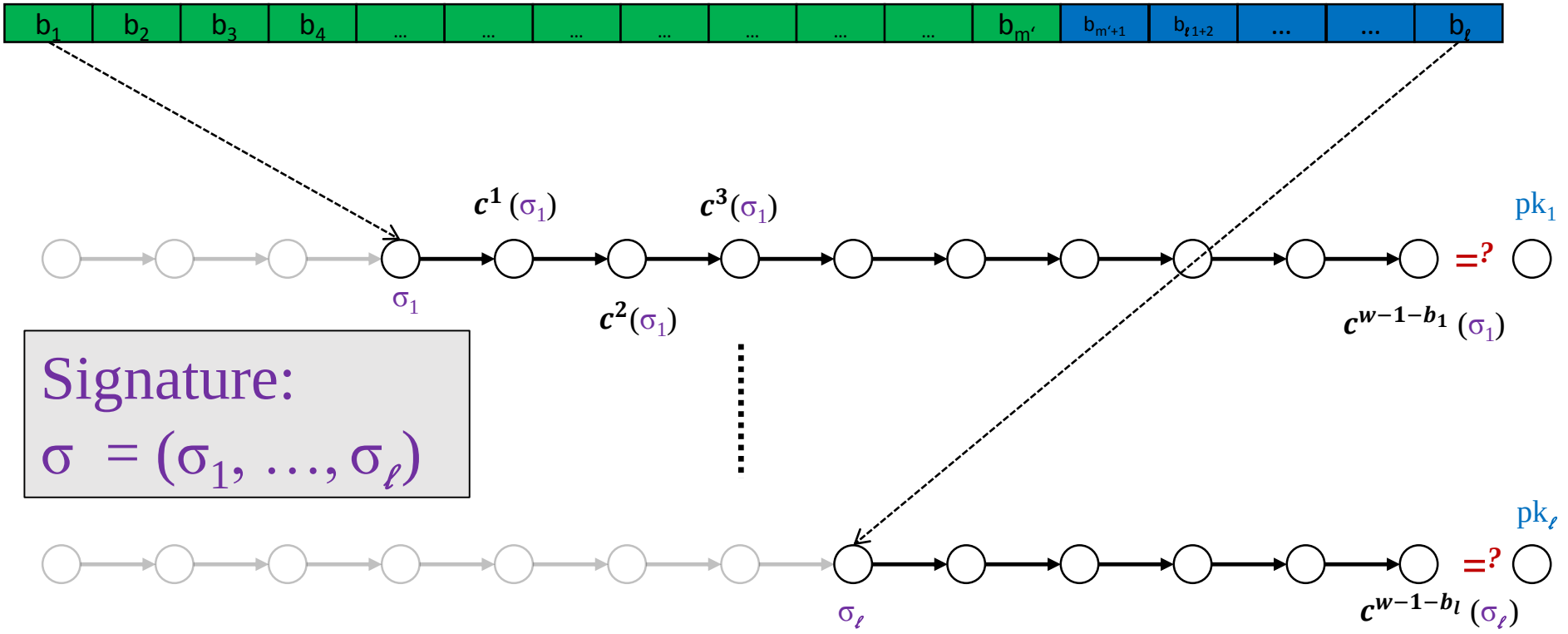


WOTS Signature generation



WOTS Signature Verification

Verifier knows: M, w



WOTS Function Chains

For $x \in \{0,1\}^n$ define $c^0(x) = x$ and

- WOTS: $c^i(x) = h_k(c^{i-1}(x))$
- WOTS$^{\$}$: $c^i(x) = h_{c^{i-1}(x)}(r)$
- WOTS^{+}: $c^i(x) = h_k(c^{i-1}(x) \oplus r_i)$

WOTS Security

Theorem (informally):

W-OTS *is strongly unforgeable under chosen message attacks if* H_n *is a collision resistant family of undetectable one-way functions.*

W-OTS$^{\$}$ *is existentially unforgeable under chosen message attacks if* H_n *is a pseudorandom function family.*

W-OTS^{+} *is strongly unforgeable under chosen message attacks if* H_n *is a* 2^{nd}*-preimage resistant family of undetectable one-way functions.*

XMSS

XMSS

Tree: Uses bitmasks

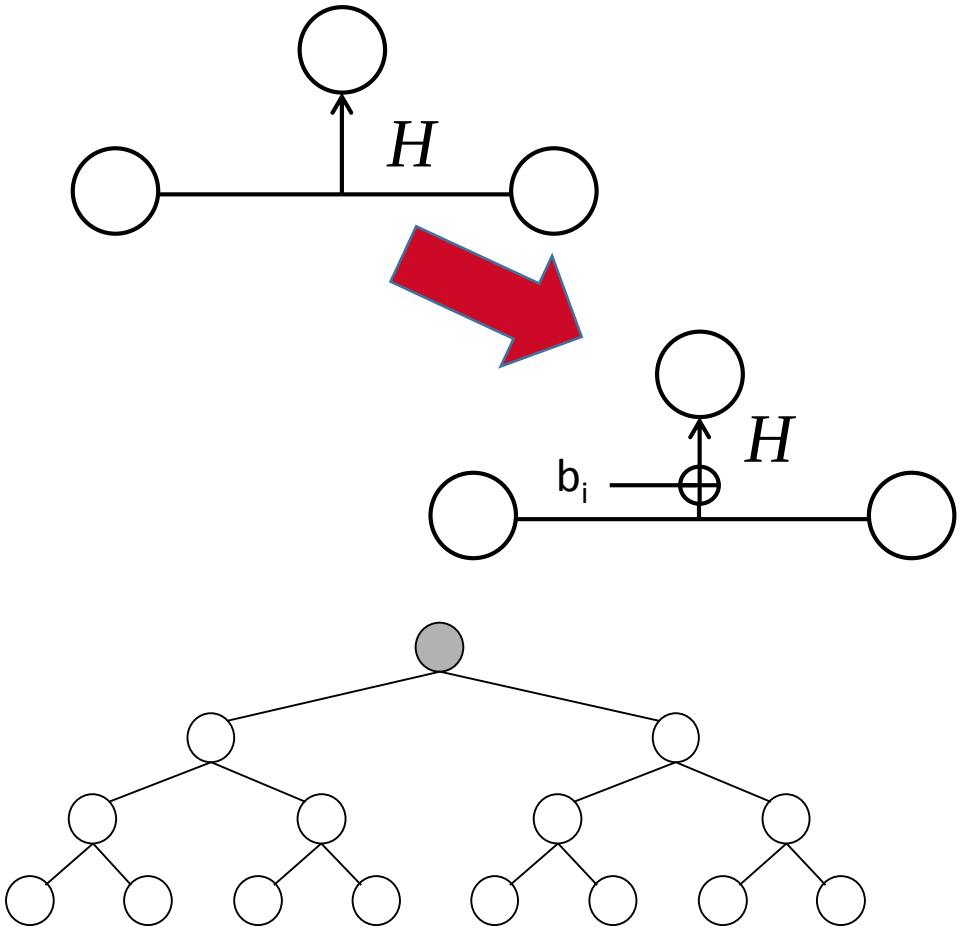
Leafs: Use binary tree with bitmasks

OTS: WOTS^{+}

Mesage digest: Randomized hashing

Collision-resilient

-> signature size halved

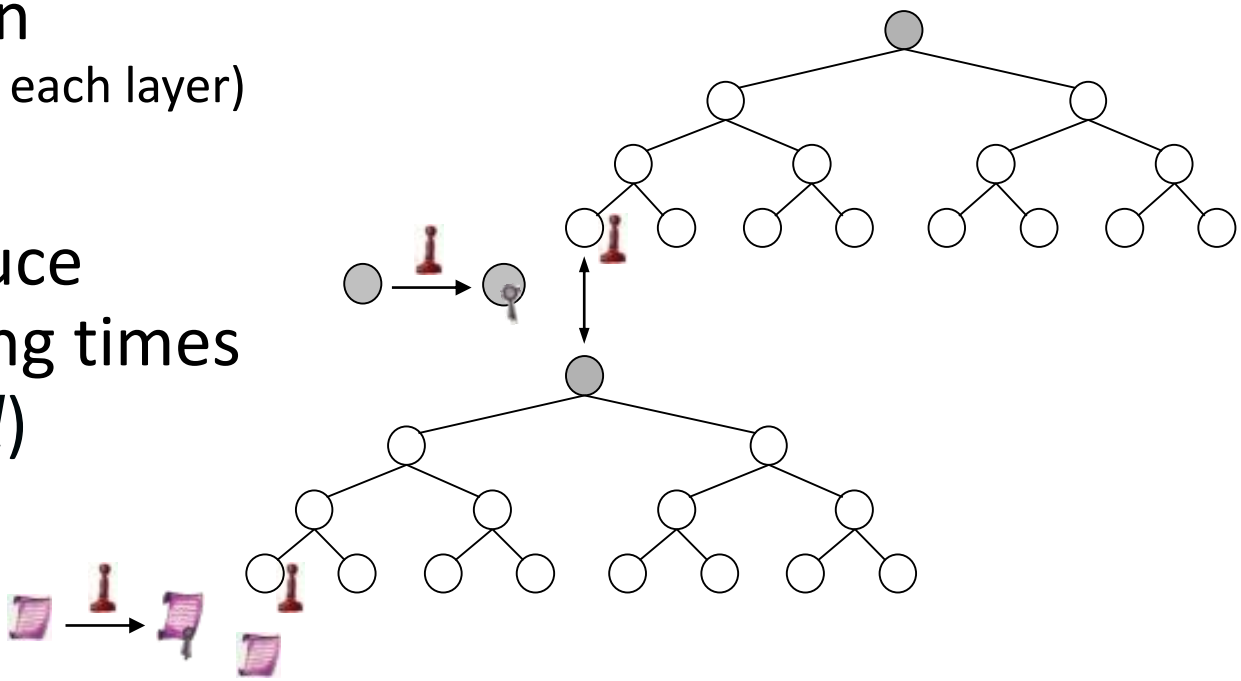


Multi-Tree XMSS

Uses multiple layers of trees

-> Key generation
(= Building first tree on each layer)
$\Theta(2^h) \rightarrow \Theta(d2^{h/d})$

-> Allows to reduce
worst-case signing times
$\Theta(\mathrm{h}/2) \rightarrow \Theta(h/2d)$



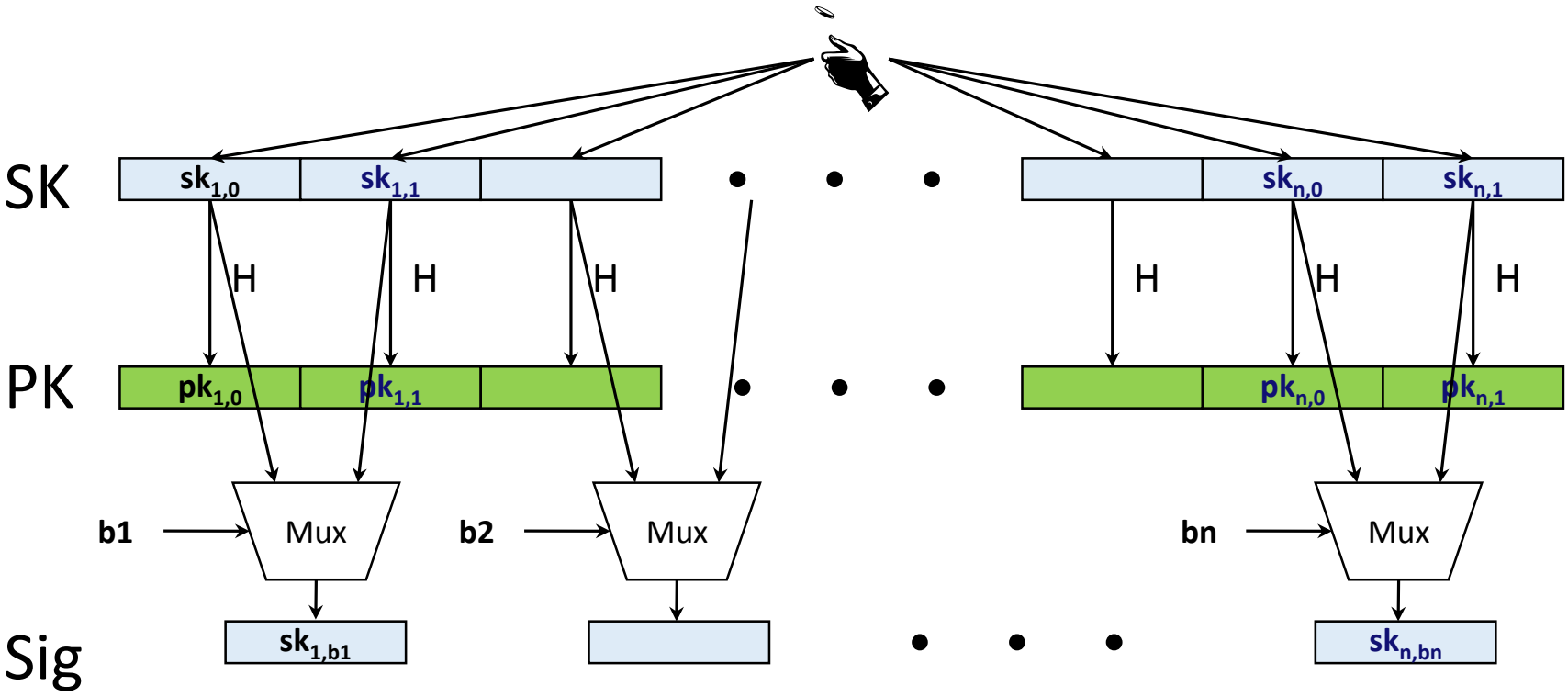

ELIMINATE
THE STATE

Protest?

Few-Time Signature Schemes

Recap LD-OTS

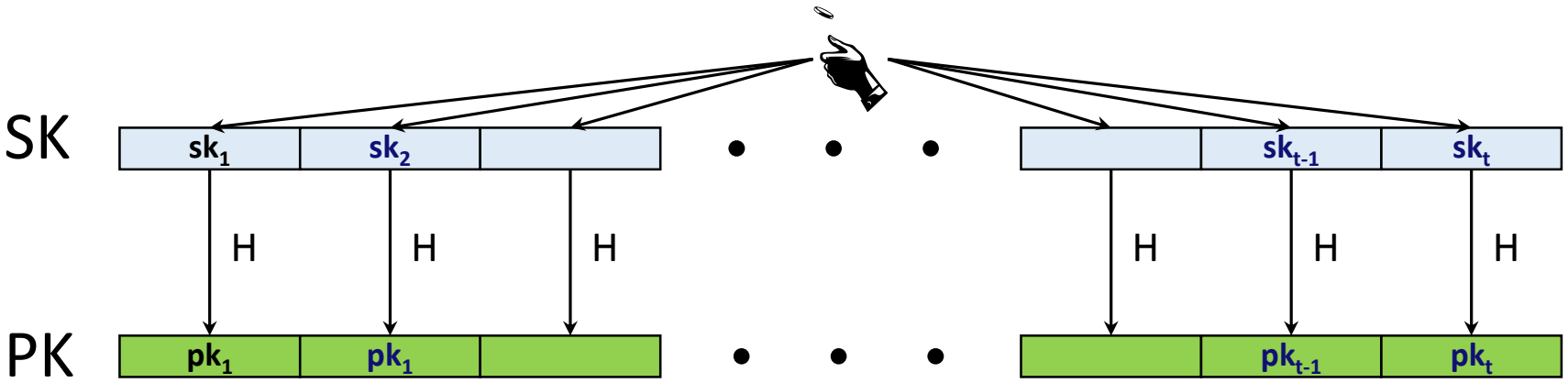
Message M = b1,…,bn, OWF H * = n bit



HORS [RR02]

Message M, OWF H, CRHF H’ [*] = n bit

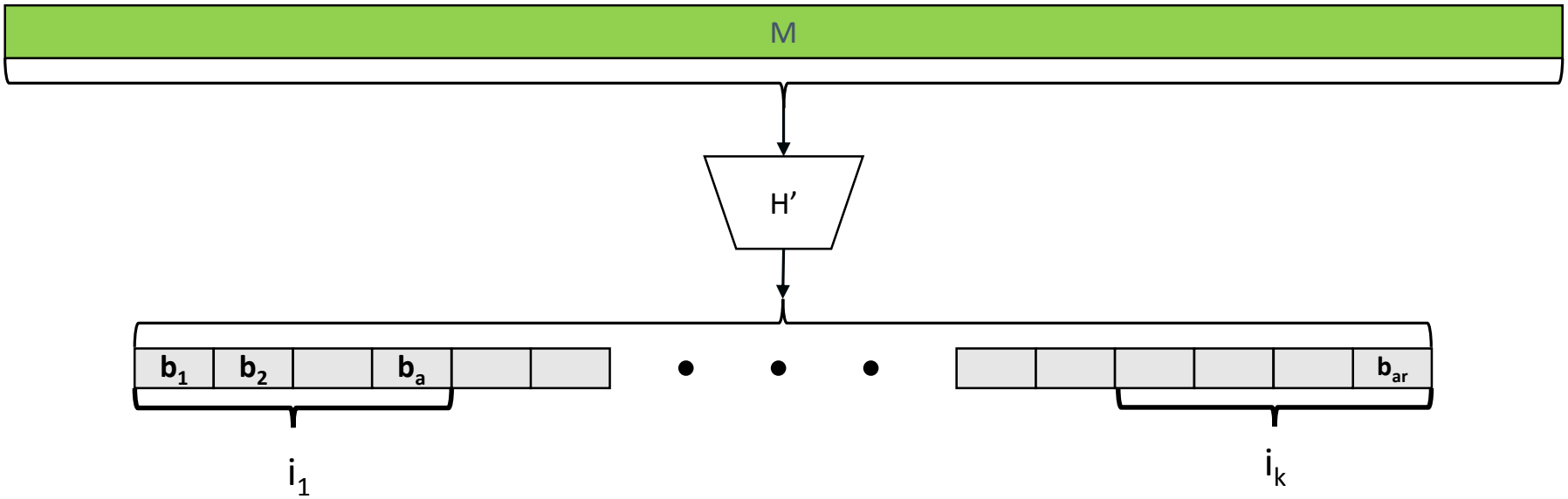
Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS mapping function

Message M, OWF H, CRHF H’ * = n bit

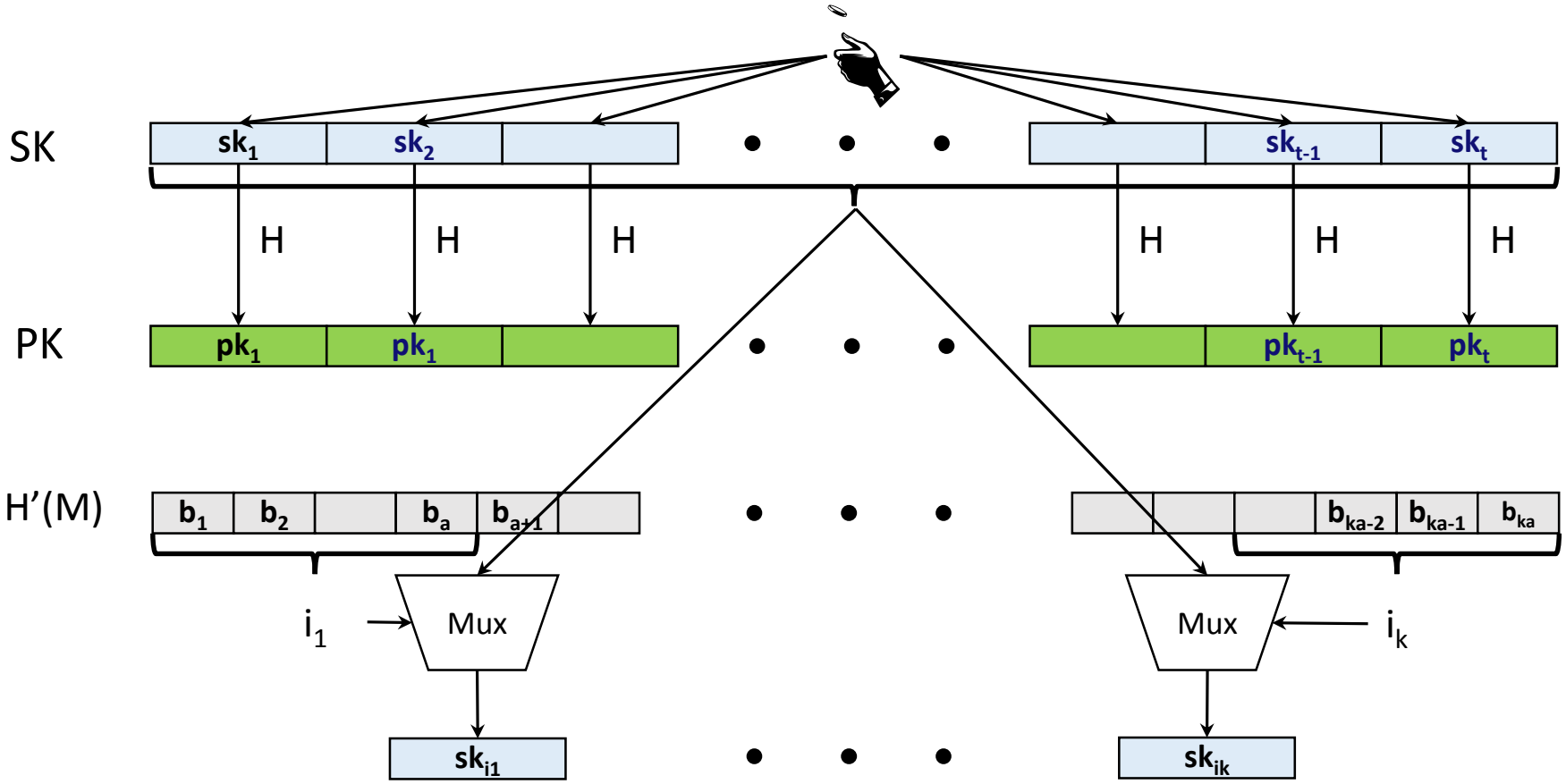
Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS

Message M, OWF H, CRHF H’ [*] = n bit

Parameters $t=2^a$,k, with m = ka (typical a=16, k=32)



HORS Security

- M mapped to k element index set $M^i \in \{1, \ldots, t\}^k$
- Each signature publishes k out of t secrets
- Either break one-wayness or…

- r-Subset-Resilience: After seeing index sets M_j^i for r messages $msg_j, 1 \leq j \leq r$, hard to find $msg_{r+1} \neq msg_j$ such that $M_{r+1}^i \in \bigcup_{1 \leq j \leq r} M_j^i$.

- Best generic attack: $\text{Succ}_{\text{r-SSR}}(A, q) = q \left(\frac{rk}{t}\right)^k$

$\longrightarrow$ Security shrinks with each signature!

HORST

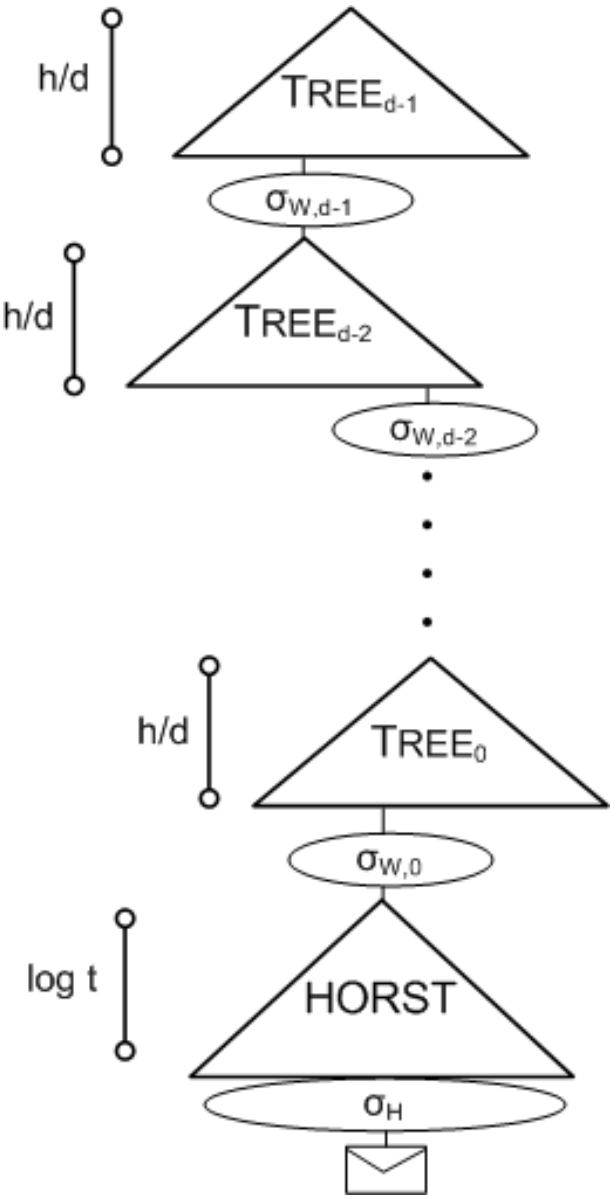
Using HORS with MSS requires adding PK (tn) to MSS signature.

HORST: Merkle Tree on top of HORS-PK

- New PK = Root
- Publish Authentication Paths for HORS signature values
- PK can be computed from Sig
- With optimizations: tn $\longrightarrow$ (k(log t − x + 1) + 2^x)n
  - E.g. SPHINCS-256: 2 MB $\longrightarrow$ 16 KB
- Use randomized message hash

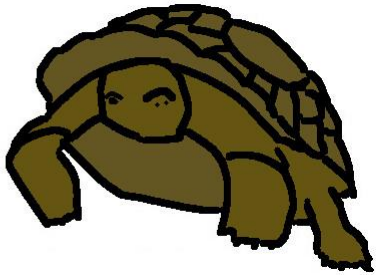
SPHINCS

- Stateless Scheme
- XMSS^{MT} + HORST
  + (pseudo-)random index
- Collision-resilient
- Deterministic signing
- SPHINCS-256:
  - 128-bit post-quantum secure
  - Hundrest of signatures / sec
  - 41 kb signature
  - 1 kb keys



PQ-Crypto is currently a hot topic

PQCRYPTO
ICT-645622



SEVENTH FRAMEWORK
PROGRAMME

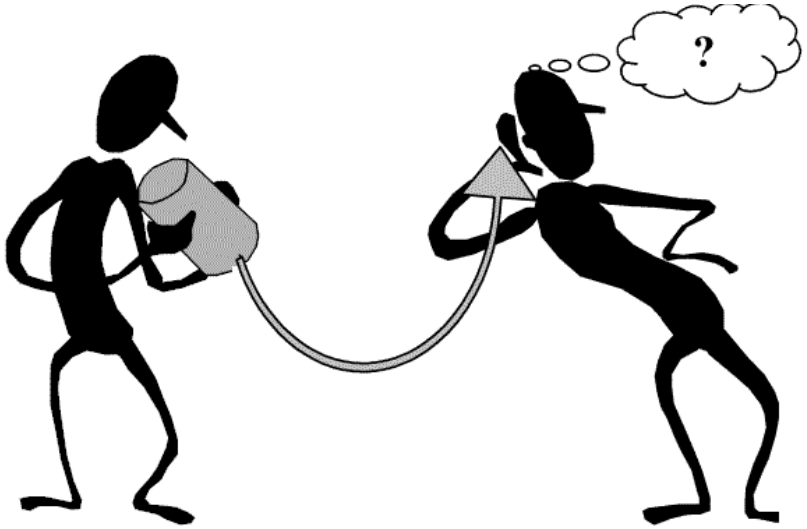
PQCrypto 2016

IETF®

NIST

ETSI
World Class Standards

Thank you!
Questions?



For references & further literature see
https://huelsing.wordpress.com/hash-based-signature-schemes/literature/