iCIS | Digital Security
Radboud University

# Post Quantum Cryptography for the IoT

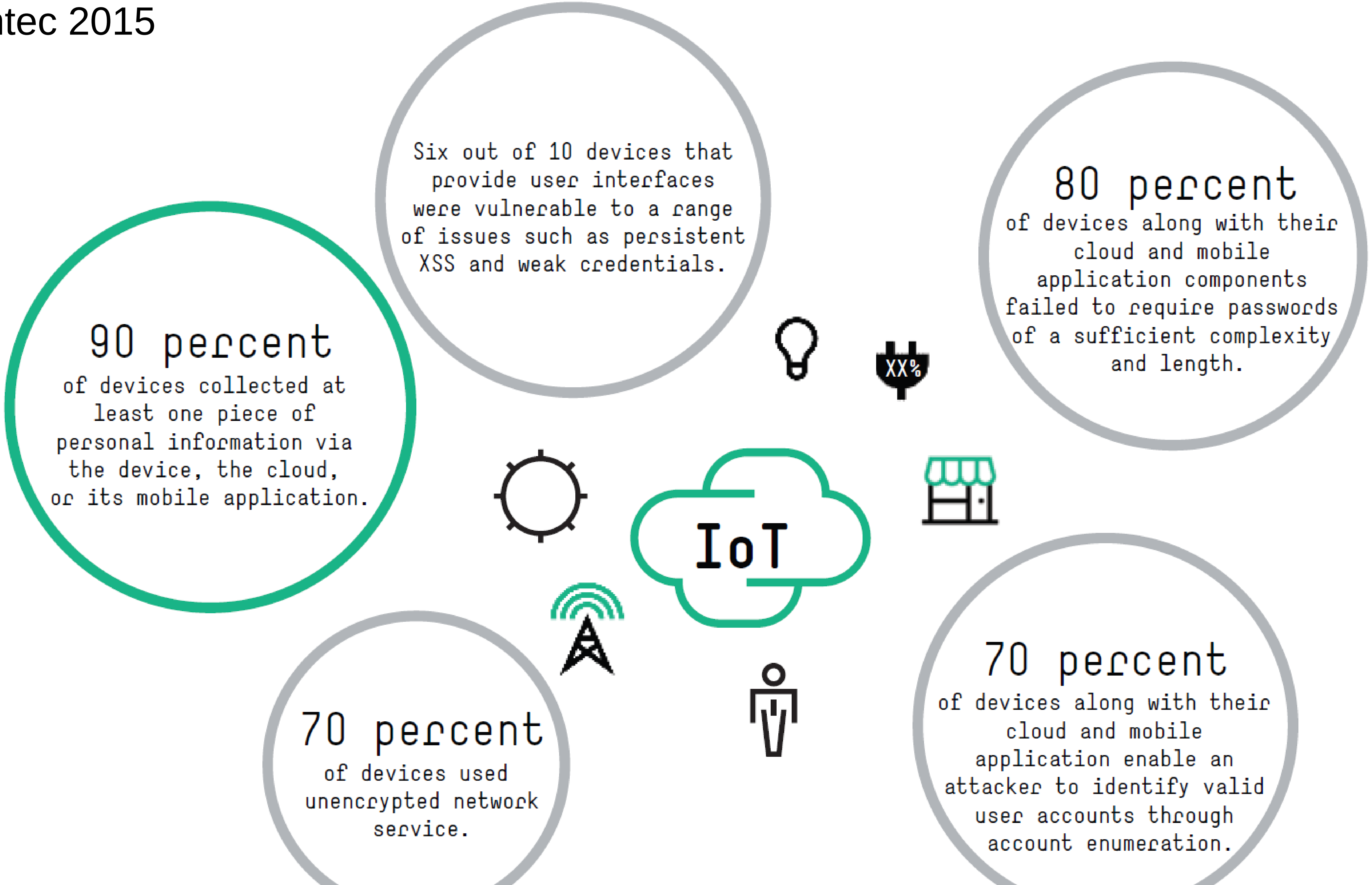
Simona Samardjiska
Digital Security Group – Radboud University

# Crypto in the IoT - Where do we stand?

**Few years ago**:
- OWASP 2014, HP study 2015, Symantec 2015

The OWASP Internet of Things Top 10 - 2014

- I1 Insecure Web Interface
- I2 Insufficient Authentication/Authorization
- I3 Insecure Network Services
- I4 Lack of Transport Encryption
- I5 Privacy Concerns
- I6 Insecure Cloud Interface
- I7 Insecure Mobile Interface
- I8 Insufficient Security Configurability
- I9 Insecure Software/Firmware
- I10 Poor Physical Security




HP study 2015

# Crypto in the IoT - Where do we stand?

**Few years ago**:

- OWASP 2014, HP study 2015, Symantec 2015
- Crypto not used or used improperly

- The global picture is more or less **still the same**
  – **October 2016**: Dyn DNS provider DDoS attack - Mirai malware botnet (DVRs and webcams) brings down Twitter, Amazon, Reddit, Spotify, Netflix, PlayStation Network
  – **April 2017**: BrickerBot, Persirai, …

IOT CRIME DIARY

FROM THE EDITORS AT CYBERSECURITY VENTURES

Q1 2017

IoTCrimes.com provides business and technology executives, chief information security officers (CISOs), IT security teams, and the cyber community with a quarterly diary of noteworthy Internet of Things (IoT) hacking and breach activity, and the latest innovations aimed at thwarting IoT crimes.

  – WikiLeaks, **March 2017**: *"...exposes how the Central Intelligence Agency hacks smartphones, computer operating systems, message applications and internet-connected televisions…"*
  – Altman Vilandrie & Company, **April 2017**: "*Almost half of all companies in the US using an IoT network have been the victims of recent security breaches"*

# IoT soup crypto challenges

- Crypto is a solution for many of the IoT security issues - But it is costly!

- **Major problem – constrained environment**

  - Memory constrains
    - Typically several KB
    - 8 bit NXP RS08: 64B-16B RAM

  - Energy and power consumption
    - RFID tags, solar powered sensors

  - Chip area
    - FPGA – LUTs, flip-flops, multiplexers
    - ASIC – NAND gates (GE)
      - In RFID 200-2000 GE for security

  - Latency
  - Limited set of instructions

- Many devices should be very cheap
  - Yet, Nist approved ATECC508A supports ECDH and ECDSA for <0.8$, and is 5mm2


- One size fits all approach not possible
  - Still standards necessary!

# Solutions

- **Application specific cryptography**
  - Different platforms
  - Different usage
  - Different critical security issue
  - Different performance requirements

Standards necessary
for each and every one!

- **Lightweight cryptography**
  - Trade-off between security and performance
  - FELICS project www.cryptolux.org/index.php/FELICS - benchmarking lightweight crypto
  - NIST recommendations and (soon) standards
    - NISTIR 8114: Report on Lightweight Cryptography (March 2017)
    - NIST-Approved Cryptographic Primitives in Constrained Environments

- **Transport layer security**
  - Wide implementation of DTLS
  - PKI for IoT
  - Key management, key generation, key distribution

# The quantum computer threat

- A **universal quantum computer** - Deutsch '85
  - Based on the principles of quantum mechanics
  - Capable of efficiently simulating an arbitrary physical system

# The quantum computer threat

- A **universal quantum computer** - Deutsch ‘85
  - Based on the principles of quantum mechanics
  - Capable of efficiently simulating an arbitrary physical system

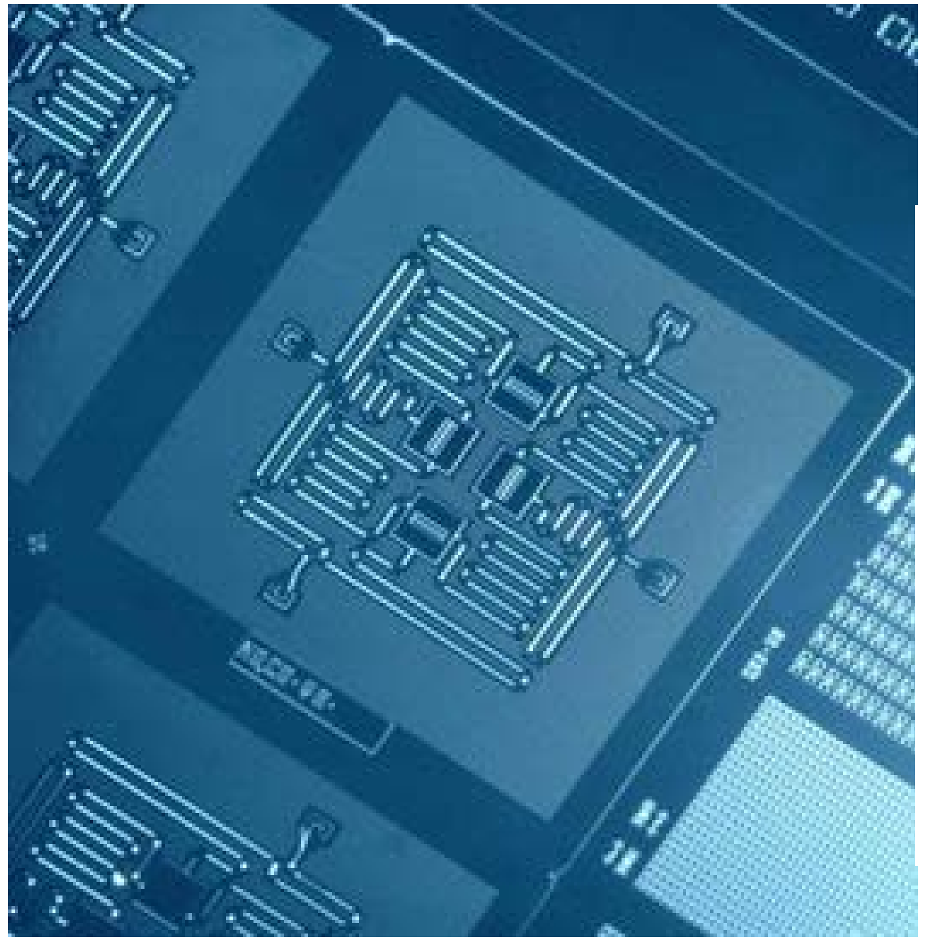


Photo: IBM Research

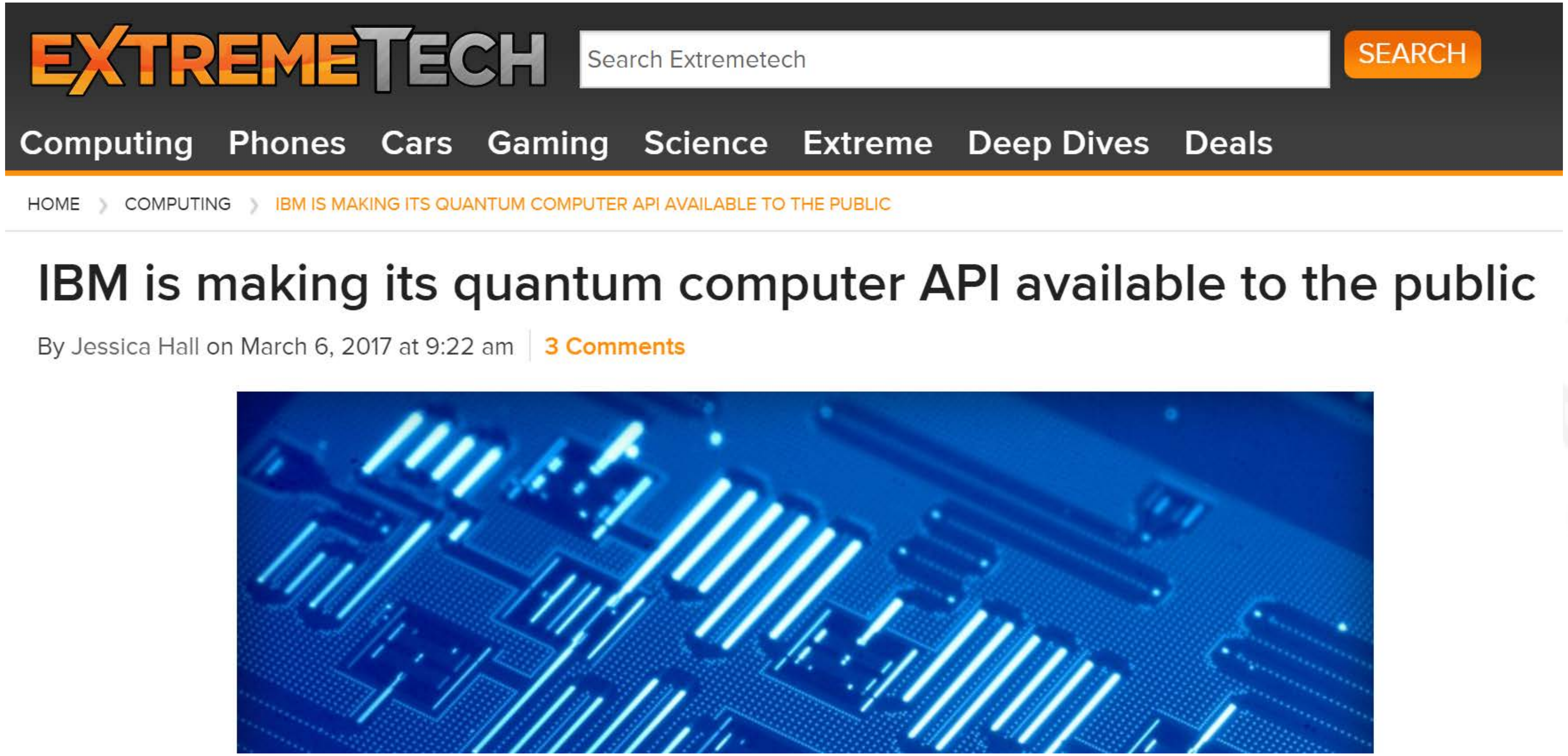
IEEE SPECTRUM

“With our recent four-qubit network, we built a system that allows us to detect both types of quantum errors,” says Jerry Chow, manager of experimental quantum computing at IBM’s Thomas J. Watson Research Center, in Yorktown Heights, N.Y. Chow, who, along with his IBM colleagues detailed their experiments in the 29 April issue of the journal Nature Communications, says, “This is the first demonstration of a system that has the ability to detect both bit-flip errors and phase errors” that exist in quantum computing systems.

The IBM system consists of four quantum bits, or qubits, arranged in a 2-by-2 configuration on a chip measuring about 1.6 square centimeters (0.25 square

# The quantum computer threat

- A **universal quantum computer** - Deutsch ‘85
  - Based on the principles of quantum mechanics
  - Capable of efficiently simulating an arbitrary physical system


EXTREMETECH

Search Extremetech SEARCH

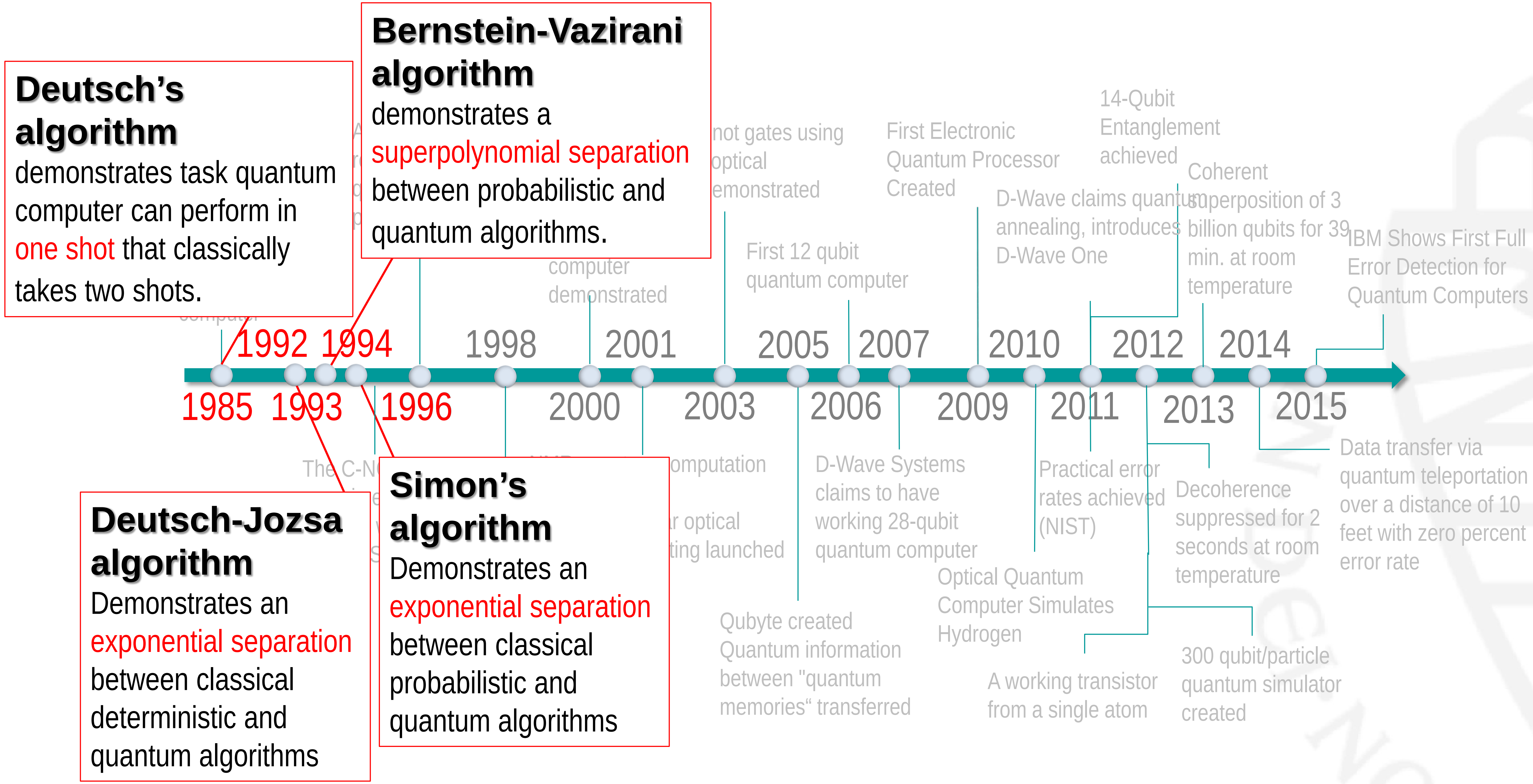
Computing Phones Cars Gaming Science Extreme Deep Dives Deals

HOME › COMPUTING › IBM IS MAKING ITS QUANTUM COMPUTER API AVAILABLE TO THE PUBLIC

IBM is making its quantum computer API available to the public

By Jessica Hall on March 6, 2017 at 9:22 am | 3 Comments

# Quantum algorithms breakthroughs

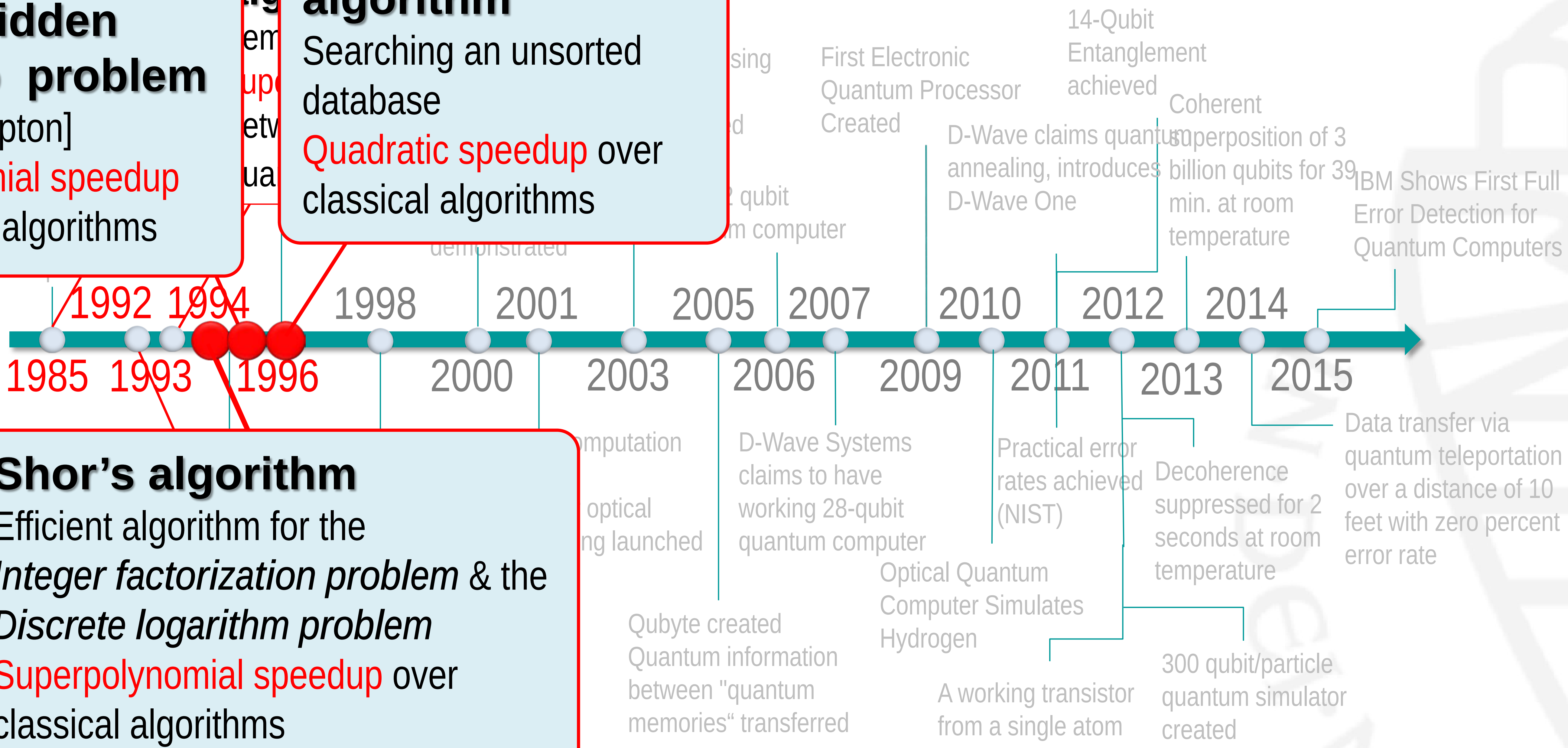


# Quantum algorithms breakthroughs

**Abelian hidden subgroup problem**
[Boneh and Lipton]
Superpolynomial speedup over classical algorithms

**Grover's algorithm**
Searching an unsorted database
Quadratic speedup over classical algorithms

**Shor's algorithm**
Efficient algorithm for the *Integer factorization problem* & the *Discrete logarithm problem*
Superpolynomial speedup over classical algorithms



# Quantum algorithms breakthroughs

**Abelian hidden subgroup problem**
[Boneh and Lipton]
Superpolynomial speedup over classical algorithms

**Grover's algorithm**
Searching an unsorted database
Quadratic speedup over classical algorithms

**Breakthrough Algorithms**

- Not only examples, but of critical practical value
- Contemporary security relies on these problems
- Start the era of broader interest in quantum computing and quantum technology

**Shor's algorithm**
Efficient algorithm for the *Integer factorization problem* & the *Discrete logarithm problem*
Superpolynomial speedup over classical algorithms

# Today’s cryptography in use?

## Algorithms we use:

- *RSA encryption scheme*
- *ElGamal encryption/signature schemes*
  - *DSA – digital signature*
- *Diffie-Hellman (DH) key exchange*
  - *MQV key agreement*
- *Elliptic curve cryptography*
  - *ECDSA, EdDSA*
  - *ECDH, ECMQV*
- *Pairing based cryptography*
  - *Tripartite Key exchange*
  - *Identity based encryption / signatures / key exchange*
  - *Attribute based encryption*

# Today's cryptography in use?

**Algorithms we use:**

- *RSA encryption scheme*
- *ElGamal encryption/signature schemes*
  - *DSA – digital signature*
- *Diffie-Hellman (DH) key exchange*
  - *MQV key agreement*
- *Elliptic curve cryptography*
  - *ECDSA, EdDSA*
  - *ECDH, ECMQV*
- *Pairing based cryptography*
  - *Tripartite Key exchange*
  - *Identity based encryption / signatures / key exchange*
  - *Attribute based encryption*

**Practically implemented in:**

- *PKI / PGP /*
- *Cryptographic protocols*
  - *SSL/TLS (HTTPS, FTPS)*
  - *SSH (SFTP, SCP)*
  - *IPsec (IKE)*
  - *IEEE 802.11*
  - *……*
  - *Commitments, Zero Knowledge*
  - *Electronic voting*
  - *Digital cash/credentials*
  - *Multiparty computation*
  - *……*

# Today’s cryptography in use?

***Broken by Quantum Algorithms for the Hidden subgroup problem***

| Algorithm | Key Length | Effective Key Strength / Security Level | |
|---|---|---|---|
| | | Conventional Computing | Quantum Computing |
| RSA-1024 | 1024 bits | 80 bits | 0 bits |
| RSA-2048 | 2048 bits | 112 bits | 0 bits |
| ECC-256 | 256 bits | 128 bits | 0 bits |
| ECC-384 | 384 bits | 256 bits | 0 bits |

Effective key strength for conventional computing derived from NIST SP 800-57 “Recommendation for Key Management”

# Today’s cryptography in use?

*Influenced by Search and collision (Grover – like) Algorithms*

**Doubling of key size**
**(Search algorithm)**

- *Block ciphers*
  - *AES, IDEA, Blowfish, GOST…*
- *Stream ciphers*
  - *CryptMT, Salsa20, Trivium, Edon80…*
- *Hash functions (preimages)*
  - *SHA-1, SHA-2, SHA-3*
  - *Hash based signatures*
- *(All symmetric key primitives)*
  - *MACs, HMACs, PRNGs, AE ciphers…*
- *Primitives based on NP-hard problems*
  - *Code-based, Lattice-based, Multivariate systems*

# Today's cryptography in use?

*Influenced by Search and collision (Grover – like) Algorithms*

**Doubling of key size (Search algorithm)**

- *Block ciphers*
  - *AES, IDEA, Blowfish, GOST…*
- *Stream ciphers*
  - *CryptMT, Salsa20, Trivium, Edon80…*
- *Hash functions (preimages)*
  - *SHA-1, SHA-2, SHA-3*
  - *Hash based signatures*
- *(All symmetric key primitives)*
  - *MACs, HMACs, PRNGs, AE ciphers…*
- *Primitives based on NP-hard problems*
  - *Code-based, Lattice-based, Multivariate systems*

**Birthday bound $\sqrt{N} \rightarrow \sqrt[3]{N}$**

**Collision algorithm:**

- *Hash functions (collisions)*
- *Primitives based on NP-hard problems*
  - *Generalized birthday attacks (Information Set Decoding) on Code-based/Lattice-based cryptosystems*

# Today's cryptography in use?

***Not trivial, but manageable!***

| Algorithm | Key Length | Effective Key Strength / Security Level | |
|---|---|---|---|
| | | Conventional | Quantum |
| AES-128 | 128 bits | 128 bits | 64 bits |
| AES-256 | 256 bits | 256 bits | 128 bits |

| Algorithm | Security Level | |
|---|---|---|
| | Conventional (Preimage/Collisions) | Quantum (Preimage/Collisions) |
| SHA-256 | 256/128 bits | 128/85 bits |
| SHA-512 | 512/256 bits | 256/170 bits |

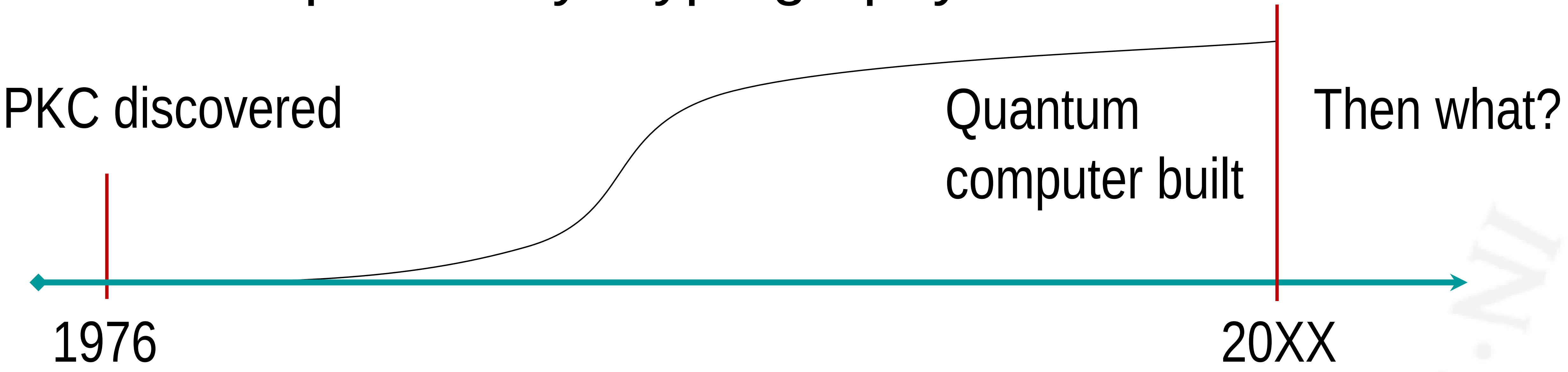
Effective key strength for conventional computing derived from NIST SP 800-57 "Recommendation for Key Management"

It's **rather unlikely** that (under the assumption that they are ever built) **quantum computers will kill ALL classical cryptography**…

…At least not symmetric cryptography!

It's **rather unlikely** that (under the assumption that they are ever built) **quantum computers will kill ALL classical cryptography**…
…At least not symmetric cryptography!

What about public key cryptography?



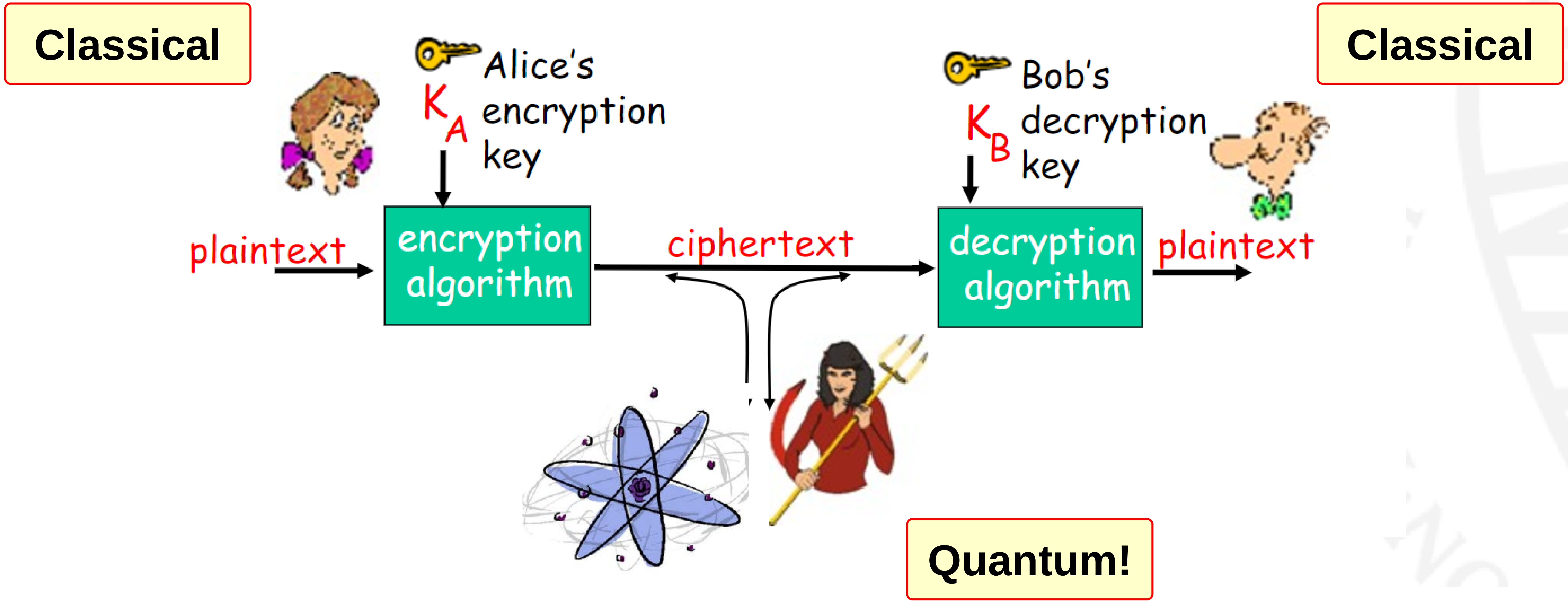

Will we need **quantum cryptography?**

**Or**

Is it possible to have **strong classical cryptography** in the quantum world?

# Post Quantum Cryptography

Cryptosystems believed to be secure against quantum computer attacks



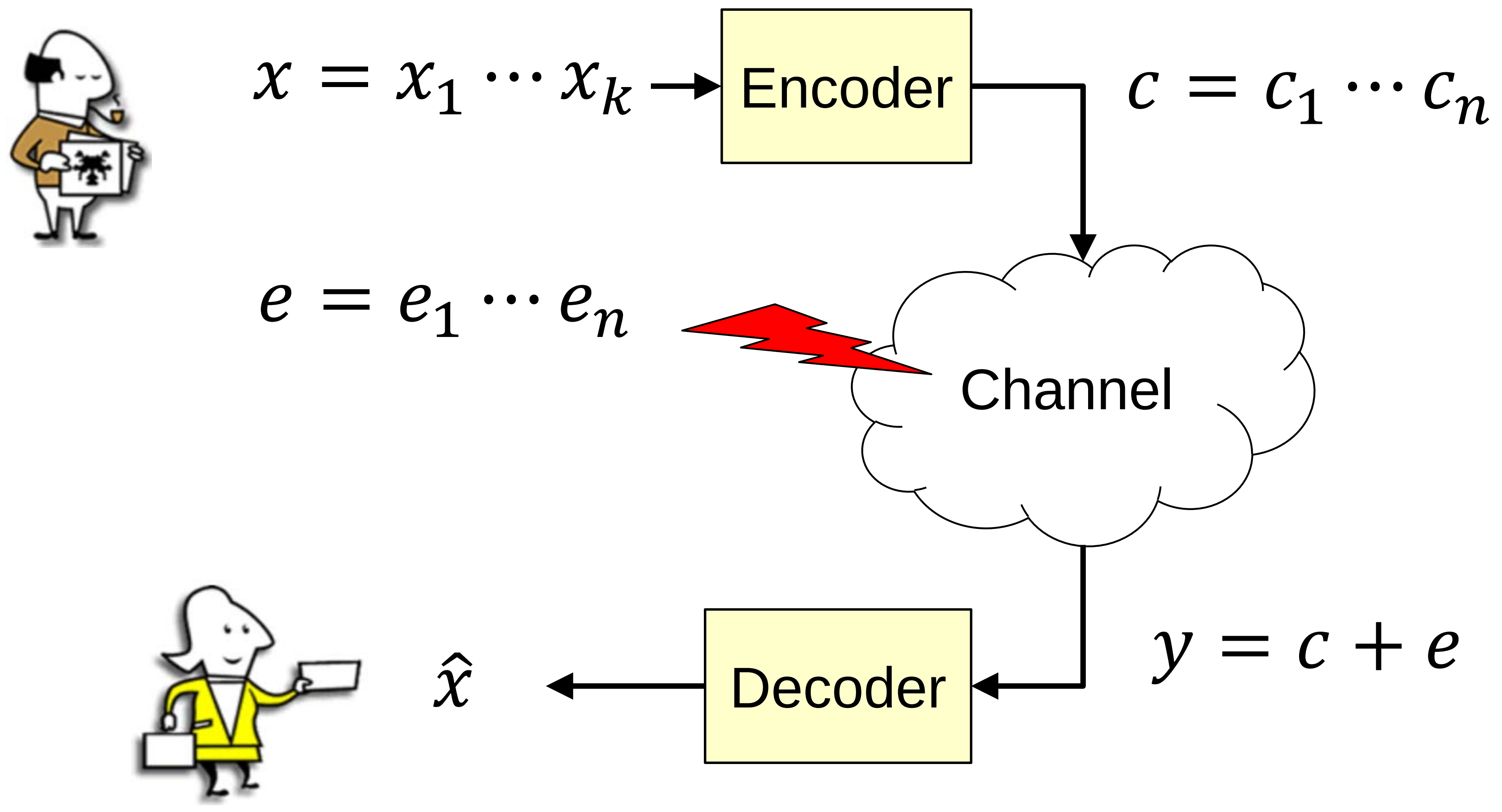
# Post Quantum Cryptography

**Cryptosystems believed to be secure against quantum computer attacks**

- ***Code-based systems*** (Syndrome decoding)
  - Encryption
- ***Multivariate Quadratic systems*** (Polynomial system solving - MQ)
  - Signatures
- ***Lattice-based systems*** (Hard problems on lattices – LWE, SVP)
  - Encryption, signatures, key agreement
- ***Hash-based systems*** (Hash functions)
  - Signatures
- ***Isogeny based systems*** (isogenies on supersingular elliptic curves)
  - Key agreement

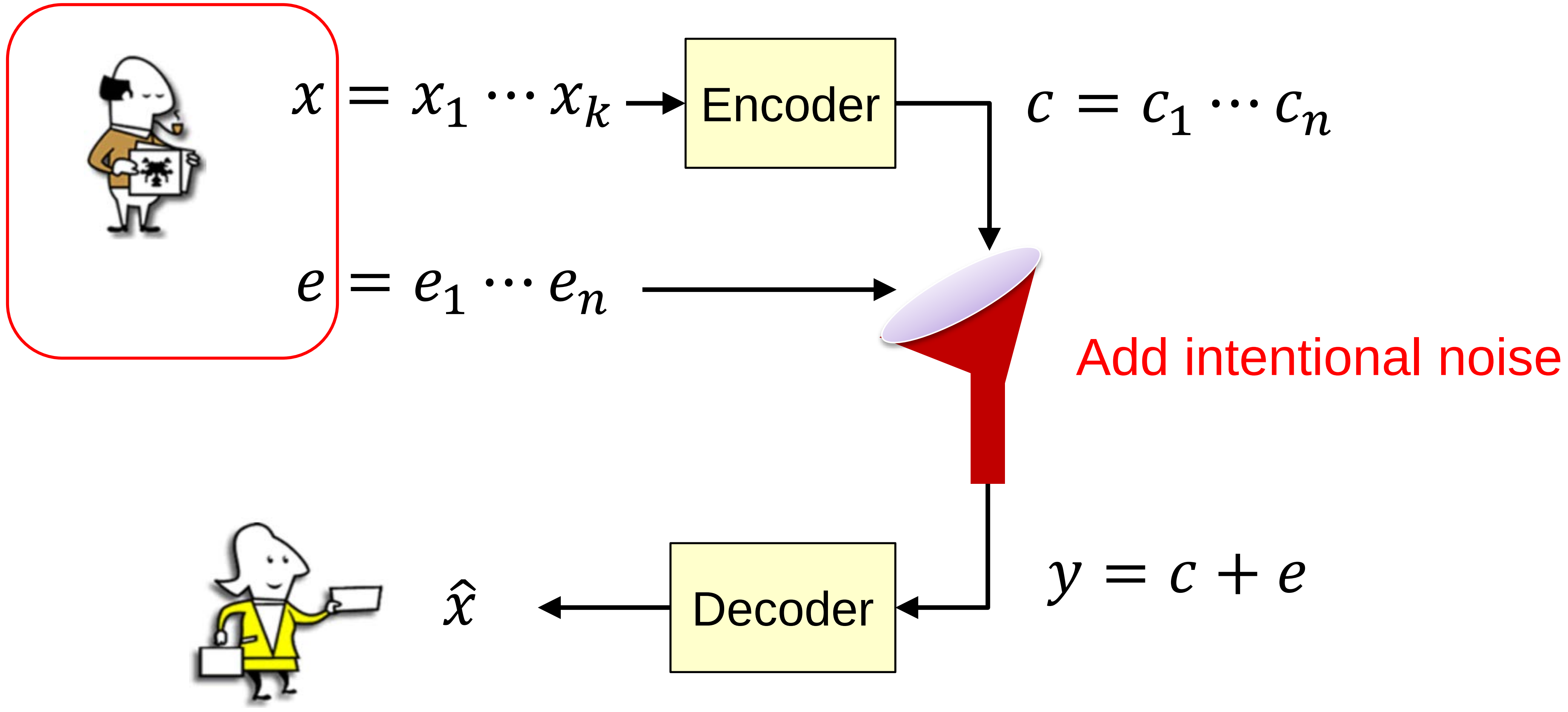
# Code-based Cryptosystems

- Coding theory essentials
- Noisy channel communication:



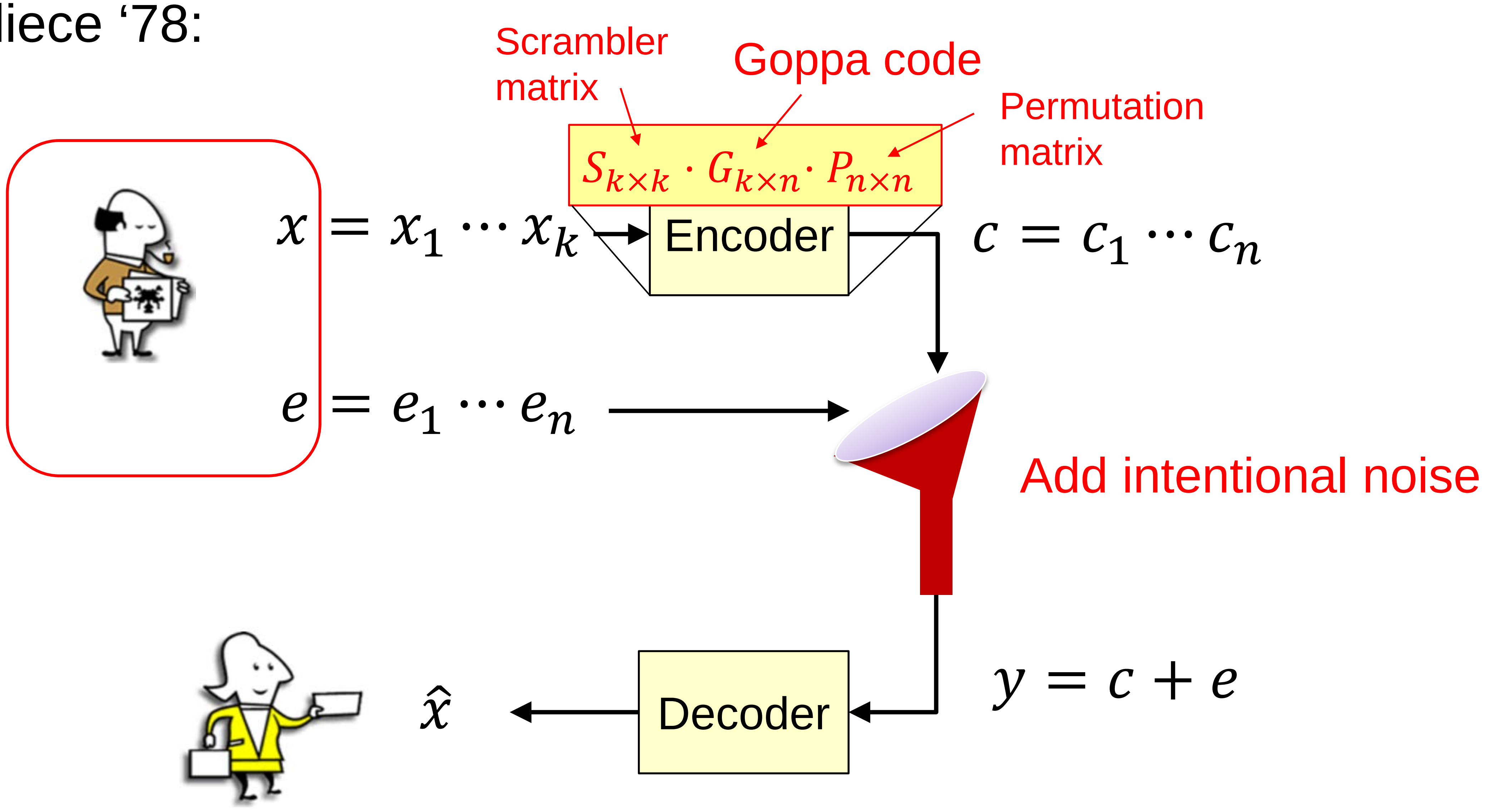
# Code-based Cryptosystems

- Coding theory essentials
- In cryptography:



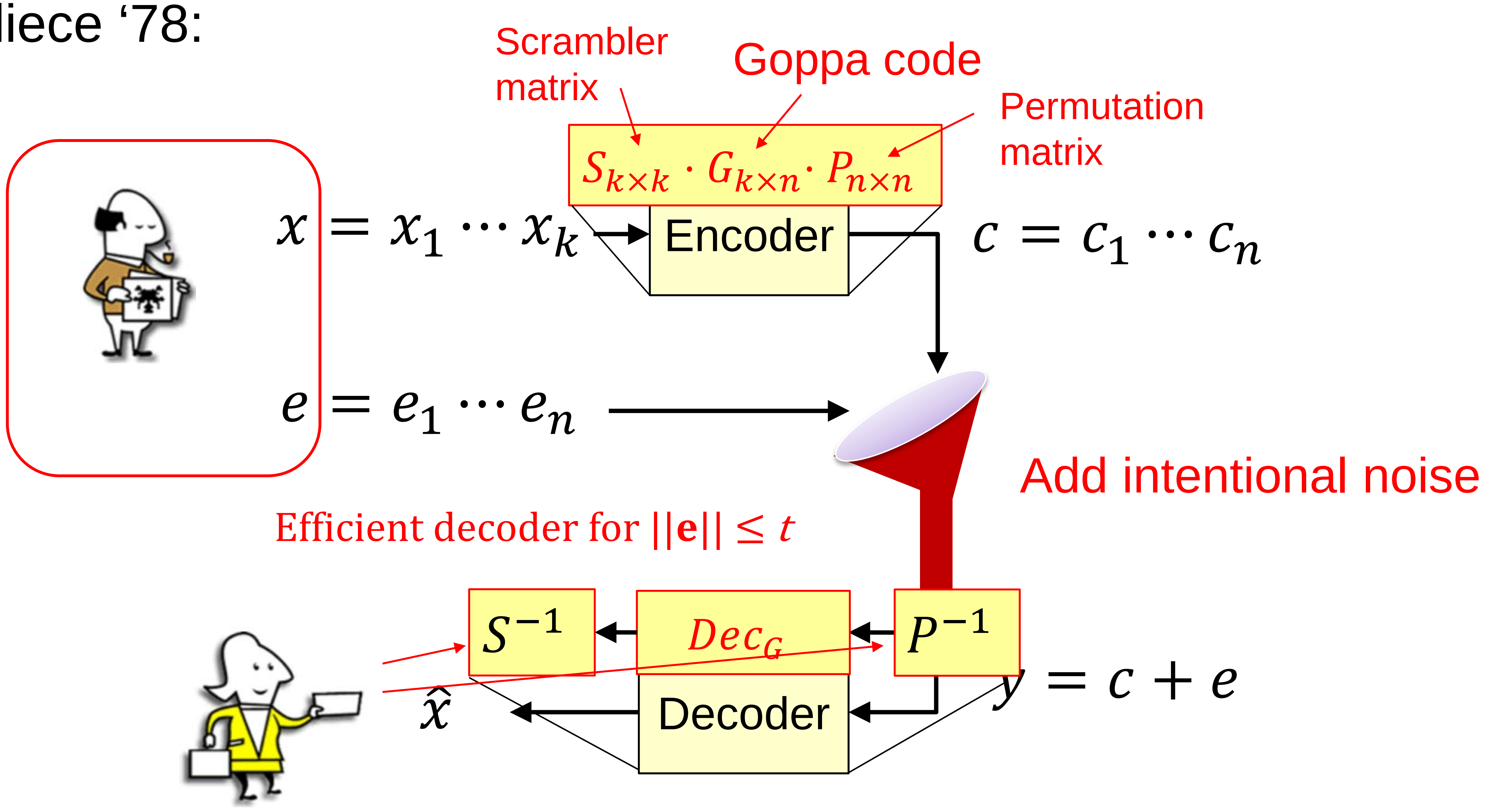
# Code-based Cryptosystems

- Hard underlying problem (NP hard): Decoding random linear codes
- No reduction to the hard problem – instead, related problems believed to be hard
- Confidence in encryption schemes
- McEliece '78:



# Code-based Cryptosystems

- Hard underlying problem (NP hard): Decoding random linear codes
- No reduction to the hard problem – instead, related problems believed to be hard
- Confidence in encryption schemes
- McEliece '78:



# Code-based Cryptosystems - Parameters

- McEliece '78 and dual system Niederreiter [Becker, Joux, May, & Meurer, 12] [Bernstein, 09], Implementation McBits [Bernstein, Chou, & Schwabe, 13]

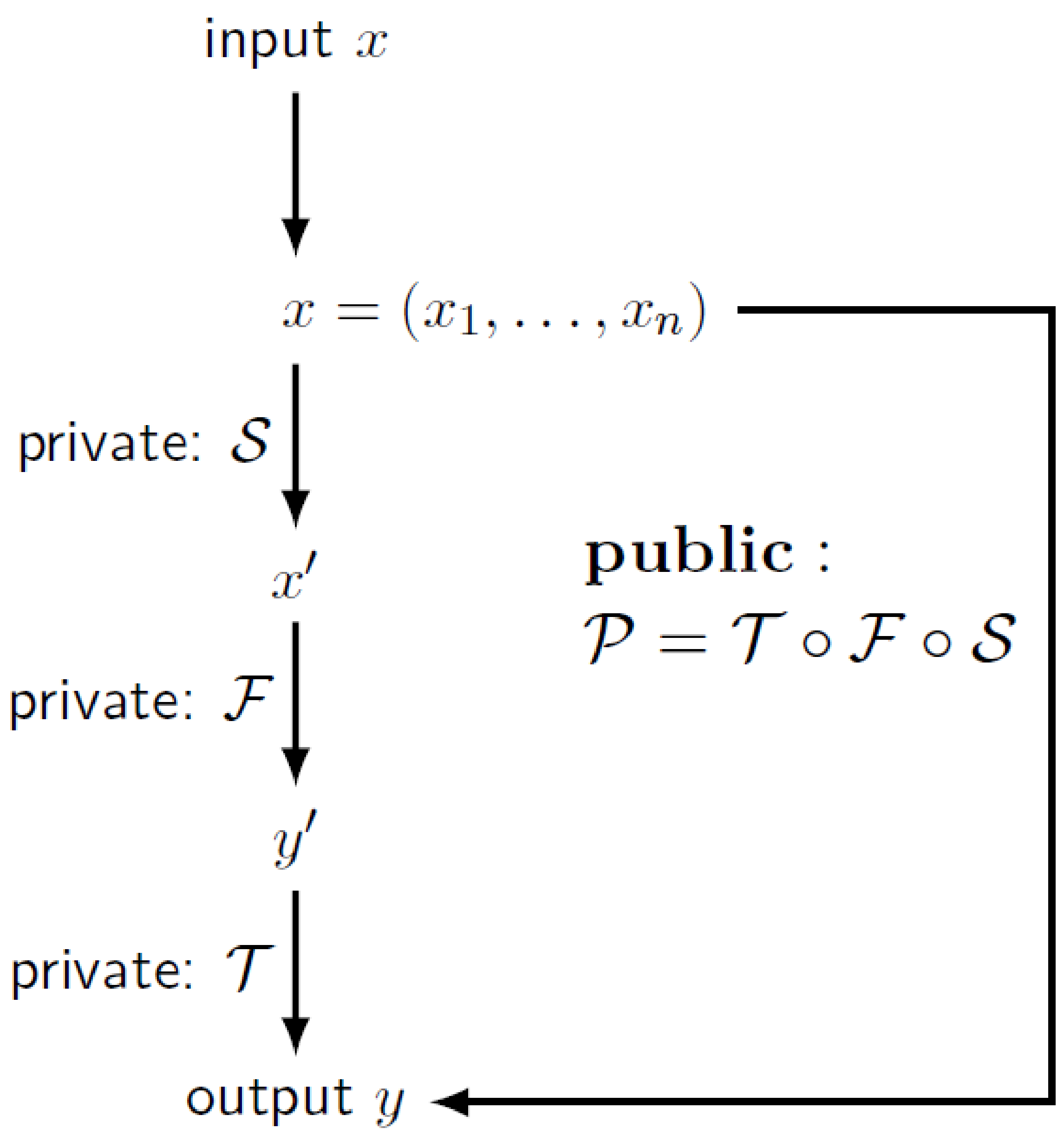
| $m, t$ | McEliece | | Niederreiter | | Key size | Classical security | PQ Security* | Decoding (cycles) |
|---|---|---|---|---|---|---|---|---|
| | Cipher | Message | Cipher | Message | | | | |
| 10, 50 | 1024 | 524 | 500 | 284 | 32 KB | 52 | 52 | |
| 11,40 | 2048 | 1608 | 440 | 280 | 88 KB | 81 | 75 | 29.4 K |
| 12,50 | 4096 | 3496 | 600 | 385 | 277 KB | 120 | 105 | |

- QC-MDPC [Misoczki, Tillich, Sendrier, & Barreto, 13], Rank-Metric codes [Loidreau, 17]

| | $(n, k, t)$ | Cipher | Message | Key size | Security |
|---|---|---|---|---|---|
| QC-MDPC | (9602, 4801, 84) | 9602 | 4801 | 4801 | 80 (classical) |
| QC-MDPC | (19714, 9857,134) | 19714 | 9857 | 9857 | 128 (classical) |
| Loidreau | (64,40,4) | 6144 | 3840 | 11500 | 140 (cl) 80 (pq) |
| Loidreau | (120,80,4) | 15360 | 10240 | 51000 | 260 (cl) 140 (pq) |

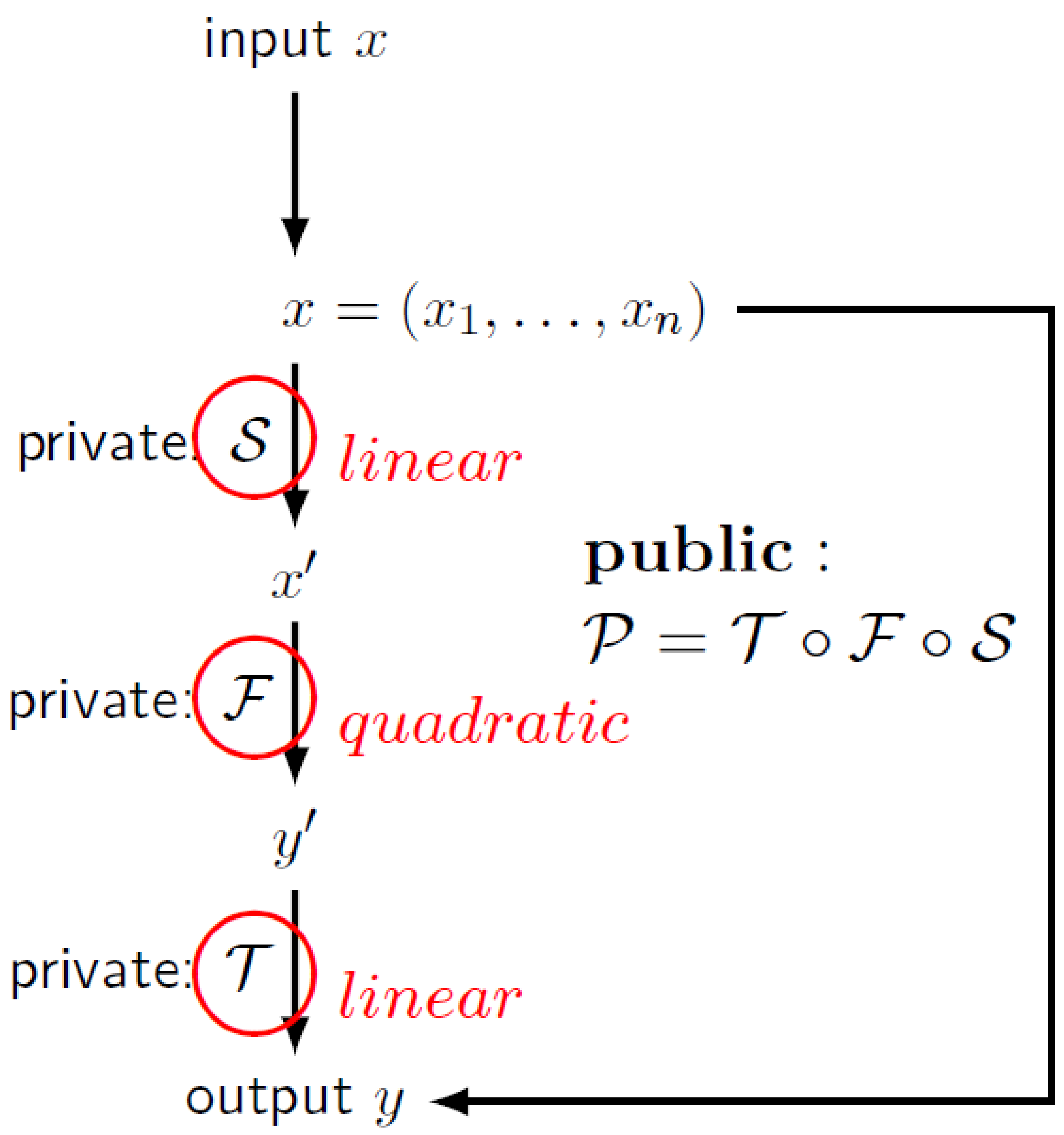
# *MQ* (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures



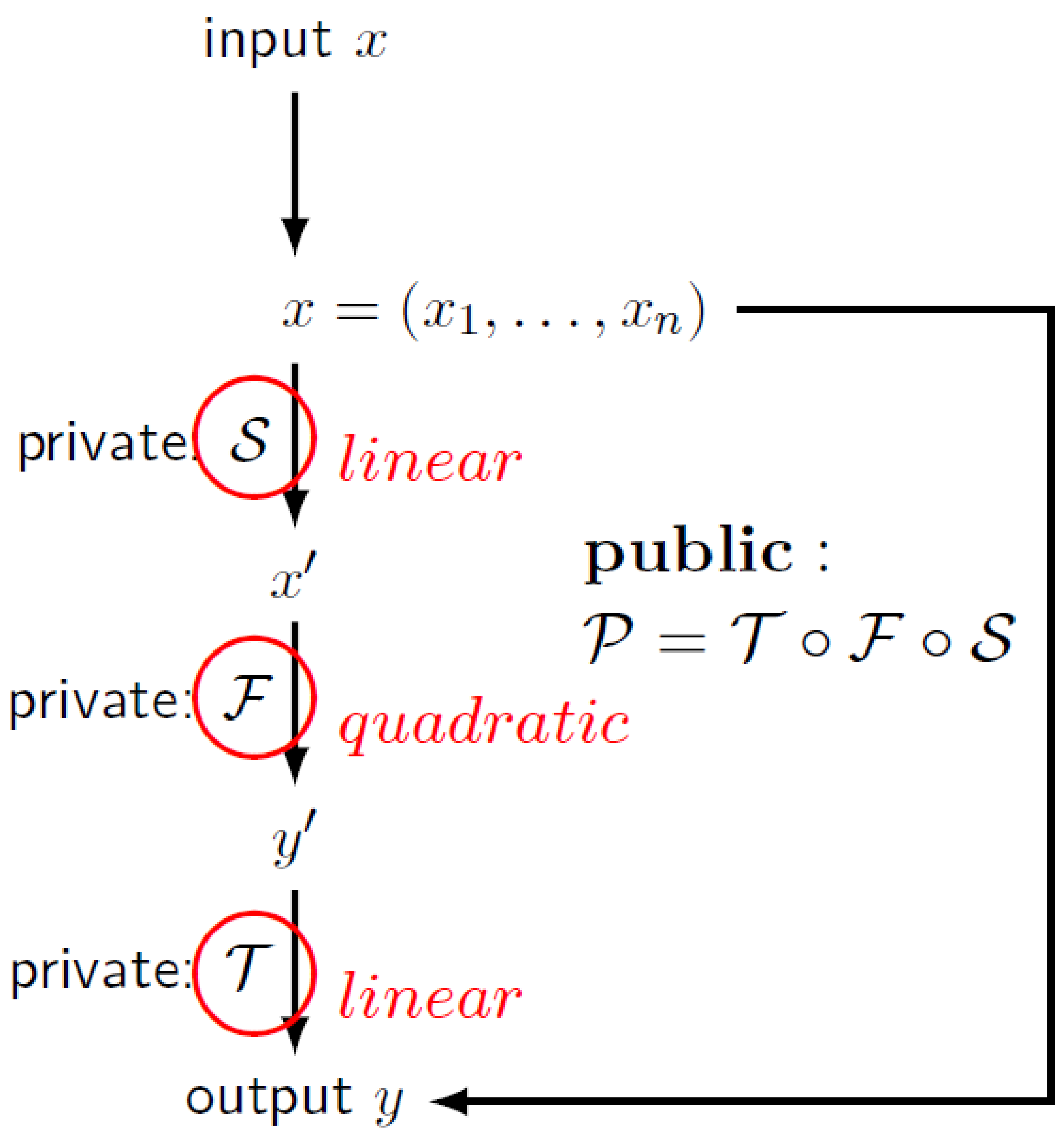
# *MQ* (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures



# *MQ* (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures

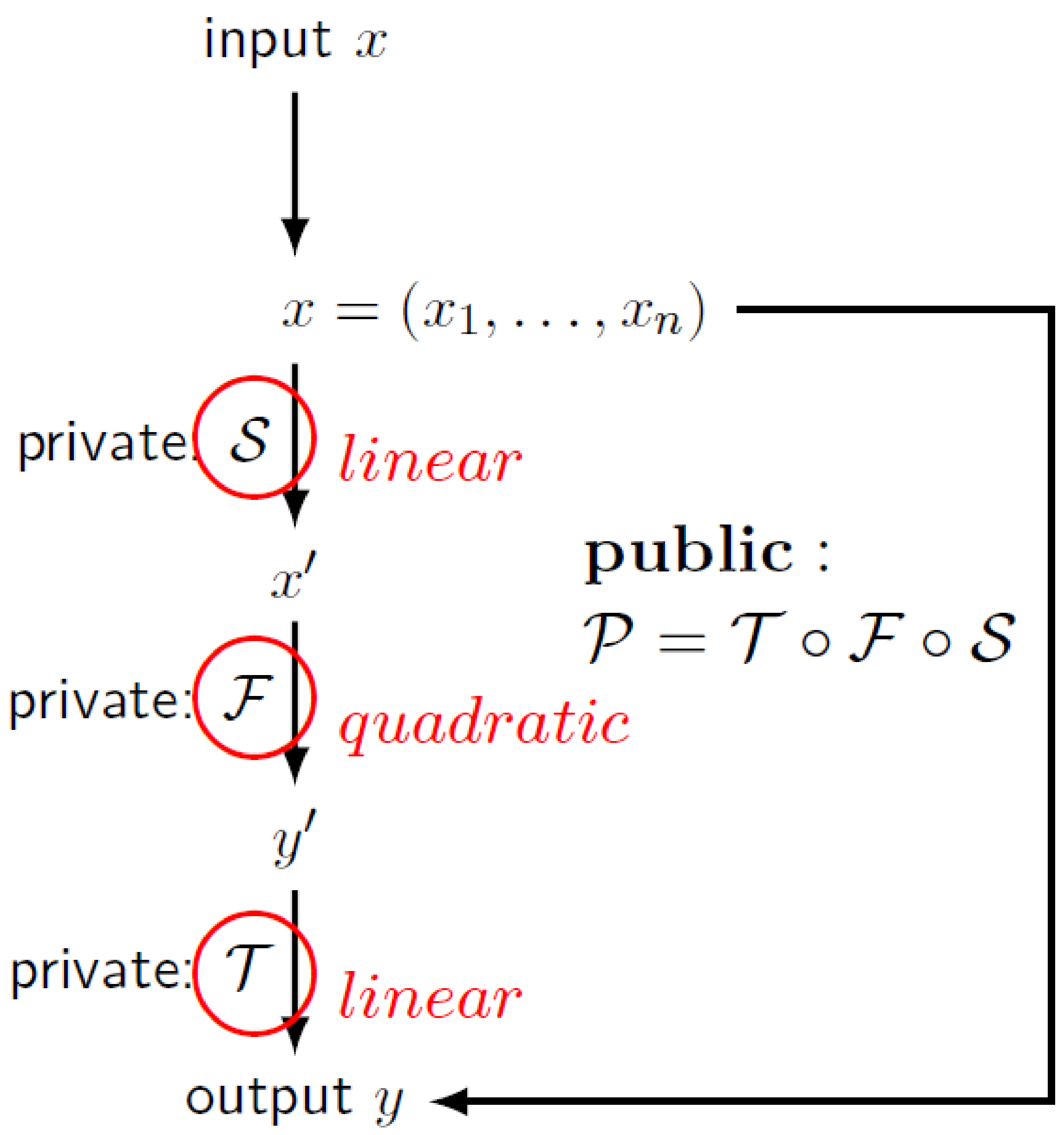



Public $\mathcal{P}$

$$p_1(x_1, \ldots, x_n)$$
$$p_2(x_1, \ldots, x_n)$$
$$\ldots$$
$$p_m(x_1, \ldots, x_n)$$

# *MQ* (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)
- (**Mainstream**) No reduction to the hard problem – related problems believed to be hard
- Confidence in signatures




$\mathcal{PoSSo}$:

**Input:**

$p_1, p_2, \dots, p_m \in \mathbb{F}_q[x_1, \dots, x_n]$

**Question:**

Find - if any - $(u_1, \dots, u_n) \in \mathbb{F}_q^n$ st.

$$\begin{cases} p_1(u_1, \dots, u_n) = 0 \\ p_2(u_1, \dots, u_n) = 0 \\ \dots \\ p_m(u_1, \dots, u_n) = 0 \end{cases}$$

# *MQ* (multivariate quadratic) Cryptosystems

- Fast, simple operations, short signatures
- Large keys, no security proofs
- Parameters for Gui [Petzoldt, Chen, Yang, Tao, Ding, 15], Rainbow [Ding, Schmidt, 04]
- Implementation [Chen, Li, Peng, Yang, Cheng, 17]

| Security (post quantum) | Signature scheme | Public key (kB) | Private key (kB) | Signature size (bit) | Sign() k cycles | Verify() k cycles |
|---|---|---|---|---|---|---|
| 80 | Gui(GF(2),120,9,3,3,2) | 110.7 | 3.8 | 129 | | |
| 100 | Gui(GF(2),161,9,6,7,2) | 271.8 | 7.5 | 181 | | |
| 128 | GUI(4,120,17,8,8,2) | 225.8 | 9.6 | 288 | 7,992.8 | 342.5 |
| 80 | Rainbow(GF(256),19,12,13) | 25.3 | 19.3 | 352 | | |
| 100 | Rainbow(GF(16),25,25,25) | 65.9 | 43.2 | 288 | | |
| 128 | Rainbow(GF(31),28,28,28) | 123.2 | 74.5 | 420 | 77.4 | 70.8 |

# *MQ* (multivariate quadratic) Cryptosystems

- Hard underlying problem (NP hard): Polynomial system solving (PoSSo)

**Two new provably secure signatures**

- **MQDSS** [Chen, Hülsing, Rijneveld, S, Schwabe, 16] – security proof in the ROM
- **Sofia** [Chen, Hülsing, Rijneveld, S, Schwabe, 17] – security proof in the Quantum ROM

| Security (post quantum) | Signature scheme | Public key (B) | Private key (B) | Signature size (KB) | Sign() k cycles | Verify() k cycles |
|---|---|---|---|---|---|---|
| 128 (ROM) | MQDSS-31-64 | 72 | 64 | 40 | 8,510.6 | 5,752.6 |
| 128 (QROM) | Sofia-4-128 | 64 | 32 | 123 | 21,305.5 | 15,492.6 |

- Transform from provably secure Identification schemes

# MQDSS

IDS

| $\mathcal{P}$ | | $\mathcal{V}$ |
|---|---|---|
| $\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$ | $\xrightarrow{\mathsf{com}}$ | |
| | $\xleftarrow{\mathsf{ch}_1}$ | $\mathsf{ch}_1 \leftarrow_R \mathsf{ChS}_1(1^k)$ |
| $\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$ | $\xrightarrow{\mathsf{resp}_1}$ | |
| | $\xleftarrow{\mathsf{ch}_2}$ | $\mathsf{ch}_2 \leftarrow_R \mathsf{ChS}_2(1^k)$ |
| $\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$ | $\xrightarrow{\mathsf{resp}_2}$ | |
| | | $b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$ |

↓ ↓ ↓

FS signature

| Signer | Verifier |
|---|---|
| $\mathsf{com} \leftarrow \mathcal{P}_0(\mathsf{sk})$ | |
| $\mathsf{ch}_1 \leftarrow H_1(m, \mathsf{com})$ | $\mathsf{ch}_1 \leftarrow H_1(m, \mathsf{com})$ |
| $\mathsf{resp}_1 \leftarrow \mathcal{P}_1(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1)$ | $\mathsf{ch}_2 \leftarrow H_2(m, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1)$ |
| $\mathsf{ch}_2 \leftarrow H_2(m, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1)$ | $b \leftarrow \mathsf{Vf}(\mathsf{pk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2, \mathsf{resp}_2)$ |
| $\mathsf{resp}_2 \leftarrow \mathcal{P}_2(\mathsf{sk}, \mathsf{com}, \mathsf{ch}_1, \mathsf{resp}_1, \mathsf{ch}_2)$ | |
| **output** : $\sigma = (\mathsf{com}, \mathsf{resp}_1, \mathsf{resp}_2)$ | **output** : $b$ |

# Lattice-based Cryptosystems

- Encryption, signatures, key exchange
- Many different hard problems

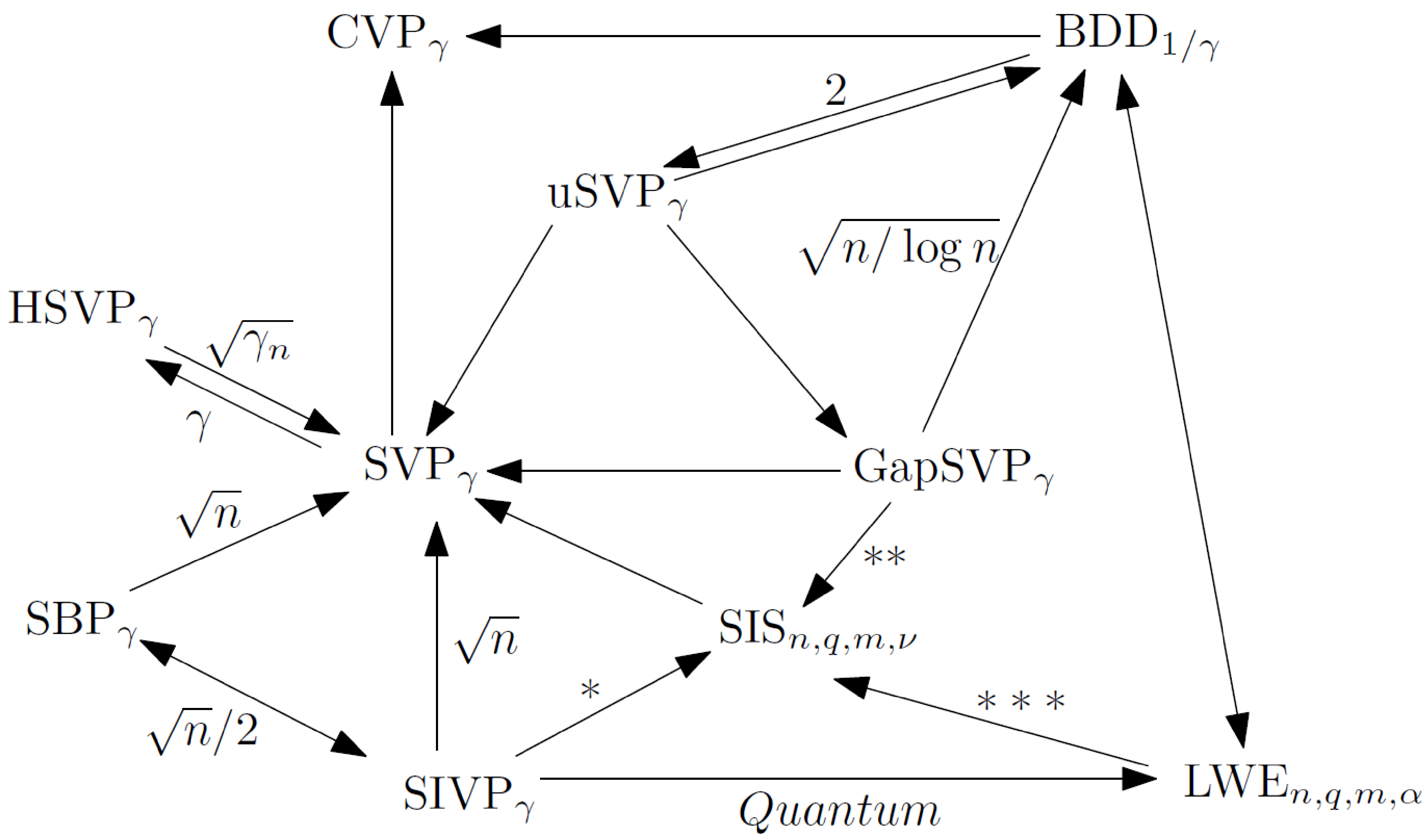



Fig. from Joop van de Pol's MSc-thesis

# Lattice-based Cryptosystems

- Learning with errors (LWE)
- Variants R-LWE, Module-LWE, LPN, …
  - Additional structure undermines security claims

- Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Let $\chi$ be an *error distribution* on $\mathcal{R}_q$
- Let $\mathbf{s} \in \mathcal{R}_q$ be secret
- Attacker is given pairs $(\mathbf{a}, \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a}$ uniformly random from $\mathcal{R}_q$
  - $\mathbf{e}$ sampled from $\chi$
- Task for the attacker: find $\mathbf{s}$
- Common choice for $\chi$: discrete Gaussian

# Lattice-based Cryptosystems

- Learning with errors (LWE)
- Variants R-LWE, Module-LWE, LPN, …
  - Additional structure undermines security claims

- Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Let $\chi$ be an *error distribution* on $\mathcal{R}_q$
- Let $\mathbf{s} \in \mathcal{R}_q$ be secret
- Attacker is given pairs $(\mathbf{a}, \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a}$ uniformly random from $\mathcal{R}_q$
  - $\mathbf{e}$ sampled from $\chi$
- Task for the attacker: find $\mathbf{s}$
- Common choice for $\chi$: discrete Gaussian

| Alice (server) | | Bob (client) |
| --- | --- | --- |
| $\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$ | | $\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$ |
| $\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$ | $\xrightarrow{\mathbf{b}}$ | $\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$ |
| | $\xleftarrow{\mathbf{u}}$ | |

| | | | |
| --- | --- | --- | --- |
| Alice has | $\mathbf{v}$ | $= \mathbf{us}$ | $= \mathbf{ass}' + \mathbf{e}'\mathbf{s}$ |
| Bob has | $\mathbf{v}'$ | $= \mathbf{bs}'$ | $= \mathbf{ass}' + \mathbf{es}'$ |

# Lattice-based Cryptosystems

- Learning with errors (LWE)
- Variants R-LWE, Module-LWE, LPN, …
  - Additional structure undermines security claims

- Let $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- Let $\chi$ be an *error distribution* on $\mathcal{R}_q$
- Let $\mathbf{s} \in \mathcal{R}_q$ be secret
- Attacker is given pairs $(\mathbf{a}, \mathbf{as} + \mathbf{e})$ with
  - $\mathbf{a}$ uniformly random from $\mathcal{R}_q$
  - $\mathbf{e}$ sampled from $\chi$
- Task for the attacker: find $\mathbf{s}$
- Common choice for $\chi$: discrete Gaussian

| Alice (server) | | Bob (client) |
|---|---|---|
| $\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$ | | $\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$ |
| $\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$ | $\xrightarrow{\mathbf{b}}$ | $\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$ |
| | $\xleftarrow{\mathbf{u}}$ | |

Alice has $\mathbf{v} = \mathbf{us} = \mathbf{ass}' + \mathbf{e}'\mathbf{s}$

Bob has $\mathbf{v}' = \mathbf{bs}' = \mathbf{ass}' + \mathbf{es}'$

approximately same (v, v′)

small (e′s, es′)

# Lattice-based Cryptosystems

- FRODO [Bos, Costello, Ducas, Mironov, Naehrig, Nikolaenko, Raghunathan, Stebila, 16]
- NewHope [Alkim, Ducas, Pöppelmann, Schwabe, 16]
  - Google Experiment for Chrome 2016: New hope + X25519 used in Chrome Canary for access to some Google services
- NTRU Prime [Bernstein, Chuengsatiansup, Lange, van Vredendaal, 16]
- Kyber [Bos, Ducas, Kiltz, Lepoint, Lyubashevsky, Schanck, Schwabe, Stehlé, 17]

| Scheme | Security bits/(type) | Hard problem | KeyGen (cycles) | Enc (cycles) | Dec (cycles) | Public key (bytes) | Private key (bytes) | Ciphertext (bytes) |
|---|---|---|---|---|---|---|---|---|
| FRODO | 130 (pass.) | LWE | 2 938 K | 3 484 K | 338 K | 11 296 | 11280 | 11288 |
| NewHope | 255 (pass.) | Ring-LWE | 88 920 | 110 986 | 19 422 | 1824 | 1792 | 2048 |
| NTRU Prime | 129 (CCA) | NTRU like | | > 51488 | | 1232 | 1417 | 1141 |
| Kyber | 161 (CCA) | Module-LWE | 77 892 | 119 652 | 125 736 | 1088 | 2400 | 1184 |

# Hash-based Signatures



- **Only secure hash function needed** (security well understood, standard model proof)
- Merkle, 89

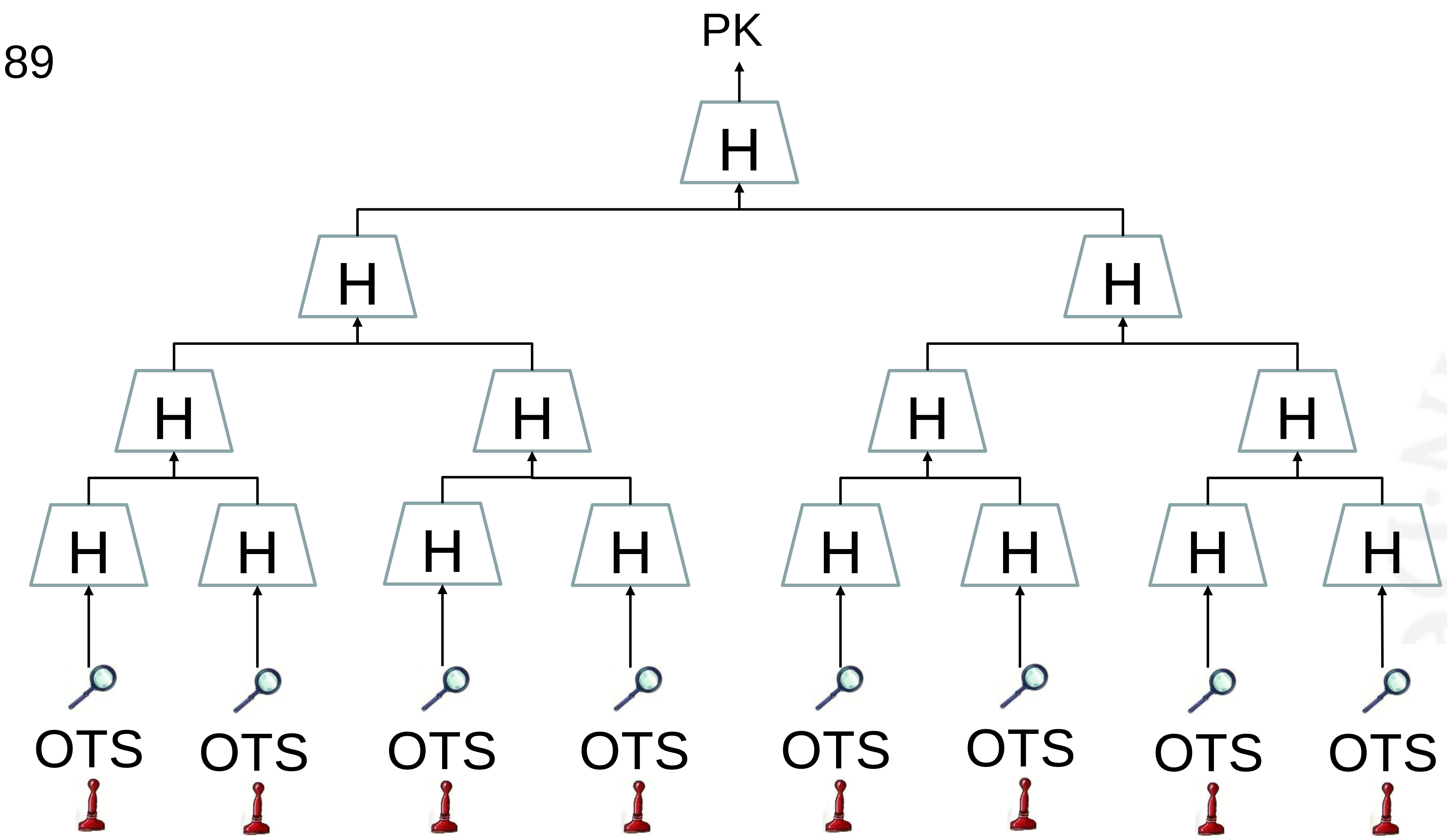



Figure: Andreas Hülsing

# Hash-based Signatures

- **Only secure hash function needed** (security well understood, standard model proof)
- Merkle, 89

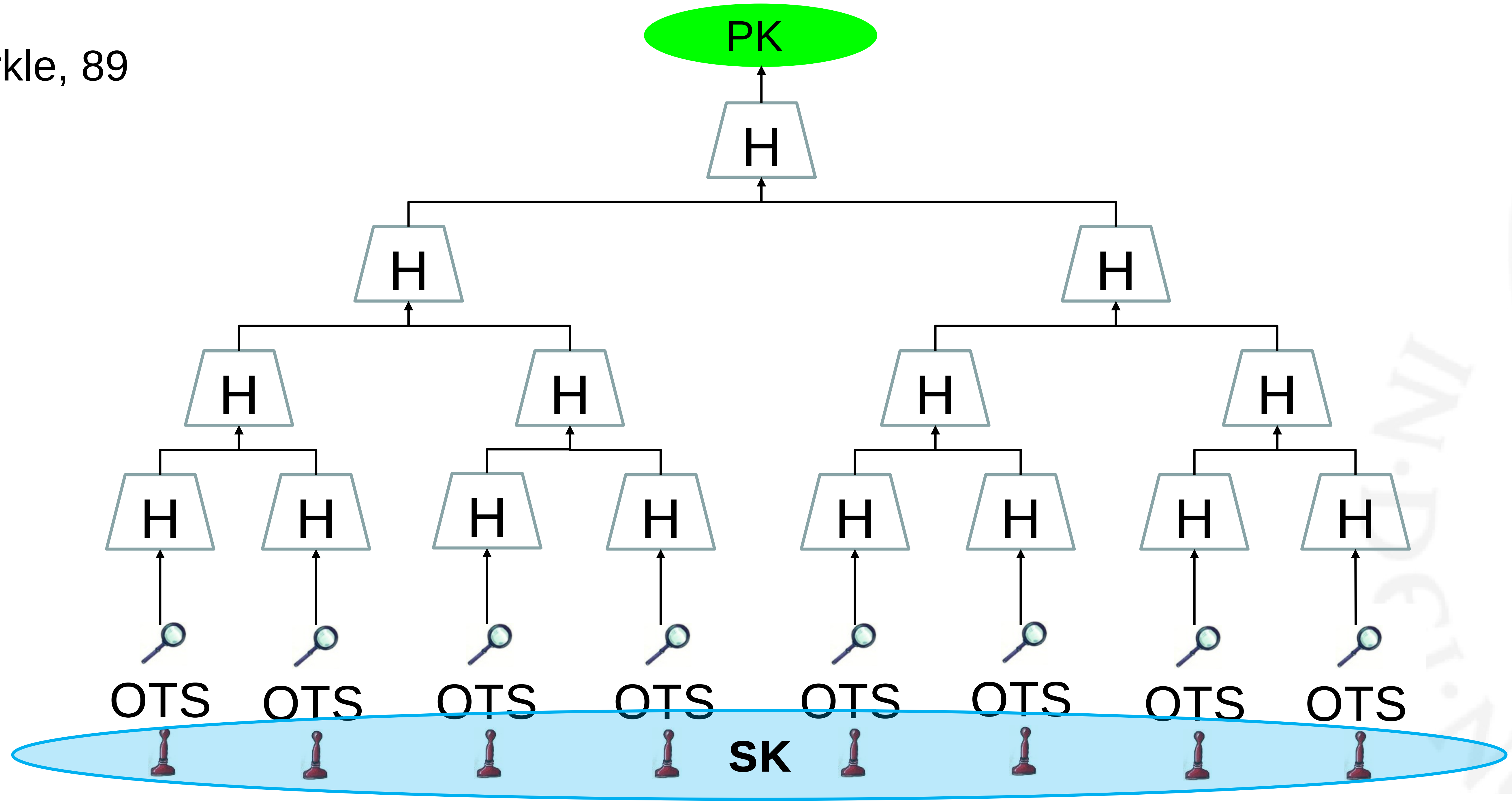



Figure: Andreas Hülsing

# Hash-based Signatures

- **Only secure hash function needed** (security well understood, standard model proof)
- Merkle, 89

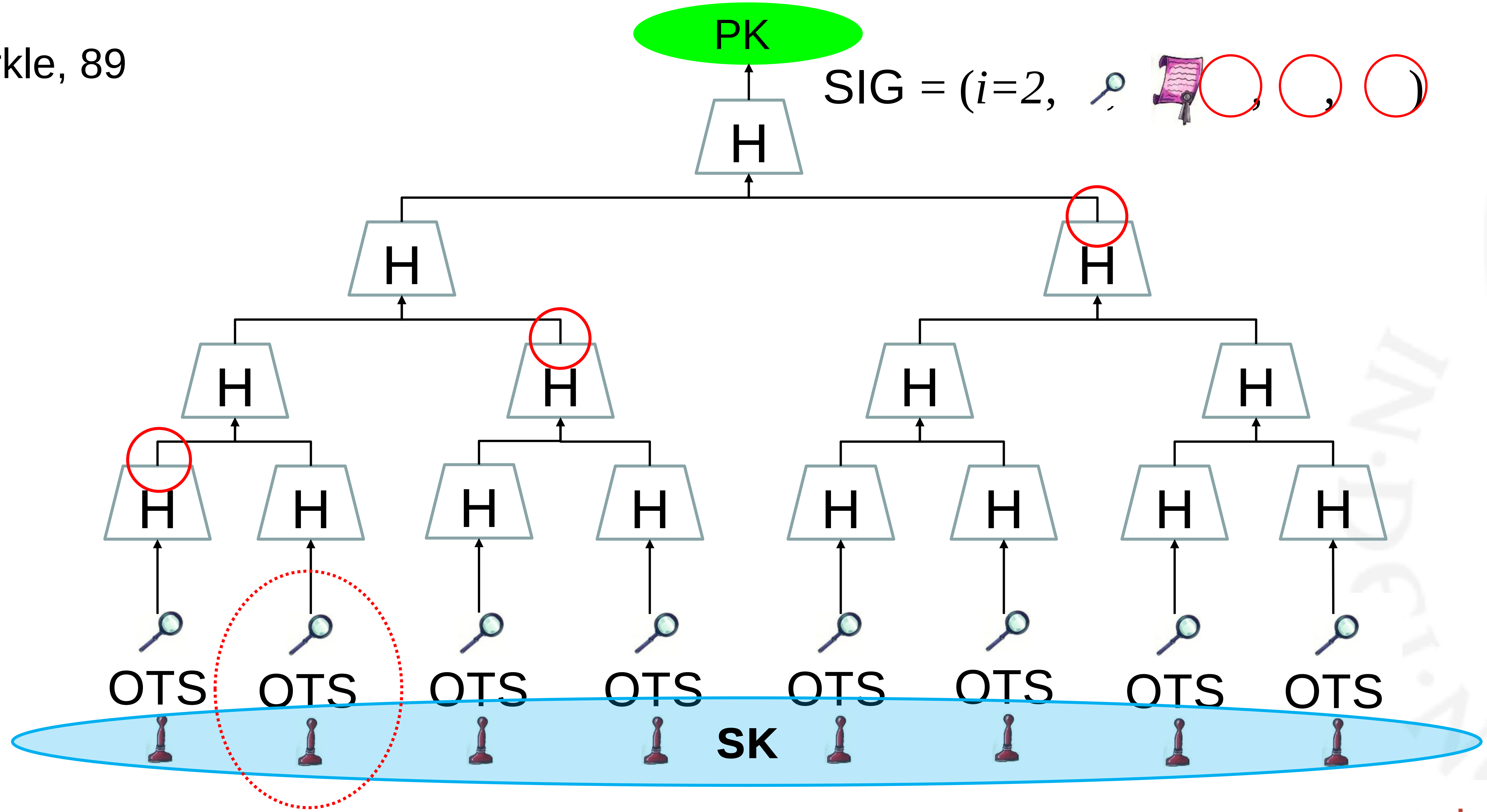



Figure: Andreas Hülsing

# Hash-based Signatures

- Most trusted post quantum signatures
- Two Internet drafts (drafts for RFCs), one in „waiting for ISRG review“

- XMSS – stateful, but forward secrecy [Buchmann, Dahmen, Hülsing, 11]
- SPHINCS – stateless [Bernstein, Hopwood, Hülsing, Lange, Niederhagen, Papachristodoulou, Schneider, Schwabe, O’Hearn, 15]

| | Sign (ms) | Verify (ms) | Signature (byte) | Public Key (byte) | Secret Key (byte) | Bit Security |
|---|---|---|---|---|---|---|
| XMSS-SHA-2 | 35.60 | 1.98 | 2084 | 1700 | 3,364 | 157 |
| XMSS-AES-NI | 0.52 | 0.07 | 2452 | 916 | 1,684 | 84 |
| SPHINCS-256 | 13.56 | 0.39 | 41000 | 1056 | 1088 | 128 |

# Challenges in Post Quantum Cryptography

- **Key sizes, signature sizes and speed**
  - Huge public keys, or signatures …. Or slow
  - ex. ECC 256b key vs McElliece 500KB key
  - ex. ECC 80B signature vs MQDSS 40KB signature

- **Software and hardware implementation**
  - Optimizations, physical security

- **Standardization**
  - What is the right choice of algorithm?

- **Deployment**
  - In TLS, DTLS, constrained devices, storage…
  - Will take a long time…

PQCRYPTO
ICT-645622

# Challenges in Post Quantum Cryptography **For the IoT**

- **Key sizes, signature sizes and speed**
  - Huge public keys, or signatures …. Or slow
  - ex. ECC 256b key vs McElliece 500KB key
  - ex. ECC 80B signature vs MQDSS 40KB signature
- **Software and hardware implementation**
  - Optimizations, physical security
- **Standardization**
  - What is the right choice of algorithm?
- **Deployment**
  - In TLS, DTLS, constrained devices, storage…
  - Will take a long time…

PQCRYPTO
ICT-645622

# Post Quantum Crypto for the IoT is not fantasy



## Ready for tomorrow: Infineon demonstrates first post-quantum cryptography on a contactless security chip

May 30, 2017 | Business & Financial Press

Security experts at Infineon's Munich headquarters and the Center of Excellence for contactless technologies in Graz, Austria, made a breakthrough in this field. They implemented a post-quantum key exchange scheme on a commercially available contactless smart card chip. Key exchange schemes are used to establish an encrypted channel between two parties. The deployed algorithm is a variant of "New Hope", a quantum-resistant cryptosystem also › **explored successfully by Google** on a development version of the Chrome browser.

"The phantom of the quantum computer is keeping academia and the IT industry on high alert," said Thomas Pöppelmann from Infineon's Chip Card & Security Division, who has been co-developing the New Hope algorithm. "At Infineon, we are proud to be the first to transfer PQC onto contactless smart cards. Our challenges comprised the small chip size and limited memory capacity to store and execute such a complex algorithm as well as the transaction speed." Thomas Pöppelmann and his co-researchers received the prestigious › **Facebook Internet Defense Prize 2016** for the development of New Hope.

# Post Quantum Crypto for the IoT is not fantasy

- **MQ signatures** - short, fast - traditional choice for constrained devices
- Rainbow hardware implementation [Tang et al., 11]
  - ALTERA Stratix II FPGA
  - Only 198 cycles for signing

- Rainbow impl. [Czypek, Heyse, Thomae, 12]
  - Atmel AVR ATxMega128a1 microchip
  - 32MHz, 8-bit architecture
  - 128KB Flash, 128KB SRAM

| | Sign (s) | Verify (s) | Pub.key | Sig |
|---|---|---|---|---|
| Rainbow(36,21,22) | 0.25 | 0.28 | **136 kB** | 43 B |
| Ed25519* | 1.02 | 0.73 | 32 B | 64 B |

  - * NaCl for AVR microcontrollers http://nacl.cr.yp.to/.

# Post Quantum Crypto for the IoT is not fantasy

- **MQ signatures** - short, fast - traditional choice for constrained devices
- Rainbow hardware implementation [Tang et al., 11]
  - ALTERA Stratix II FPGA
  - Only 198 cycles for signing

- Rainbow impl. [Czypek, Heyse, Thomae, 12]
  - Atmel AVR ATxMega128a1 microchip
  - 32MHz, 8-bit architecture
  - 128KB Flash, 128KB SRAM

| | Sign (s) | Verify (s) | Pub.key | Sig |
|---|---|---|---|---|
| Rainbow(36,21,22) | 0.25 | 0.28 | **136 kB** | 43 B |
| Ed25519* | 1.02 | 0.73 | 32 B | 64 B |

  - * NaCl for AVR microcontrollers http://nacl.cr.yp.to/.

- Armed SPHINCS [Hülsing, Rijneveld, Schwabe, 15]
  - STM32L100C development board
  - ARM Cortex M3, ARMv7-M
  - 32MHz, 32-bit architecture, 16 regs
  - 256KB Flash, 16KB RAM

| | Sign (s) | Verify (s) | Signature | memory |
|---|---|---|---|---|
| $XMSS^{MT}$ | 0.61 | 16 | 28288 | |
| SPHINCS-256 | 18.4 | 0.51 | **41 kB** | 7 kB |

# Timeline

- Fall 2016 – formal Call For Proposals
- Nov 2017 – Deadline for submissions
- 3-5 years – Analysis phase
  - NIST will report its findings
- 2 years later – Draft standards ready

NIST National Institute of Standards and Technology
Information Technology Laboratory

SEARCH: Search

CONTACT SITE MAP

Computer Security Division CSD
Computer Security Resource Center CSRC

CSRC Home | About | Projects / Research | Publications | News & Events

**Post-Quantum Cryptography Project**

Submission Requirements

Minimum Acceptability Requirements

CSRC HOME > GROUPS > CT > POST-QUANTUM CRYPTOGRAPHY PROJECT

## POST-QUANTUM CRYPTO STANDARDIZATION

### Call For Proposals Announcement

The National Institute of Standards and Technology (NIST) has initiated a process to solicit, evaluate, and standardize one or more quantum-resistant public-key cryptographic algorithms. Currently, public-key cryptographic algorithms are specified in FIPS 186-4, Digital Signature Standard, as well as special publications SP 800-56A Revision 2, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography and SP 800-56B Revision 1, Recommendation for Pair-Wises Key-Establishment Schemes Using Integer Factorization Cryptography. However, these algorithms are vulnerable to attacks from large-scale quantum computers (see NISTIR 8105 Report on Post Quantum Cryptography). It is intended that the new public-key cryptography standards will specify one or more additional unclassified, publicly disclosed digital signature, public-key encryption, and key-establishment algorithms that are available worldwide, and are capable of protecting sensitive government information well into the foreseeable future, including after the advent of quantum computers.

# Thank you for listening!

?

*If computers that you build are quantum,*
*Then spies everywhere will all want 'em.*
*Our codes will all fail,*
*And they'll read our email,*
*Till we get crypto that's quantum,*
*and daunt 'em.*

Jennifer and PeterShor

*To read our E-mail, how mean*
*of the spies and their quantum machine;*
*be comforted though,*
*they do not yet know*
*how to factorize twelve or fifteen.*

*Volker Strassen*