Post-Quantum Cryptography

Dr. Thomas Pöppelmann

Infineon Technologies AG

Agenda

1. Introduction
2. Post-Quantum Cryptography
3. Lattice-Based Cryptography
4. Hash- and Code-Based Cryptography
5. Outlook

Agenda

1. Introduction
2. Post-Quantum Cryptography
3. Lattice-Based Cryptography
4. Hash- and Code-Based Cryptography
5. Outlook

Background on quantum computers

IBM's quantum cloud computer goes commercial
http://www.nature.com/news/ibm-s-quantum-cloud-computer-goes-commercial-1.21585

Intel Delivers 17-Qubit Superconducting Chip with Advanced Packaging to QuTech
https://newsroom.intel.com/news/intel-delivers-17-qubit-superconducting-chip-advanced-packaging-qutech/

Quantum computers

- Idea: Use quantum mechanical effects for computation
- Different from classical computers with quantum bits (qubits), quantum gates and some restrictions (no cloning, reversibility)
- Universal quantum computers expected in 15-20 years
  - Current goal is to increase number of stable qubits
  - 2016: 5-qubit computer by IBM
  - 2017: 17-qubit computer by IBM and recently by Intel

Possible specialized applications of quantum computers

- Optimization problems
- Quantum chemistry
- Cryptanalysis

The threat of quantum computers to cryptography

Quantum cryptanalysis on a universal quantum computer

Currently used **asymmetric** cryptosystems (RSA/ECC) breakable by using **Shor's algorithm**

- Classical world (currently): ECC-256 has 128-bit of security
- Quantum world (in 15-20 years): ECC-256 has almost 0-bit of security

Bit-security level for **symmetric** cryptography is halved by **Grover's algorithm**

- Classical world (currently): AES-128 has 128-bit of security
- Quantum world (in 15-20 years): AES-128 has only 64 to 80 bits of security

Quantum world (in 15-20 years)

Heavily affected:
RSA, ECDSA, ECDH

Affected:
AES-128, 3DES

Currently considered appropriately safe:
AES-256, SHA512, SHA3-512

The NSA's view and the quantum landscape

NSA Announcement

The NSA Information Assurance Directorate (IAD) announced on 19 August 2015 that a transition to post-quantum cryptography is upcoming for US governmental computer systems:

"IAD will initiate a transition to quantum resistant algorithms in the not too distant future. Based on experience in deploying Suite B, we have determined to start planning and communicating early about the upcoming transition to quantum resistant algorithms."

Research Landscape

- EU has announced a one billion euro flagship project [1]
- 7000 researchers and 1.5 billion euro funding for quantum technology research in 2015 according to [2]
- NIST started quantum resistant cryptography standardization
- ETSI established quantum safe crypto (QSC) group
- H2020 projects on quantum safe crypto (SAFEcrypto, PQCRYPTO)

IAD: https://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm
[1] https://ec.europa.eu/digital-single-market/en/news/european-commission-will-launch-eu1-billion-quantum-technologies-flagship
[2] http://qcit.committees.comsoc.org/files/2017/05/Industry-perspectives-of-Quantum-Technologies.pdf

Agenda

1. Introduction
2. Post-Quantum Cryptography
3. Lattice-Based Cryptography
4. Hash- and Code-Based Cryptography
5. Outlook

Post-quantum cryptography and quantum cryptography

Post-Quantum Cryptography and Quantum Cryptography are not the same

Post-Quantum Cryptography

- Conventional cryptography deployable without quantum computers (i.e. on classical computer)
- Believed/hoped to provide security against classical and quantum computer attacks (absence of attacks)
- Requires new mathematical hardness assumptions for public-key crypto

Quantum Cryptography

- Mainly Quantum Key Distribution (QKD) to secure communication using quantum mechanics
- Security relies on quantum mechanics not computational assumption
- Physical restrictions like fiber-optical cable (max. 100-300 km) or line of sight between communicating parties

Infineon is actively pursuing intensive research on post-quantum cryptography

Post-quantum crypto: The families

Five popular families known to build post-quantum asymmetric cryptography

Family (assumption)	Signatures	Encryption or Key Exchange	Description
Hash-based	x	-	› Based on security of symmetric hash function; number of signatures limited per public/private key for stateful schemes
Multivariate Quadratic-based	x	- (*)	› Based on multivariate polynomial equations; large public keys (27.9 kBytes to 75 kBytes); some schemes broken
Code-based	- (*)	x	› Old (1978) and trusted but large public-keys; less trust in more efficient variants (e.g., QC-MDPC)
Lattice-based	x	x	› Old proposals (NTRU in 1996) and newer ones (LWE/RLWE); good performance and reasonable sizes for key/signature/ciphertext (~1-4 kBytes)
Isogeny-based	x	x	› Related to ECC (reuse); slow but small ciphertexts/keys; relatively new field of research

Not the focus of this talk

(Family/assumption: RSA = factorization assumption; ECC = discrete logarithm assumption)

(*) Proposals exist but they are currently not considered competitive

Applications of post-quantum cryptography

Internet protocols

- Lots of critical applications rely on Transport Layer Security (TLS)
- Introduction of PQC proposals might happen soon
- Google's Adam Langley:
  "It's likely that TLS will want a post-quantum key-agreement in the future"

High-security communication

- Deployment and experiments expected soon
- Danger of adversary storing communication and later decryption
- Immediate measures: Pre-placed 256-bit symmetric keys

Automotive

- Increasing connectivity and long lifetimes
- Crypto agility and mechanisms for upgrade to PQC advisable in the long term

Internet of things (IoT)

- Important for devices with a longer lifetime, e.g. in industry automation or smart home

The NIST process

The National Institute of Standards and Technology (NIST) started a standardization effort:

- Competition-like process
- Researchers can submit key exchange, PKE, signature schemes
- Selection metrics: “security”, “cost”, “algorithm and implementation characteristics”

The timeline is:

- End of Nov 2017 – Deadline for submissions
- April 12-13, 2018 “First PQC Standardization Conference”
- 3-5 years – Analysis phase
- 2 years later – Draft standards ready (2023-2025)

We are close to the deadline

- So far NIST process appears to work (some late refinements of software framework happened)
- Several projects have announced their submission (see https://post-quantum.ch/); Some overlap expected
- Expected near term outcome: Consolidated view on PQC landscape with lots of implementations and practical instantiations

NIST: http://csrc.nist.gov/groups/ST/post-quantum-crypto/

Agenda

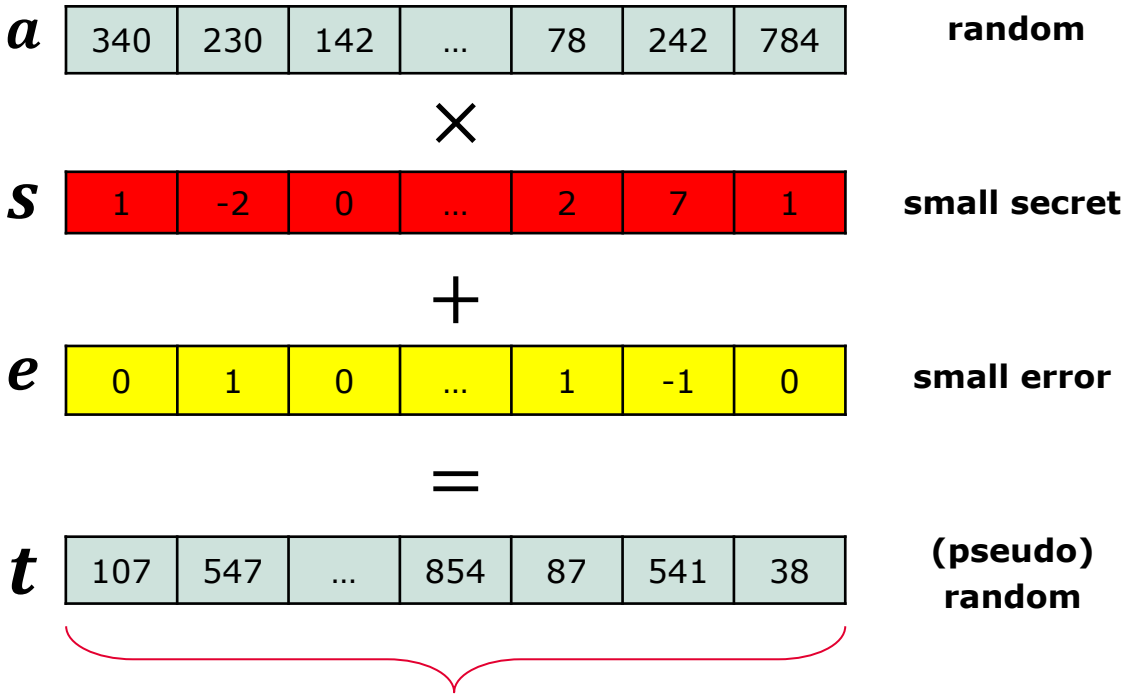
1. Introduction
2. Post-Quantum Cryptography
3. Lattice-Based Cryptography
4. Hash- and Code-Based Cryptography
5. Outlook

Lattice-based cryptography

Lattice-based cryptography

- Cryptography based on the hardness of lattice problems
  - A lattice of dimension n is a maximal discrete subgroup of $\mathbb{R}^n$
  - Set of points defined by integer combination of basis vectors
- Active field of research and useful for public-key encryption, signatures, and more
  - **NTRU** (1996)
  - **Kyber**, **Newhope** and **Frodo** key exchange schemes or public-key encryption
  - **Dilithium**, **BLISS**, and **Tesla** signatures
- Advantages: Efficiency and flexibility
  - In general: Operations on vectors and matrices modulo relatively small integers
  - Additional structure for better performance in lattices (might lead to attacks) – Ring Learning with Errors (**RLWE**)
  - Ideal lattices allow very fast operation on small polynomials (1024 coefficients with less than 16-bit)

Ring Learning with Errors



- **Ideal lattices** correspond to ideals in the ring $R_q = Z_q[x]/(x^n+1)$. Aka: Work with polynomials with n coefficients that are reduced modulo q and $x^n + 1$.
- **A Ring Learning With Errors** (RLWE) sample is: $\mathbf{t} = \boldsymbol{as} + \boldsymbol{e} \in R_q$ for uniform $\boldsymbol{a} \in R_q$ and small $\boldsymbol{s}, \boldsymbol{e} \leftarrow D_\sigma$
  - Search-RLWE: Find **s** when given $\mathbf{t}$ and $\mathbf{a}$
  - Decision-RLWE: Distinguish $\mathbf{t}$ from uniform when given $\mathbf{t}$ and $\mathbf{a}$

Key exchange: Diffie-Hellman with noise

Dlog

Constant $a \in \mathbb{Z}_p$ $s_a, s_b, b, u \in \mathbb{Z}_p$		
Alice (Server)		Bob (Client)
$s_a \xleftarrow{\$} \{0, \ldots, p-1\}$		$s_b \xleftarrow{\$} \{0, \ldots, p-1\}$
$b \leftarrow a^{s_a} \bmod p$	$\xrightarrow{b}$ $\xleftarrow{u}$	$u \leftarrow a^{s_b} \bmod p$
$k_a = u^{s_a} = a^{s_a s_b} \bmod p$		$k_b = b^{s_b} = a^{s_a s_b} \bmod p$
$k_a = k_b$		

Traditional Diffie-Hellman over Z_p

- Traditional Diffie-Hellman is based on discrete logarithm problem (Dlog)
- Find s for $a^s \, mod \, p$ for a being a generator and p a large prime

RLWE

Constant $\mathbf{a} \in \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n+1)$ $s_a, \mathbf{e}, \mathbf{b}, \mathbf{u}, s_b, \mathbf{e}' \in \mathcal{R}_q$		
Alice (Server)		Bob (Client)
$s_a, \mathbf{e} \xleftarrow{\$} \chi$		$s_b, \mathbf{e}' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}s_a + \mathbf{e}$	$\xrightarrow{b}$ $\xleftarrow{u}$	$\mathbf{u} \leftarrow \mathbf{a}s_b + \mathbf{e}'$
$k_a = \mathbf{u}s_a = \mathbf{a}s_a s_b + \mathbf{e}'s_a$		$k_b = \mathbf{b}s_b = \mathbf{a}s_a s_b + \mathbf{e}s_b$
$k_a \approx k_b$		

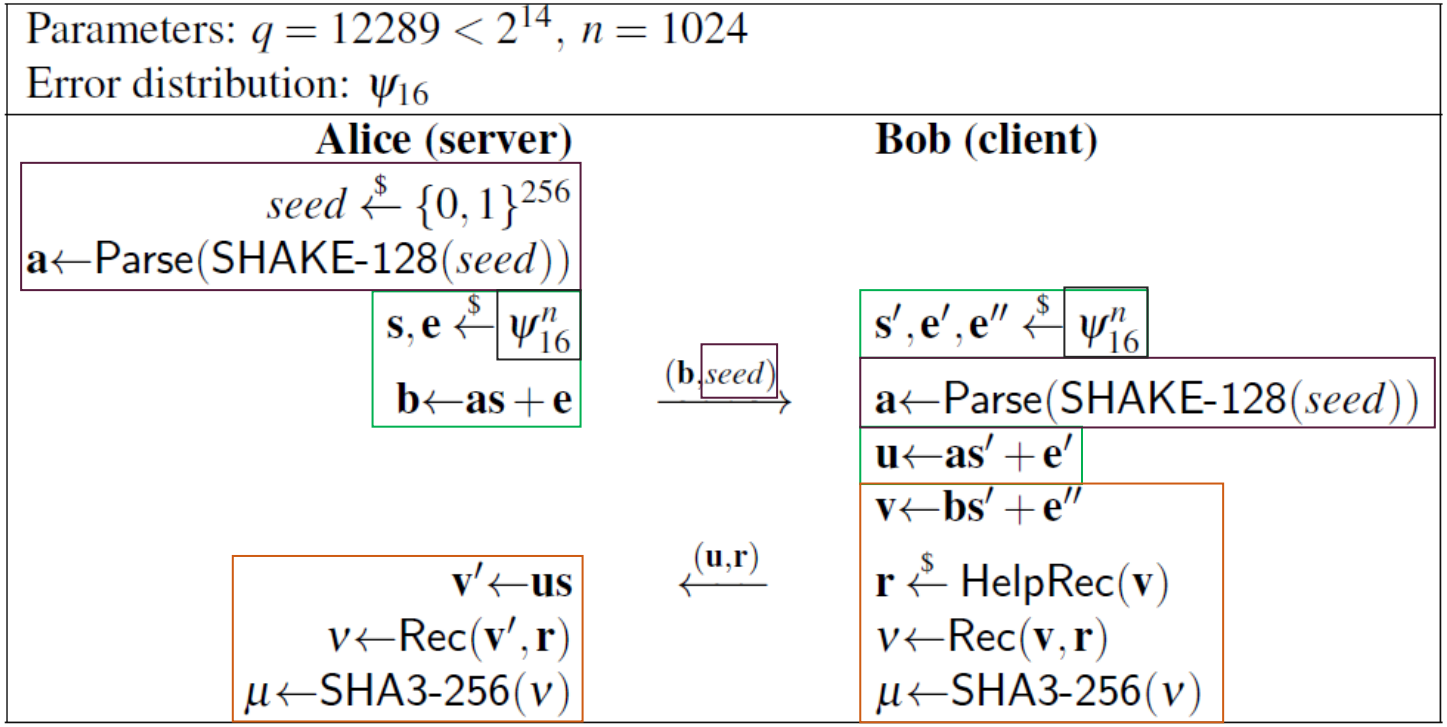
Noise

"Diffie-Hellman with noise" over $R_q = Z_q[x]/x^n+1$

- "Diffie-Hellman with noise" based in Ring-Learning with Errors problem (RLWE)
- Find s when given $as + e \in R_q$ for uniform a and small s and e
- Due to noise only approximate key is obtained -> methods needed to agree on exact key

NewHope: Protocol

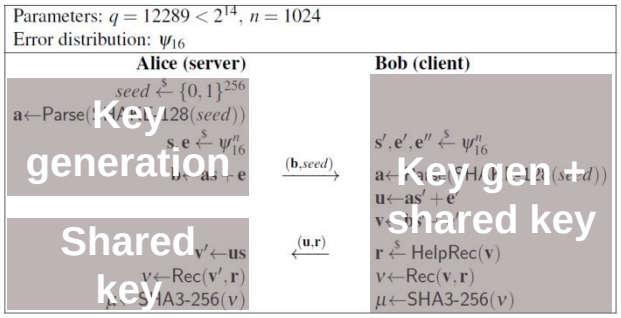
Paper by Erdem Alkim (RU Nijmegen), Léo Ducas (CWI Amsterdam), Thomas Pöppelmann (IFX), Peter Schwabe (RU Nijmegen)
presented at USENIX Security'16



- Compute "Diffie-Hellman with noise" over polynomials in R_q with estimated 255 bits of security
- Choose a fresh parameter $\boldsymbol{a}$ for every protocol run to protect against backdoors (Parse)
- Error distribution $\psi_k = \sum_{i=0}^{k-1} b_i - b_i'$ for $b_i, b_i' \in \{0,1\}$ (binomial; no discrete Gaussians)
- HelpRec/Rec are reconciliation functions to deal with noise and to extract the final key μ

NewHope: Implementation

- Reference C implementation
  - Arithmetic on 16-bit and 32-bit integers
  - No division (/) or modulo (%) operator
- Intel Assembly implementation
  - Speed up NTT using vectorized double arithmetic
  - Uses AES-256 instructions for noise sampling
- ARM Cortex M4 implementation [1]
  - Optimized using ARM assembly



NewHope requires the execution of three steps.
Each party sends roughly 2000 bytes.

NewHope on Intel CPUs

Operation	Reference	Optimized
Key generation (server)	0.128 ms	0.044 ms
Key gen + shared key (client)	0.192 ms	0.056 ms
Shared key (server)	0.043 ms	0.0095 ms

Assuming a CPU @ 2 GHz (0.056 ms => 17800 executions/s)
Exemplary EC Diffie-Hellman (ECDH) implementation is 0.075 ms

NewHope on Cortex-M4 uC [1]

Operation	Cortex-M4
Key generation (server)	9.6 ms
Key gen + shared key (client)	14.8 ms
Shared key (server)	1.79 ms

Microcontroller @ 100 MHz (14.8 ms => 67 executions/s)
Exemplary EC Diffie-Hellman (ECDH) implementation is 16 ms

[1] Alkim, Jakubeit, Schwabe:NewHope on ARM Cortex-M. SPACE 2016

Next steps after ephemeral key exchange

What are the next steps for lattice crypto?

- Lattice-based cryptoschemes like NewHope work well for ephemeral key exchange (other proposals as well: Kyber, Frodo, Lizard, etc.)
- What about public-key encryption and side-channels?

"…, the implementation security aspect of lattice-based cryptography is still a vastly unexplored and open topic." [1]

[1] Primas, Pessl, Mangard: Single-Trace Side-Channel Attacks on Masked Lattice-Based Encryption. CHES 2017

Ring-LWE Encryption [LPR10]

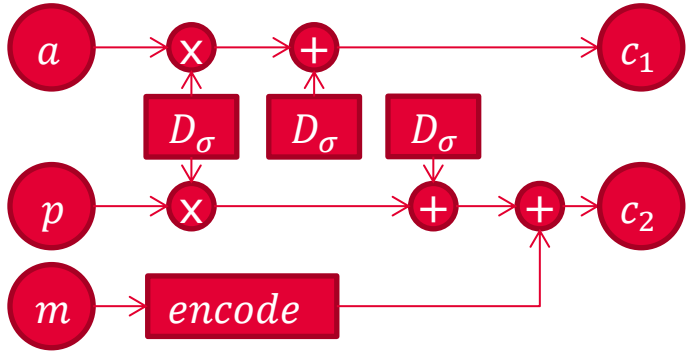
RLWE.CPA.Gen: Choose $\boldsymbol{a} \leftarrow R_q$ and $\boldsymbol{r}_1, \boldsymbol{r}_2 \leftarrow D_\sigma$; *pk*: $\boldsymbol{p} = \boldsymbol{r}_1 - \boldsymbol{a} \cdot \boldsymbol{r}_2 \in \mathrm{R_q}$; *sk*: $\boldsymbol{r}_2$

RLWE.CPA.Enc$(\boldsymbol{a}, \boldsymbol{p}, m \in \{0,1\}^n)$:

$\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3 \leftarrow D_\sigma$.

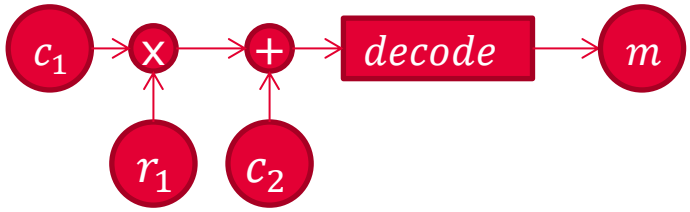
$\bar{\boldsymbol{m}} = encode(m)$.

Return $[\boldsymbol{c}_1 = \boldsymbol{a} \cdot \boldsymbol{e}_1 + \boldsymbol{e}_2, \boldsymbol{c}_2 = \boldsymbol{p} \cdot \boldsymbol{e}_1 + \boldsymbol{e}_3 + \bar{\boldsymbol{m}}]$



RLWE.CPA.Dec$(c = [\boldsymbol{c}_1, \boldsymbol{c}_2], \boldsymbol{r}_2)$:

Return $decode(\boldsymbol{c}_1 \cdot \boldsymbol{r}_2 + \boldsymbol{c}_2)$



Correctness: $\boldsymbol{c}_1\boldsymbol{r}_2 + \boldsymbol{c}_2 = (\boldsymbol{a}\boldsymbol{e}_1 + \boldsymbol{e}_2)\boldsymbol{r}_2 + \boldsymbol{p}\boldsymbol{e}_1 + \boldsymbol{e}_3 + \bar{\boldsymbol{m}}$

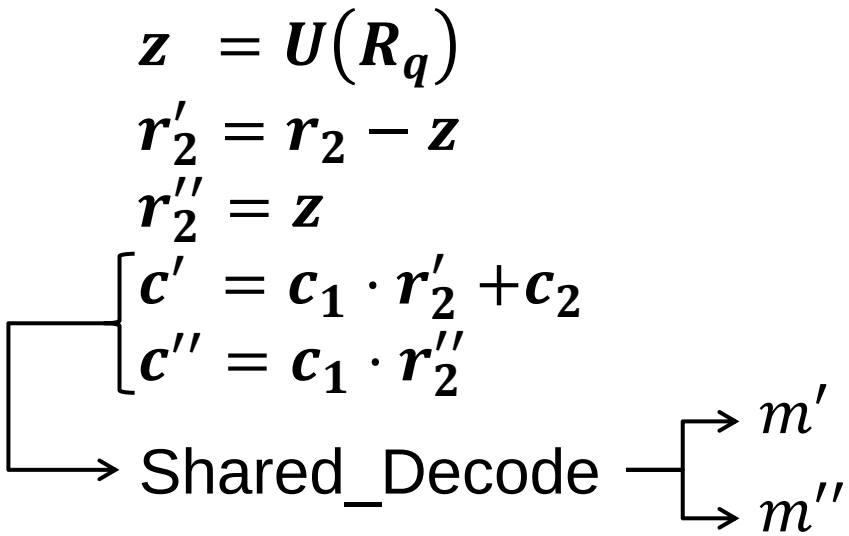
$= \boldsymbol{r}_2\boldsymbol{a}\boldsymbol{e}_1 + \boldsymbol{r}_2\boldsymbol{e}_2 + \boldsymbol{r}_1\boldsymbol{e}_1 - \boldsymbol{r}_2\boldsymbol{a}\boldsymbol{e}_1 + \boldsymbol{e}_3 + \bar{\boldsymbol{m}} = \bar{\boldsymbol{m}} + \boldsymbol{r}_2\boldsymbol{e}_2 + \boldsymbol{r}_1\boldsymbol{e}_1 + \mathrm{e}_3$

large **small**

[LPR10] Lyubashevsky, Peikert, Regev:On Ideal Lattices and Learning with Errors over Rings. EUROCRYPT 2010

RLWE masking: The straightforward approach

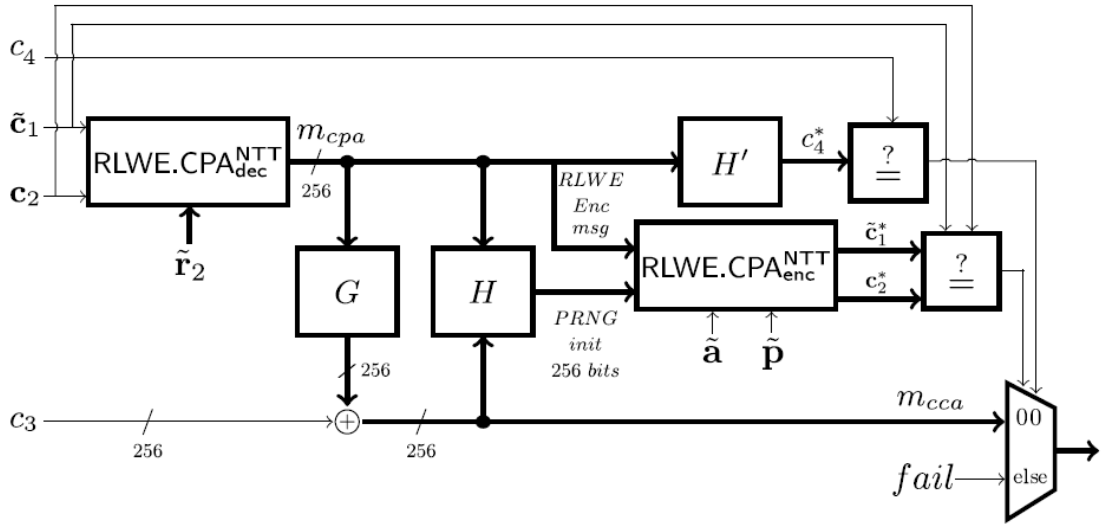
$$decode(\boldsymbol{c_1} \cdot \boldsymbol{r_2} + \boldsymbol{c_2})$$



- RLWE decryption is easy to mask/randomize
  - Splitting of the secret key into shares is simple
  - Polynomial arithmetic is fast
  - Shared decoder is challenging but can be done and has been demonstrated

However, this is only half of the picture!

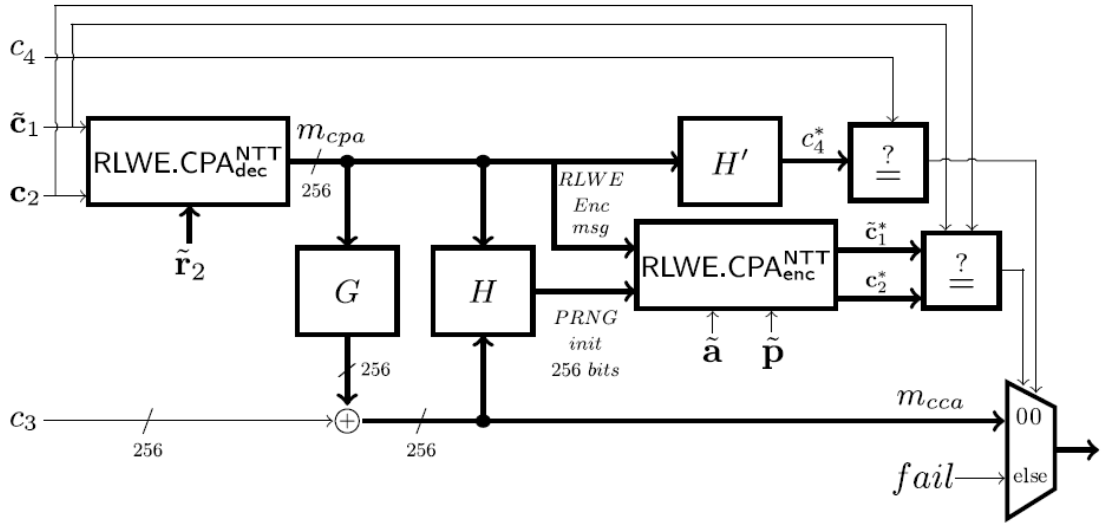
CCA2 conversion for RLWE encryption



- The RLWE encryption scheme is only secured against **chosen plaintext attacks** (CPA)
- **Conversion with re-encryption required** to achieve sematic security against chosen ciphertext attacks (CCA2)
- What would a side-channel/DPA adversary do? SCA or CCA attack?
  - CCA attack is simple and does not require any equipment
  - Side-channel research has to consider the transformation and implementers have to consider the cost

$$decode(\boldsymbol{c}_1 \cdot \boldsymbol{r}_2 + \boldsymbol{c}_2)$$

Masked implementation of RLWE encryption



- We propose a masked implementation with CCA conversion with security proof in the probing model [1]
- All bold boxes in the CCA-secured decryption have to be masked
  - Hash functions G,H,H' (i.e., SHA3)
  - Decryption routine including message decoding
  - Encryption routine including the noise sampler and message encoding
  - Final comparisons

[1] Oder, Schneider, Pöppelmann, Güneysu: Practical CCA2-Secure and Masked Ring-LWE Implementation. IACR ePrint 2016: 1109 (recently updated)

Results: Masked implementation

Implementation results for our masked RLWE Encryption implementation [1]

Implementation results
- RLWE instantiation similar to NewHope (n=1024) with 233 bits of security
- Hiding as additional countermeasure
- Unmasked decryption is 4.4 million cycles (overhead factor of 6)
- Masked noise sampling is expensive (1.1 -> 6 million cycles when masked)

Operation	Cycles on Cortex-M4	Time @100 MHz
Key generation	2.669.559	26 ms
Encryption	4.176.684	42 ms
Decryption (masked)	25.334.493	253 ms

NIST P-256 elliptic curve 81.60.000 cycles (81 ms); for countermeasures add factor 1.5; From "Practical Results of ECC Side Channel Countermeasures on an ARM Cortex M3 Processor", Samotyja and Lemke-Rust

Future work
- Analysis of second order resistance
- Analysis of fault attacks (interesting in combination with CCA2 conversion)
- Performance improvements
- Look at Kyber (MLWE), Frodo (LWE), and friends

[1] Oder, Schneider, Pöppelmann, Güneysu: Practical CCA2-Secure and Masked Ring-LWE Implementation. IACR ePrint 2016: 1109 (recently updated)

Agenda

1. Introduction
2. Post-Quantum Cryptography
3. Lattice-Based Cryptography
4. Hash- and Code-Based Cryptography
5. Outlook

Hash-based signatures

Stateful-hash based signatures

- Appears highly suitable for firmware protection (applications with few signatures per public key)
- eXtended Merkle Signature Scheme (XMSS) in late stage at IETF [1] with signatures between 1 KByte and 5 KBytes
- Genua GmbH announced that product updates will be protected with post-quantum signatures

Schemes are stable since a few years. Challenges are mostly implementation and integration into products. Some works on side-channel [2] protection.

Stateless-hash based signatures

- Most popular stateless scheme is SPHINCS
- Signature size is 41 Kbyte
- Signature can be streamed out of an embedded device [3]

No works on side-channel or fault protection. No fundamental improvements to SPHINCS signature size since introduction but needed for usage in embedded devices.

[1] https://tools.ietf.org/html/draft-irtf-cfrg-xmss-hash-based-signatures-10
[2] Thomas Eisenbarth, Ingo von Maurich, Xin Ye:Faster Hash-Based Signatures with Bounded Leakage. SAC 2013
[3] Andreas Hülsing, Joost Rijneveld, Peter Schwabe: ARMed SPHINCS - Computing a 41 KB Signature in 16 KB of RAM. PKC 2016

Code-based cryptography

McEliece/Niederreiter with binary Goppa codes

- Well vetted but public keys between 0.5 to 1 Mbytes and slow key generation
- Encryption is very fast in software and hardware
- Research on implementation available

Too large for embedded devices. Already some works on side-channel attacks [1,2].

Tricky: As key generation is slow you need a CCA2 conversion.

McEliece/Niederreiter with more structured codes

- Example is McEliece with QC-MDPC (Moderate Density Parity-Check) codes
- Smaller public keys and relatively fast on embedded devices
- Suffers from (small) decryption error probability that interferes with conversion [4]

Newer and more structure but survived cryptanalysis for some years now. First works on side-channel resistance [3] but more works required.

[1] Chen, Eisenbarth, von Maurich, Steinwandt: Horizontal and Vertical Side Channel Analysis of a McEliece Cryptosystem. IEEE Trans. Inf. Forensics and Sec. 2016
[2] Chen, Eisenbarth, von Maurich, Steinwandt: Differential Power Analysis of a McEliece Cryptosystem. ACNS 2015: 538-556
[3] von Maurich, Güneysu: Towards Side-Channel Resistant Implementations of QC-MDPC McEliece Encryption on Constrained Devices. PQCrypto 2014
[4] Guo, Johansson, Stankovski: A Key Recovery Attack on MDPC with CCA Security Using Decoding Errors. ASIACRYPT 2016

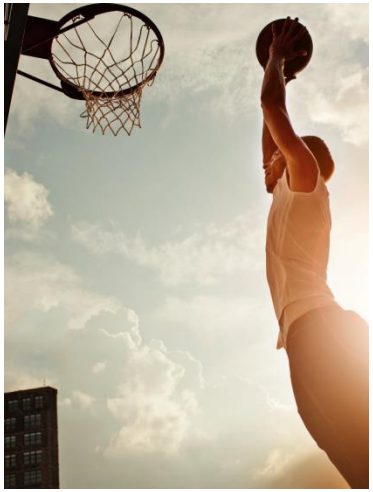
Agenda

1. Introduction
2. Post-Quantum Cryptography
3. Lattice-Based Cryptography
4. Hash- and Code-Based Cryptography
5. Outlook

Overview

Personal overview and assessment

- Stateful hash-based signatures ready for primetime where restrictions do not prevent usage (firmware protection)
- Ephemeral key-exchange is where lattice-based cryptography really shines
  - Frodo (LWE), NewHope (RLWE), Kyber (MLWE) are simple, fast, and efficient
  - Small decryption error probability can be tolerated without causing problems (=efficiency)
- Public-key encryption with lattices is a bit more tricky
  - NTRU works quite well here due to efficient conversion
  - For LWE/RLWE the CCA2 conversion introduces a lot of complexity and decryption errors have to be negligible
  - Side-channel protection can become very complex
- Code-based cryptography
  - In ephemeral key exchange no issue with decryption errors but more expensive key generation
  - CCA2 conversion and error-free decryption seem challenging for structured codes



PQC challenges and opportunities for the community and industry

Personal view on challenges and opportunities

- Shrink the signature size on stateless hash-based signature schemes like SPHINCS
- Cryptanalysis of PQC (the struggle of efficiency vs. structure)
  - LWE vs. RLWE
  - Binary Goppa vs. QC-MDPC and other variants
  - Quantum attacks on all families
- Implementation security for all proposals in the NIST process
  - Side-channel attacks in all variations
  - Fault attacks in all variations
- Practical deployment of hash-based signatures for firmware updates
- Integration of PQC schemes into protocols
  - PQC schemes have different strengths and weaknesses
  - PQC may not fit into all protocols due to key size

Outlook

Post-quantum cryptography is needed to secure a quantum world

In a quantum world we will have:

- More cryptographic standards and maybe even de-facto standardization
- Probably different schemes for encryption, signatures, and key exchange required
- Larger keys, signatures and ciphertexts

PQC will have a disruptive impact on the security industry – industry and academia have to act on it now to be prepared

More information on PQC:

http://www.infineon.com/post-quantum-crypto

Thank you!

Thank you for your attention!

Any questions?

Infineon is hiring:

- Software Developer for Cryptographic Software in Augsburg
- Software Concept Engineer Automotive Ethernet and Network Security in Munich
- Senior Staff Engineer for Automotive Microcontroller Security Applications in Munich
- Embedded Software Test Developer for Java Card Operating System in Augsburg

Visit https://www.infineon.com/cms/de/careers/jobsearch/jobsearch/
or contact me via email (thomas.poeppelmann@infineon.com)

Part of your life. Part of tomorrow.

Infineon