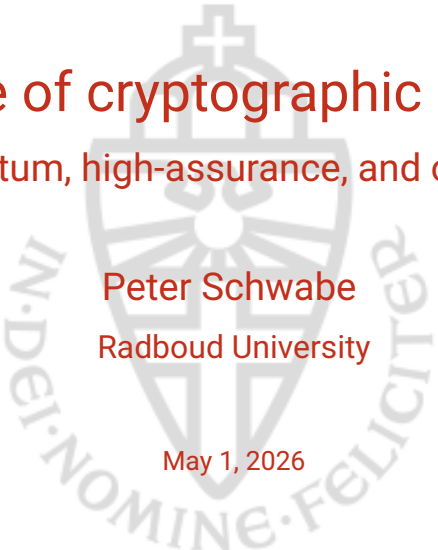
The Future of cryptographic engineering

post-quantum, high-assurance, and open-source

Peter Schwabe

Radboud University

May 1, 2026



“Inaugural”

*“An inaugural lecture is a formal public lecture given at a university by a **newly appointed** full professor, [...] It marks the professor’s official **introduction** to the academic community.”*

—Wikipedia on “Inaugural lecture” (emphasis added)

“Inaugural”

“Is this the full prof promotion from before Corona or another thing? Congrats in any case.”

—An answer to my invitation mail

Cryptographic *Engineering*

1. Research motivated by **real-world challenges**
2. **Constructive** research: we like to build stuff
3. Scientific approach: Use **models and abstractions**, validate those in the lab

The post-quantum (re-)volution

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

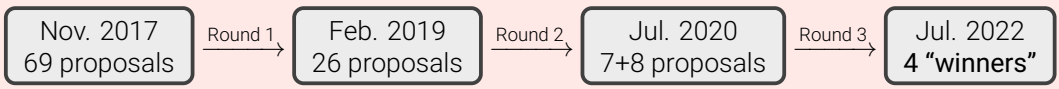
Peter W. Shor†

Abstract

A digital computer is generally believed to be an efficient universal computing device; that is, it is believed able to simulate any physical computing device with an increase in computation time by at most a polynomial factor. This may not be 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.

The post-quantum (re-)volution

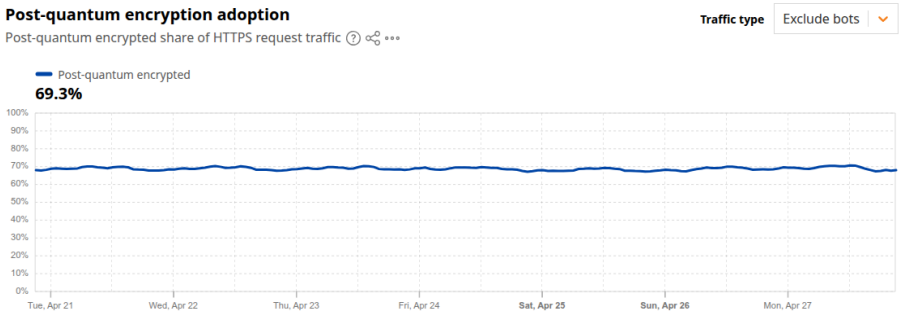
NIST PQC



“The public-key encryption and key-establishment algorithm that will be standardized is CRYSTALS-KYBER. The digital signatures that will be standardized are CRYSTALS-Dilithium, FALCON, and SPHINCS$^+$. While there are multiple signature algorithms selected, NIST recommends CRYSTALS-Dilithium as the primary algorithm to be implemented”

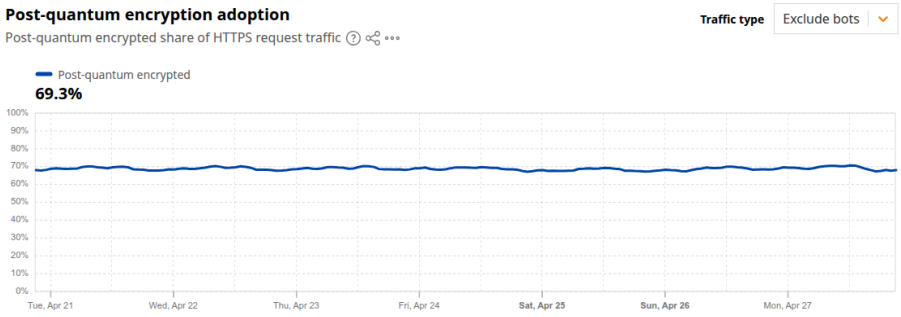
—NIST IR 8413 (July 2022)

Where are we now?



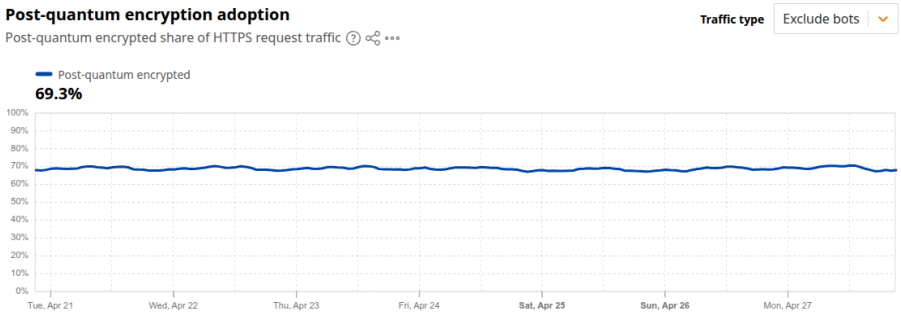
From https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption

Where are we now?



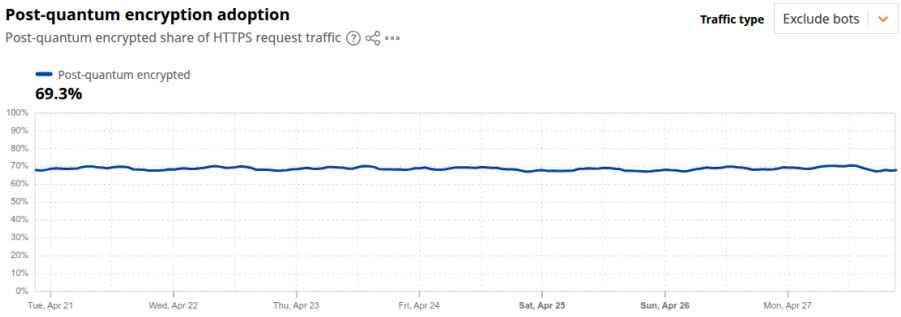
From https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption

Where are we now?



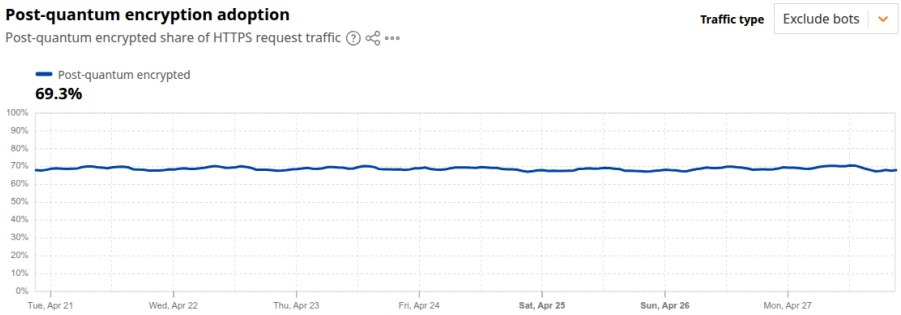
From https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption

Where are we now?



From https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption

Where are we now?



From https://radar.cloudflare.com/adoption-and-usage#post-quantum-encryption-adoption

So, what remains to be done in *research*?

Where are we now?


```
crypto == pqc
```

EPOQUE: Engineering post-quantum cryptography

Secure implementations

- Optimize for tiny embedded systems and large Intel/AMD processors
- Optimize for speed and size, understand tradeoffs
- Side-channel attacks **and countermeasures**
- Fault attacks **and countermeasures**

Post-quantum protocols

- Re-think protocols with post-quantum primitives
- (How) can we live without Diffie-Hellman?
- Do we need efficient key generation?
- Do we need fast signatures?

PQC research – “EPOQUE ctd.”

Implementation security and **protocol migration**
are more relevant than ever!

PQC research – “EPOQUE ctd.”

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 1, pp. 60–88. DOI:10.46586/tches.v2023.i1.60-88

Adapting Belief Propagation to Counter Shuffling of NTTs

Julius Hermelink[1,2], Silvan Streit[3],
Emanuele Strieder[4] and Katharina Thieme[5]

PQC research – “EPOQUE ctd.”

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 1, pp. 60–88. DOI:10.46586/tches.v2023.i1.60-88

Ada

Belief Propagation Meets Lattice Reduction: Security Estimates for Error-Tolerant Key Recovery from Decryption Errors

Julius Hermelink[1†] Erik Mårtensson[2,3], Simona Samardjiska[4], Peter Pessl[5], Gabi Dreo Rodosek[6]

PQC research – “EPOQUE ctd.”

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 1, pp. 60–88. DOI:10.46586/tches.v2023.i1.60-88

The Insecurity of Masked Comparisons: SCAs on ML-KEM's FO-Transform

Julius Hermelink
Max Planck Institute for Security and Privacy
Bochum, Germany
julius.hermelink@mpi-sp.org

Kai-Chun Ning
Max Planck Institute for Security and Privacy
Bochum, Germany
kai-chun.ning@mpi-sp.org

Richard Petri
Max Planck Institute for Security and Privacy
Bochum, Germany
rp@rpls.de

Emanuele Strieder
Fraunhofer AISEC
Garching, Germany
Technical University of Munich
Munich, Germany
emanuele.strieder@aisec.fraunhofer.de

PQC research – “EPOQUE ctd.”

IACR Transactions on Cryptographic Hardware and Embedded Systems
ISSN 2569-2925, Vol. 2023, No. 1, pp. 60–88. DOI:10.46586/tches.v2023.i1.60-88

Beli… Ada… Se…

Ju…

The Insecurity of Masked Comparisons: SCAs on ML-KEM's FO-Transform

Max Pl…

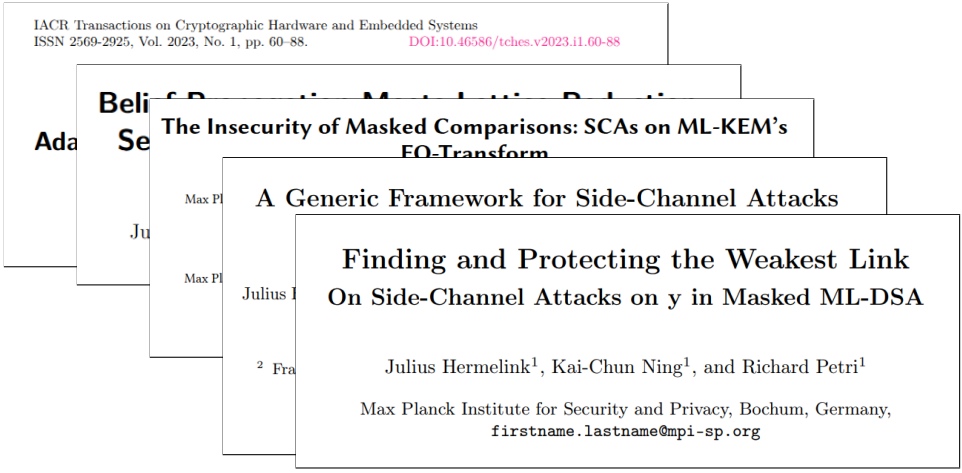
Max Pl…

A Generic Framework for Side-Channel Attacks against LWE-based Cryptosystems

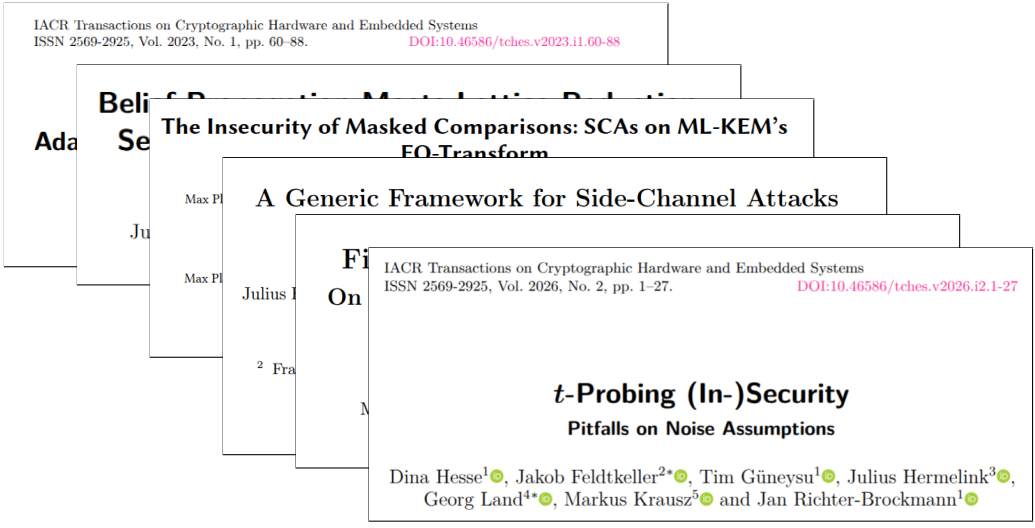
Julius Hermelink[1], Silvan Streit[2,3], Erik Mårtensson[4,5], and Richard Petri[1]

[1] Max Planck Institute for Security and Privacy, Bochum, Germany, {julius.hermelink,richard.petri}@mpi-sp.org
[2] Fraunhofer AISEC, Garching, Germany, silvan.streit@aisec.fraunhofer.de
[3] Technical University of Munich (TUM), Munich, Germany
[4] Lund University, Lund, Sweden, erik.martensson@eit.lth.se
[5] Advenica AB, Malmö, Sweden

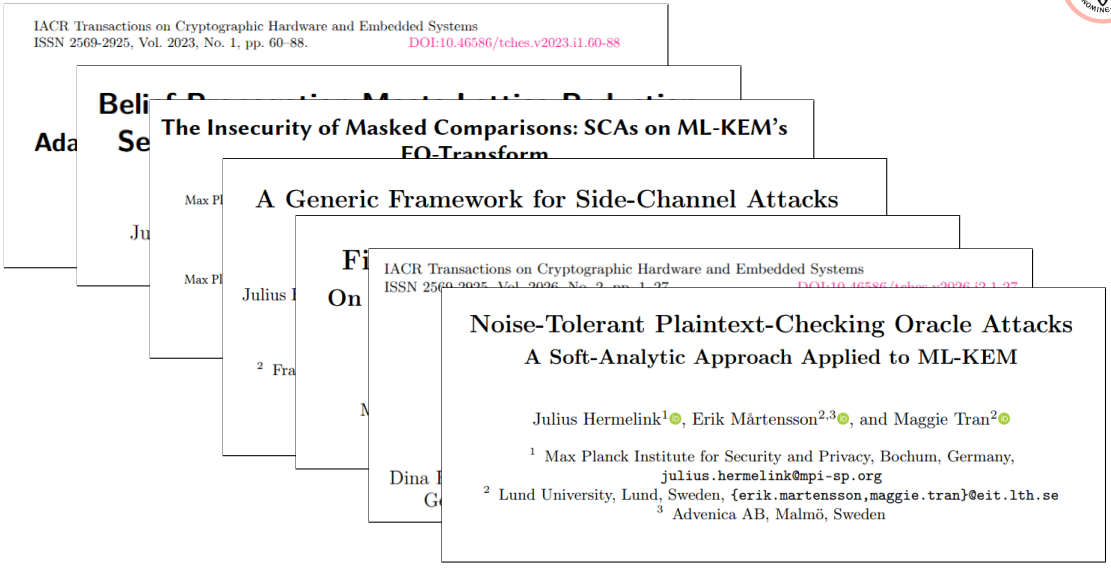
PQC research – “EPOQUE ctd.”



PQC research – “EPOQUE ctd.”



PQC research – “EPOQUE ctd.”



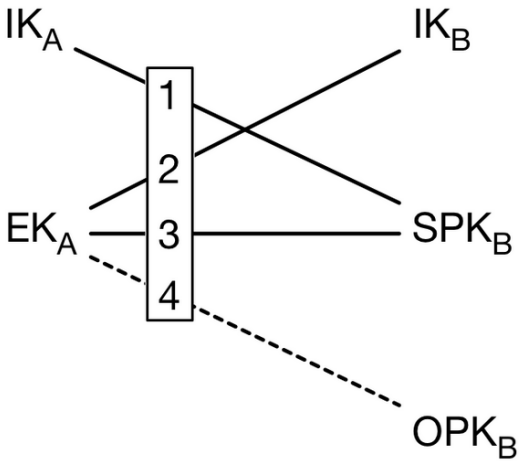
PQC research – “EPOQUE ctd.”

- 2023: Signal upgrades handshake to PQ
- Only goal: protect against HNDL

PQC research – “EPOQUE ctd.”

- 2023: Signal upgrades handshake to PQ
- Only goal: protect against HNDL
- Original X3DH protocol

$$sk = \text{KDF}(\text{DH}_1|\text{DH}_2|\text{DH}_3|\text{DH}_4)$$



https://signal.org/docs/specifications/x3dh/

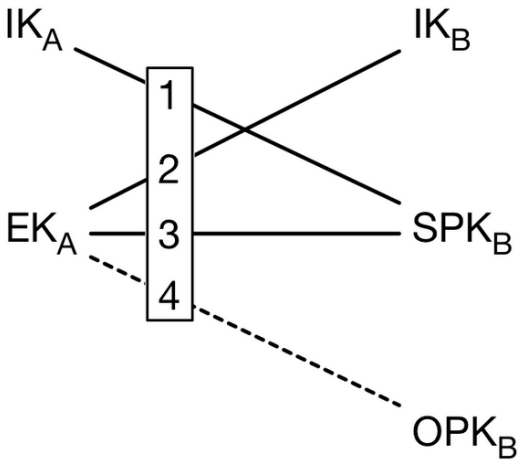
PQC research – “EPOQUE ctd.”

- 2023: Signal upgrades handshake to PQ
- Only goal: protect against HNDL
- Original X3DH protocol

$$sk = \text{KDF}(\text{DH}_1|\text{DH}_2|\text{DH}_3|\text{DH}_4)$$

- Upgraded PQXDH:
  - Additionally obtain SS from PQ KEM
  - Compute final shared key as

$$sk = \text{KDF}(\text{DH}_1|\text{DH}_2|\text{DH}_3|\text{DH}_4|\text{SS})$$



https://signal.org/docs/specifications/x3dh/

PQC research – “EPOQUE ctd.”

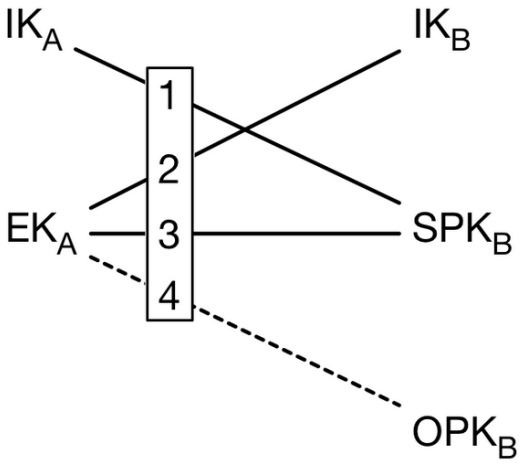
- 2023: Signal upgrades handshake to PQ
- Only goal: protect against HNDL
- Original X3DH protocol

$$sk = \text{KDF}(\text{DH}_1|\text{DH}_2|\text{DH}_3|\text{DH}_4)$$

- Upgraded PQXDH:
  - Additionally obtain SS from PQ KEM
  - Compute final shared key as

$$sk = \text{KDF}(\text{DH}_1|\text{DH}_2|\text{DH}_3|\text{DH}_4|\text{SS})$$

- Two (potential) attacks discovered by Bhargavan, Jacomme, Kiefer, and Schmidt



https://signal.org/docs/specifications/x3dh/

Beyond “EPOQUE”

More (KEMs) and signatures

- Research on block ciphers did not stop with FIPS-43-6 (DES)
- NIST PQC selected HQC as additional KEM in 2025
- NIST PQC continues selection of signatures (“on-ramp”)
- Can we get smaller, faster, more secure PQC?

Beyond “EPOQUE”

More (KEMs) and signatures

- Research on block ciphers did not stop with FIPS-43-6 (DES)
- NIST PQC selected HQC as additional KEM in 2025
- NIST PQC continues selection of signatures (“on-ramp”)
- Can we get smaller, faster, more secure PQC?

Advanced PQC

- KEMs and signatures are just a small part of public-key crypto
- PQ threshold schemes → NIST threshold competition
- PQ zero-knowledge and anonymous credentials

Challenges and Tools

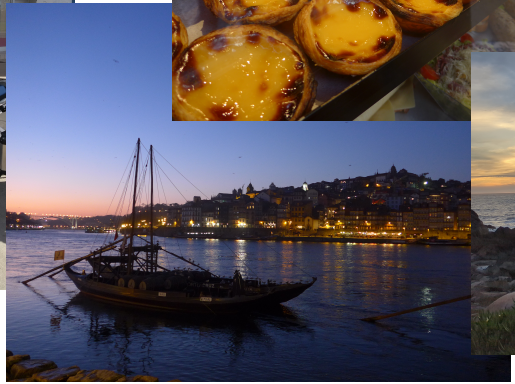
Challenges

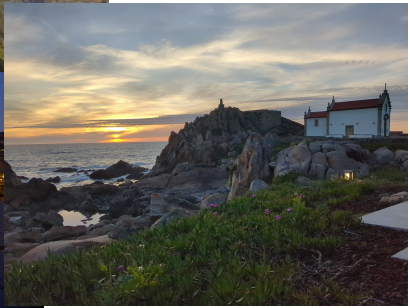
- (Proven to be) secure implementations (including SCA security)
- Clear and meaningful security assumptions
- Verified security proofs using those assumptions
- Secure interfaces that connect to other components

Tools

- Today: Write software in C and assembly, ideally test/audit
- Need to move from 20th to 21st century:
  - Move from C to Rust (?)
  - Move from assembly to, e.g., jasmin (?)
  - Only use domain-specific languages (?)
  - Statically verify security properties during SW build

2020, the crazy year





2020, the crazy year

Goal: Formally verified implementation of Kyber

6 years later

≈9 papers, 30+ collaborators

Basavesh Ammanaghatta Shivakumar, Santiago Arranz Olmos, José Bacelar Almeida, Gustavo Xavier Delerue Marinho Alves, Manuel Barbosa, Francisca Barros, Gilles Barthe, Lionel Blatter, Chitchanok Chuengsatiansup, Ignacio Cuevas, François Dupressoir, Luís Esquível, Ruben Gonzalez, Benjamin Grégoire, Andreas Hülsing, Vincent Hwang, Jan Jancar, Matthias Kannwischer, Vincent Laporte, Jean-Christophe Léchenet, Ting-han Lim, Cameron Low, Tiago Oliveira, Hugo Pacheco, Swarn Priya, Miguel Quaresma, Rolfe Schmidt, Antoine Séré, Lucas Tabary-Maujean, Pierre-Yves Strub, Yuval Yarom, Zhiyuan Zhang, Jieyu Zheng

6 years later

≈9 papers, 30+ collaborators

Basavesh Ammanaghatta Shivakumar, Santiago Arranz Olmos, José Bacelar Almeida, Gustavo Xavier Delerue Marinho Alves, Manuel Barbosa, Francisca Barros, Gilles Barthe, Lionel Blatter, Chitchanok Chuengsatiansup, Ignacio Cuevas, François Dupressoir, Luís Esquível, Ruben Gonzalez, Benjamin Grégoire, Andreas Hülsing, Vincent Hwang, Jan Jancar, Matthias Kannwischer, Vincent Laporte, Jean-Christophe Léchenet, Ting-han Lim, Cameron Low, Tiago Oliveira, Hugo Pacheco, Swarn Priya, Miguel Quaresma, Rolfe Schmidt, Antoine Séré, Lucas Tabary-Maujean, Pierre-Yves Strub, Yuval Yarom, Zhiyuan Zhang, Jieyu Zheng

FORMOSA CRYPTO

BOSTON UNIVERSITY

University of BRISTOL

CASA
Cyber Security in the Age of Large-Scale Adversaries

CryptoExperts

h_da formerly university of applied sciences | EUT+

HPI Hasso Plattner Institut
Digital Engineering · Universität Potsdam

imdea software

INESCTEC

Inria

Universidade do Minho

Max Planck Institute for Security and Privacy

Universität Potsdam

U.PORTO
Faculdade de Ciências
Universidade do Porto

PQ SHIELD

Radboud Universiteit

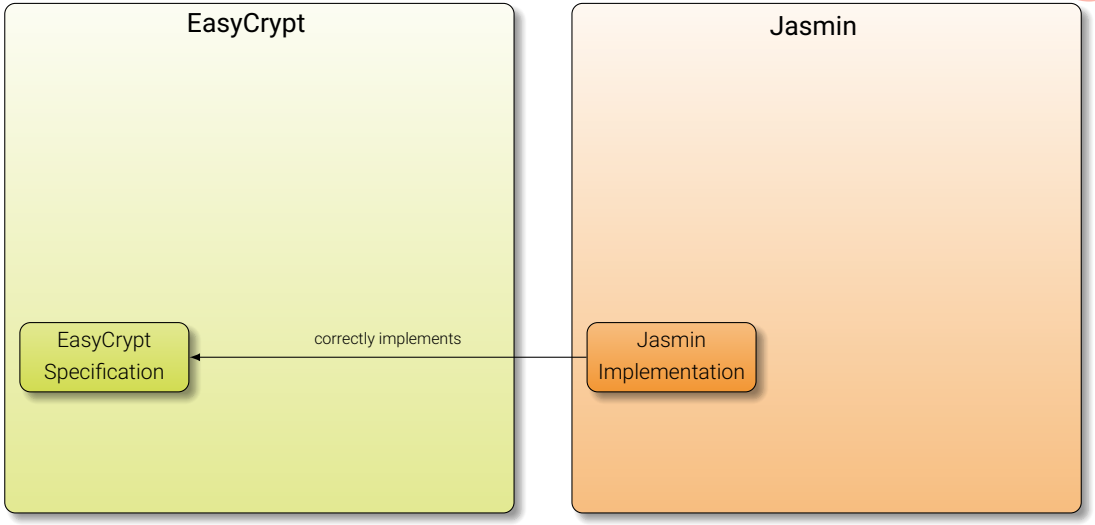
Rosenpass

SANDBOX AQ

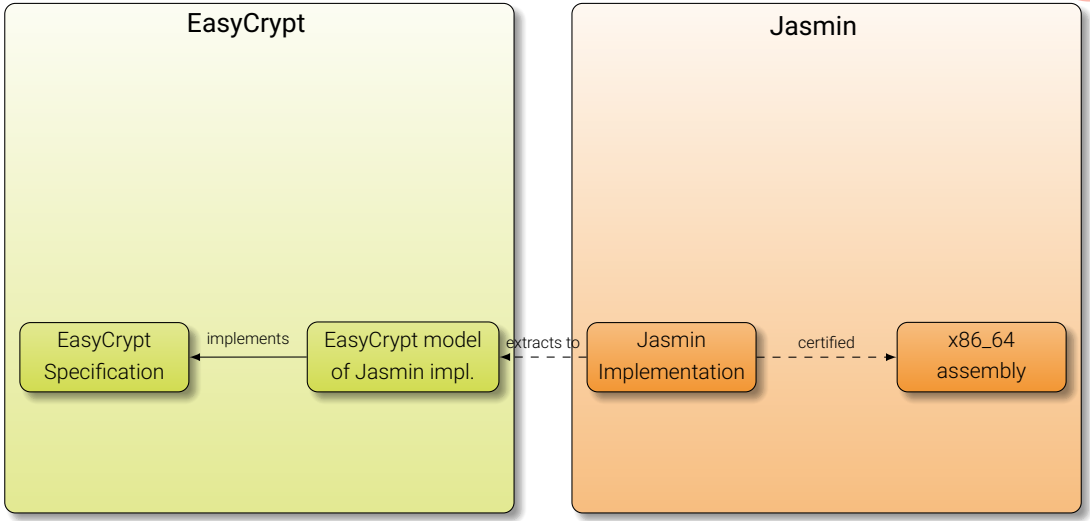
TU/e Eindhoven University of Technology

UCLouvain

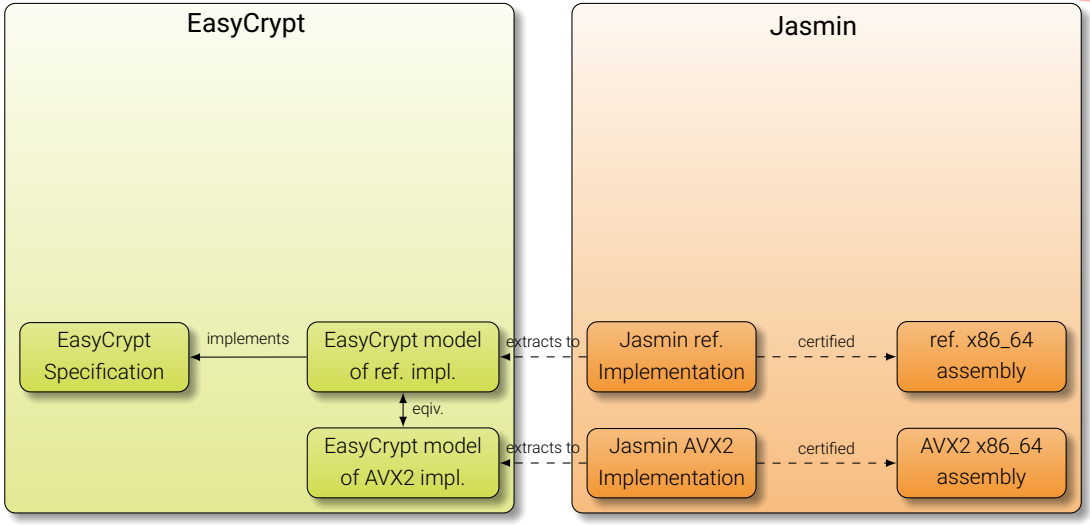
Verifying Kyber



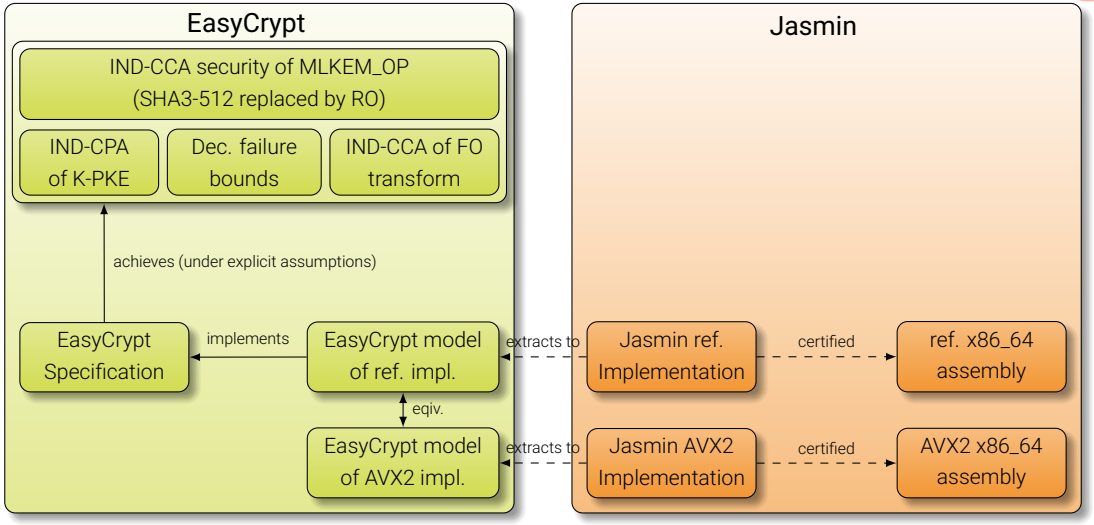
Verifying Kyber



Verifying Kyber



Verifying Kyber



Verifying Kyber

Hey Diego, we have a very important question. How hard would it be for us to change the title of our Formally verified Kyber paper at this point?

Specifically, change "Part 1" to "Episode IV"

Also, greetings from Lisbon, also from Manuel! 18:44

Verifying Kyber

Hey Diego, we have a very important question. How hard would it be for us to change the title of our Formally verified Kyber paper at this point?

Specifically, change "Part 1" to "Episode IV"

Also, greetings from Lisbon, also from Manuel! 18:44

Omg

That's the coolest and silliest request ever. Let me talk to Marcel but it should be fine 18:45

Principled implementation security

- Future proof “constant-time”
  - Information-flow type system in Jasmin
  - Only use DOIT instructions on secret data
  - Influenced update to DOIT instructions in Feb. 2026

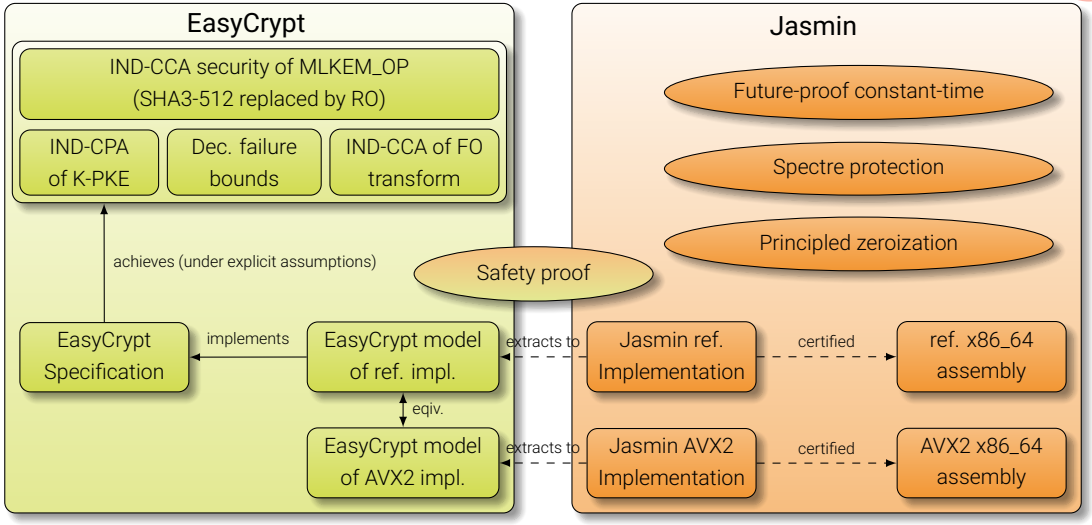
Principled implementation security

- Future proof “constant-time”
  - Information-flow type system in Jasmin
  - Only use DOIT instructions on secret data
  - Influenced update to DOIT instructions in Feb. 2026
- Systematic protection against Spectre attacks
  - Only DOIT on secret data also during transient execution
  - Rewrite function returns to address Spectre-RSB
  - Performance overhead of $<10\%$

Principled implementation security

- Future proof “constant-time”
  - Information-flow type system in Jasmin
  - Only use DOIT instructions on secret data
  - Influenced update to DOIT instructions in Feb. 2026
- Systematic protection against Spectre attacks
  - Only DOIT on secret data also during transient execution
  - Rewrite function returns to address Spectre-RSB
  - Performance overhead of $<10\%$
- Principled zeroization
  - Erase all (sensitive) data from stack and registers
  - Automatic by Jasmin compiler
  - Leverages various aspects of Jasmin

Principled implementation security



PQC as a catalyst for high-assurance crypto

"We started using Kyber in production last year with our updated handshake protocol [...] Kyber is becoming an important building block for us.

Currently we rely on the PQClean implementation [...]

We are trying to decide what Kyber/ML-KEM implementation to use going forward. [...]

Is there anything that you think we should consider as we evaluate our options?

—Rolfe Schmidt, July 2024

The road ahead

Many fun challenges

- Consider more powerful attackers
- Super-optimization for Jasmin (CryptOpt, SLOTHY. . .)
- Link primitive and protocol proofs
- Go from protocols to applications

The road ahead

Many fun challenges

- Consider more powerful attackers
- Super-optimization for Jasmin (CryptOpt, SLOTHY. . .)
- Link primitive and protocol proofs
- Go from protocols to applications

One elephant in the room: **Scale!**

- Cannot take 6 years per primitive
- Already massively improved tooling
- Protocols more complex
- Applications *even* more complex

The road ahead

Many fun challenges

- Consider more powerful attackers
- Super-optimization for Jasmin (CryptOpt, SLOTHY. . .)
- Link primitive and protocol proofs
- Go from protocols to applications

One elephant in the room: **Scale!**

- Cannot take 6 years per primitive
- Already massively improved tooling
- Protocols more complex
- Applications *even* more complex
- **GenAI to the rescue?**

What I didn’t see coming in 2019: open source

Hang on, open source?

"It is very important to me that all my results can be freely used by anyone without any restrictions. I therefore place all my software in the public domain and make it available online."

—My Veni proposal (2013)

"Since the beginning of his academic career, the PI makes all results of his research, including software, freely available. All software on the PI's website `https://cryptojedi.org` *is in the public domain (if requested via the Creative Commons Zero Waiver) to maximize reusability of results."*

—My ERC proposal (2017)

The frustration of a software guy

"The fault attacker considered in this paper is thus limited to injecting only one fault per scalar multiplication. We assume that this fault may either skip a short block of consecutive instructions, set an arbitrary register to zero, or set an arbitrary register value to a random value [...] we also exclude the instruction pointer from the set of "arbitrary" registers that the attacker may fault."

—Batina, Chmielewski, Haase, Samwel, Schwabe, 2023.

The frustration of a software guy

- Some attacks cannot be addressed in software (alone)
- Hardware we can work with does not have suitable protections
- Hardware with suitable protections is **very** closed

The frustration of a software guy

https://dining.ucsb.edu/dining-commons/de-la-guerra-dining-commons

The frustration of a software guy

SCA-secure ECC in software – mission impossible?

The frustration of a software guy

SCA-secure ECC in software – mission impossible?

Mission impossible.

Open-source silicon

- RISC-V: open-source ISA
  - Started in 2010 at UC Berkeley
  - Multiple open-source implementations
  - Multiple commercial chips, now widely deployed
  - Security-minded design, but not purpose-built for crypto

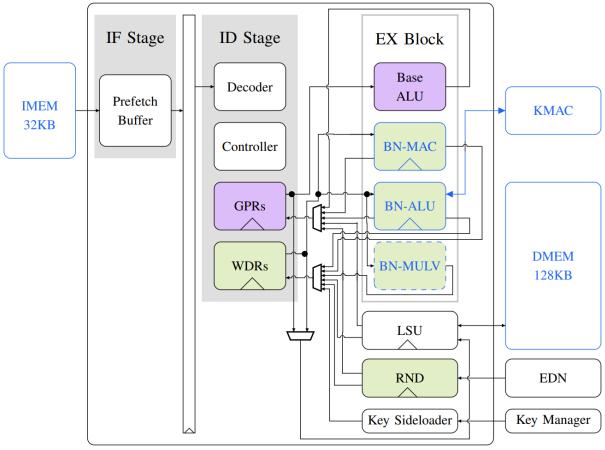
Open-source silicon

- RISC-V: open-source ISA
  - Started in 2010 at UC Berkeley
  - Multiple open-source implementations
  - Multiple commercial chips, now widely deployed
  - Security-minded design, but not purpose-built for crypto
- OpenTitan
  - Public launch in 2019
  - Open-source HW root of trust
  - Involvement from academia and industry
  - **Purpose-built for crypto**

PQC as a catalyst for open-source silicon

Migrating open-source silicon to PQC

- Start with PQ secure boot (SLH-DSA)
- HW/SW support ML-KEM & ML-DSA
- Fork of OpenTitan
- OTBN → ACC
- Small area increase, massive speed boost
- Preparing industry and research tapeouts



Industry Partners

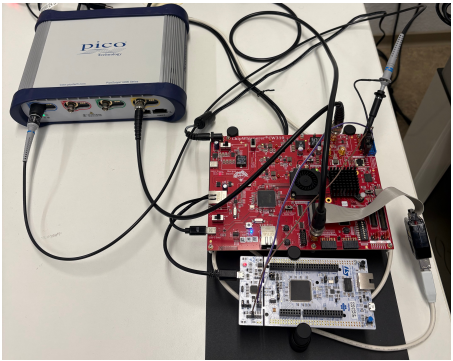
zeroRISC

Rivos

PQC as a catalyst for open-source silicon

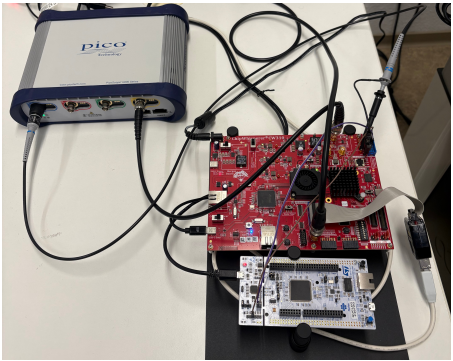
More powerful attacker: From software SCA to hardware SCA

- What can/should HW do for SW?
- What can/should SW do for HW?



More powerful attacker: From software SCA to hardware SCA

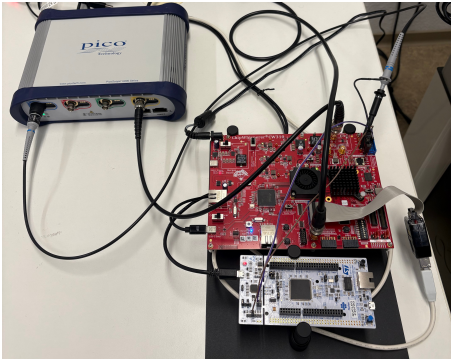
- What can/should HW do for SW?
- What can/should SW do for HW?
- What are the tradeoffs for principled SCA resistance?
  - What kind of noise actually matters?
  - Cost for increasing that noise (in HW and/or SW)
  - Cost for amplifying noise (masking order)
- Similar questions for fault injection. . .



Principled implementation security (part 2)

More powerful attacker: From software SCA to hardware SCA

- What can/should HW do for SW?
- What can/should SW do for HW?
- What are the tradeoffs for principled SCA resistance?
  - What kind of noise actually matters?
  - Cost for increasing that noise (in HW and/or SW)
  - Cost for amplifying noise (masking order)
- Similar questions for fault injection. . .



Goal: End-to-end tooling for principled SCA/FI security on open-source silicon

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,
- running on the ACC,

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,
- running on the ACC,
- with computer verified proofs from gate-level to IND-CCA, SUF-CMA, etc.,

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,
- running on the ACC,
- with computer verified proofs from gate-level to IND-CCA, SUF-CMA, etc.,
- with formal models of powerful real-world implementation attacks,

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,
- running on the ACC,
- with computer verified proofs from gate-level to IND-CCA, SUF-CMA, etc.,
- with formal models of powerful real-world implementation attacks,
- carrying computer-verified proofs of security against these attacks,

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,
- running on the ACC,
- with computer verified proofs from gate-level to IND-CCA, SUF-CMA, etc.,
- with formal models of powerful real-world implementation attacks,
- carrying computer-verified proofs of security against these attacks,
- offering well-defined interfaces to, e.g., Signal and PQ-eIDAS, and

Bringing it all together

Think

- ML-KEM, ML-DSA, and post-quantum anonymous credentials,
- implemented in Jasmin,
- running on the ACC,
- with computer verified proofs from gate-level to IND-CCA, SUF-CMA, etc.,
- with formal models of powerful real-world implementation attacks,
- carrying computer-verified proofs of security against these attacks,
- offering well-defined interfaces to, e.g., Signal and PQ-eIDAS, and
- proofs of security extending to these protocols and applications.

A massive team effort

Aaron Kaiser, Ádám Vécsi, Adrien Koutsos, Alfredo Martins, Amber Sprenkels, Amin Abdulrahman, Anirban Chakraborty, Andreas Hülsing, Andreas Zankl, Anna Stefano Narivelomanana, Antoine Séré, Azade Rezaeezade, Basavesh Ammanaghatta Shivakumar, Bas Westerbaan, Benjamin Dowling, Benjamin Gregoire, Benjamin Grégoire, Benoît Viguier, Björn Haase, Bor de Kock, Cameron Low, Chitchanok Chuengsatiansup, Daniel De Almeida Braga, Daniel Genkin, David Mateos Romero, David Pichardie, David Wu, Deirdre Connolly, Denisa Greconici, Dominik Hartmann, Douglas Stebila, Eike Kiltz, Erdem Alkim, Fabio Campos, Felix Mölder, Felix Oberhansl, Fiona Weber, Francisca Barros Francisco Rodríguez-Henríquez, François Dupressoir, Freek Wiedijk, Gilles Barthe, Giulio Malavolta, Goutam Tamvada, Gustavo Xavier Delerue Marinho Alves, Hoang Nguyen Hien Pham, Hugo Pacheco, Ignacio Cuevas, Ko Stoffelen, Kristin Habersang, Jack Barnes, Jade Philipoom, Jan Jancar, Jean-Christophe Léchenet, Jesús-Javier Chi-Domínguez, Jieyu Zheng, João Diogo Duarte, Joël Alwen, Joost Rijneveld, Jorge Chavez-Saab, José Bacelar Almeida, Julius Hermelink, Justus Bergermann, Kai-Chun Ning, Karolin Varner, Kevin Liao, Krijn Reijnders, Lejla Batina, Lucas Tabary-Maujean, Luís Esquível, Łukasz Chmielewski, Manuel Barbosa, Marcel Böhme, Marcel Fourné, Marco Guarnieri, Marta Mularczyk, Matthias J. Kannwischer, Michael Meyer, Miguel Quaresma, Mohamed Sabt, Monika Trimoska, Niels Samwel, Noemi Terzo, Parisa Amiri Eliasi, Philip R. Zimmermann, Phillip Gajland, Pierre-Alain Fouque, Pierre-Yves Strub, Pol Van Aubel, Richard Petri, Rolfe Schmidt, Ruben Gonzalez, Ruben Niederhagen, Rui Qi Sim, Sabrina Manickam, Santiago Arranz Olmos, Sioli O'Connell, Solène Mirliaz, Sunjay Cauligi, Swarn Priya, Swechchha Adhikari, Tamara Rezk, Thing-han Lim, Ting-Yun Yeh, Thom Wiggers, Tiago Oliveira, Timmy Weerwag, Tobias Stelzer, Vincent Hwang, Vincent Krämer, Vincent Laporte, William Wang, Ushana Bandyopadhyay, Yasemin Acar, Yawning Angel, Yuval Yarom, Zhiyuan Zhang . . . **and many more before 2020**

A massive team effort

From EPOQUE to epoqɔ

https://epoqc.org

stay tuned. . .