# Post-Quantum Cryptography

Johannes Buchmann and Nina Bindel

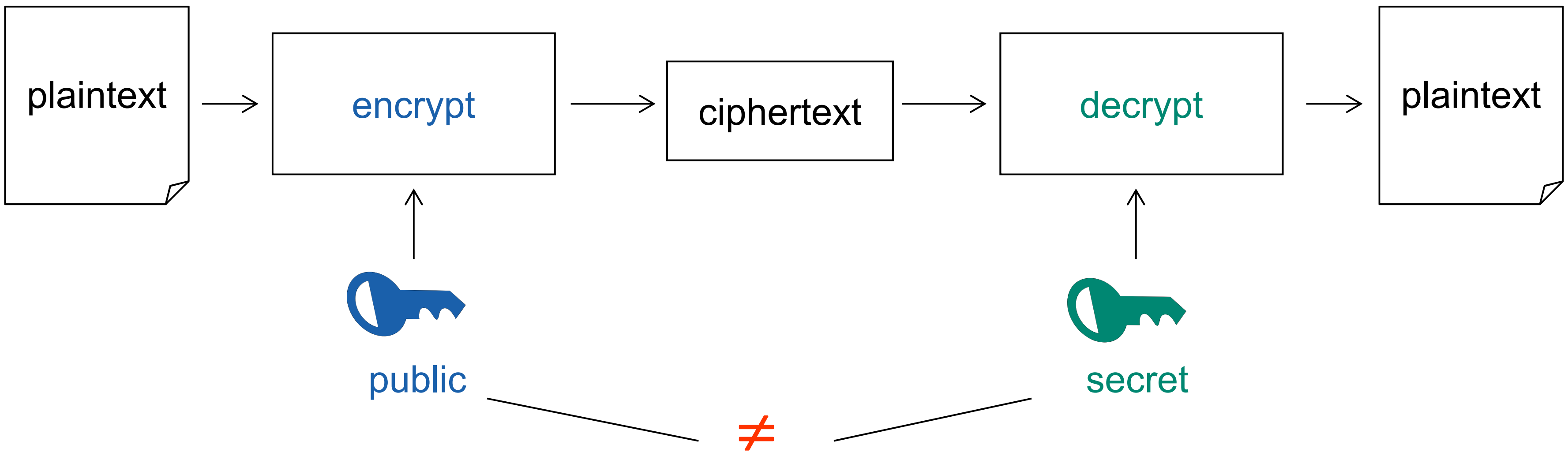
TECHNISCHE UNIVERSITÄT DARMSTADT



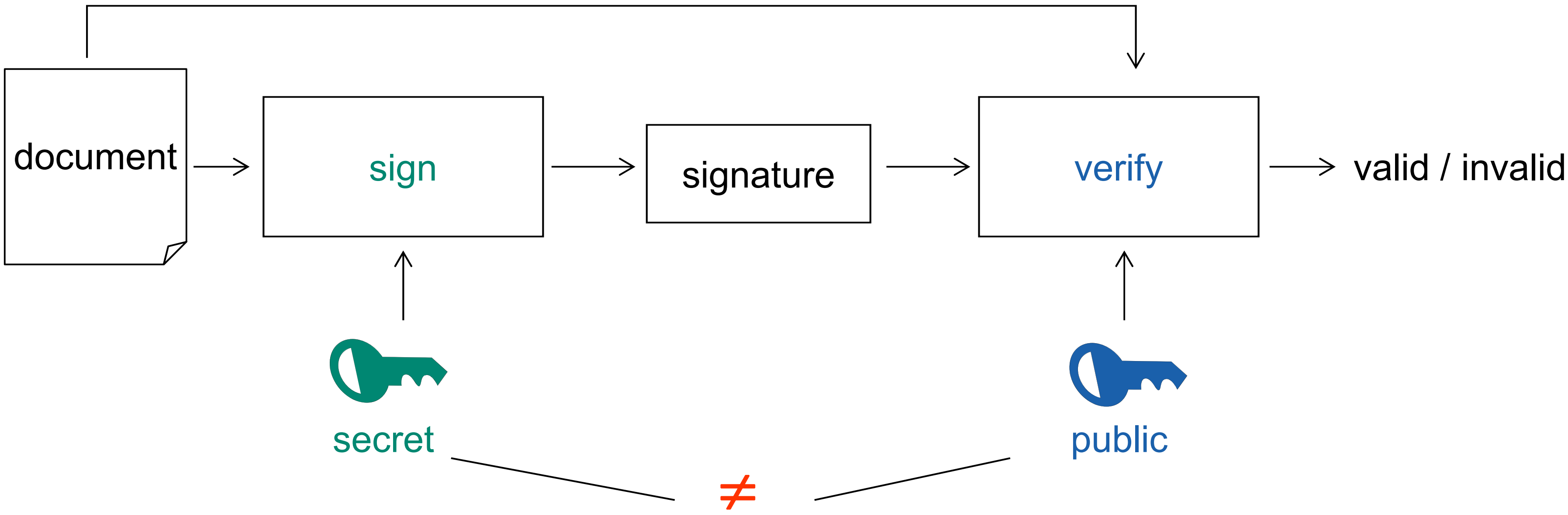
CROSSING

# Public-key cryptography

# Public-key encryption



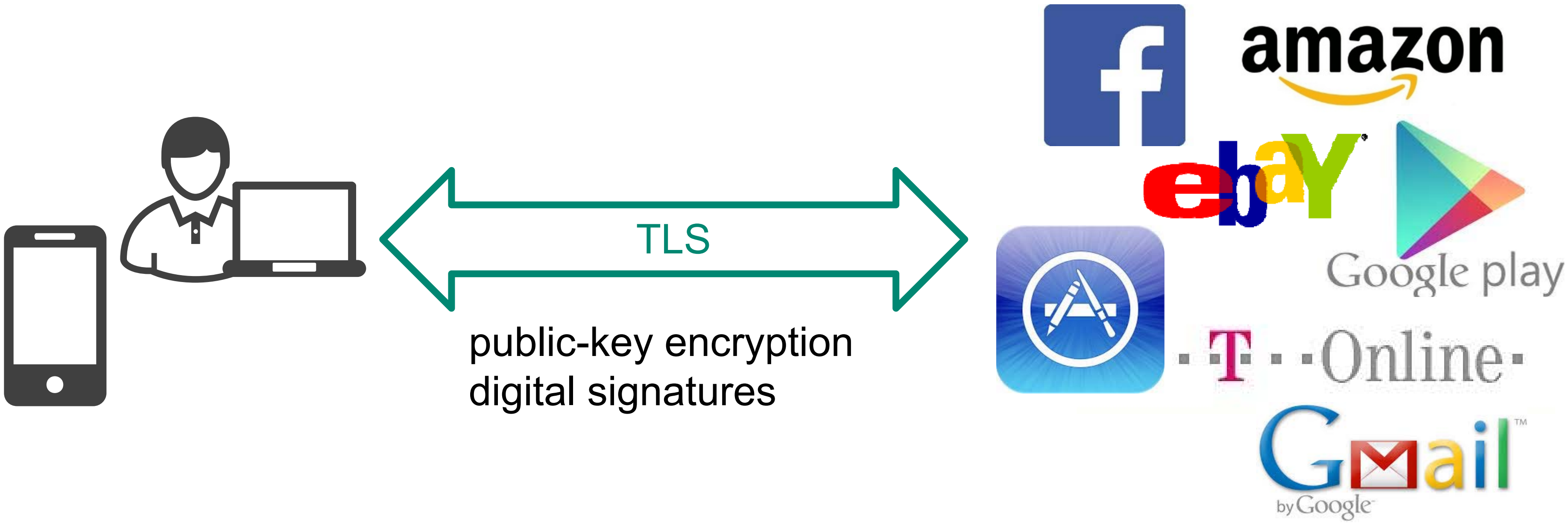
# Digital signatures



# IT-security requires public-key cryptography

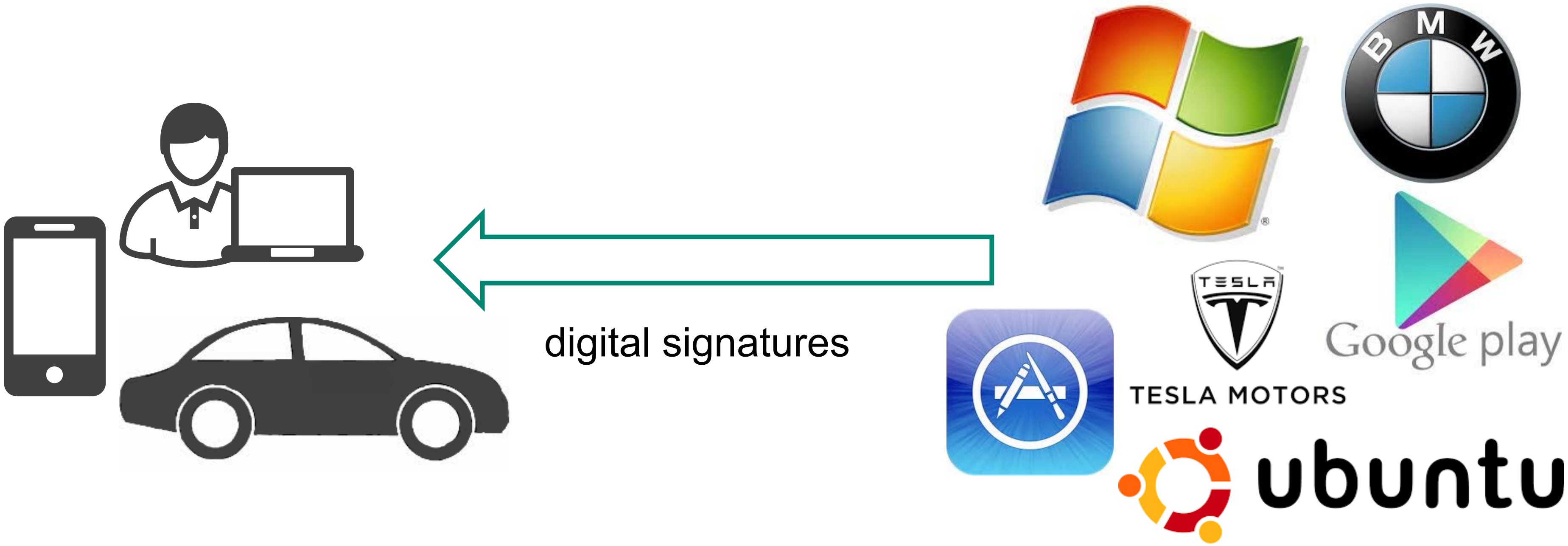
# TLS



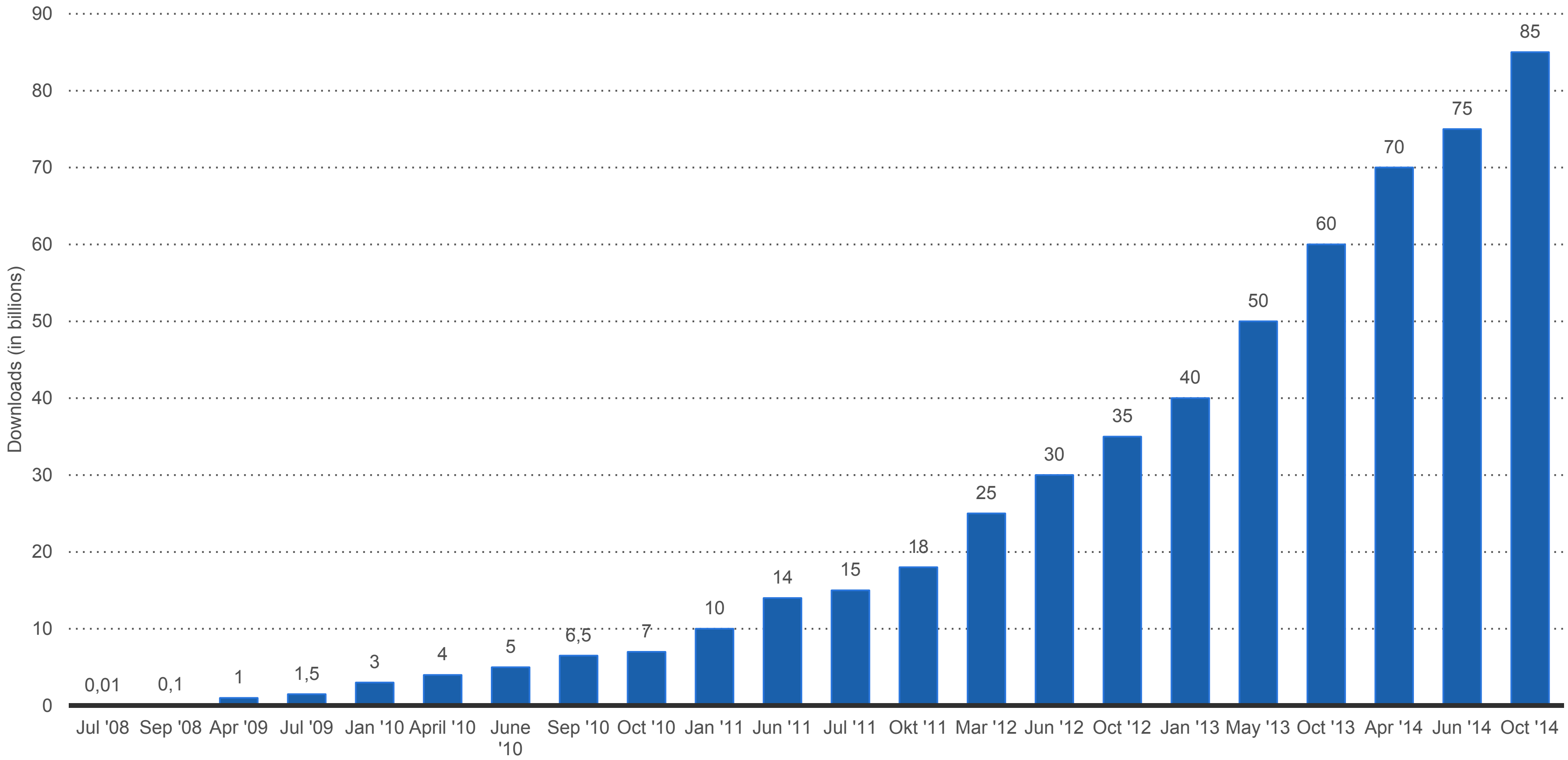



**Billions daily!**

# Software downloads



# Number of worldwide downloads from Apple App Store July 2008 - October 2014 (in billions)




Source: Apple

# Current public-key cryptography

# "Generic" RSA

Public key: finite Group $G$, exponent $e$, $\gcd(e, |G|) = 1$

Secret key: $|G|$

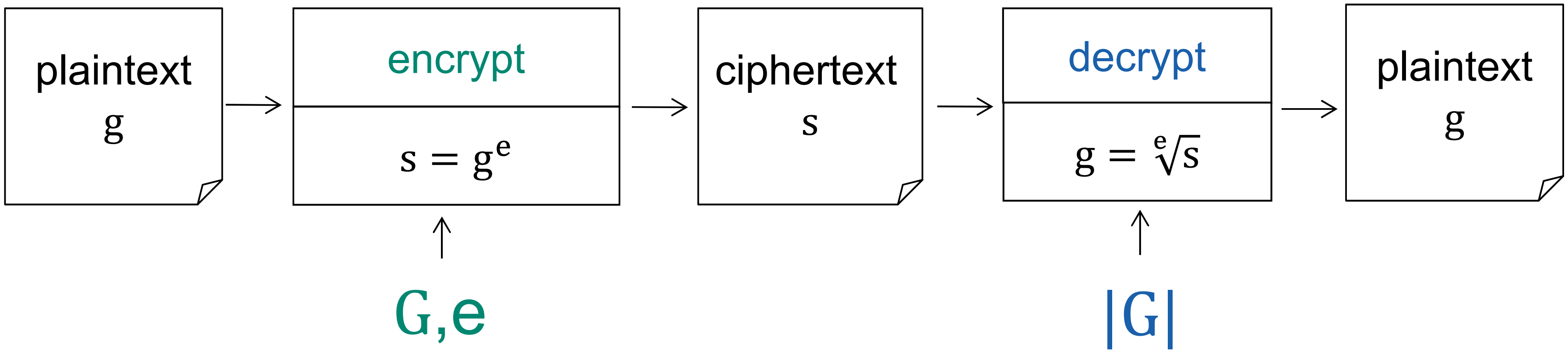
Allows to compute: $\sqrt[e]{g} = g^{e^{-1} \bmod |G|}, g \in G$

# “Generic” RSA encryption

Public key: finite Group $G$, exponent $e$, $\gcd(e, |G|) = 1$

Secret key: $|G|$

Allows to compute: $\sqrt[e]{g} = g^{e^{-1} \bmod |G|}, g \in G$



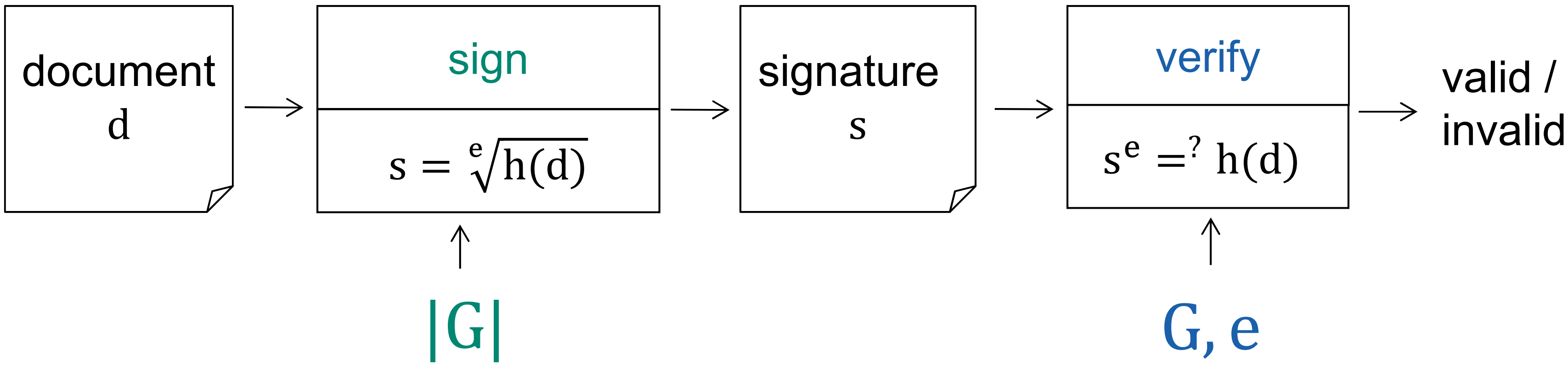
# "Generic" RSA signature

Public key: finite Group $G$, exponent $e$, $\gcd(e, |G|) = 1$

Secret key: $|G|$

Allows to compute: $\sqrt[e]{g} = g^{e^{-1} \bmod |G|}, g \in G$

Hash function $h: \{0,1\}^* \rightarrow G$






CROSSING

# RSA: How to keep $|G|$ secret?

Public key: $e, p, q$ primes, $n = pq$, $G = (\mathbb{Z}/n\mathbb{Z})^*$

Secret key: $|G| = (p-1)(q-1)$

relies on hardness of integer factorization

→ only known method to keep $|G|$ secret

# Factorization complexity

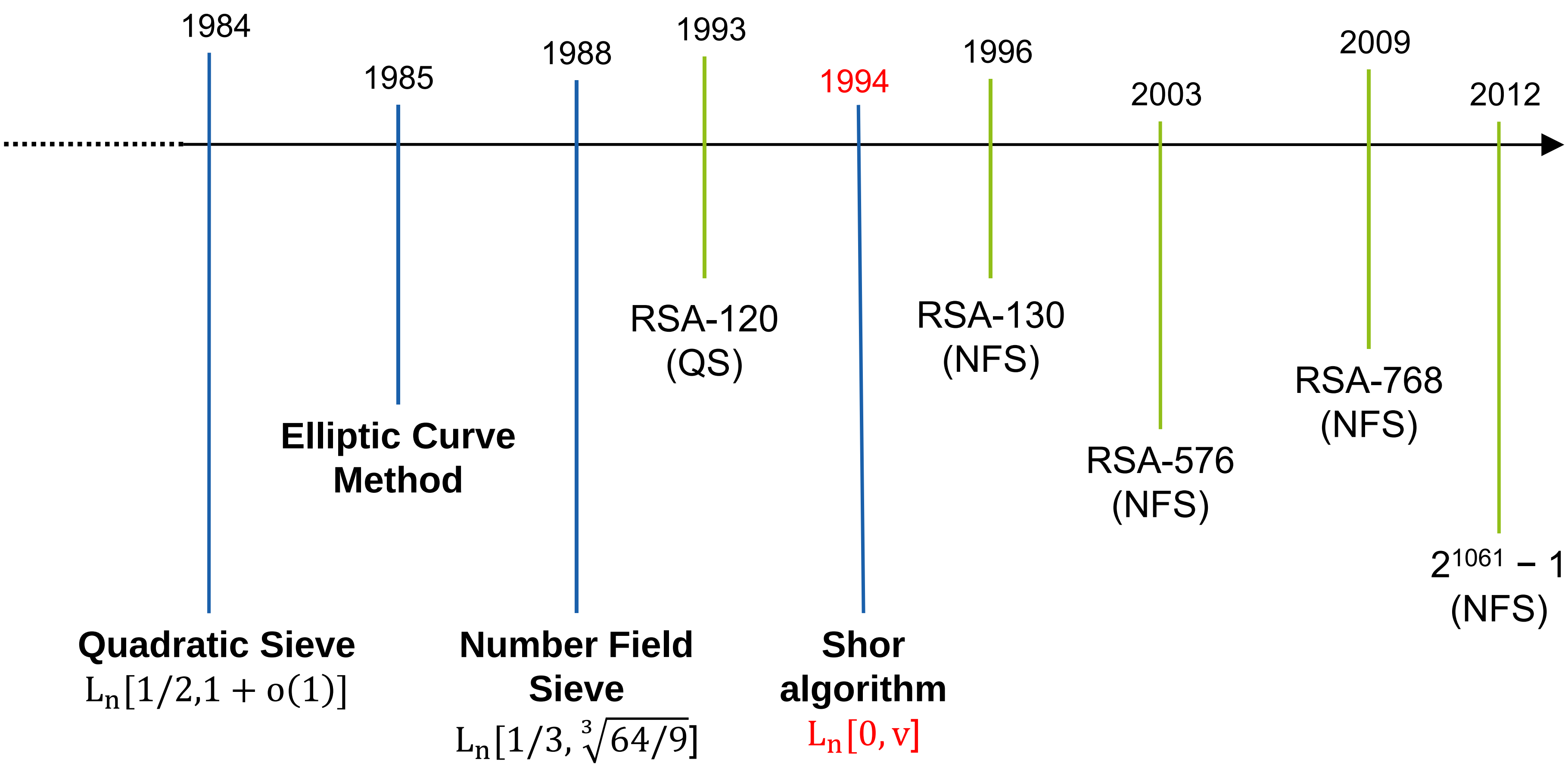
$$L_n[u,v] = e^{v(\log n)^u (\log\log n)^{(1-u)}}$$

$L_n[0,v] = (\log n)^v$ polynomial

$L_n[1,v] = (e^{\log n})^v$ exponential

# Factorization progress

TECHNISCHE UNIVERSITÄT DARMSTADT






CROSSING

# ElGamal encryption and signatures

TECHNISCHE UNIVERSITÄT DARMSTADT

Rely on **Discrete Logarithm** Problem:

Given: Group $G = \langle g \rangle$, $h \in G$

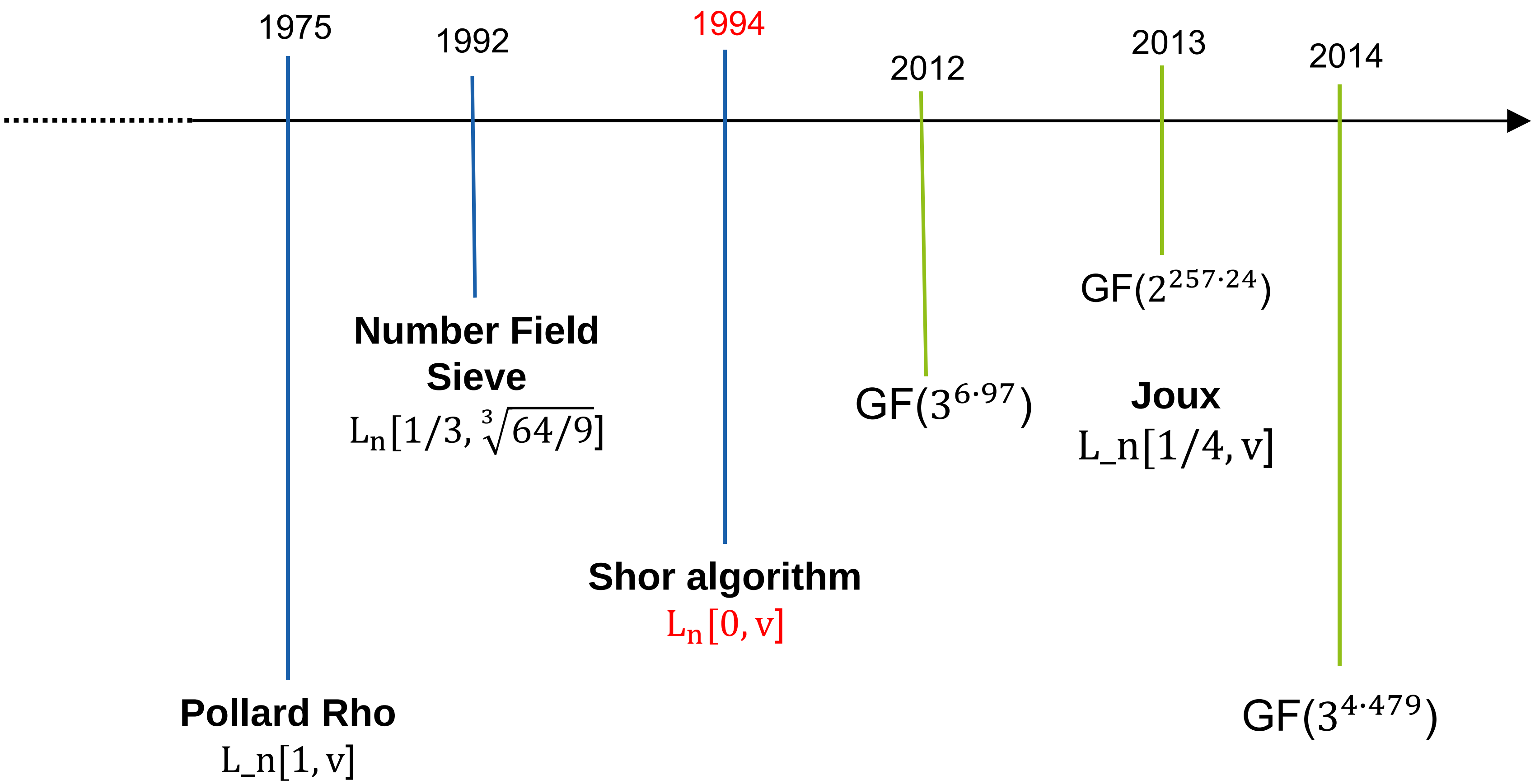
Find: $x \in \mathbb{Z}$ with $h = g^x$

Choices for $G$: -GF$(p^n)^*$

- group of points of elliptic curves over GF$(p^n)$

# Algorithms for solving $\mathrm{GF}(p^n)^*$-DL

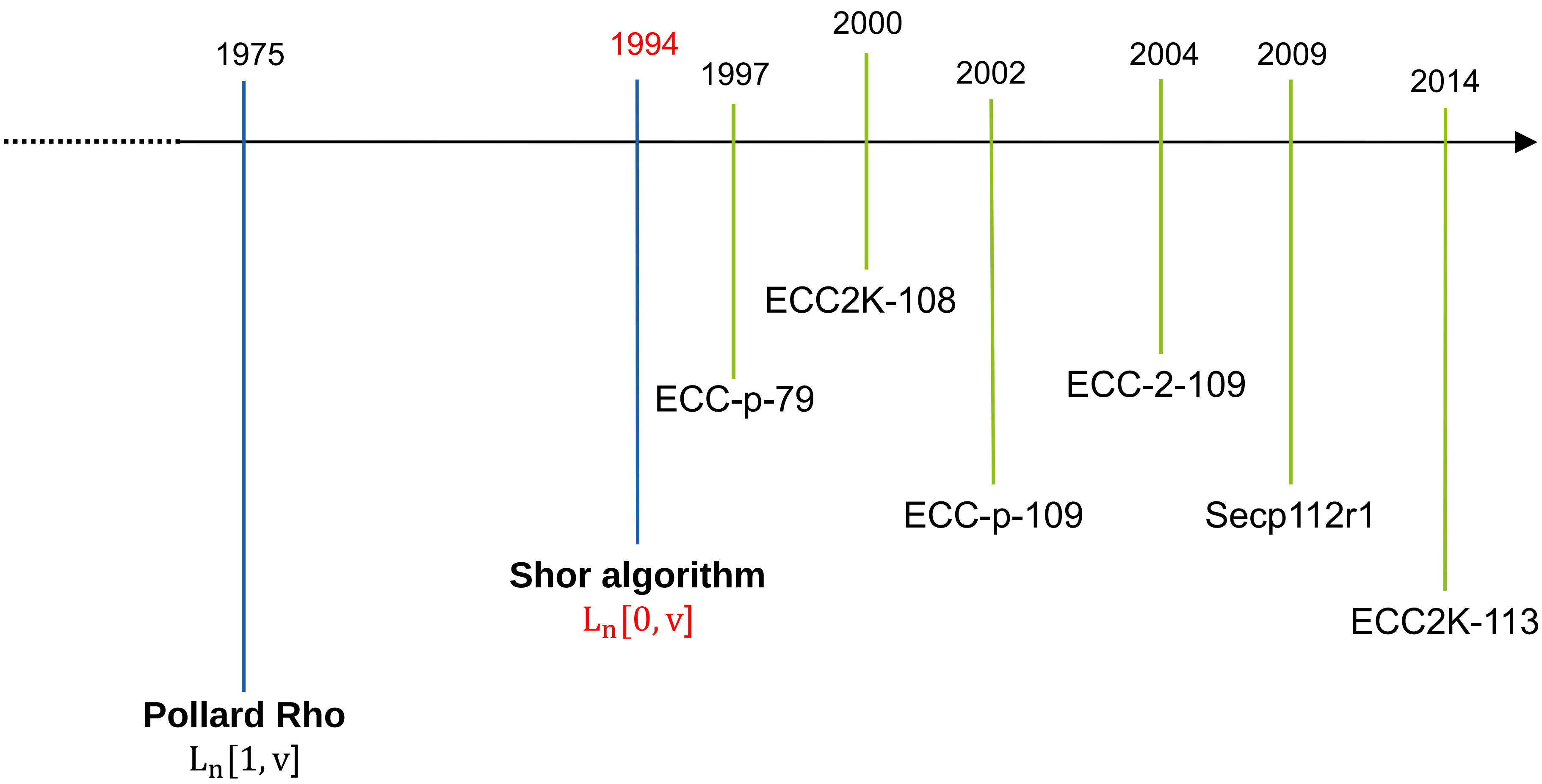
TECHNISCHE UNIVERSITÄT DARMSTADT






CROSSING

# Algorithms for solving EC-DL



# The quantum computer threat

# Shor’s algorithm 1997

TECHNISCHE UNIVERSITÄT DARMSTADT

Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*

Peter W. Shor†

**RSA and ElGamal insecure**

A digital computer is g
device; that is, it is believ
an increase in computatior
true when quantum mechanics is taken into consideration. This paper considers factoring integers and finding discrete logarithms, two problems which are generally thought to be hard on a classical computer and which have been used as the basis of several proposed cryptosystems. Efficient randomized algorithms are given for these two problems on a hypothetical quantum computer. These algorithms take a number of steps polynomial in the input size, e.g., the number of digits of the integer to be factored.

**Keywords:** algorithmic number theory, prime factorization, discrete logarithms, Church’s thesis, quantum computers, foundations of quantum mechanics, spin systems, Fourier transforms

**AMS subject classifications:** 81P10, 11Y05, 68Q10, 03D10

# Quantum computer realistic?

The Washington Post | PostTV | Politics | Opinions | Local | Sports | National | World | Bus

National Security

In the News | Drones | American Airlines | Benghazi | Ferris wheel | ‘American Idol’

Senate report: Benghazi attack was preventable

VIDEO | Top Springsteen political moments

MA Stat

**NSA seeks to build quantum computer that could crack most types of encryption**

By Steven Rich and Barton Gellman, Published: January 2 E-mail the writers

# Quantum computer realistic

TECHNISCHE UNIVERSITÄT DARMSTADT

ScienceD

TRENDING: 5 myths and 5 mistakes job-hunting coders must know about · Android · IT Outsourcing · Resources/White Papers

COMPUTERWORLD

Popular Now:

…ing processors

MORE LIKE THIS

IBM spending $3 billion to rethink decades-old computer design

Money talks, and that's all quantum maker D-Wave has to say

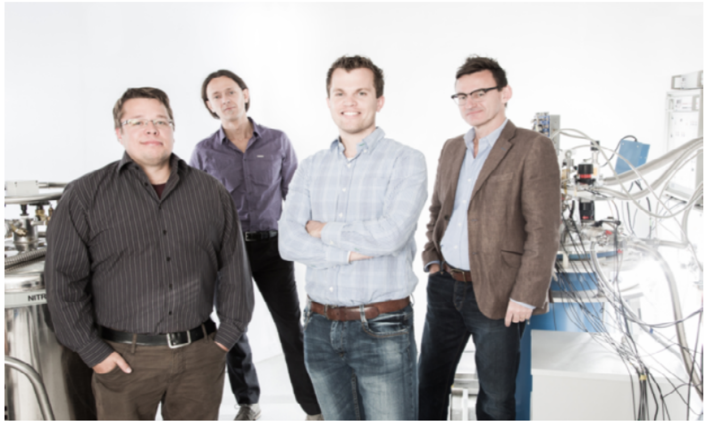
TRENDING: 5 myths and 5 mistakes job-hunting coders must know about · Android · IT Outsourcing · Resources/White Papers

COMPUTERWORLD

Popular Now:

NEWS

## Researchers use silicon to push quantum computing toward reality



*Researchers at the University of New South Wales are pushing forward the possibility of developing a true quantum computer. From left, Juha Muhonen, Andrea Morello, Menno Veldhorst and Andrew Dzurak have been researching ways to use silicon to develop quantum bits.*

*Credit: University of New South Wales*

MORE LIKE THIS

Quantum rewrites the rules of computing

IBM spending $3 billion to rethink decades-old computer design

Money talks, and that's all quantum maker D-Wave has to say

New tech could let quantum machines tackle huge problems

By Sharon Gaudin FOLLOW

Computerworld | Oct 23, 2014 9:27 AM PT

Researchers in Australia have developed silicon-wrapped quantum technology

…ia, Santa

…ystems.

…s to perform



CROSSING

WIKIPEDIA
The Free Encyclopedia

Main page
Contents
Featured content
Current events
Random article
Donate to Wikipedia
Wikimedia Shop

Interaction
Help
About Wikipedia
Community portal
Recent changes
Contact page

Tools
What links here
Related changes
Upload file
Special pages
Permanent link
Page information
Wikidata item
Cite this page

Print/export
Create a book
Download as PDF
Printable version

Languages

Article Talk | Read Edit View history Search

# Quantum computer

From Wikipedia, the free encyclopedia

A **quantum computer** is a computation system that makes direct use of quantum-mechanical phenomena, such as superposition and entanglement, to perform operations on data.[1] Quantum computers are different from digital computers based on transistors. Whereas digital computers require data to be encoded into binary digits (bits), each of which is always in one of two definite states (0 or 1), quantum computation uses qubits (quantum bits), which can be in superpositions of states. A theoretical model is the quantum Turing machine, also known as the universal quantum computer. Quantum computers share theoretical similarities with non-deterministic and probabilistic computers; one example is the ability to be in more than one state simultaneously. The field of quantum computing was first introduced by Yuri Manin in 1980[2] and Richard Feynman in 1982.[3][4] A quantum computer with spins as quantum bits was also formulated for use as a quantum space–time in 1968.[5]

As of 2014, quantum computing is still in its infancy but experiments have been carried out in which quantum computational operations were executed on a very small number of qubits.[6] Both practical and theoretical research continues, and many national governments and military funding agencies support quantum computing research to develop quantum computers for both civilian and national security purposes, such as cryptanalysis.[7]

Large-scale quantum computers will be able to solve certain problems much quicker than any classical computer using the best currently known algorithms, like integer factorization using Shor's algorithm or the simulation of quantum many-body systems. There exist quantum algorithms, such as Simon's algorithm, that run faster than any possible probabilistic classical algorithm.[8] Given sufficient computational resources, however, a classical computer could be made to simulate any quantum algorithm, as quantum computation does not violate the Church–Turing thesis.[9]

The Bloch representa fundament computers

Contents [hide]





CROSSING

# Post-quantum cryptography

# Performance requirements

| Secure until | Security level | RSA modulus/finite field size | Elliptic curve |
|---|---|---|---|
| 2015 | 80 | 1248 | 160 |
| 2025 | 96 | 1776 | 192 |
| 2030 | 112 | 2493 | 224 |
| 2040 | 128 | 3248 | 256 |

Ecrypt recommendations

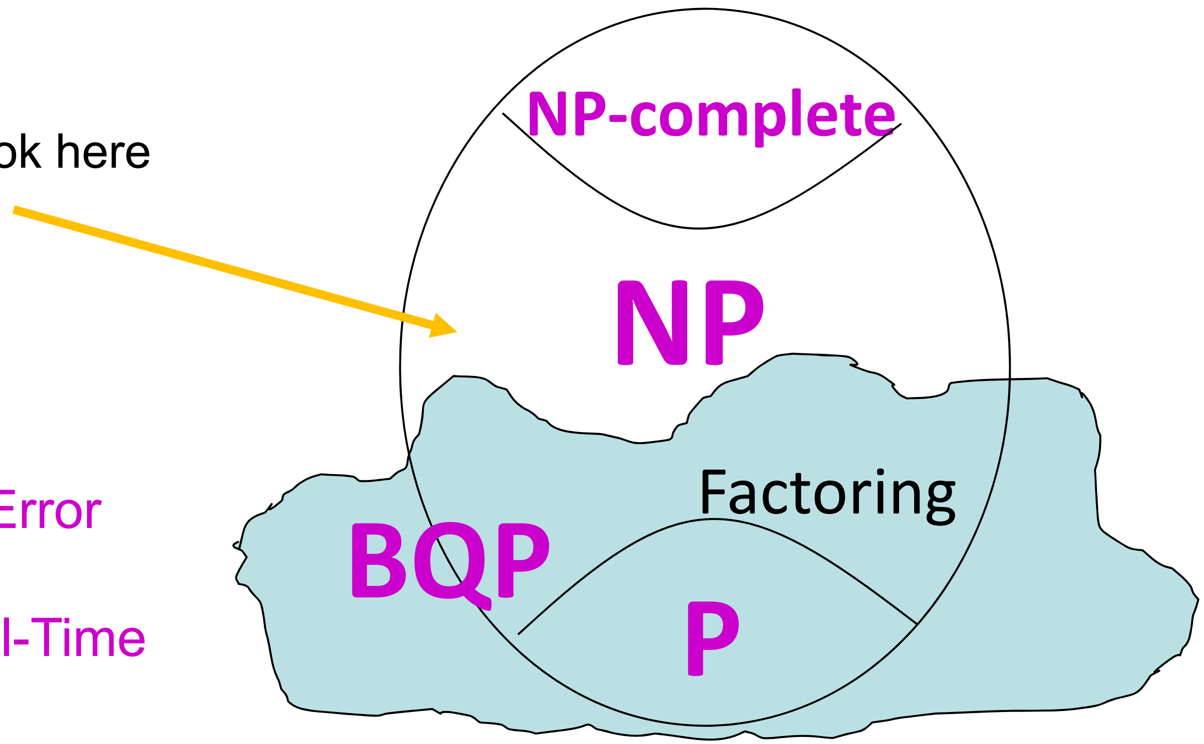
- Space for keys and signatures: a few kilobytes
- Small ciphertext expansion
- Times: milliseconds

# Post-quantum problems?

No provable quantum resistence

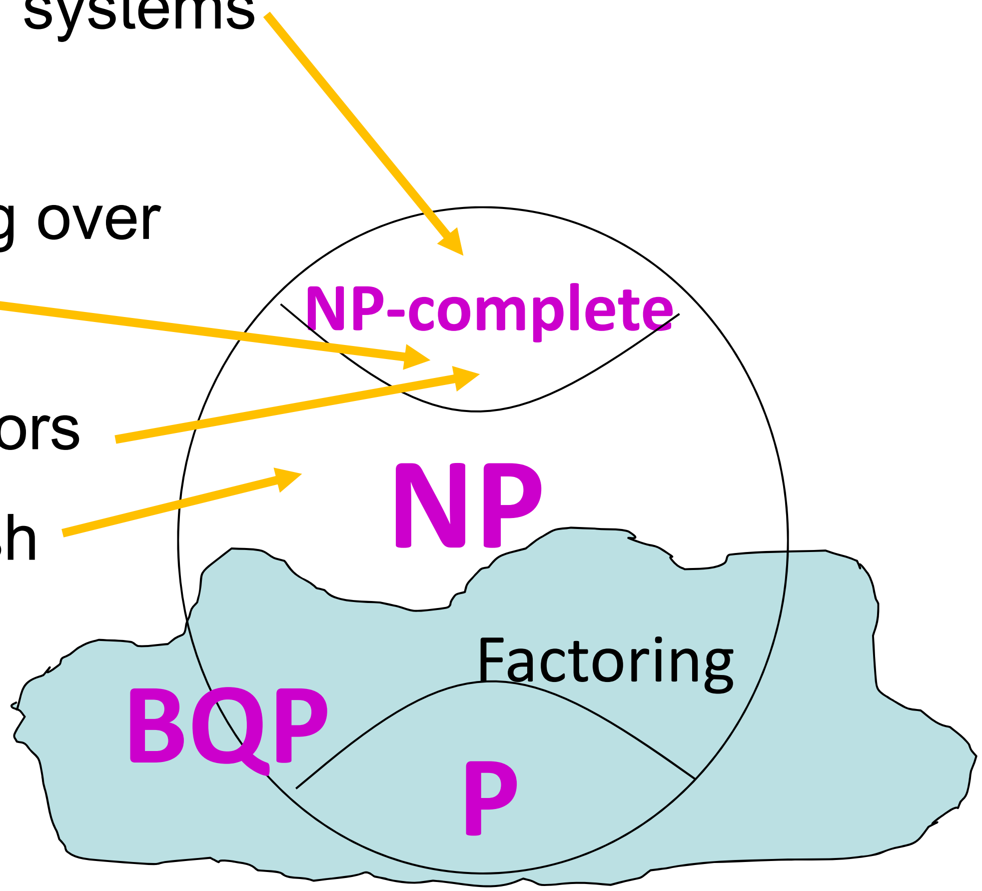
We must look here

Bounded-Error
Quantum
Polynomial-Time



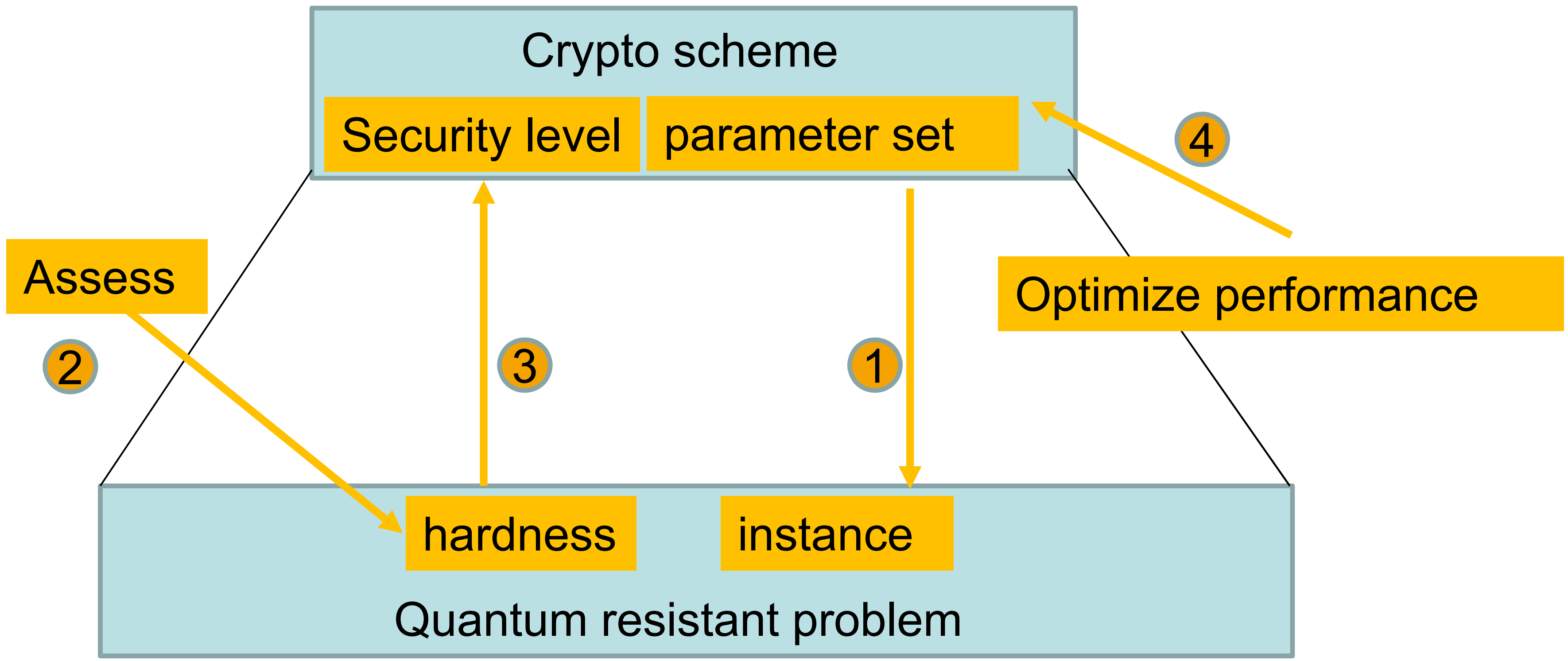
## Candidates

- Solving non-linear equation systems over finite fields
- Bounded distance decoding over finite fields
- Short and close lattice vectors
- Breaking cryptographic hash functions
- Quantum key exchange



# Strategy




Crypto scheme
Security level
parameter set
4
Assess
Optimize performance
2
3
1
hardness
instance
Quantum resistant problem

# Multivariate cryptography

# MQ problem

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

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

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

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

# MQ-Problem

TECHNISCHE UNIVERSITÄT DARMSTADT

Given: $n, m, p_1, \ldots, p_m \in F[x_1, \ldots, x_n]$ quadratic, $F$ finite field

Find: $y_1, \ldots, y_n \in F$, such that

$$p_1(y_1, \ldots, y_n) = \ldots = p_m(y_1, \ldots, y_n) = 0$$

MP is NP-complete (Garey, Johnson 1979) (decision version)



CROSSING

# Multivariate signatures



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

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

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

Secret Key: $S, P$, T allows to compute $G^1 = T^{-1} \circ P^{-1} \circ S^{-1}$

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

Verifying: $G(s) \stackrel{?}{=} m$

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

# Code-based cryptography

# Bounded distance decoding problem

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

Find:
- $x \in C\colon \operatorname{dist}(x, y) \leq t$

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

# McEliece cryptosystem (1978)

TECHNISCHE UNIVERSITÄT DARMSTADT

$S, G, P$ 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



CROSSING

# Lattice-based cryptography

# Why lattice-based cryptography?

- Expected to resist quantum computer attacks
- Worst-to-average-case reduction
- Permits fully homomorphic encryption

# Lattice problems

$n \in \mathbb{N}, L = \mathbb{Z}b_1 + \cdots + \mathbb{Z}b_n \subseteq \mathbb{R}^n$ lattice; B = ($b_1, \ldots, b_n$) basis

## α-Shortest Vector Problem (SVP)

Given: $\alpha > 1$, lattice $L = L(B)$ basis $B$

Find: $v \in L$ nonzero such that $||v|| \leq \alpha\lambda_1(L)$

## α-Closest Vector Problem (CVP)

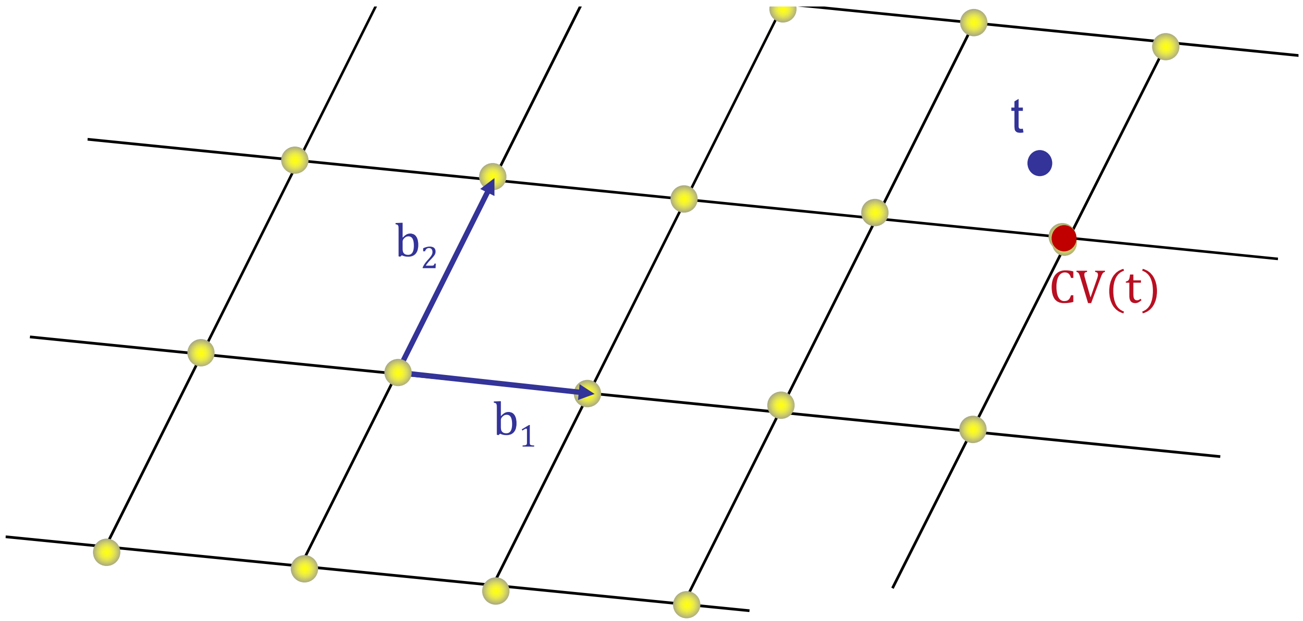
Given: $\alpha > 1$, lattice $L = L(B)$ basis $B$, $t$

Find: $v \in L$ such that $\left||t - v|\right| \leq \alpha \min_{w \in L} ||t - w||$

# 2-dimensional $\alpha$CVP

Given: $B = (b_1, b_2),\ t, \alpha$

Find: $CV(t) \in L(B)$: $\big\| t - CV(t) \big\| \leq \alpha \min_{w \in L} \| t - w \|$

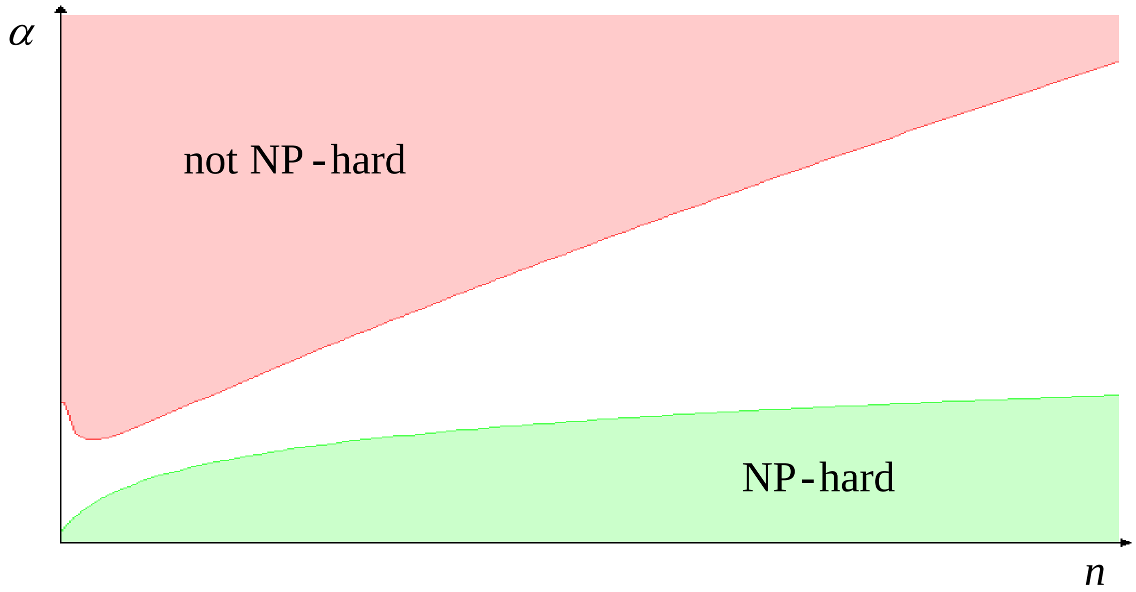


# Complexity of α-CVP

TECHNISCHE UNIVERSITÄT DARMSTADT

Arora et al. (1997):

$$\log(n)^c\text{-CVP is NP-hard for all } c$$



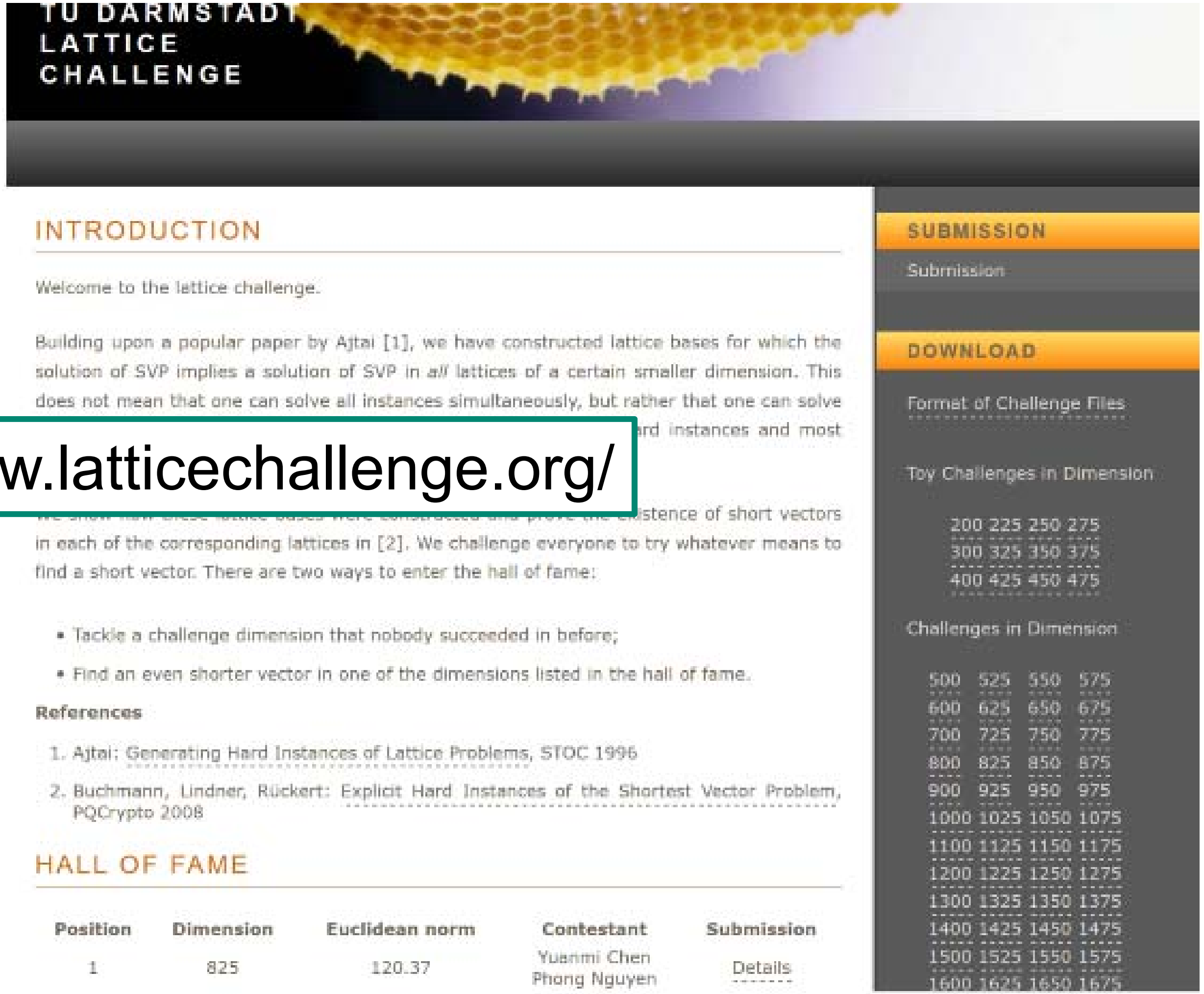

Goldreich, Goldwasser (2000):

$$\Omega\left(\sqrt{n}/\log(n)\right)\text{-CVP is not NP-hard or } \mathbf{coNP} \subseteq \mathbf{AM}$$



CROSSING

# Practical complexity

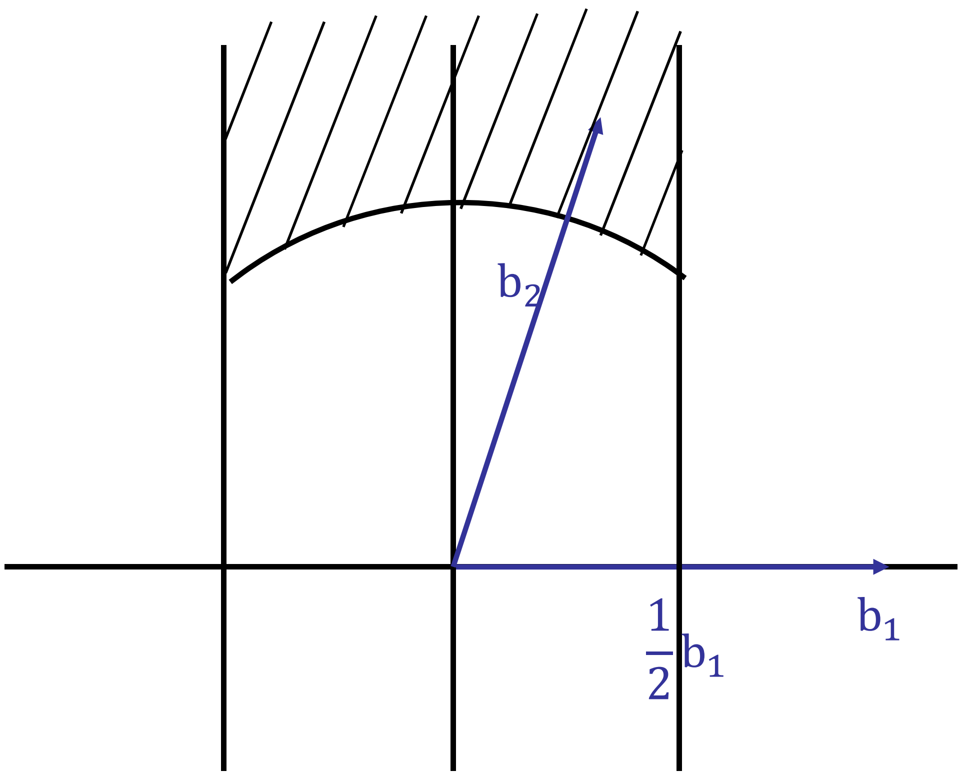



http://www.latticechallenge.org/

# The idea of lattice-based cryptography

- **GGH Sign 1995**
- NTRU Encrypt 1996
- NTRU Sign 2003

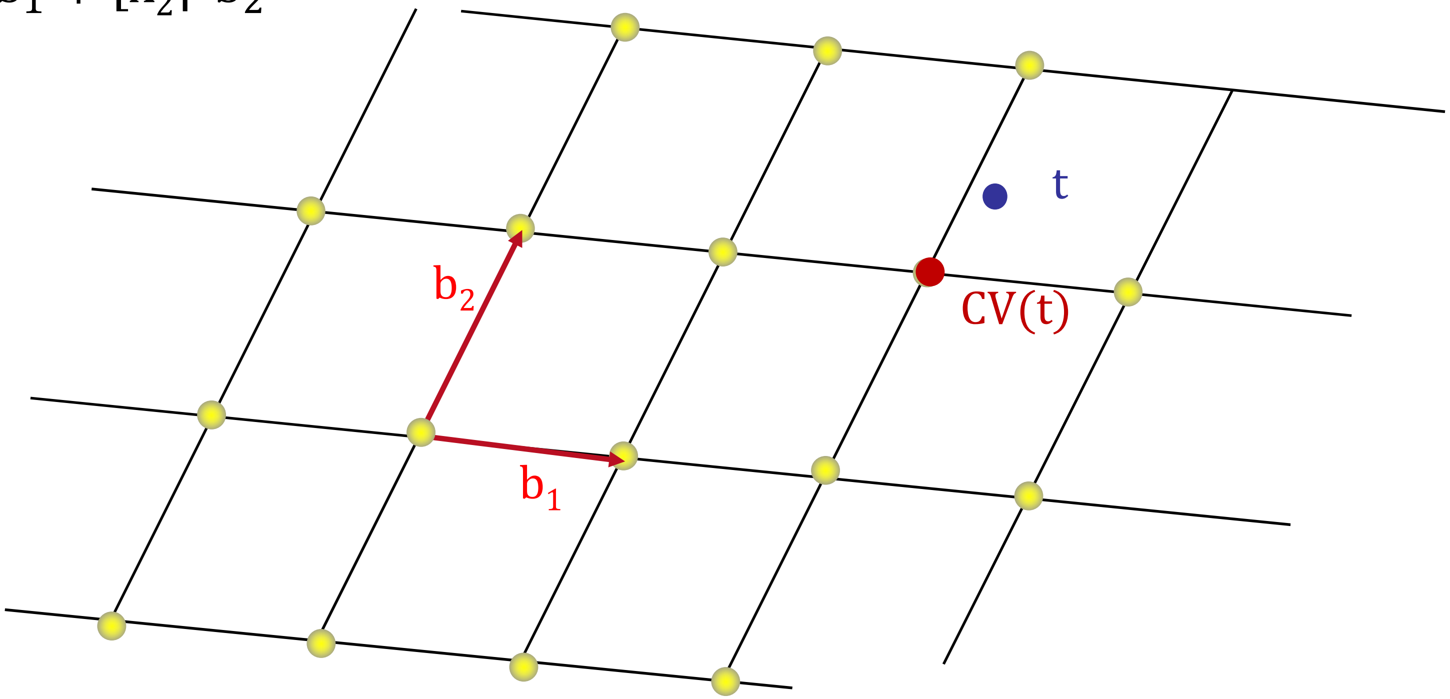
# Reduced bases (Gauß 1801)



# $(\mathbf{b_1}, \mathbf{b_2})$ reduced ⇒ CVP easy

$t = x_1 b_1 + x_2 b_2$

$CV(t) = \lfloor x_1 \rceil b_1 + \lfloor x_2 \rceil\, b_2$



# $\mathbf{B} = (\mathbf{b_1}, \mathbf{b_2})$ not reduced ⇒ CVP hard



$L = \mathbb{Z}^2$, $B = \left(\begin{pmatrix}1\\0\end{pmatrix}, \begin{pmatrix}0\\1\end{pmatrix}\right)$, $t = \begin{pmatrix}3.4\\-2.3\end{pmatrix}$, $\mathrm{CVP}(t) = \begin{pmatrix}3\\-2\end{pmatrix}$

Another basis $B' = \left(\begin{pmatrix}100\\99\end{pmatrix}, \begin{pmatrix}99\\98\end{pmatrix}\right)$

$t = \begin{pmatrix}3.4\\-2.3\end{pmatrix} = -560.9 \cdot \begin{pmatrix}100\\99\end{pmatrix} + 566.6 \cdot \begin{pmatrix}99\\98\end{pmatrix}$

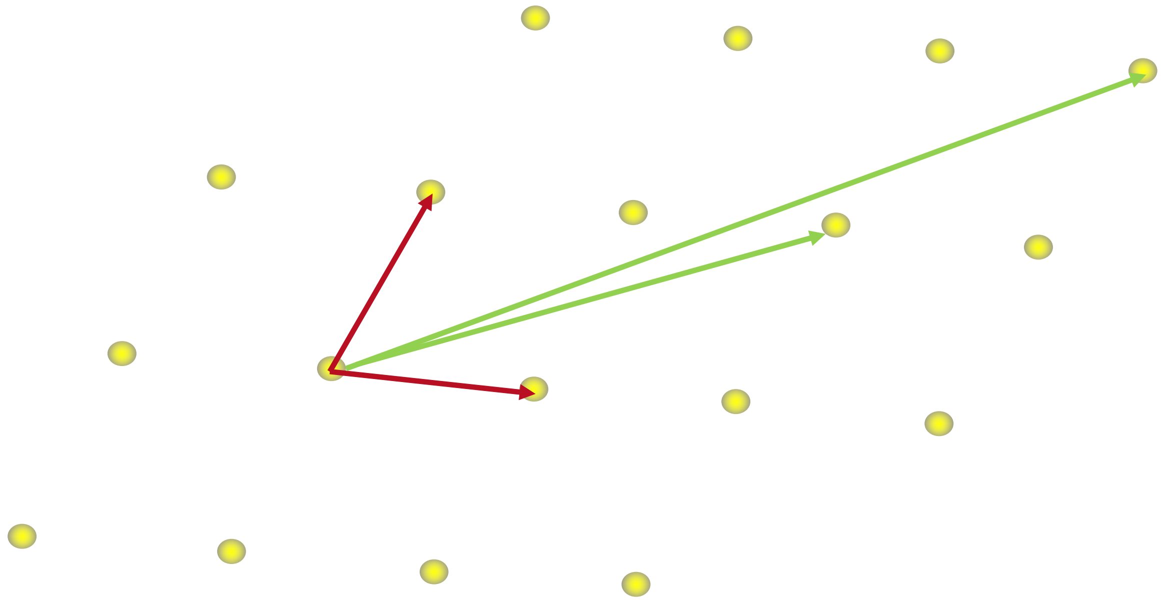
$-561 \cdot \begin{pmatrix}100\\99\end{pmatrix} + 567 \cdot \begin{pmatrix}99\\98\end{pmatrix} = \begin{pmatrix}33\\27\end{pmatrix} \neq \begin{pmatrix}3\\-2\end{pmatrix} = \mathrm{CVP}(t)$

# Key generation

Key generation: $\mathrm{n} \in \mathbb{N}, \mathrm{L} \subseteq \mathbb{R}^{\mathrm{n}}$ lattice

Secret key: „reduced“ basis $\mathrm{B}$ of $\mathrm{L}$. (Allows to efficiently solve CVP.)

Public key: „bad” basis $\mathrm{B'}$ of $\mathrm{L}$. (Does not.)



# Public-key encryption



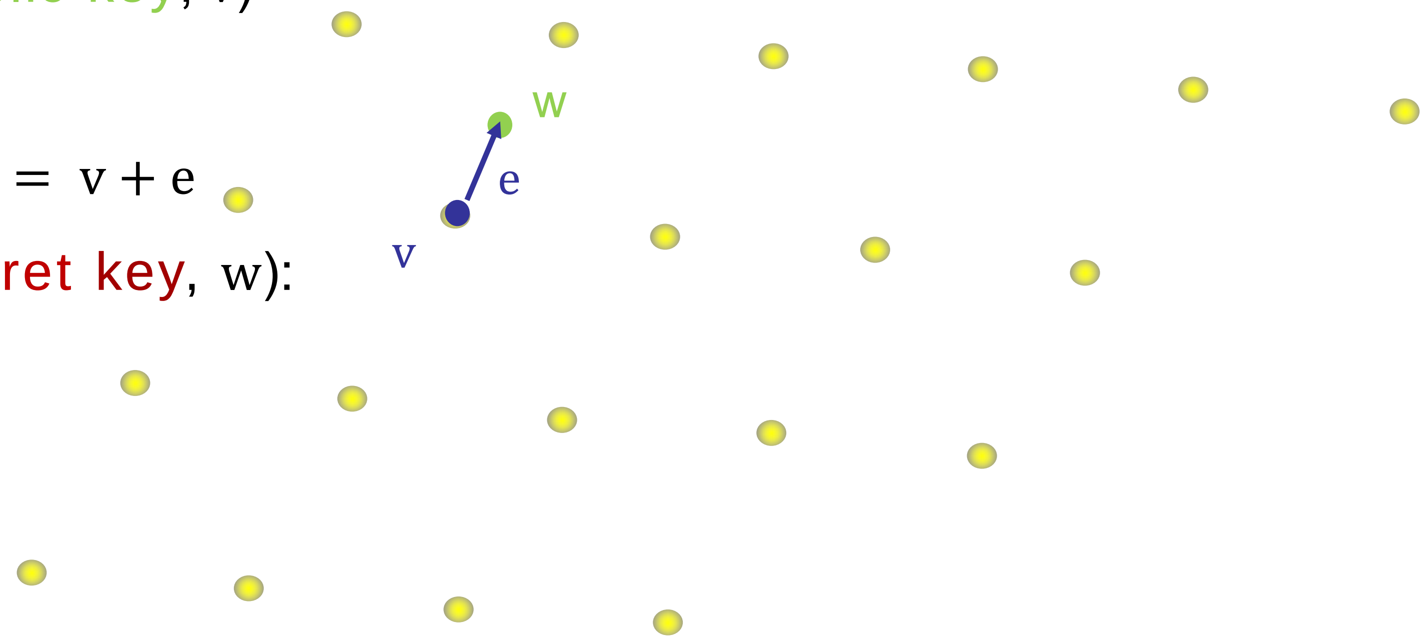
Plaintext $v \in L$

Encryption(public key, $v$)

- small $e \in \mathbb{R}^n$
- ciphertext $w = v + e$

Decryption(secret key, $w$):

- $v = CV(w)$



# Digital signature

Public: Cryptographic hash function $h: \{0,1\} \to \mathbb{R}^n$

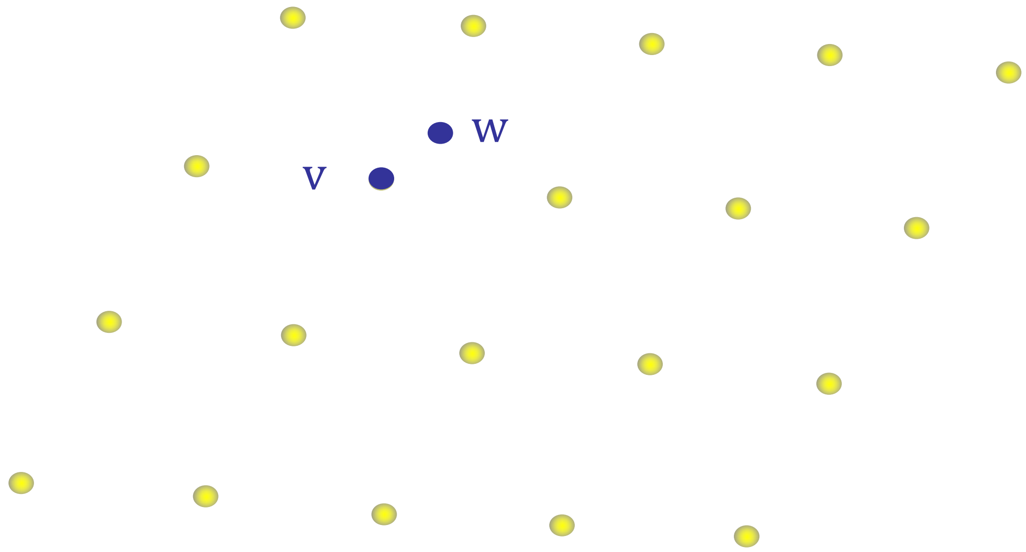
Sign(secret key, document $d$):

$$w = h(d)$$

$$v = CV(w)$$

Verify(public key, $v, w$):

$v$ close to $w$ ?

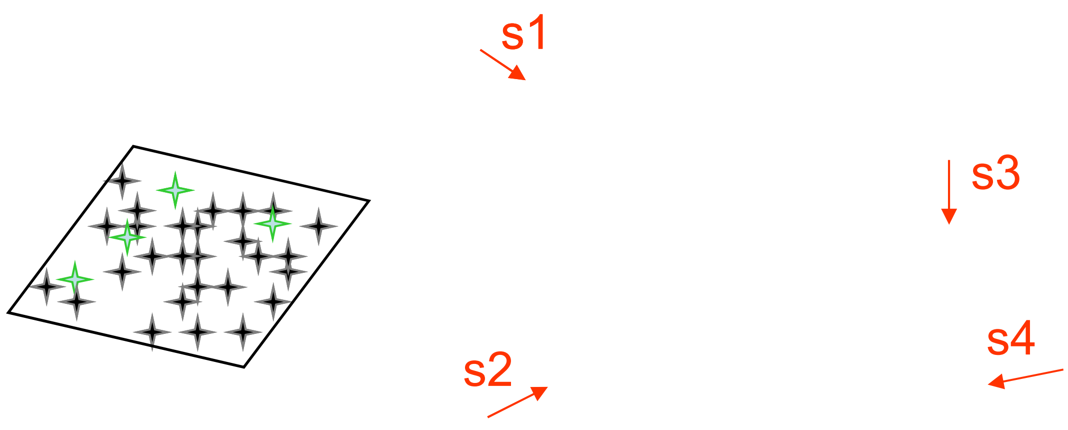





CROSSING

# Learning the secret key

## Nguyen and Regev 2006




NTRU-251 broken using ≈ 400 signatures

GGH-400 broken using ≈ 160.000 signatures



TECHNISCHE UNIVERSITÄT DARMSTADT
CROSSING

# Performance

- NTRU encrypt 1996: fast and small

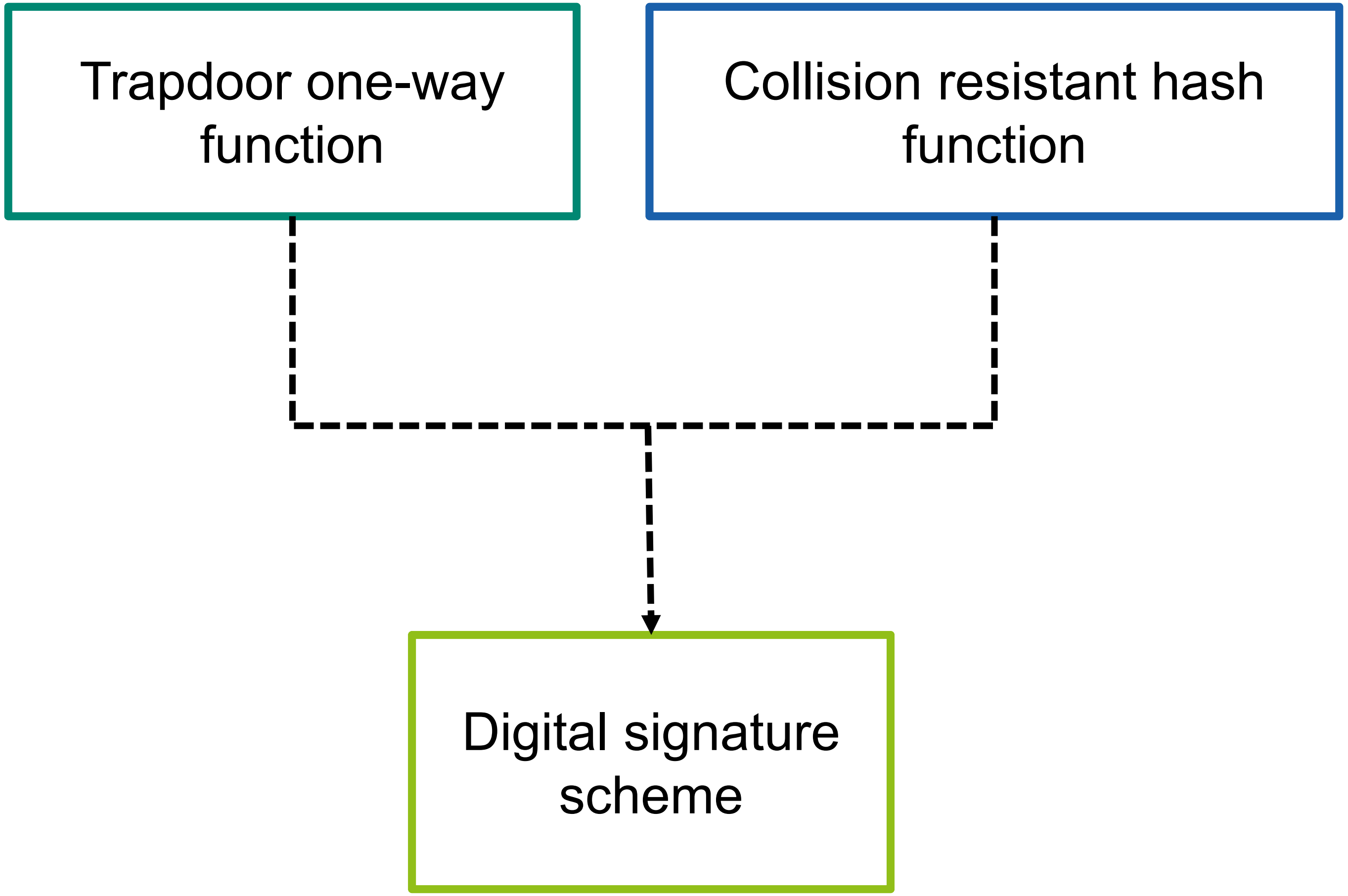
The provable schemes to be studied more

- Bliss 2013 and Bai/Galbraith 2014 signature with improvements of Bindel: fast but large signatures

- Lindner, Peikert 2010 encryption with improvements of Göpfert: fast but ciphertext expansion

# Hash-based signatures

# Typical construction

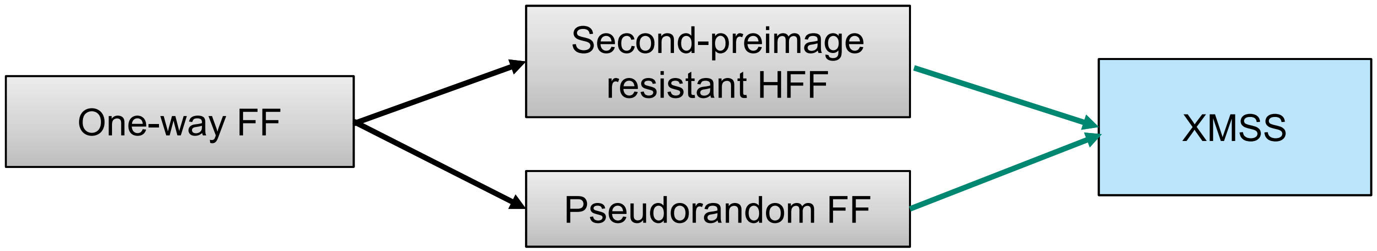


# Trapdoor one-way functions hard to construct but not required

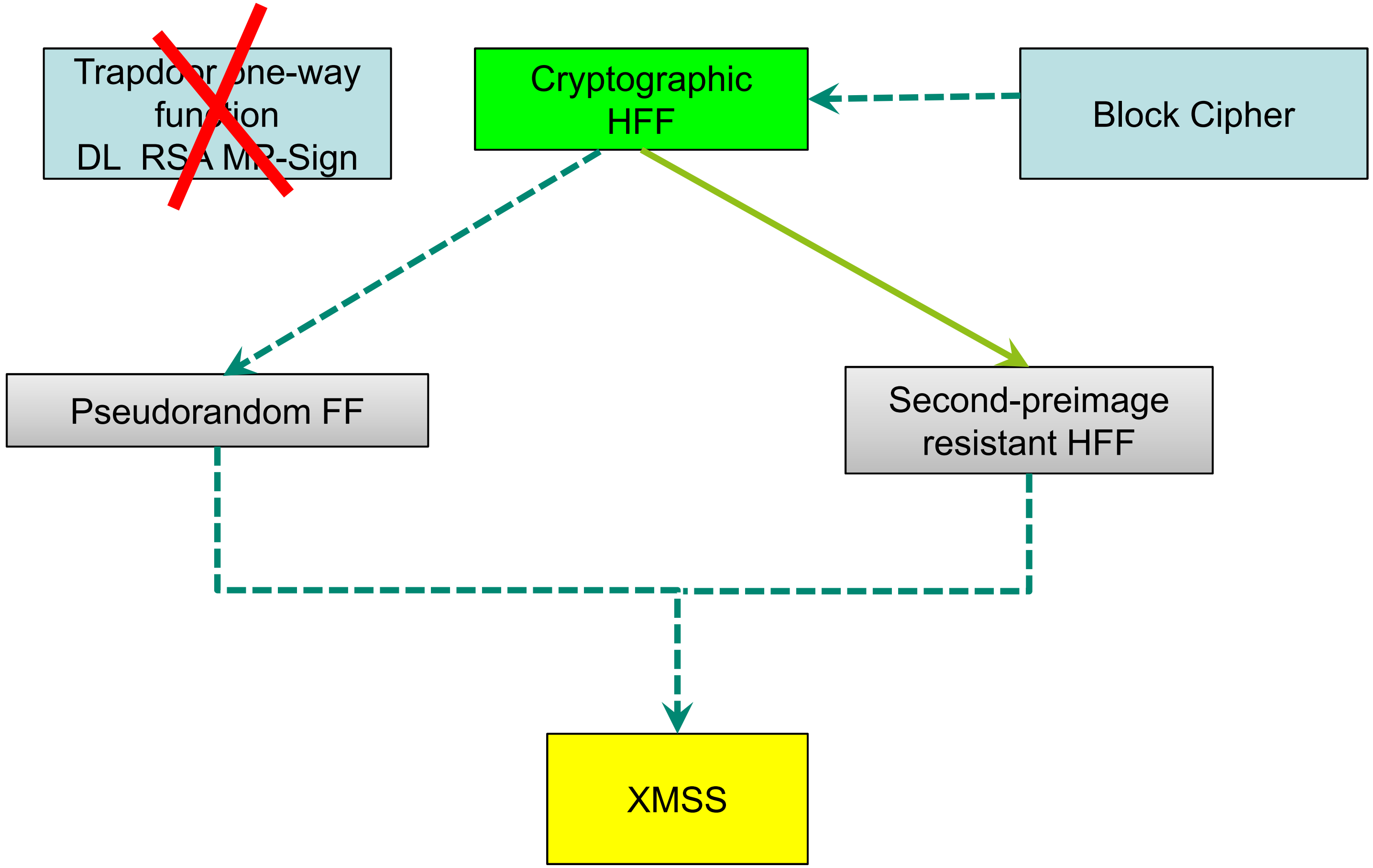
# XMSS signature

## JB, Coronado, Dahmen, Hülsing

- Based on Merkle signature scheme
- Has minimal security requirements



# XMSS in practice



# Hash functions & Blockciphers

TECHNISCHE UNIVERSITÄT DARMSTADT

AES
Blowfish
3DES
Twofish
Threefish
Serpent
IDEA
RC5
RC6
…

SHA-2
SHA-3
BLAKE
Grøstl
JH
Keccak
Skein
VSH
MCH
MSCQ
SWIFFTX
RFSB
…

# XMSS performance

TECHNISCHE UNIVERSITÄT DARMSTADT

| | 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,<br>w = 64, |
| XMSS-AES-NI | **0.52** | **0.07** | 19,616 | 7,328 | 1,684 | 84 | h = 20,<br>w = 4 |
| XMSS-AES | 1.06 | 0.11 | 19,616 | 7,328 | 1,684 | 84 | h = 20,<br>w = 4 |
| RSA 2048 | **3.08** | **0.09** | ≤ 2,048 | ≤ 4,096 | ≤ 512 | 87 | |



CROSSING

# XMSS transfer project

**Denis Butin, Stefan Gazdag**

Practical Hash-based Signatures

http://www.square-up.org/

Soviel ist sicher.

# Conclusion

# Todos

- Standardize and integrate into standard applications: XMSS + NTRU-Encrypt/McEliece

CROSSING
http://www.crossing.tu-darmstadt.de

- Provide/optimize security proofs

- Study computational problems in the presence of modern computing architectures -> parameter selection

- Optimize schemes for secure parameters - consider side channels.

- Integrate with quantum key exchange.


THANK YOU
VERY MUCH