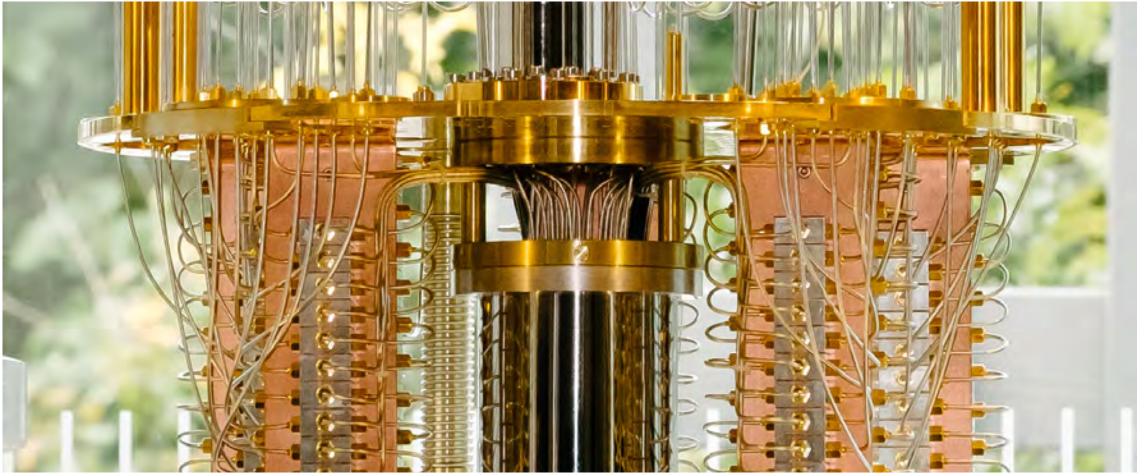
Post-Quantum Cryptography

Dr. Ruben Niederhagen, February 8, 2016

Fraunhofer SIT



Introduction
Quantum Computers

Using quantum states for computation:

Introduced in 1985 by David Deutsch [3].

- Operate on *qubits*
- using *gates*
- that perform *reversible* operations
- exploiting *entanglement* and *superposition*.

Introduction
Quantum Computers

Using quantum states for computation:

Introduced in 1985 by David Deutsch [3].

- Operate on *qubits*
- using *gates*
- that perform *reversible* operations
- exploiting *entanglement* and *superposition*.

Theoretical (since $\approx$ 1900):

- qubit: $\mathbb{C}^2$
- gate: unitary matrix over $\mathbb{C}$

Physical (since $\approx$ 1990s):

- qubit: photon, electron, atom, quantum dots...
- gate: phase shifter, EM field, laser, ...

Introduction

Quantum Computers

Using quantum states for computation:

Introduced in 1985 by David Deutsch [3].

- Operate on *qubits*
- using *gates*
- that perform *reversible* operations
- exploiting *entanglement* and *superposition*.

Theoretical (since ≈ 1900): ✓

- qubit: $\mathbb{C}^2$
- gate: unitary matrix over $\mathbb{C}$

Physical (since ≈ 1990s):

- qubit: photon, electron, atom, quantum dots...
- gate: phase shifter, EM field, laser, ...

Using quantum states for computation:

Introduced in 1985 by David Deutsch [3].

- Operate on *qubits*
- using *gates*
- that perform *reversible* operations
- exploiting *entanglement* and *superposition*.

Theoretical (since ≈ 1900): ✓

- qubit: $\mathbb{C}^2$
- gate: unitary matrix over $\mathbb{C}$

Physical (since ≈ 1990s): ?

- qubit: photon, electron, atom, quantum dots...
- gate: phase shifter, EM field, laser, ...

Quantum algorithms:

- Simon's algorithm, Deutsch–Jozsa algorithm, . . .
- Grover's algorithm: search in $\sqrt{n}$ time.
- Shor's algorithm: discrete logarithm and integer factorization in polynomial time (solve the abelian hidden subgroup problem).

Fraunhofer SIT

Introduction
Quantum Computers

Quantum algorithms:

- Simon's algorithm, Deutsch–Jozsa algorithm, . . .
- Grover's algorithm: search in $\sqrt{n}$ time.
- Shor's algorithm: discrete logarithm and integer factorization in polynomial time (solve the abelian hidden subgroup problem).

Effect on current cryptography:

- Grover reduces a brute force attack on AES-128 from time $c \cdot 2^{128}$ to time $c' \cdot 2^{64}$; similar for hash-functions.
  $\Rightarrow$ **Use 256-bit primitives!**

Quantum algorithms:

- Simon's algorithm, Deutsch–Jozsa algorithm, . . .
- Grover's algorithm: search in $\sqrt{n}$ time.
- Shor's algorithm: discrete logarithm and integer factorization in polynomial time (solve the abelian hidden subgroup problem).

Effect on current cryptography:

- Grover reduces a brute force attack on AES-128 from time $c \cdot 2^{128}$ to time $c' \cdot 2^{64}$; similar for hash-functions.
  $\Rightarrow$ **Use 256-bit primitives!**
- Shor **breaks all RSA, ECC, DHE, ECDHE, DSA, ECDSA, ..!**

Introduction
Quantum Computers

The Internet is broken, secure communication is broken; what now?

Fraunhofer
SIT

The Internet is broken, secure communication is broken; what now?

The physicist says:

Use quantum technologies to fight quantum technology!

The Internet is broken, secure communication is broken; what now?

The physicist says:

Use quantum technologies to fight quantum technology!

The cryptographer says:

Just base your crypto on math that quantum computers can't break.

Introduction
"Quantum Cryptography"

"Quantum Cryptography" is

- mainly limited to *quantum key distribution*,

Introduction

“Quantum Cryptography”

“Quantum Cryptography” is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),

Introduction
“Quantum Cryptography”

“Quantum Cryptography” is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),
- requires direct fiber-optical connection or line of sight,

Introduction

“Quantum Cryptography”

“Quantum Cryptography” is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),
- requires direct fiber-optical connection or line of sight,
- has a problem with large distances,

Introduction
“Quantum Cryptography”

“Quantum Cryptography” is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),
- requires direct fiber-optical connection or line of sight,
- has a problem with large distances,
- needs new infrastructure and new technology,

Fraunhofer SIT

Introduction
“Quantum Cryptography”

“Quantum Cryptography” is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),
- requires direct fiber-optical connection or line of sight,
- has a problem with large distances,
- needs new infrastructure and new technology,
- does not work for mobile phones, sensor networks, cars, ...

Introduction
"Quantum Cryptography"

"Quantum Cryptography" is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),
- requires direct fiber-optical connection or line of sight,
- has a problem with large distances,
- needs new infrastructure and new technology,
- does not work for mobile phones, sensor networks, cars, ...
- does not scale well, and

Introduction
"Quantum Cryptography"

"Quantum Cryptography" is

- mainly limited to *quantum key distribution*,
- provides no authentication (apart from PUF technologies),
- requires direct fiber-optical connection or line of sight,
- has a problem with large distances,
- needs new infrastructure and new technology,
- does not work for mobile phones, sensor networks, cars, ...
- does not scale well, and
- is not really necessary if one does not insist in *physical principles* but is fine with **math and computational complexity**.

Introduction
Post-Quantum Cryptography

Main task of post-quantum cryptography [2]:

Find mathematically hard problems that

- cannot be broken by classical computers,

Introduction
Post-Quantum Cryptography

Main task of post-quantum cryptography [2]:

Find mathematically hard problems that

- cannot be broken by classical computers,
- cannot be broken by quantum computers,

Introduction
Post-Quantum Cryptography

Main task of post-quantum cryptography [2]:

Find mathematically hard problems that

- cannot be broken by classical computers,
- cannot be broken by quantum computers,
- provide a trapdoor for asymmetric crypto, and

Introduction
Post-Quantum Cryptography

Main task of post-quantum cryptography [2]:

Find mathematically hard problems that

- cannot be broken by classical computers,
- cannot be broken by quantum computers,
- provide a trapdoor for asymmetric crypto, and
- can be used efficiently in terms of
  - time,
  - memory, and
  - communication.

Introduction
Post-Quantum Cryptography

Current approaches are:

- code-based cryptography,
- multivariate cryptography,
- hash-based cryptography,
- lattice-based cryptography, and
- supersingular elliptic curve isogenies.

Code-based Cryptography

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

01101100

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Code-based Cryptography
Error-Correcting Codes

$$01101100 \xrightarrow[\text{encode}]{} 10011001001$$

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Fraunhofer SIT

$$01101100 \xrightarrow[\text{encode}]{} 10011001001 \overset{\text{transmitt}}{\dashrightarrow} 10010\,001011$$

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Fraunhofer SIT

Code-based Cryptography

Error-Correcting Codes

$$01101100 \xrightarrow[\text{encode}]{} 10011001001 \dashrightarrow_{\text{transmitt}} 10011\ 001001$$

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Code-based Cryptography
Error-Correcting Codes

$$01101100 \xrightarrow[\text{encode}]{} 10011001001 \dashrightarrow[\text{transmitt}]{} 1001\mathbf{1}\ 0010\mathbf{0}1 \xrightarrow[\text{decode}]{} 01101100$$

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Code-based Cryptography
Error-Correcting Codes

$$01101100 \xrightarrow[\text{encode}]{} 10011001001 \dashrightarrow_{\text{transmitt}} 10011\ 001001 \xrightarrow[\text{decode}]{} 01101100$$

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Practical application requires *efficient* encoding and decoding algorithms.

Fraunhofer SIT

Code-based Cryptography

Error-Correcting Codes

$$01101100 \xrightarrow[\text{encode}]{} 10011001001 \dashrightarrow_{\text{transmitt}} 10011\ 001001 \xrightarrow[\text{decode}]{} 01101100$$

Error correction on a noisy channel:

Add redundant information to the message that allows to detect and correct bit-errors.

Practical application requires *efficient* encoding and decoding algorithms.

Encoding: Multiply message vector with *generator matrix*.

Decoding: Use *decoding algorithm* of the code.

Fraunhofer SIT

Code-based Cryptography
McEliece Crypto System

- System Parameters: $n, t \in \mathbb{N}$, where $t \ll n$.
- Key Generation:

$$\begin{aligned} \mathsf{G} &: \ k \times n \text{ generator matrix of a code } \mathcal{G}, \\ \mathsf{S} &: \ k \times k \text{ random non-singular matrix}, \\ \mathsf{P} &: \ n \times n \text{ random permutation matrix}. \end{aligned}$$

  Compute $k \times n$ matrix $\mathsf{G}^{\mathsf{pub}} = \mathsf{SGP}$.

- Public Key: $(\mathsf{G}^{\mathsf{pub}}, t)$
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$
  where $D_{\mathcal{G}}$ is an efficient decoding algorithm for $\mathcal{G}$.

Code-based Cryptography
McEliece Crypto System

- Public Key: $(\mathsf{G}^{\mathsf{pub}}, t)$ (recall: $\mathsf{G}^{\mathsf{pub}} = \mathsf{SGP}$)
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$.

Code-based Cryptography
McEliece Crypto System

- Public Key: $(\mathsf{G}^{\mathsf{pub}}, t)$ (recall: $\mathsf{G}^{\mathsf{pub}} = \mathsf{SGP}$)
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$.
- Encryption: to encrypt message $\mathsf{m} \in \mathbb{F}_2^k$,
  randomly choose $\mathsf{e} \in \mathbb{F}_2^n$ of weight t; compute

$$\mathsf{c} = \mathsf{m}\mathsf{G}^{\mathsf{pub}} \oplus \mathsf{e}.$$

Code-based Cryptography

McEliece Crypto System

- Public Key: $(\mathsf{G}^{\mathsf{pub}}, t)$ (recall: $\mathsf{G}^{\mathsf{pub}} = \mathsf{SGP}$)
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$.
- Encryption: to encrypt message $\mathsf{m} \in \mathbb{F}_2^k$,
  randomly choose $\mathsf{e} \in \mathbb{F}_2^n$ of weight t; compute

$$\mathsf{c} = \mathsf{m}\mathsf{G}^{\mathsf{pub}} \oplus \mathsf{e}.$$

- Decryption: compute

$$\mathsf{c}' = \mathsf{c}\mathsf{P}^{-1} = \mathsf{mSG} \oplus \mathsf{e}\mathsf{P}^{-1},$$

use $D_{\mathcal{G}}$ to decode c' to $\mathsf{m}' = \mathsf{mS}$,
compute

$$\mathsf{m} = \mathsf{m}'\mathsf{S}^{-1} = \mathsf{mSS}^{-1}.$$

McEliece problem:

Given a McEliece public key $(\mathsf{G}^{\mathsf{pub}}, t)$, $\mathsf{G}^{\mathsf{pub}} \in \{0,1\}^{k \times n}$ and a cipher text $\mathsf{c} \in \{0,1\}^n$, find a message $\mathsf{m} \in \{0,1\}^k$ with $w_{\mathsf{H}}(\mathsf{m}\mathsf{G}^{\mathsf{pub}} - \mathsf{c}) = t$.

McEliece problem:

Given a McEliece public key $(\mathsf{G}^{\mathsf{pub}}, t)$, $\mathsf{G}^{\mathsf{pub}} \in \{0,1\}^{k \times n}$ and a cipher text $\mathsf{c} \in \{0,1\}^n$, find a message $\mathsf{m} \in \{0,1\}^k$ with $w_{\mathsf{H}}(\mathsf{m}\mathsf{G}^{\mathsf{pub}} - \mathsf{c}) = t$.

The hardness of this problem depends on the specific code.
McEliece proposes to use binary Goppa codes.

Code-based Cryptography
Niederreiter Crypto System

- System Parameters: $n, t \in \mathbb{N}$, where $t \ll n$.
- Key Generation:
  H : $(n-k) \times n$ parity check matrix of a code $\mathcal{G}$,
  P : $n \times n$ random permutation matrix.
  Compute
  S : $(n-k) \times (n-k)$ non-singular matrix, and
  $\mathrm{H}^{\mathrm{pub}}$: $(n-k) \times n$ matrix
  such that $\mathrm{SHP} = \left(\mathrm{Id}_{n-k} \mid \mathrm{H}^{\mathrm{pub}}\right)$.
- Public Key: $(\mathrm{H}^{\mathrm{pub}}, t)$
- Private Key: $(\mathrm{S}, D_{\mathcal{G}}, \mathrm{P})$
  where $D_{\mathcal{G}}$ is an efficient *syndrome* decoding algorithm for $\mathcal{G}$.

Code-based Cryptography
Niederreiter Crypto System

- Public Key: $(\mathsf{H}^{\mathsf{pub}}, t)$ (recall: $\left(\mathsf{Id}_{n-k} \mid \mathsf{H}^{\mathsf{pub}}\right) = \mathsf{SHP}$)
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$.

Code-based Cryptography
Niederreiter Crypto System

- Public Key: $(\mathsf{H}^{\mathsf{pub}}, t)$ (recall: $\left(\mathsf{Id}_{n-k} \mid \mathsf{H}^{\mathsf{pub}}\right) = \mathsf{SHP}$)
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$.
- Encryption: to encrypt message $\mathsf{e} \in \mathbb{F}_2^n$ of weight t, compute the syndrome

$$\mathsf{s} = \left(\mathsf{Id}_{n-k} \mid \mathsf{H}^{\mathsf{pub}}\right) \mathsf{e}^{\mathsf{T}}.$$

Code-based Cryptography
Niederreiter Crypto System

- Public Key: $(\mathsf{H}^{\mathsf{pub}}, t)$ (recall: $\left(\mathsf{Id}_{n-k} \mid \mathsf{H}^{\mathsf{pub}}\right) = \mathsf{SHP}$)
- Private Key: $(\mathsf{S}, D_{\mathcal{G}}, \mathsf{P})$.
- Encryption: to encrypt message $\mathsf{e} \in \mathbb{F}_2^n$ of weight t, compute the syndrome

$$\mathsf{s} = \left(\mathsf{Id}_{n-k} \mid \mathsf{H}^{\mathsf{pub}}\right) \mathsf{e}^{\mathsf{T}}.$$

- Decryption: compute

$$\mathsf{s}' = \mathsf{S}^{-1} s = \mathsf{HPe}^{\mathsf{T}},$$

use $D_{\mathcal{G}}$ to recover $\mathsf{e}' = \mathsf{Pe}^{\mathsf{T}}$, compute

$$\mathsf{e}^{\mathsf{T}} = \mathsf{P}^{-1}\mathsf{e}' = \mathsf{P}^{-1}\mathsf{Pe}^{\mathsf{T}}.$$

Fraunhofer SIT

Code-based Cryptography
McEliece and Niederreiter

Recommended parameters:

$$
\begin{aligned}
n &= 6960 \\
m &= 13 \\
t &= 119 \\
k &= n - mt = 5413
\end{aligned}
$$

Estimated security level: 266 bit.

Public key size: $(n-k)k$ bits $\approx 1,046,739$ bytes.

Code-based Cryptography

McEliece and Niederreiter

Recommended parameters:

$$
\begin{aligned}
n &= 6960 \\
m &= 13 \\
t &= 119 \\
k &= n - mt = 5413
\end{aligned}
$$

Estimated security level: 266 bit.

Public key size: $(n-k)k$ bits $\approx 1{,}046{,}739$ bytes.

Disadvantages of McEliece and Niederreiter:

- Large key size when using binary Goppa codes.

Fraunhofer SIT

Code-based Cryptography

Further improvements for code-based schemes:

Use codes with a more compact representation, e.g. cyclic codes.

Code-based Cryptography

Further improvements for code-based schemes:

Use codes with a more compact representation, e.g. cyclic codes.

Problems with decoding errors!

Code-based Cryptography

Further improvements for code-based schemes:

Use codes with a more compact representation, e.g. cyclic codes.

Problems with decoding errors!

Further code-based schemes:

- Signature schemes, e.g., CFS: large (huge?) public keys.

Code-based Cryptography

Further improvements for code-based schemes:

Use codes with a more compact representation, e.g. cyclic codes.

Problems with decoding errors!

Further code-based schemes:

- Signature schemes, e.g., CFS: large (huge?) public keys.
- Cryptographic hash functions, e.g., FSB: no competitive performance.

Code-based Cryptography

Further improvements for code-based schemes:

Use codes with a more compact representation, e.g. cyclic codes.

Problems with decoding errors!

Further code-based schemes:

- Signature schemes, e.g., CFS: large (huge?) public keys.
- Cryptographic hash functions, e.g., FSB: no competitive performance.
- Pseudo random number generators: no competitive performance?

Multivariate Cryptography

$$
\begin{aligned}
5x_1^3x_2x_3^2 + 17x_2^4x_3 + 23x_1^2x_2^4 + 13x_1 + 12x_2 + 5 &= 0\\
12x_1^2x_2^3x_3 + 15x_1x_3^3 + 25x_2x_3^3 + 5x_1 + 6x_3 + 12 &= 0\\
28x_1x_2x_3^4 + 14x_2^3x_3^2 + 16x_1x_3 + 32x_2 + 7x_3 + 10 &= 0\\
54x_1^6x_3 + 2x_1^4 + 59x_1^2x_2^3 + 42x_1^2x_3^7 + x_1 + 17 &= 0
\end{aligned}
$$

Underlying problem:

Solving a system of m multivariate polynomial equations in n variables over $\mathbb{F}_q$ is called the **MP problem**.

Multivariate Cryptography
Introduction

Underlying problem:

Solving a system of m multivariate polynomial equations in n variables over $\mathbb{F}_q$ is called the **MP problem**.

Example

$$5x_1^3x_2x_3^2 + 17x_2^4x_3 + 23x_1^2x_2^4 + 13x_1 + 12x_2 + 5 = 0$$
$$12x_1^2x_2^3x_3 + 15x_1x_3^3 + 25x_2x_3^3 + 5x_1 + 6x_3 + 12 = 0$$
$$28x_1x_2x_3^4 + 14x_2^3x_3^2 + 16x_1x_3 + 32x_2 + 7x_3 + 10 = 0$$

Underlying problem:

Solving a system of m multivariate polynomial equations in n variables over $\mathbb{F}_q$ is called the **MP problem**.

Example

$$
\begin{aligned}
5x_1^3x_2x_3^2 + 17x_2^4x_3 + 23x_1^2x_2^4 + 13x_1 + 12x_2 + 5 = 0\\
12x_1^2x_2^3x_3 + 15x_1x_3^3 + 25x_2x_3^3 + 5x_1 + 6x_3 + 12 = 0\\
28x_1x_2x_3^4 + 14x_2^3x_3^2 + 16x_1x_3 + 32x_2 + 7x_3 + 10 = 0
\end{aligned}
$$

Hardness:

The MP problem is an *NP-complete* problem even for multivariate *quadratic* systems and $q = 2$.

Multivariate Cryptography

Introduction

Underlying problem:

Solving a system of m multivariate polynomial equations in n variables over $\mathbb{F}_q$ is called the **MP problem**.

Example

$$\begin{aligned} x_3x_2 + x_2x_1 + x_2 + x_1 + 1 &= 0 \\ x_3x_1 + x_3x_2 + x_3 + x_1 &= 0 \\ x_3x_2 + x_3x_1 + x_3 + x_2 &= 0 \end{aligned}$$

Hardness:

The MP problem is an *NP-complete* problem even for multivariate *quadratic* systems and $q = 2$.

Notation:

For a set $f = (f_1, \ldots, f_m)$ of m quadratic polynomials in n variables over $\mathbb{F}_2$, let $f(x) = (f_1(x), \ldots, f_m(x)) \in \mathbb{F}_2^m$ be the solution vector of the evaluation of f for $x \in \mathbb{F}_2^n$.

Multivariate Cryptography
Introduction

Notation:

For a set $f = (f_1, \ldots, f_m)$ of m quadratic polynomials in n variables over $\mathbb{F}_2$, let $f(x) = (f_1(x), \ldots, f_m(x)) \in \mathbb{F}_2^m$ be the solution vector of the evaluation of f for $x \in \mathbb{F}_2^n$.

Definition ($\mathcal{MQ}$ over $\mathbb{F}_2$)

Let $\mathcal{MQ}(\mathbb{F}_2^n, \mathbb{F}_2^m)$ be the set of all systems of quadratic equations in n variables and m equations over $\mathbb{F}_2$.
We call one element $P \in \mathcal{MQ}(\mathbb{F}_2^n, \mathbb{F}_2^m)$ an instance of $\mathcal{MQ}$ over $\mathbb{F}_2$.

Multivariate Cryptography
Basic Idea for Multivariate Public Key Cryptography (MPKC)

- System Parameters: $m, n, \in \mathbb{N}$.
- Key Generation: choose "random" $f \in \mathcal{MQ}(\mathbb{F}_2^n, \mathbb{F}_2^m)$ such that f^{-1} is secretly known.
- Public Key: f.
- Private Key: f^{-1}.

Multivariate Cryptography

Basic Idea for Multivariate Public Key Cryptography (MPKC)

- System Parameters: $m, n, \in \mathbb{N}$.
- Key Generation: choose "random" $f \in \mathcal{MQ}(\mathbb{F}_2^n, \mathbb{F}_2^m)$ such that f^{-1} is secretly known.
- Public Key: f.
- Private Key: f^{-1}.
- Encryption: to encrypt message m $\in \mathbb{F}_2^n$, compute c $= f(\mathrm{m})$.

Fraunhofer SIT

Multivariate Cryptography

Basic Idea for Multivariate Public Key Cryptography (MPKC)

- System Parameters: $m, n, \in \mathbb{N}$.
- Key Generation: choose "random" $f \in \mathcal{MQ}(\mathbb{F}_2^n, \mathbb{F}_2^m)$
  such that f^{-1} is secretly known.
- Public Key: f.
- Private Key: f^{-1}.
- Encryption: to encrypt message $\mathrm{m} \in \mathbb{F}_2^n$,
  compute $\mathrm{c} = f(\mathrm{m})$.
- Decryption: Decrypt $\mathrm{m} = f^{-1}(\mathrm{c})$.

Fraunhofer SIT

Multivariate Cryptography

Basic Idea for Multivariate Public Key Cryptography (MPKC)

- System Parameters: $m, n, \in \mathbb{N}$.
- Key Generation: choose "random" $f \in \mathcal{MQ}(\mathbb{F}_2^n, \mathbb{F}_2^m)$
  such that f^{-1} is secretly known.
- Public Key: f.
- Private Key: f^{-1}.
- Encryption: to encrypt message $\mathrm{m} \in \mathbb{F}_2^n$,
  compute $\mathrm{c} = f(\mathrm{m})$.
- Decryption: Decrypt $\mathrm{m} = f^{-1}(\mathrm{c})$.

Problem:

How do you find f and f^{-1} such that f is a hard instance of $\mathcal{MQ}$?

Design pattern

Usually, f is constructed as a sequence of invertible functions, e.g.,

$$f = r \circ s \circ t$$

with r and t multivariate linear and
s quadratic with a easy-to-invert structure.

Multivariate Cryptography

Multivariate Public Key Cryptography (MPKC)

Design pattern

Usually, f is constructed as a sequence of invertible functions, e.g.,

$$f = r \circ s \circ t$$

with r and t multivariate linear and
s quadratic with a easy-to-invert structure.

This often does **NOT** result in a hard instance of $\mathcal{MQ}$!

Multivariate Cryptography

Multivariate Public Key Cryptography (MPKC)

Design pattern

Usually, f is constructed as a sequence of invertible functions, e.g.,

$$f = r \circ s \circ t$$

with r and t multivariate linear and
s quadratic with a easy-to-invert structure.

This often does **NOT** result in a hard instance of $\mathcal{MQ}$!

Recent secure (i.e., not yet broken?) **examples:**

- Rainbow signature scheme,
- Quartz or HFEv- signature scheme,
- PMI+ public key encryption scheme.

Fraunhofer SIT

Multivariate Cryptography

Multivariate Public Key Cryptography (MPKC)

Design pattern

Usually, f is constructed as a sequence of invertible functions, e.g.,

$$f = r \circ s \circ t$$

with r and t multivariate linear and
s quadratic with a easy-to-invert structure.

This often does **NOT** result in a hard instance of $\mathcal{MQ}$!

Recent secure (i.e., not yet broken?) **examples:**

- Rainbow signature scheme,
- Quartz or HFEv- signature scheme,
- PMI+ public key encryption scheme.

$\left.\begin{array}{l} \text{Rainbow} \\ \text{Quartz / HFEv-} \end{array}\right\}$ Easier to construct.

Multivariate Cryptography
Multivariate Public Key Cryptography (MPKC)

Further MQ schemes:

- symmetric encryption schemes,
- cryptographic hash functions, and
- pseudo random number generators.

Fraunhofer SIT

Multivariate Cryptography

Multivariate Public Key Cryptography (MPKC)

Further MQ schemes:

- symmetric encryption schemes,
- cryptographic hash functions, and
- pseudo random number generators.

Concerns about MQ schemes:

- Most public-key encryption schemes have been broken!
- Efficient (sparse) $\mathcal{MQ}$ instances have problems with randomness!

Hash-based Cryptography

Hash-based Cryptography
Introduction

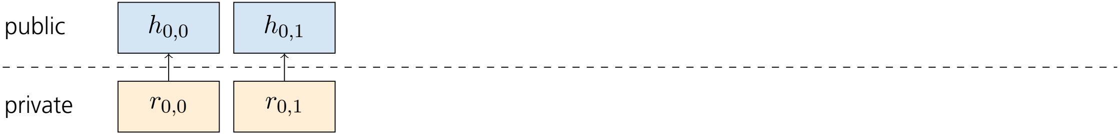
Basic idea:

Computing pre-images of a cryptographic hash function remains hard also for quantum computers (Grover).

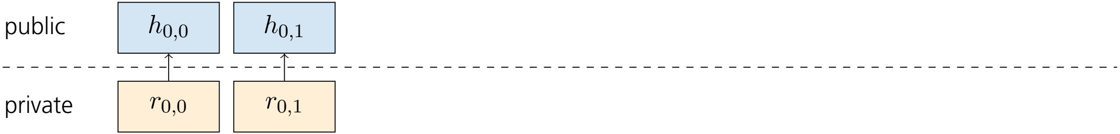
$\Rightarrow$ Use pre-image as private key, hash-value as public key.

Hash-based Cryptography

Lamport and Merkle

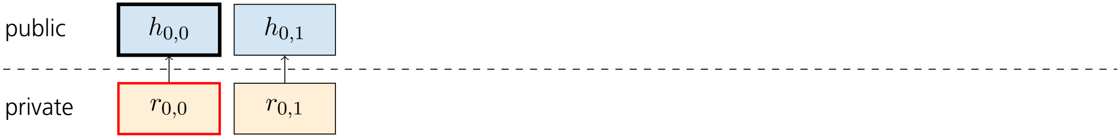


Hash-based Cryptography
Lamport and Merkle



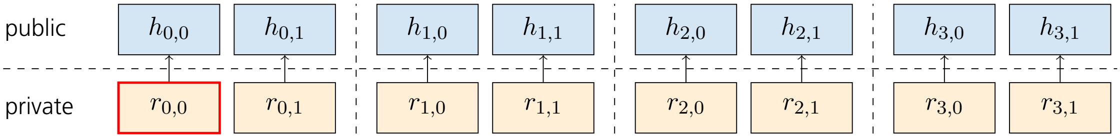
Message: 0_b

Hash-based Cryptography
Lamport and Merkle



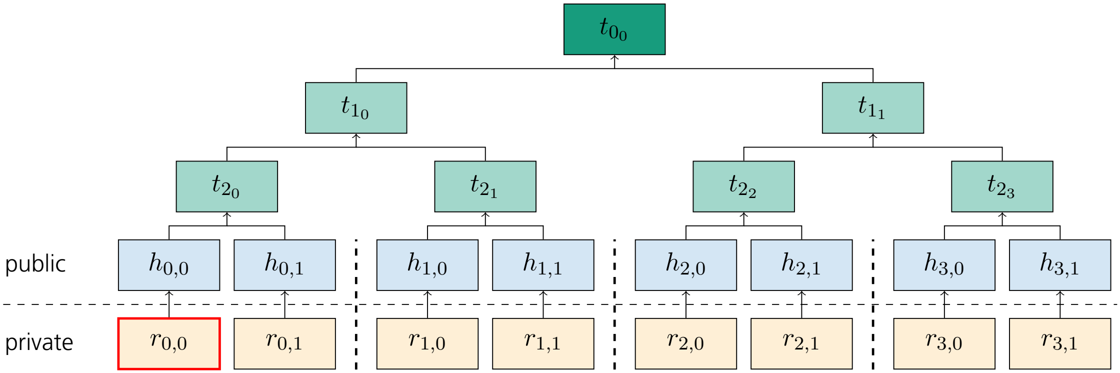
Message: 0_b

Hash-based Cryptography
Lamport and Merkle



Message: 0_b

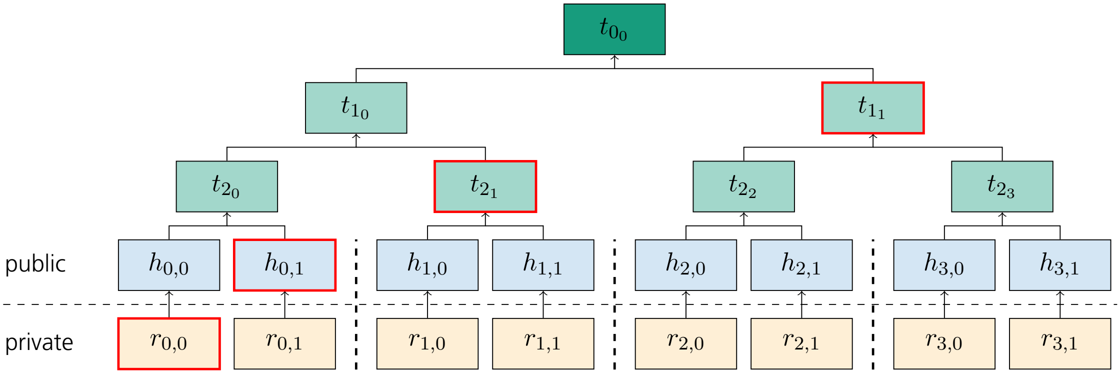
Hash-based Cryptography
Lamport and Merkle



Message: 0_b

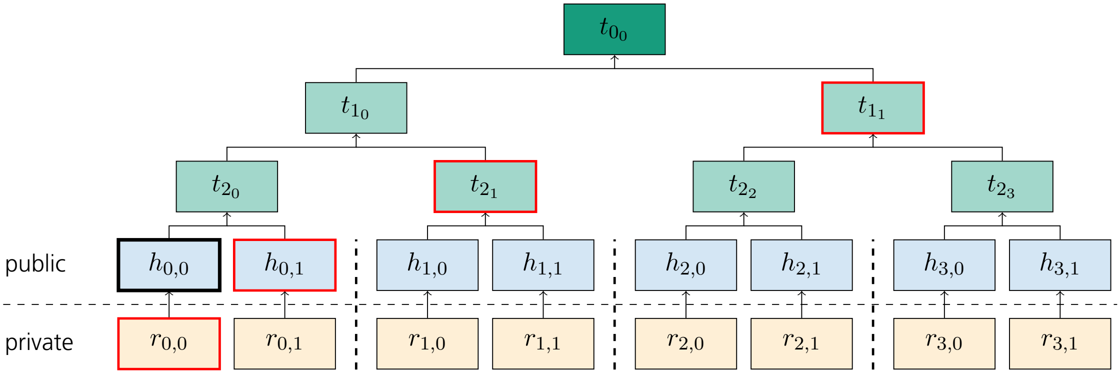
Fraunhofer SIT

Hash-based Cryptography
Lamport and Merkle



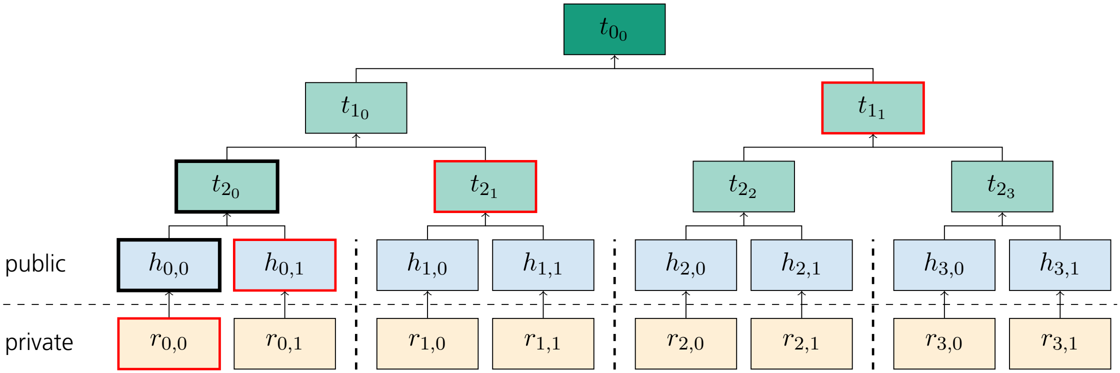
Message: 0_b

Hash-based Cryptography
Lamport and Merkle



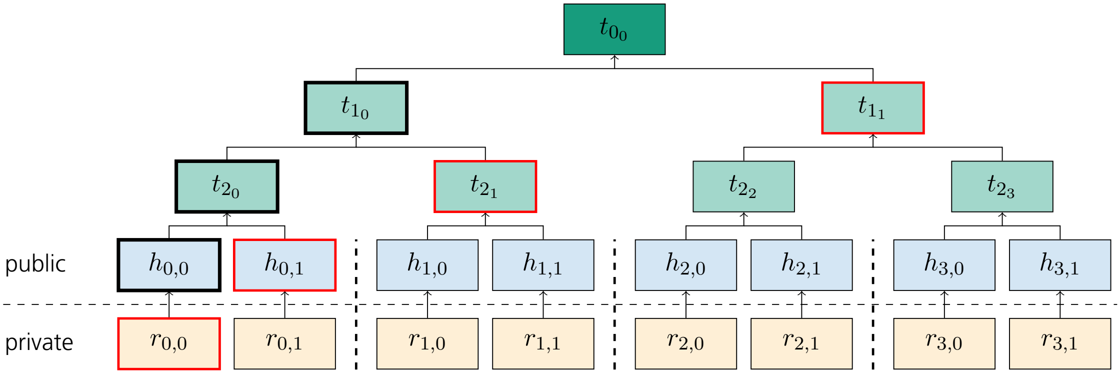
Message: 0_b

Hash-based Cryptography
Lamport and Merkle



Message: 0_b

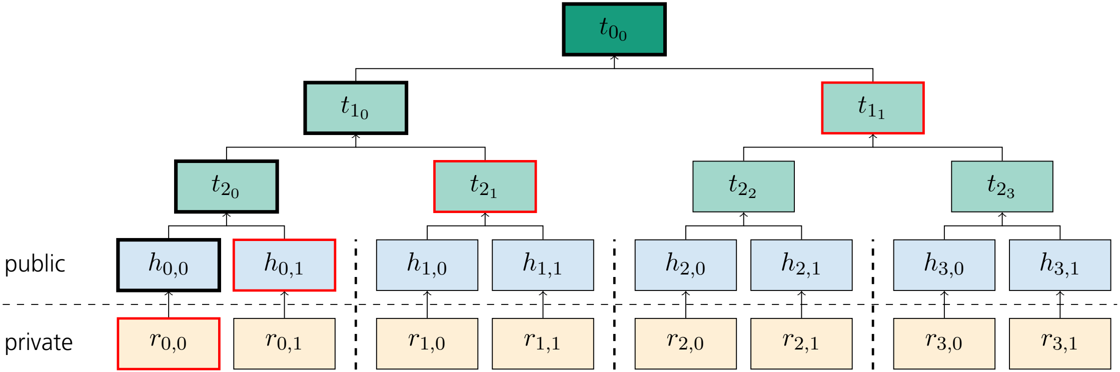
Hash-based Cryptography
Lamport and Merkle



Message: 0_b

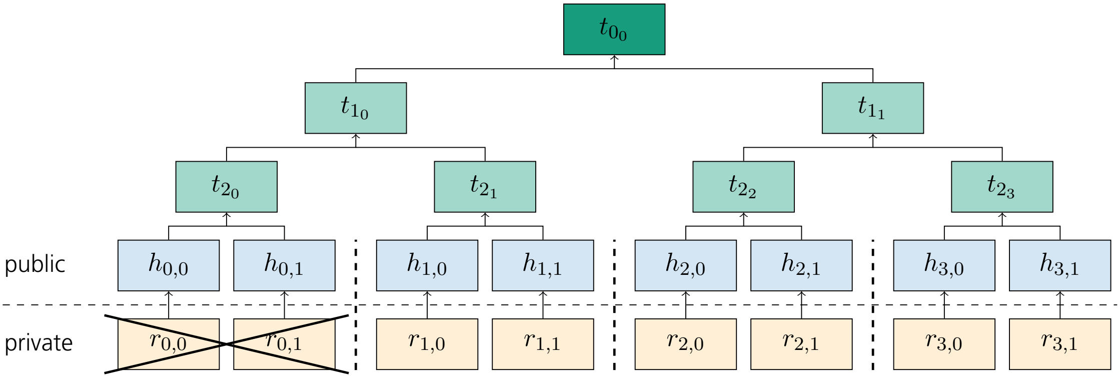
Fraunhofer SIT

Hash-based Cryptography
Lamport and Merkle

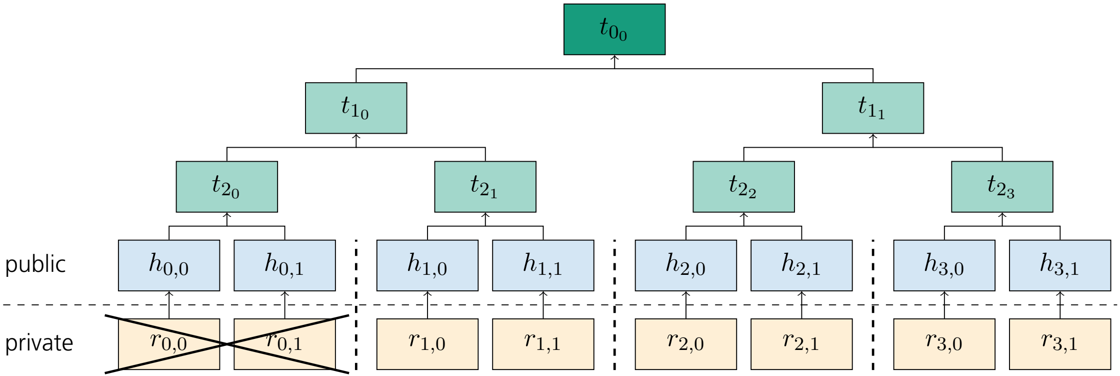


Message: 0_b

Hash-based Cryptography
Lamport and Merkle

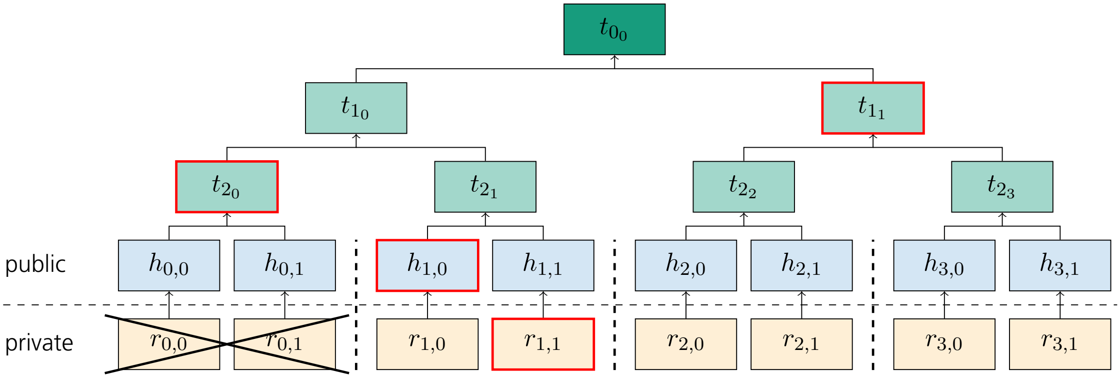


Hash-based Cryptography
Lamport and Merkle



Message: 1_b

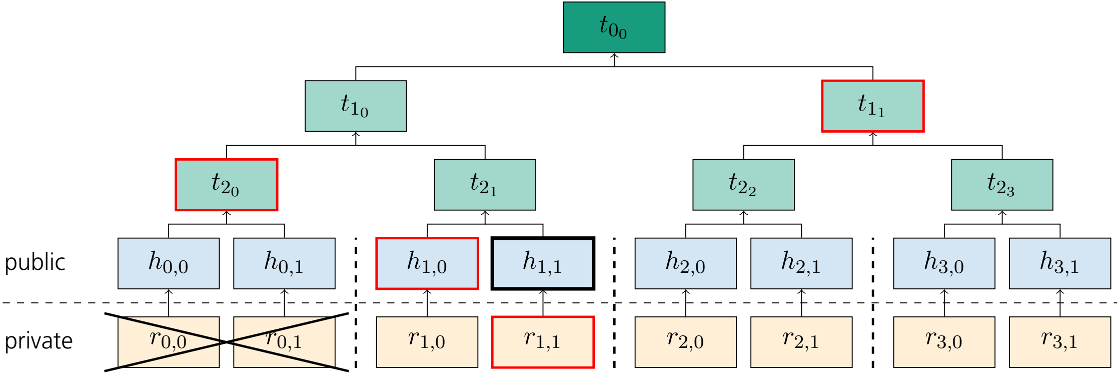
Hash-based Cryptography
Lamport and Merkle



Message: 1_b

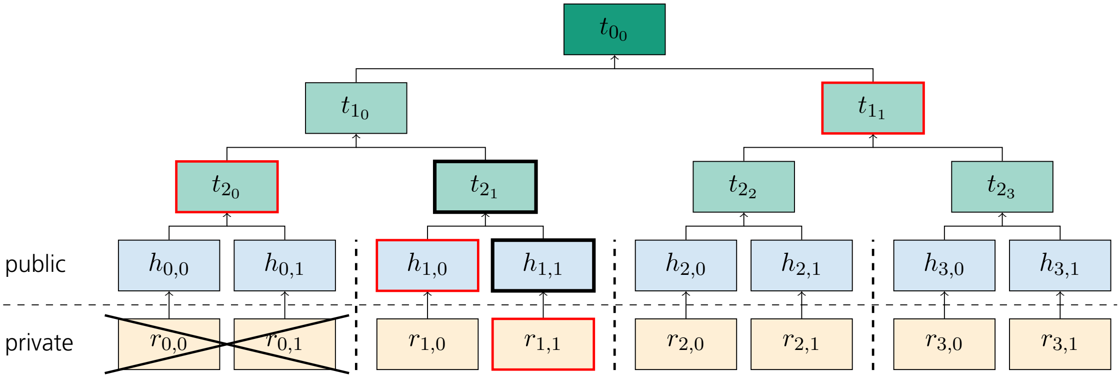
Fraunhofer SIT

Hash-based Cryptography
Lamport and Merkle



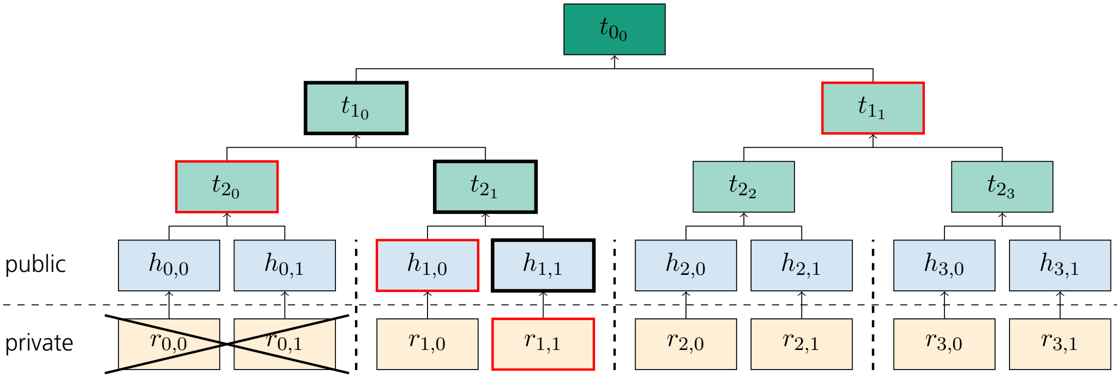
Message: 1_b

Hash-based Cryptography
Lamport and Merkle



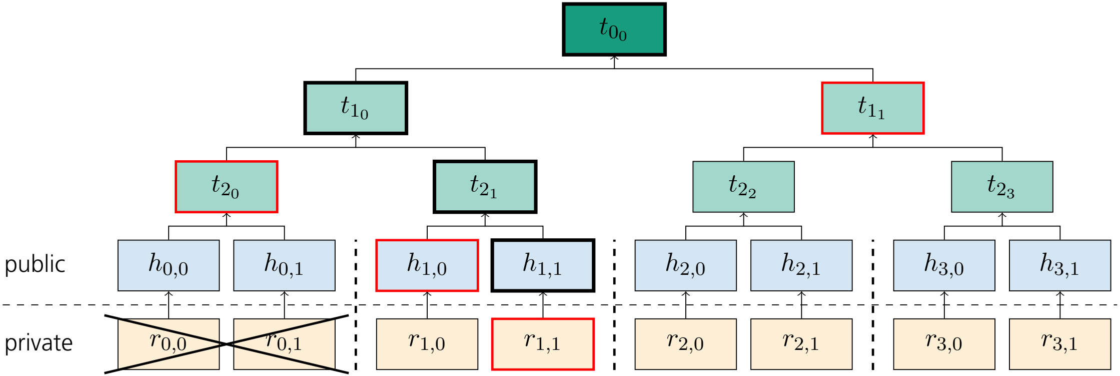
Message: 1_b

Hash-based Cryptography
Lamport and Merkle



Message: 1_b

Hash-based Cryptography
Lamport and Merkle

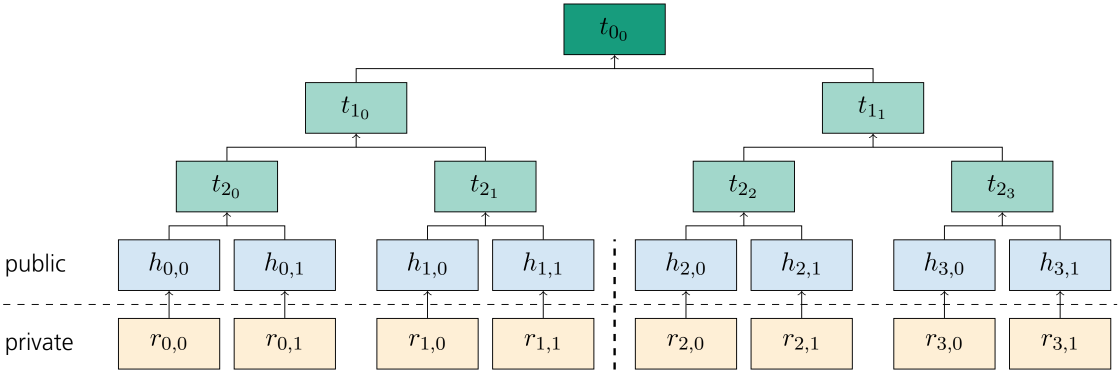


Message: 1_b

Fraunhofer SIT

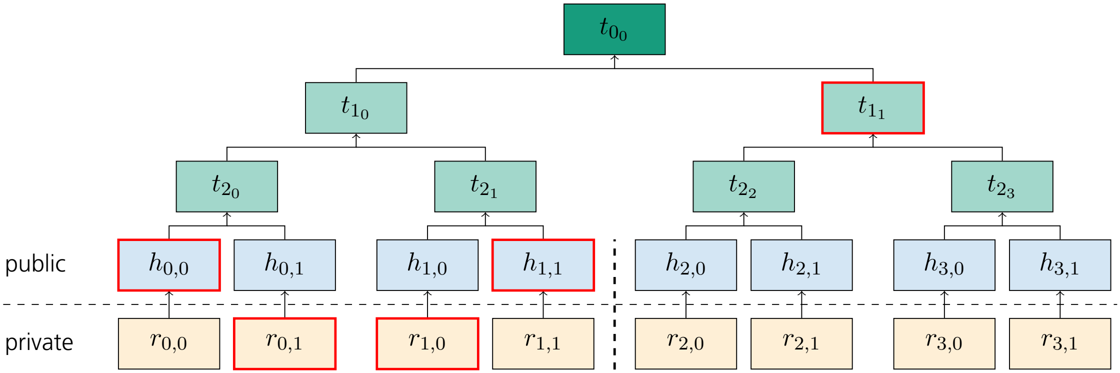
Hash-based Cryptography

Lamport and Merkle



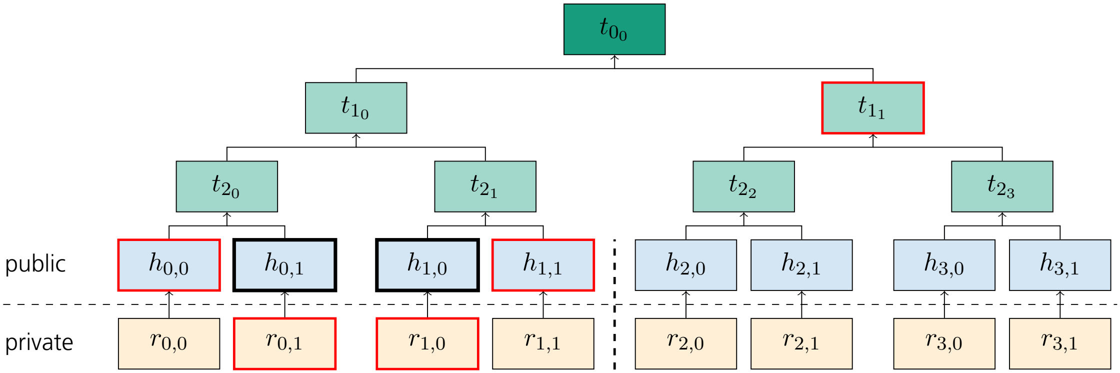
Message: 10_b

Hash-based Cryptography
Lamport and Merkle



Message: 10_b

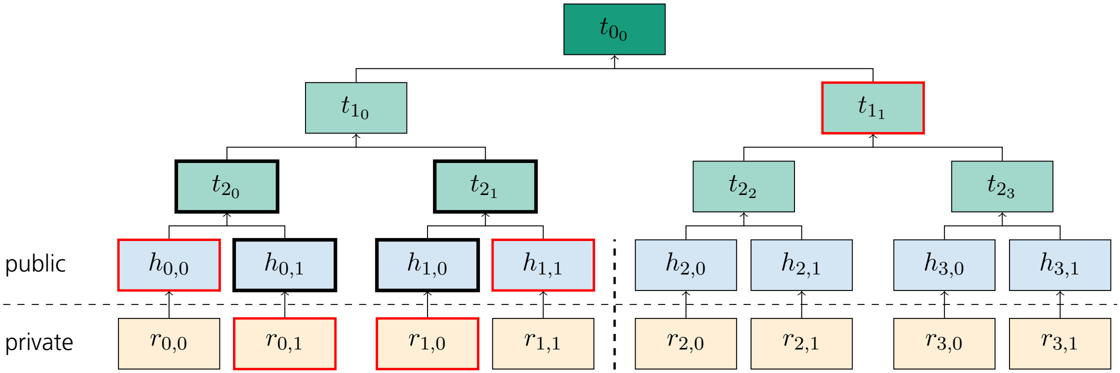
Hash-based Cryptography
Lamport and Merkle



Message: 10_b

Hash-based Cryptography

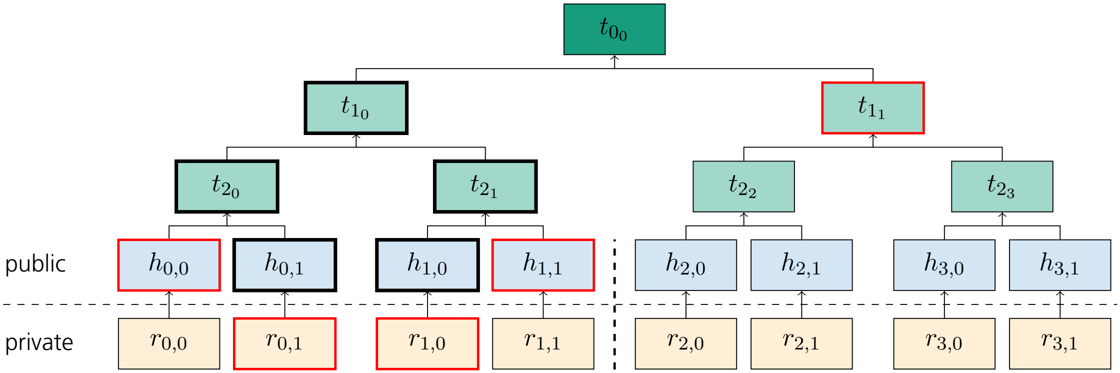
Lamport and Merkle



Message: 10_b

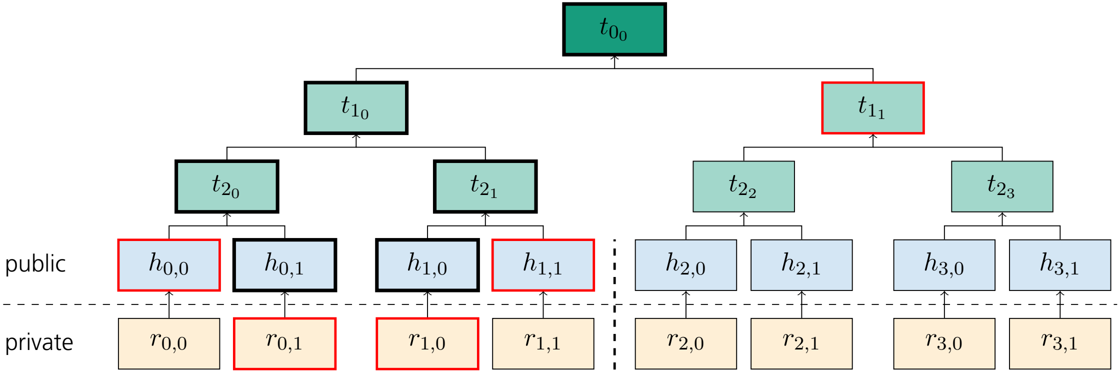
Fraunhofer SIT

Hash-based Cryptography
Lamport and Merkle



Message: 10_b

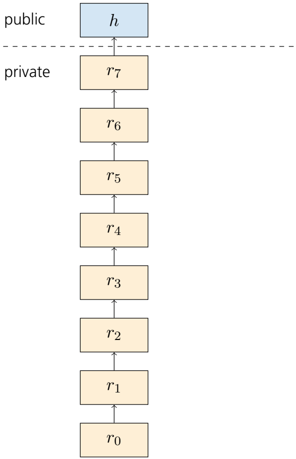
Hash-based Cryptography
Lamport and Merkle



Message: 10_b

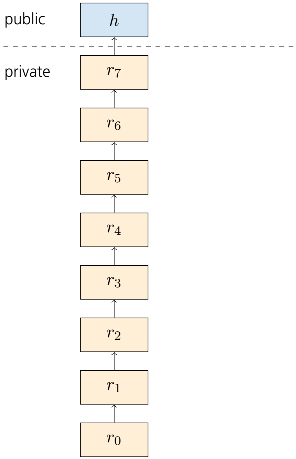
Hash-based Cryptography

(Simplified) Winternitz One-Time Scheme (WOTS)



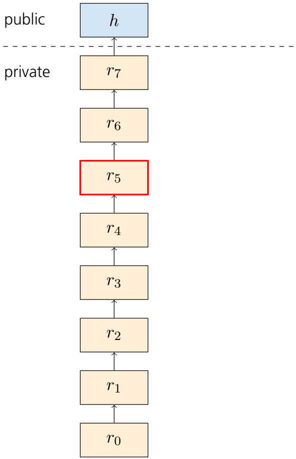
Hash-based Cryptography
(Simplified) Winternitz One-Time Scheme (WOTS)

Message: $101_b = 5$



Hash-based Cryptography

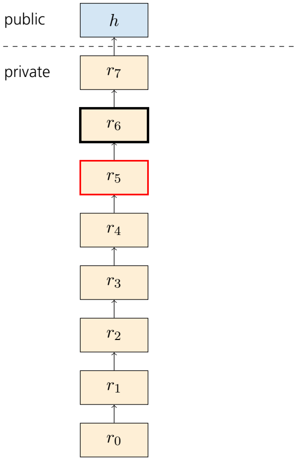
(Simplified) Winternitz One-Time Scheme (WOTS)



Message: $101_b = 5$

Hash-based Cryptography

(Simplified) Winternitz One-Time Scheme (WOTS)

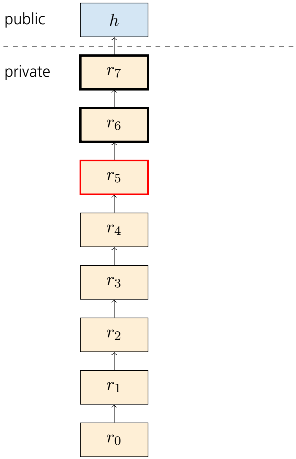


Message: $101_b = 5$

Hash-based Cryptography

(Simplified) Winternitz One-Time Scheme (WOTS)

Message: $101_b = 5$

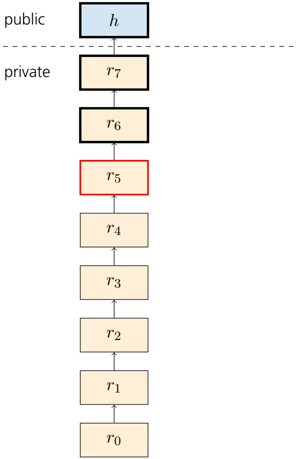


Fraunhofer SIT

Hash-based Cryptography

(Simplified) Winternitz One-Time Scheme (WOTS)

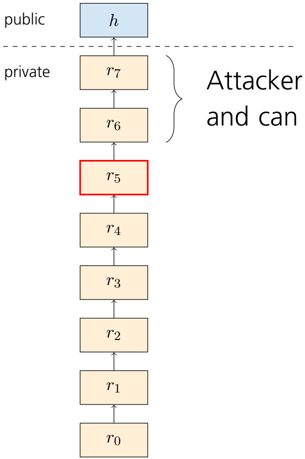
Message: $101_b = 5$



Hash-based Cryptography

(Simplified) Winternitz One-Time Scheme (WOTS)

Message: $101_b = 5$

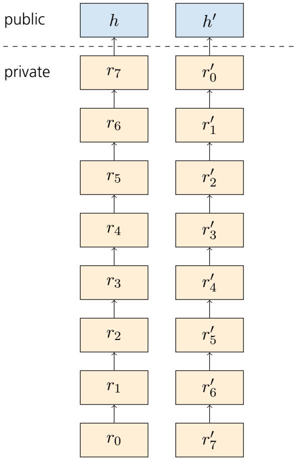


Attacker learns private keys and can sign 110_b and 111_b!

Hash-based Cryptography

(Simplified) Winternitz One-Time Scheme (WOTS)

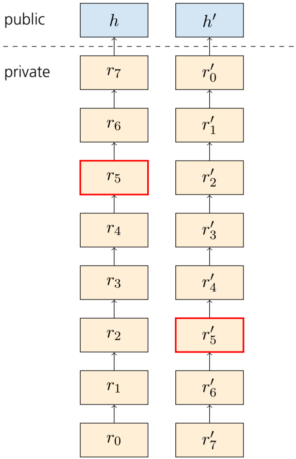
Message: $101_b = 5$



Hash-based Cryptography

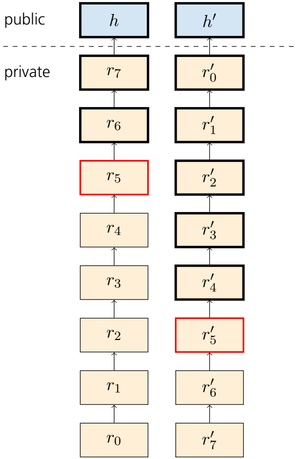
(Simplified) Winternitz One-Time Scheme (WOTS)

Message: $101_b = 5$



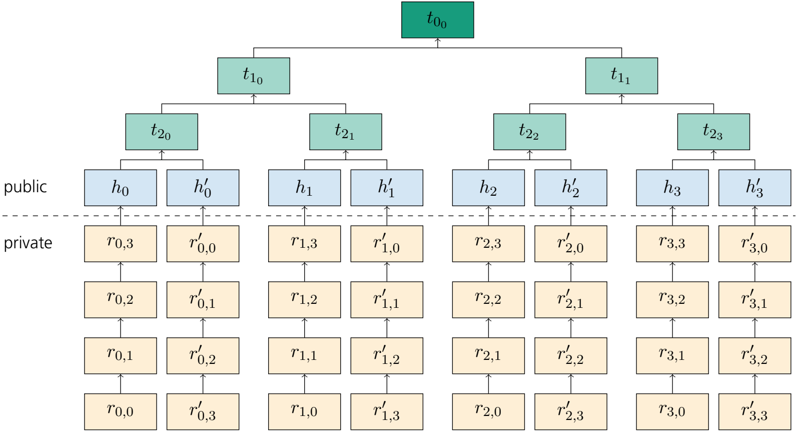
Hash-based Cryptography
(Simplified) Winternitz One-Time Scheme (WOTS)

Message: $101_b = 5$



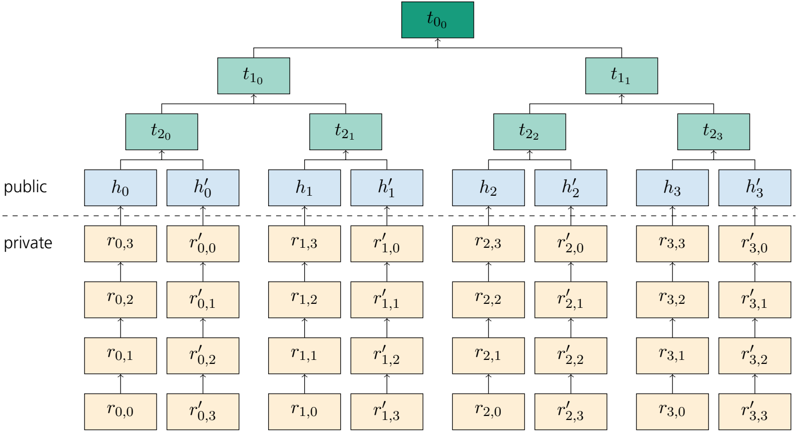
Hash-based Cryptography

(Simplified) Winternitz and Merkle Tree



Hash-based Cryptography

(Simplified) Winternitz and Merkle Tree

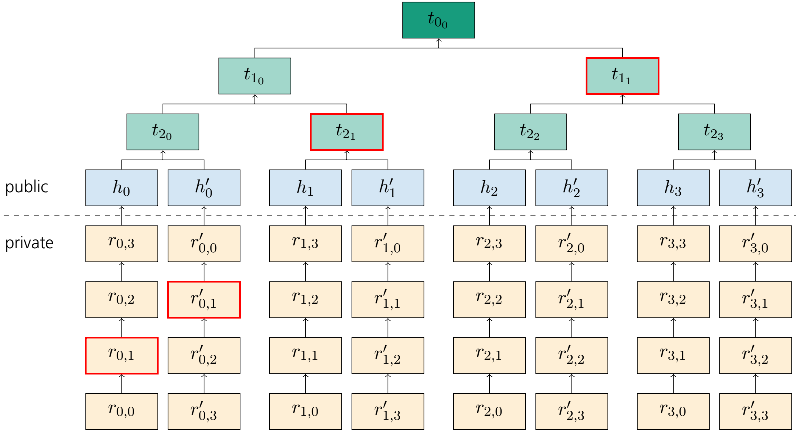


Message: $01_b = 1$

Fraunhofer SIT

Hash-based Cryptography

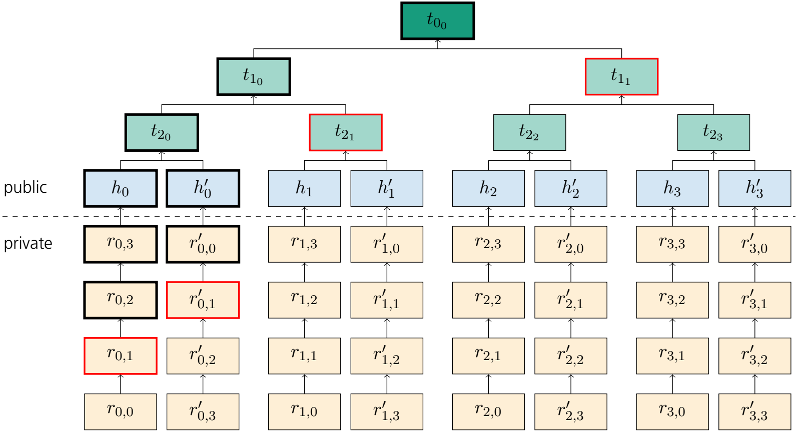
(Simplified) Winternitz and Merkle Tree



Message: $01_b = 1$

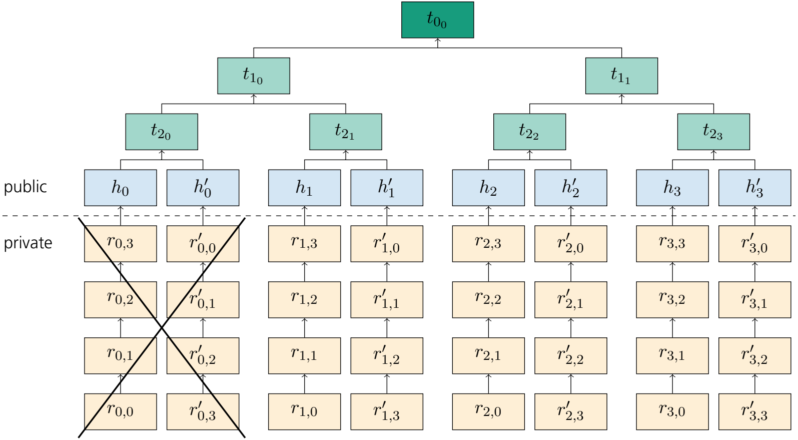
Hash-based Cryptography

(Simplified) Winternitz and Merkle Tree



Message: $01_b = 1$

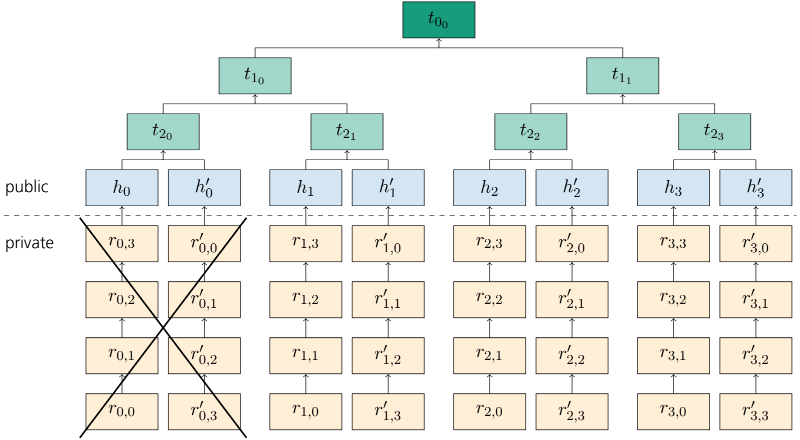
Hash-based Cryptography
(Simplified) Winternitz and Merkle Tree



Fraunhofer SIT

Hash-based Cryptography

(Simplified) Winternitz and Merkle Tree

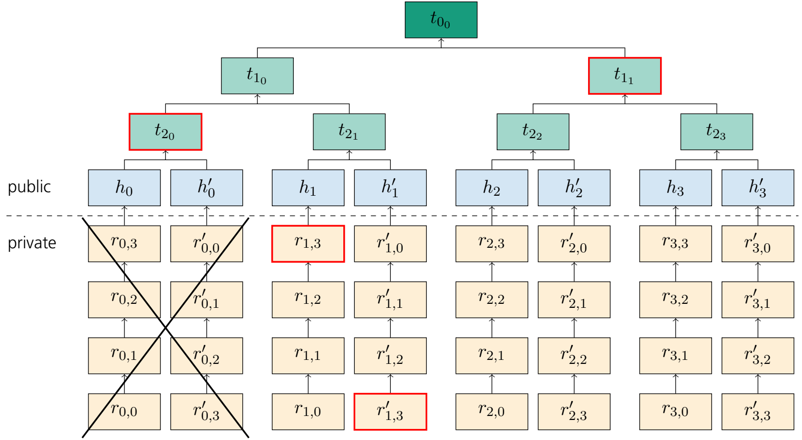


Message: $11_b = 3$

Fraunhofer SIT

Hash-based Cryptography

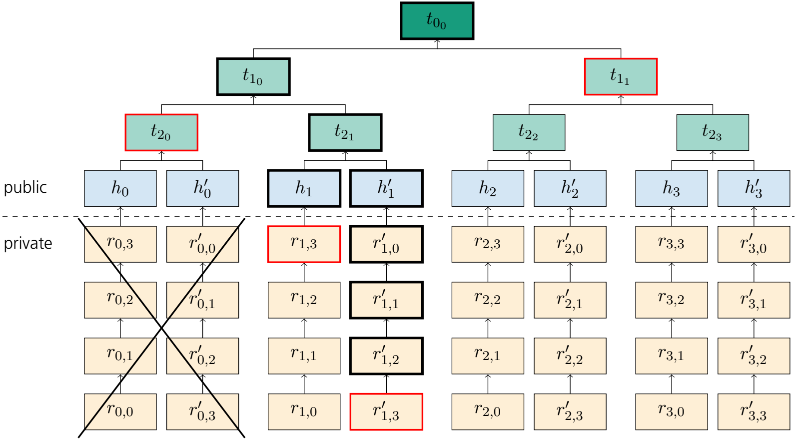
(Simplified) Winternitz and Merkle Tree



Message: $11_b = 3$

Hash-based Cryptography

(Simplified) Winternitz and Merkle Tree



Message: $11_b = 3$

Fraunhofer SIT

Hash-based Cryptography

Summary:

- Only helpful for Signatures.

Fraunhofer
SIT

Hash-based Cryptography

Summary:

- Only helpful for Signatures.
- Number of signatures per public key is limited.

Hash-based Cryptography

Summary:

- Only helpful for Signatures.
- Number of signatures per public key is limited.
- Tree structures allow to sign many messages, e.g., XMSS.

Hash-based Cryptography

Summary:

- Only helpful for Signatures.
- Number of signatures per public key is limited.
- Tree structures allow to sign many messages, e.g., XMSS.
- There are sate free schemes, e.g., SPHINCS.

Hash-based Cryptography

Summary:

- Only helpful for Signatures.
- Number of signatures per public key is limited.
- Tree structures allow to sign many messages, e.g., XMSS.
- There are sate free schemes, e.g., SPHINCS.
- Key generation is expensive.

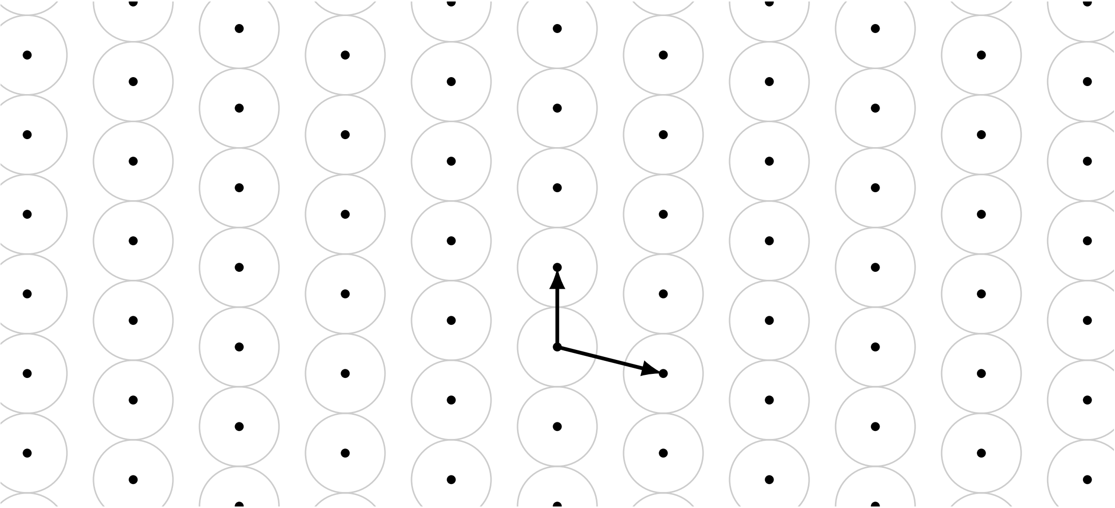
Hash-based Cryptography

Summary:

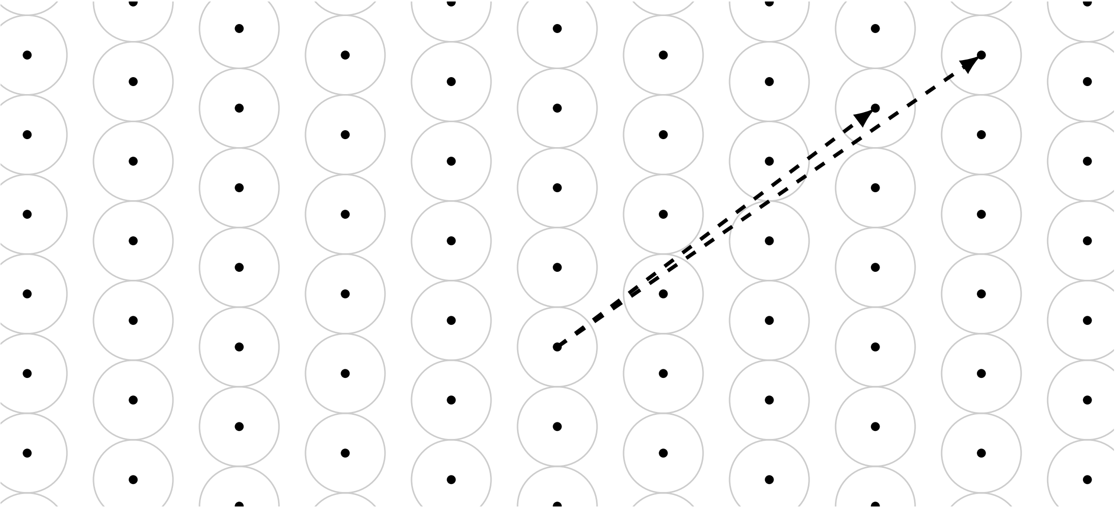
- Only helpful for Signatures.
- Number of signatures per public key is limited.
- Tree structures allow to sign many messages, e.g., XMSS.
- There are sate free schemes, e.g., SPHINCS.
- Key generation is expensive.
- Signatures are relatively large.

Lattice-based Cryptography

Lattice-based Cryptography
Introduction

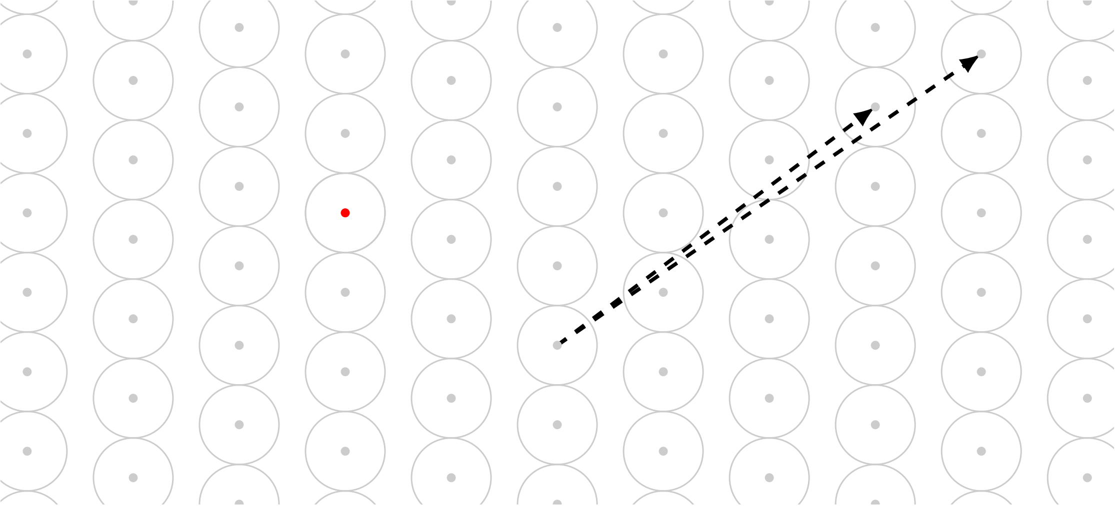


Lattice-based Cryptography
Introduction



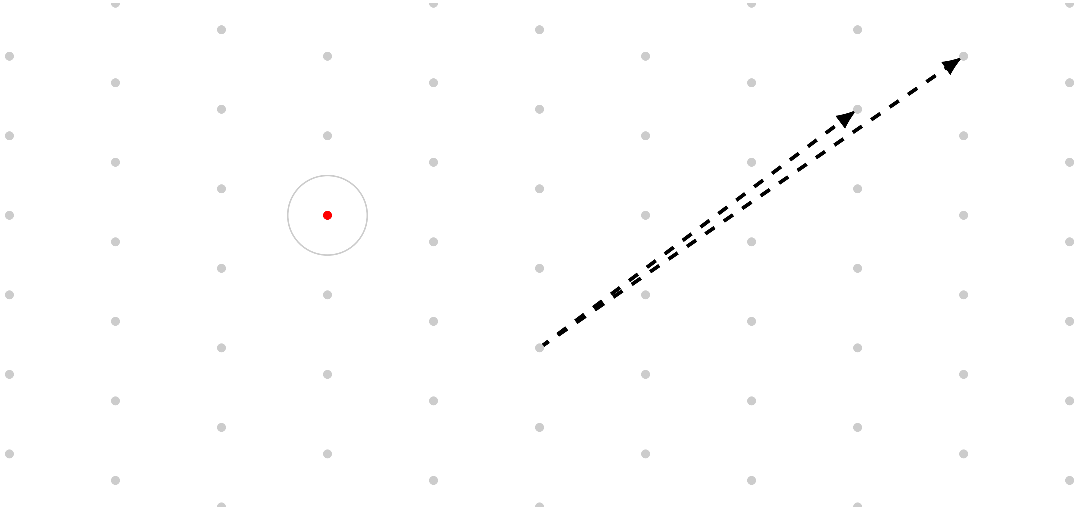
Fraunhofer SIT

Lattice-based Cryptography
Introduction



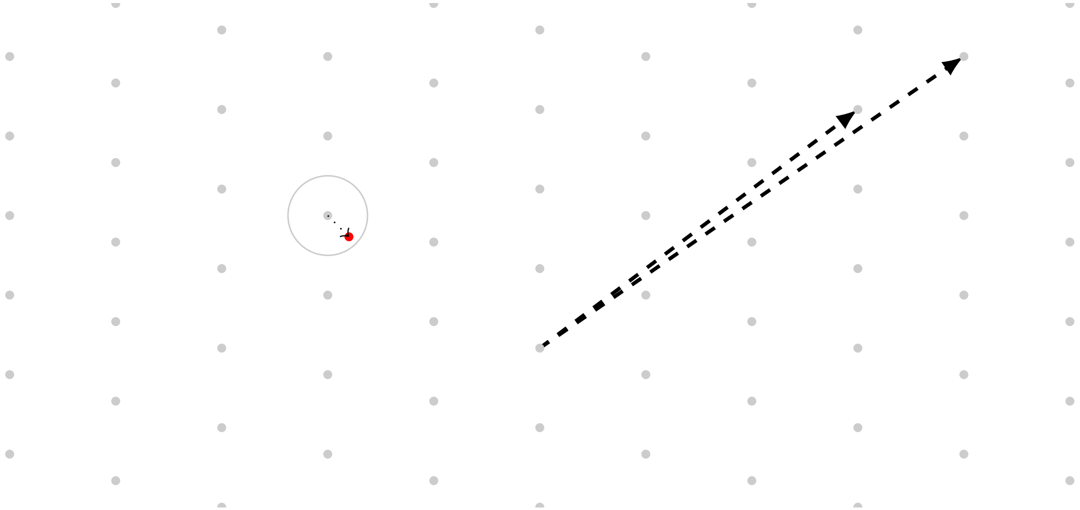
Fraunhofer SIT

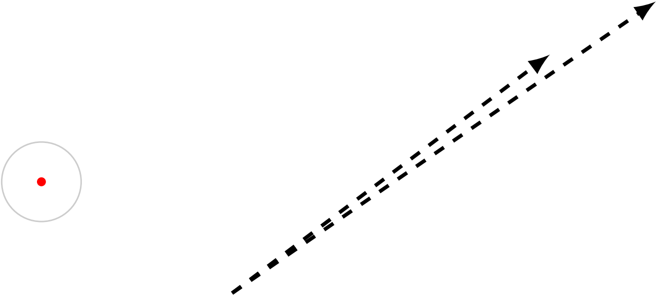
Lattice-based Cryptography
Introduction



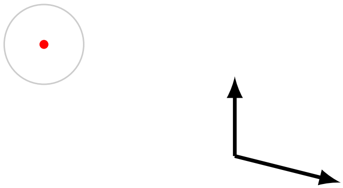
Fraunhofer SIT

Lattice-based Cryptography
Introduction

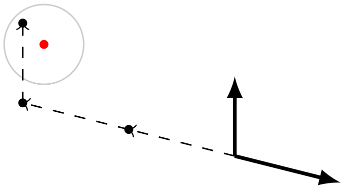




Lattice-based Cryptography
Introduction



Lattice-based Cryptography
Introduction



Fraunhofer SIT

Underlying hard problems:

- CVP: closest vector problem,
- SVP: shortest vector problem,
- LWE: learning with errors.

Lattice-based Cryptography
Introduction

Underlying hard problems:

- CVP: closest vector problem,
- SVP: shortest vector problem,
- LWE: learning with errors.

Popular lattice-based schemes:

- public key encryption: NTRU, NTRU prime;
- key exchange: New Hope (experimentally used by Google).

Security proofs of lattice-based schemes:

- There are security proofs and
  worst-case to average-case reductions.

Security proofs of lattice-based schemes:

- There are security proofs and worst-case to average-case reductions.
- Security proofs are not tight:
  Security parameters are chosen based on *best-known* attacks, not based on security proofs.

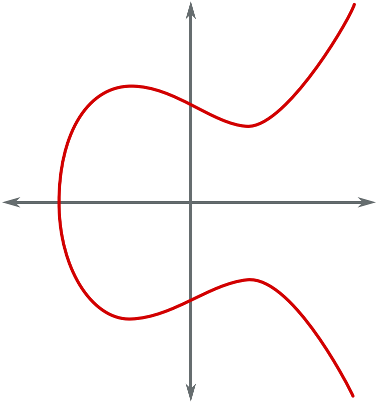
Security proofs of lattice-based schemes:

- There are security proofs and worst-case to average-case reductions.
- Security proofs are not tight: Security parameters are chosen based on *best-known* attacks, not based on security proofs.

Problems with lattice-based schemes:

- Attack-complexity not yet deeply understood,
- attacks are improved frequently.

Supersingular Isogenies



$$y^2 = x^3 - x + 1$$

Supersingular Isogenies

Overview

$$\begin{tikzcd} & E_A \arrow[dr, "\psi'(E_A)"] & \\ E \arrow[ur, "\phi(E)"] \arrow[dr, "\psi(E)"'] & & E_{AB} \\ & E_B \arrow[ur, "\phi'(E_B)"'] & \end{tikzcd}$$

Basic idea:

- Use secret mappings (isogenies) between elliptic curves to compute a shared secret.
- Does not operate on points of a curve but on curves using maps.

Supersingular Isogenies
Overview

$$\begin{tikzcd}
& E_A \arrow[dr, "\psi'(E_A)"] & \\
E \arrow[ur, "\phi(E)"] \arrow[dr, "\psi(E)"'] & & E_{AB} \\
& E_B \arrow[ur, "\phi'(E_B)"'] &
\end{tikzcd}$$

Features:

- **DH-like PQ key exchange scheme.**
- \+ Small communication overhead.
- – High computational cost.

Fraunhofer SIT

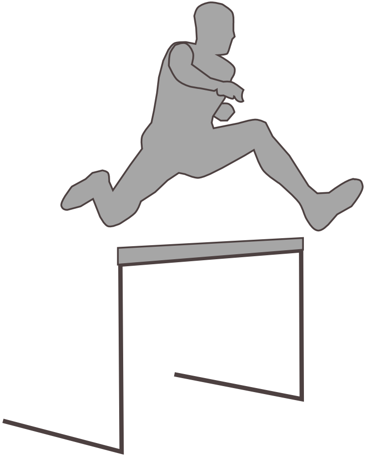
Supersingular Isogenies
Overview

$$
\begin{tikzcd}
 & E_A \arrow[dr, "\psi'(E_A)"] & \\
E \arrow[ur, "\phi(E)"] \arrow[dr, "\psi(E)"'] & & E_{AB} \\
 & E_B \arrow[ur, "\phi'(E_B)"'] &
\end{tikzcd}
$$

Problems:

- Very recent proposal; security not yet well understood.
- First proposal with *ordinary* curves broken by quantum computers.
- New proposal using *supersingular* curves under examination.

Performance and Challenges



Performance and Challenges
Recommendations

Initial recommendations from the “PQCRYPTO project” (2015) [1]:

- Symmetric Encryption:
  - AES-256,
  - Salsa20 with 256-bit key.

Performance and Challenges
Recommendations

Initial recommendations from the “PQCRYPTO project” (2015) [1]:

- Symmetric Encryption:
  - AES-256,
  - Salsa20 with 256-bit key.
- Public-key Encryption:
  - McEliece with binary Goppa codes using length $n = 6960$, dimension $k = 5413$, and adding $t = 119$ errors.

Performance and Challenges
Recommendations

Initial recommendations from the "PQCRYPTO project" (2015) [1]:

- Symmetric Encryption:
  - AES-256,
  - Salsa20 with 256-bit key.
- Public-key Encryption:
  - McEliece with binary Goppa codes using length $n = 6960$, dimension $k = 5413$, and adding $t = 119$ errors.
- Public-key Signatures:
  - XMSS (with state),
  - SPHINCS-256 (stateless).

Fraunhofer SIT

Performance and Challenges

NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016

Fraunhofer SIT

Performance and Challenges
NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016
April 2016	NIST releases NISTIR 8105 — Report on Post-Quantum Cryptography

Performance and Challenges

NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016
April 2016	NIST releases NISTIR 8105 — Report on Post-Quantum Cryptography
Dec. 2016	Formal Call for Proposals

Performance and Challenges
NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016
April 2016	NIST releases NISTIR 8105 — Report on Post-Quantum Cryptography
Dec. 2016	Formal Call for Proposals
Nov. 2017	Deadline for submissions

Performance and Challenges
NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016
April 2016	NIST releases NISTIR 8105 — Report on Post-Quantum Cryptography
Dec. 2016	Formal Call for Proposals
Nov. 2017	Deadline for submissions
Early 2018	Workshop — Submitter's Presentations

Performance and Challenges
NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016
April 2016	NIST releases NISTIR 8105 — Report on Post-Quantum Cryptography
Dec. 2016	Formal Call for Proposals
Nov. 2017	Deadline for submissions
Early 2018	Workshop — Submitter's Presentations
3-5 years	Analysis Phase — NIST will report findings
	1-2 workshops during this phase

Fraunhofer SIT

Performance and Challenges
NIST Post-Quantum Cryptography Standardization

Time line:

Feb. 2016	Announcement at PQCrypto 2016
April 2016	NIST releases NISTIR 8105 — Report on Post-Quantum Cryptography
Dec. 2016	Formal Call for Proposals
Nov. 2017	Deadline for submissions
Early 2018	Workshop — Submitter's Presentations
3-5 years	Analysis Phase — NIST will report findings
	1-2 workshops during this phase
2 years later	Draft Standards ready

Performance and Challenges

NIST Post-Quantum Cryptography Standardization

Round 1 Submissions:

Family	Signatures	KEM/Encryption	sum
lattice-based	5	23	28
code-based	3	17	20
multivariate	7	3	10
hash-based	2		2
"others"	3	6	9
sum	20	49	69

Performance and Challenges
Cost of PQ Schemes

Scheme	Public key size (bytes)	Data size (bytes)
Classical schemes:		
• RSA:		
– RSA-2048	256	256
– RSA-4096	512	512
• ECC:		
– 256-bit	32	32
– 512-bit	64	64
• Key exchange:		
– DH	—	256 – 512
– ECDH	—	32 – 64

Fraunhofer
SIT

Performance and Challenges
Cost of PQ Schemes

Scheme	Public key size (bytes)	Data size (bytes)
Public-key signatures:		
• Hash based:		
– XMSS (stateful)	64	2,500 – 2,820
– SPHINCS (state free)	1,056	41,000
• Multivariate based:		
– HFEv-	500,000 – 1,000,000	25 – 32
– Rainbow	148,500 – 1,321,000	64 – 147
• Lattice based:		
– Dilithium	896 – 1760	1386 – 3365
– qTESLA	2,976 – 6,432	2,720 – 5,920

Performance and Challenges
Cost of PQ Schemes

Scheme	Public key size (bytes)	Data size (bytes)
Public-key encryption:		
• Code based:		
- McEliece (binary Goppa codes)	958,482 – 1,046,739	187 – 194
- McEliece (QC-MDPC codes)	4,097	8,226
• Lattice based:		
– NTRUEncrypt	1,023 – 4,097	1023 – 4,097
– Kyber (KEM)	1,088	1,184

Fraunhofer SIT

Performance and Challenges

Cost of PQ Schemes

Scheme	Public key size (bytes)	Data size (bytes)
Key exchange:		
• Lattice based:		
– NewHope	—	1,824 – 2,048
– Kyber (KEX)	—	1,184 – 2,368
• Supersingular isogenies:		
– SIDH	—	564

Fraunhofer SIT

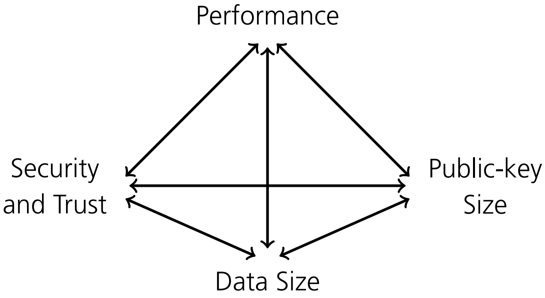
Performance and Challenges
Relative Performance

Family	Key Generation	Public Key Encryption/Verification	Private Key Decryption/Signing
Code based:	slow	fast	medium
Multivariate:	slow	fast	medium
Hash based:	slow	fast	slow
Lattice based:	fast	fast	fast
Isogenies:	————	slow (key exchange)	————
ECC-256	fast	medium	fast
RSA-3072	slow	fast	slow

Performance and Challenges
Challenges

Open research questions:

- Make trusted schemes more efficient.
- Make efficient schemes more reliable.



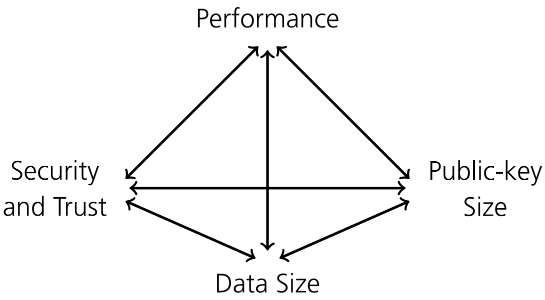
Performance and Challenges
Challenges

Open research questions:

- Make trusted schemes more efficient.
- Make efficient schemes more reliable.

Real-world PQC:

- Investigate the usability of PQC schemes in real-world applications.
- Prepare applications for the transition to PQC. $\Rightarrow$ crypto-agility



Thank you for your attention!

Literature

D. Augot, L. Batina, D. J. Bernstein, J. Bos, J. Buchmann, W. Castryck, O. Dunkelman, T. Güneysu, S. Gueron, A. Hülsing, T. Lange, M. S. E. Mohamed, C. Rechberger, P. Schwabe, N. Sendrier, F. Vercauteren, and B.-Y. Yang. *Initial recommendations of long-term secure post-quantum systems*. Tech. rep. `http://pqcrypto.eu.org/docs/initial-recommendations.pdf`. PQCRYPTO Horizon 2020 ICT-645622, Sept. 2015.

D. J. Bernstein, J. Buchmann, and E. Dahmen, eds. *Post Quantum Cryptography*. Springer, 2008.

D. Deutsch. "Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer". In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 400.1818 (1985), pp. 97–117.

Image Credits

Title page:	by IBM Research, CC BY-ND 2.0
Telegraph:	CC0 Creative Commons
Hash browns:	by Crisco 1492, CC BY-SA 3.0
Lettuce:	CC0 Creative Commons
Elliptic curve:	by Yassine Mrabet, CC BY-SA 3.0
Hurdle:	CC0 Creative Commons

Contact Information

Dr. Ruben Niederhagen

Cyber-Physical System Security

Fraunhofer-Institute for
Secure Information Technology

Address: Rheinstraße 75
64295 Darmstadt
Germany
Internet: http://www.sit.fraunhofer.de

Phone: +49 6151 869-135
Fax: +49 6151 869-224
E-Mail: ruben.niederhagen@sit.fraunhofer.de