An update on NIST's PQC standardization process

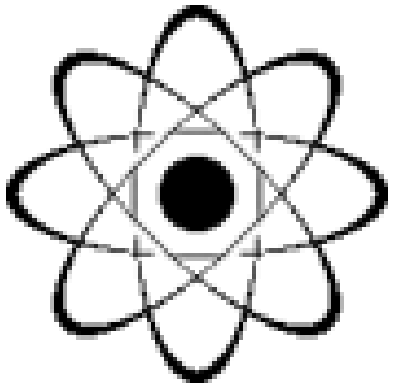
Andreas Huelsing
Eindhoven University of Technology

29.11.2022

This talk is biased

- My background is in theory and design
- I was involved in three submissions: SPHINCS+, NTRU & MQDSS

Quantum Quantum Quantum Quantum



Communication security (simplified)

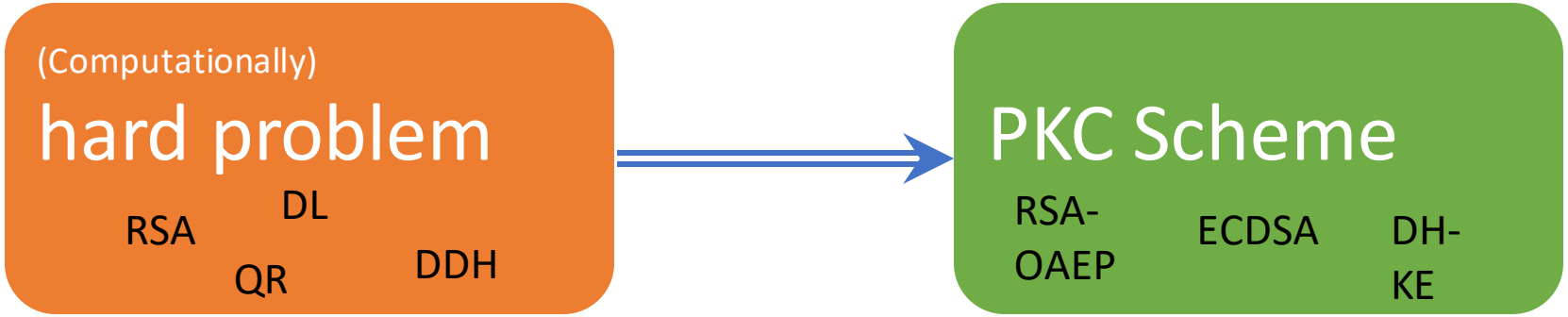
Hi

pk, Cert(pk belongs to shop)

PKC to establish shared secret sk

SKC secured communication using sk

How to build PKC



The quantum threat

Algorithms for Quantum Computation: Discrete Logarithms and Factoring

Peter W. Shor
AT&T Bell Labs
Room 2D-149
600 Mountain Ave.
Murray Hill, NJ 07974, USA

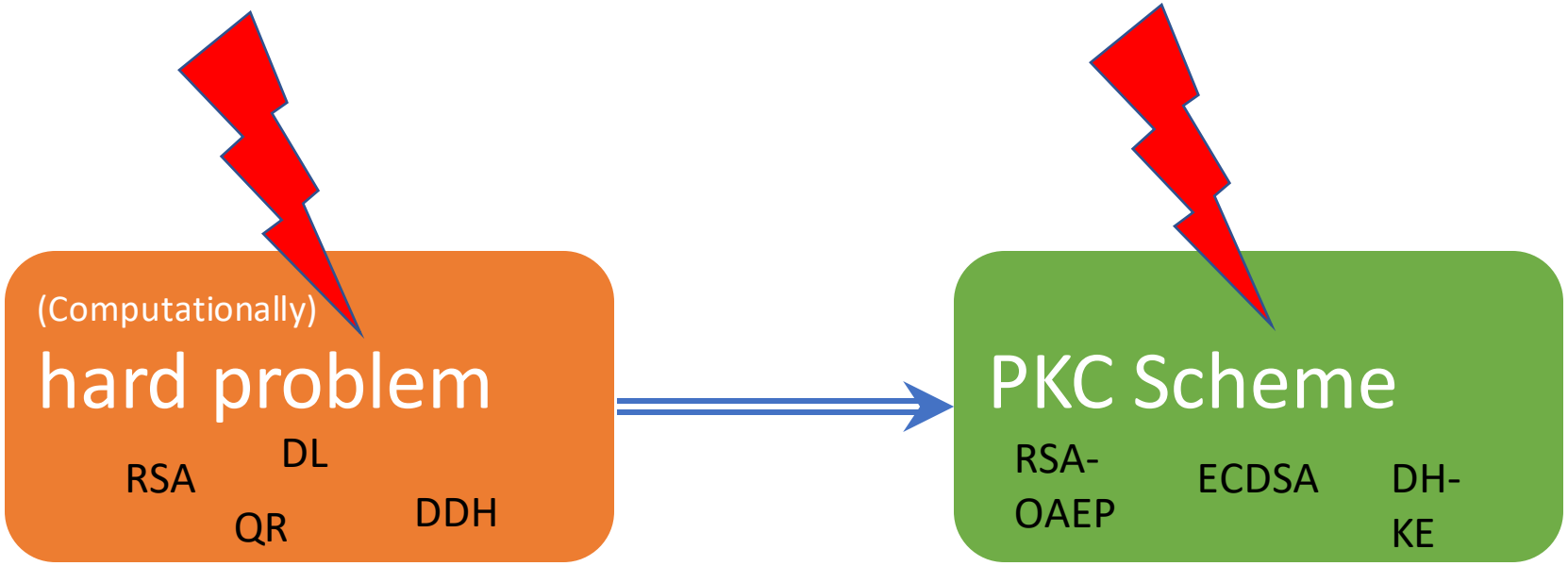
Abstract

A computer is generally considered to be a universal computational device; i.e., it is believed able to simulate any physical computational device with a cost in computation time of at most a polynomial factor. It is not clear whether this is still true when quantum mechanics is taken into consideration. Several researchers, starting with David Deutsch, have developed models for quantum mechanical computers and have investigated their computational properties. This paper gives Las Vegas algorithms for finding discrete logarithms and factoring integers on a quantum computer that take a number of steps which is polynomial in the input size, e.g., the number of digits of the integer to be factored. These two problems are generally considered hard on a classical computer and have been used as the basis of several proposed cryptosystems. (We thus give the first examples of quantum cryptanalysis.)

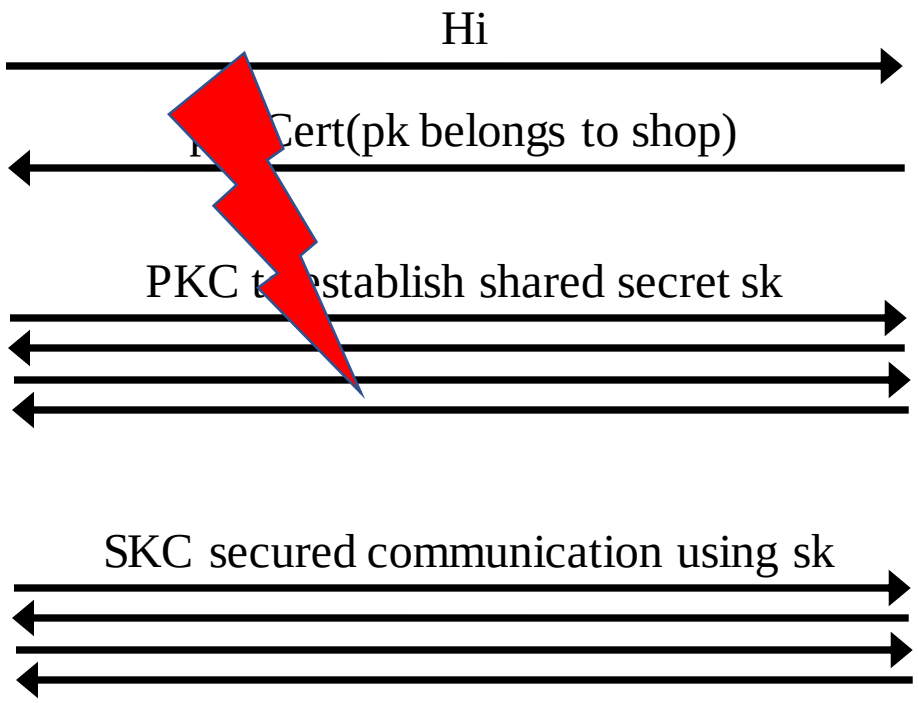
[1, 2]. Although he did not ask whether quantum mechanics conferred extra power to computation, he did show that a Turing machine could be simulated by the reversible unitary evolution of a quantum process, which is a necessary prerequisite for quantum computation. Deutsch [9, 10] was the first to give an explicit model of quantum computation. He defined both quantum Turing machines and quantum circuits and investigated some of their properties.

The next part of this paper discusses how quantum computation relates to classical complexity classes. We will thus first give a brief intuitive discussion of complexity classes for those readers who do not have this background. There are generally two resources which limit the ability of computers to solve large problems: time and space (i.e., memory). The field of analysis of algorithms considers the asymptotic demands that algorithms make for these resources as a function of the problem size. Theoretical computer scientists generally classify algorithms as effi-

How to build PKC



Communication security (simplified)



It‘s a question of risk assessment

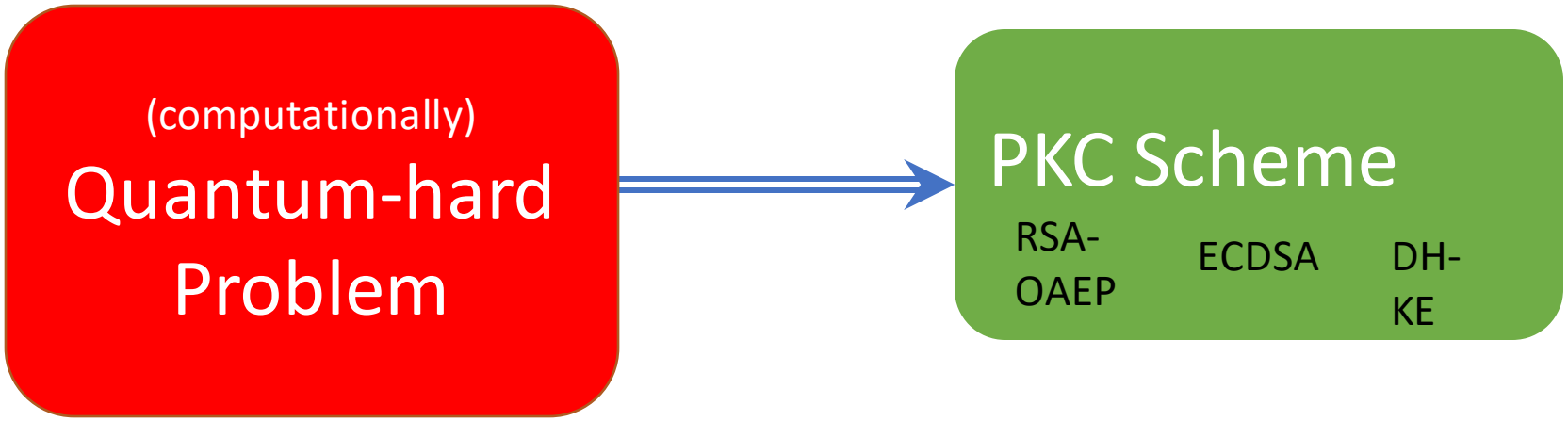
Why should we care today?
Store now, decrypt later

Long-lived systems

- Development time easily 10+ years
- Lifetime easily 10+ years

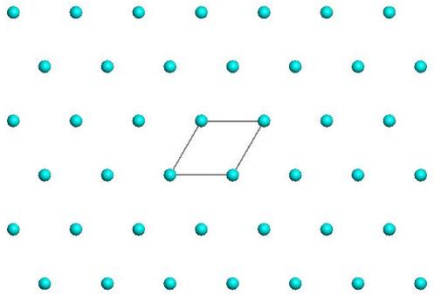
- At least make sure you got a secure update channel!

How to build PQC

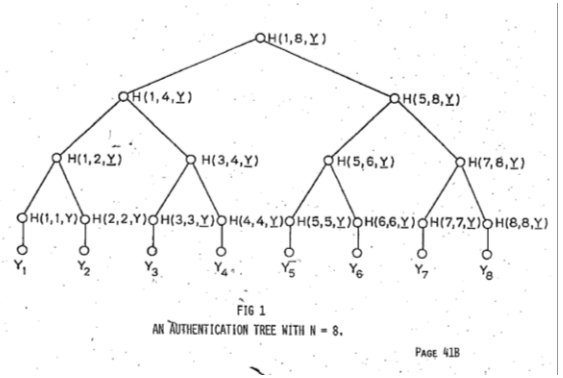


(Conjectured) Quantum-hard problems

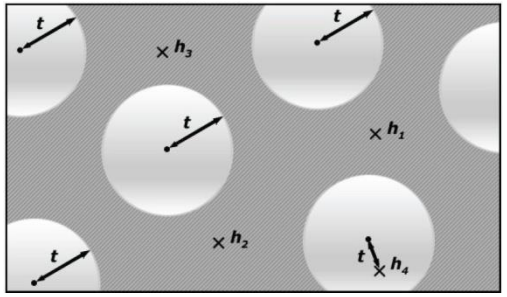
Lattice-based: SVP / CVP



Hash-based: CR / SPR / ...



Code-based: SD



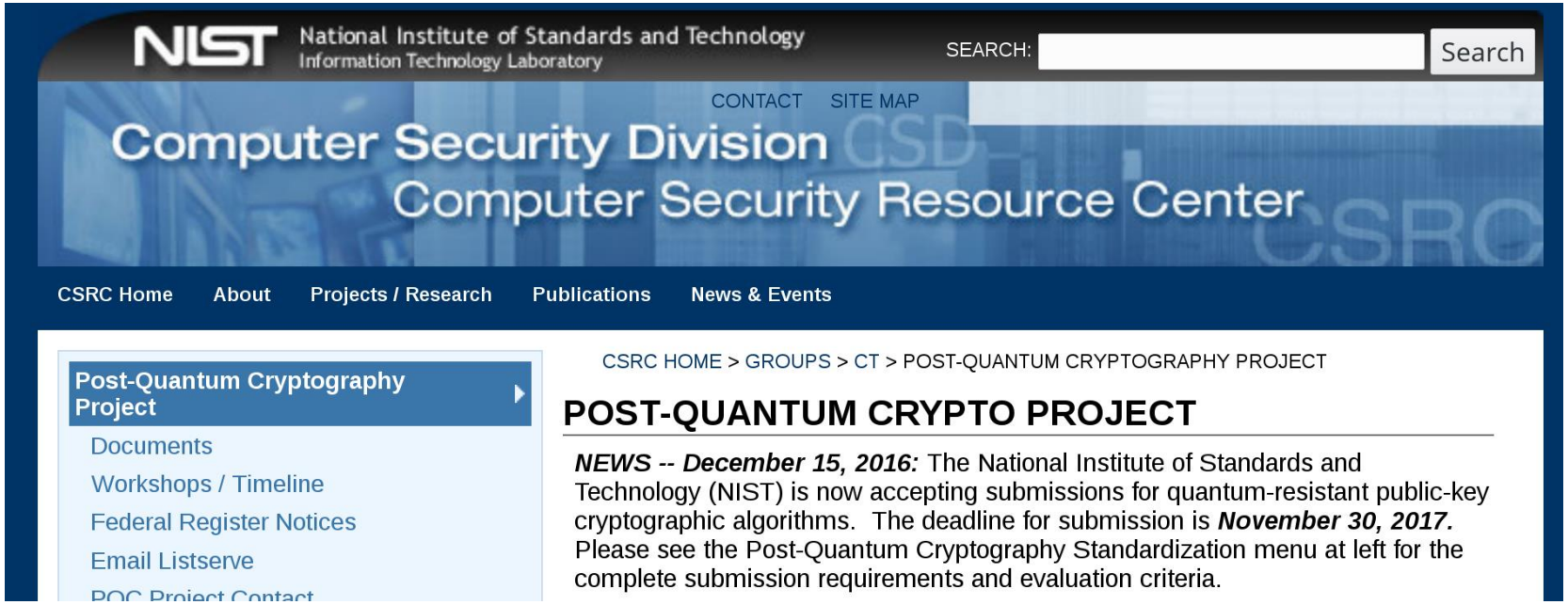
Multivariate: MQ

$$y_1 = x_1^2 + x_1x_2 + x_1x_4 + x_3$$

$$y_2 = x_3^2 + x_2x_3 + x_2x_4 + x_1 + 1$$

$$y_3 = \ldots$$

NIST Competition



"We see our role as managing a process of achieving community consensus in a transparent and timely manner" NIST's Dustin Moody 2018

The competition

What is there to do?

- Research on PQC consciously happened since early 2000's
- Handful of schemes known
- Just pick one?

A million details & trade-offs

Case study lattice-based encryption

What object? Matrix? Poly? Matrix of poly's?

Almost all proposals used Regev / LPR:

- PK = (A, b) = (A, As+e) ; SK = s
- Enc ((A,b), m):
  - $u = A^T s' + e'$
  - $v = b^T s' + e'' + \text{encoding}(m)$
- Dec (s, (u,v)):
  - $t = v - u^T s$
    $= b^T s' + e'' + \text{encoding}(m) - s'^T A s + e'^T s$
    $= (As)^T s' + e^T s' + e'' - s'^T A s + e'^T s + \text{encoding}(m) = \text{encoding}(m) + \text{small}$
  - m' = decode(t)

What distribution? Uniform? Gaussian?

What means small? $\|\cdot\|_2$? $\|\cdot\|_\infty$?

What encoding?

Which exact problem to relate security to?

e.g., lattice-based schemes:

- SIS or LWE or both?
- Plain, ring, or module version?
- For LWE, random error or deterministic error via rounding?
- What error and secret distribution?
- What norm to apply?

Only the beginning… FO (PKE -> KEM)

- Explicit or implicit rejection?
- Key confirmation hash?
- Include PK in hashes? Maybe just an identifier? The hash? First n bytes?

How to implement a random oracle that maps to a funny set?

- Rejection sampling?
- Over sampling & rounding?
- What hash function to use?

Submission

- Nov 2017: 82 submissions collected
- Dec 2017: 69 “complete & proper” proposals published
  - 45 KEM of which 21 based on lattices! (And 17 on codes)

Goal of the competition

- Which schemes are secure?
- What trade-offs, details are the best
- Which schemes are the most efficient?
- Can the schemes be implemented ..
  - ..securely?
  - ..on different platforms?

#1 goal: Verify security

Guess what?!
On the impossibility of unconditionally secure public-key encryption

Lorenz Panny

Department of Mathematics and Computer Science,
Technische Universiteit Eindhoven, The Netherlands
lorenz@yx7.cc

Abstract. We (once again) refute recurring claims about a public-key encryption scheme that allegedly provides unconditional security. This is approached from two angles: We give an information-theoretic proof of impossibility, as well as a concrete attack breaking the proposed scheme in essentially no time.

Keywords: public-key cryptography · perfect secrecy · information theory · impossibility · cryptanalysis

1 Introduction

In 2017, *Guess Again*, a public-key encryption scheme claiming *unconditional* security against passive eavesdroppers, was submitted to NIST's call for post-quantum cryptography [1]. Although we publicly broke that scheme with a fast attack script about three hours after the proposals were published by NIST [6], the authors still have not acknowledged the attack nor withdrawn their proposal (though NIST deselected it from advancing to the second round). About

https://eprint.iacr.org/2019/1228.pdf

#1 goal: Verify security

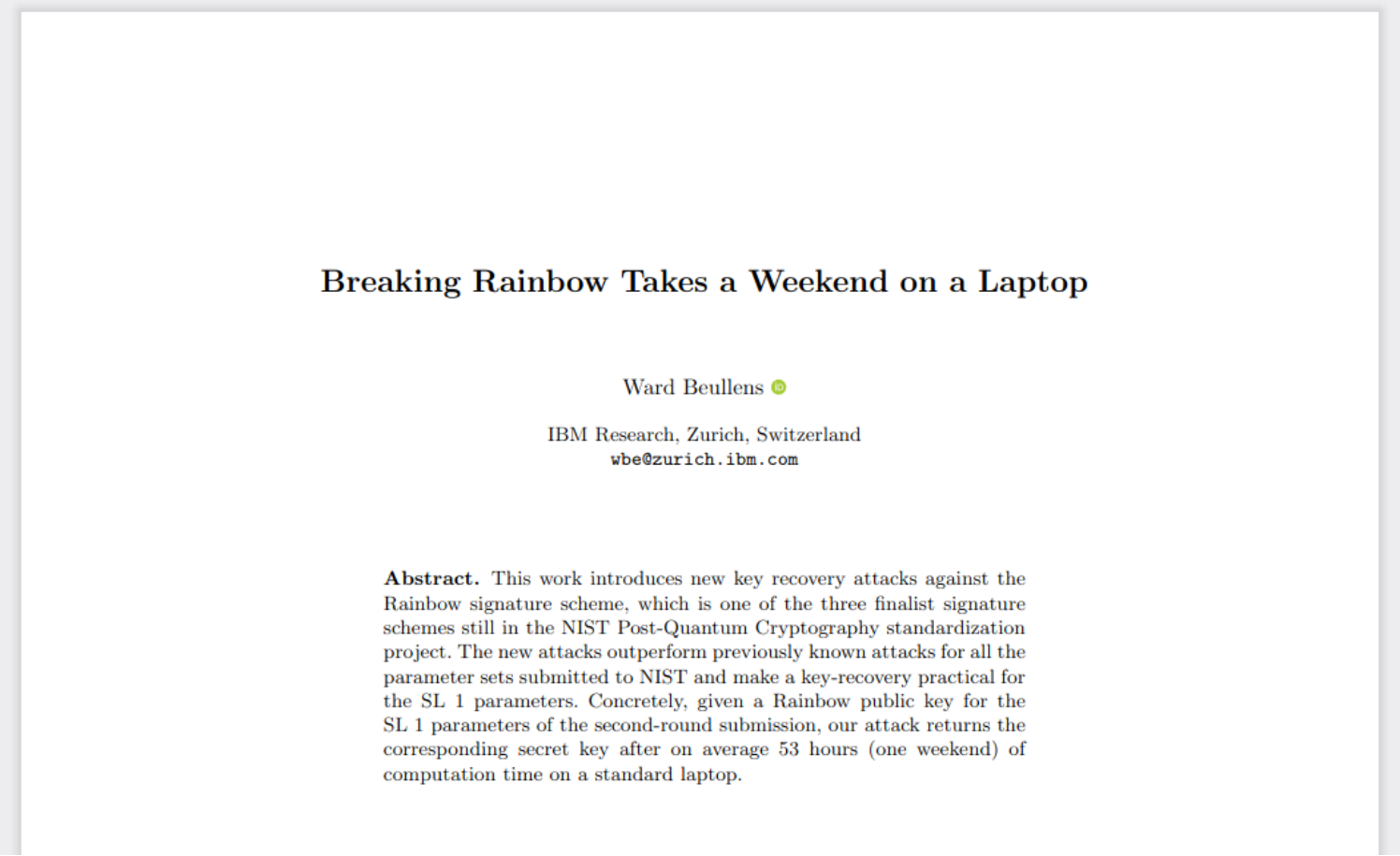
- First 3 weeks: 12 schemes broken or significantly attacked
  - 5 more withdrawn
- Next 4 months: 4 more broken or significantly attacked
- Total of 18 schemes withdrawn or rejected after 1st round due to security
- … then a long silence (except discussions about precise security levels)

#1 goal: Verify security

#1 lesson:

Be afraid of what your laptops do on the weekend!

#1 goal: Verify security



Breaking Rainbow Takes a Weekend on a Laptop

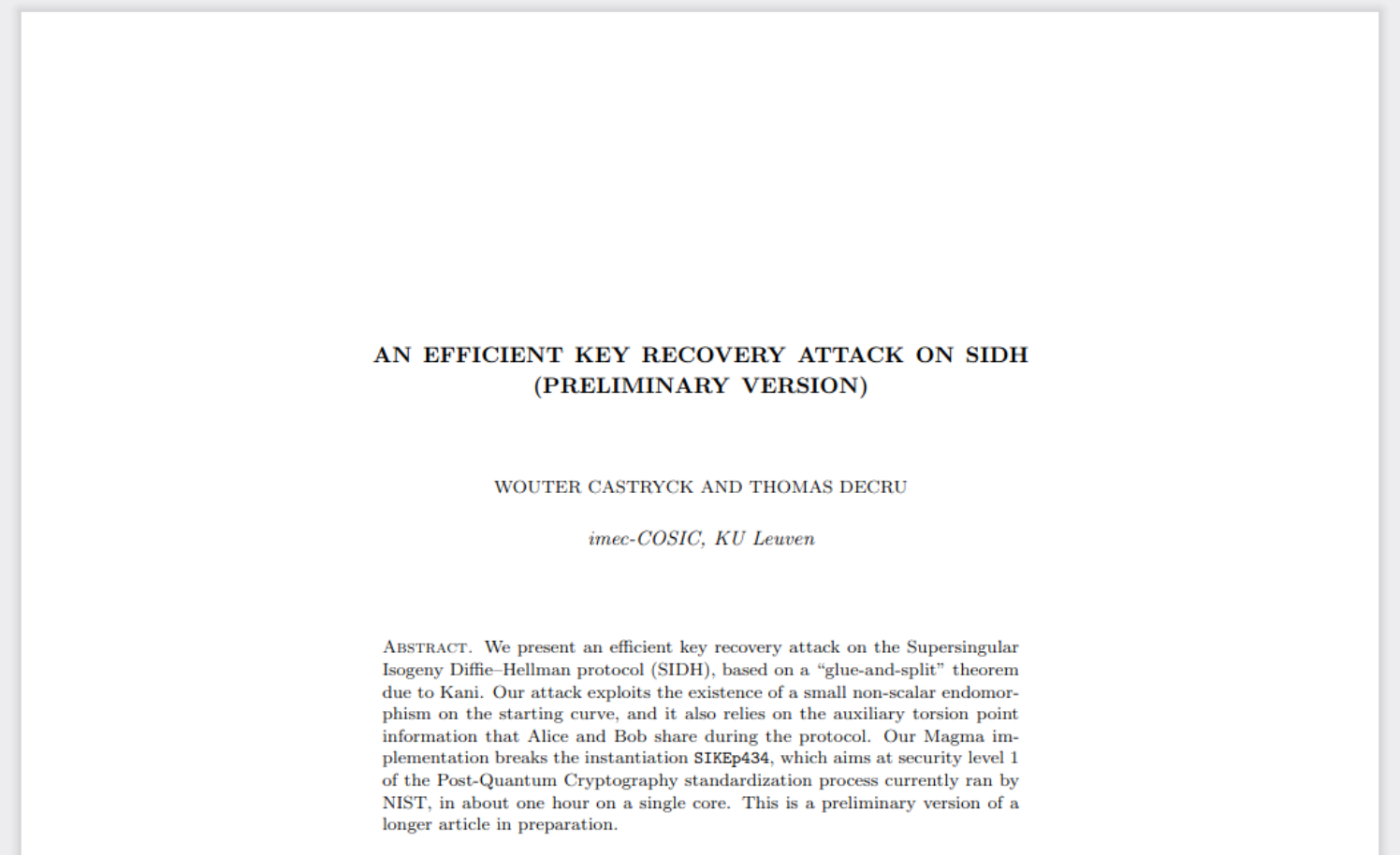
Ward Beullens

IBM Research, Zurich, Switzerland
wbe@zurich.ibm.com

Abstract. This work introduces new key recovery attacks against the Rainbow signature scheme, which is one of the three finalist signature schemes still in the NIST Post-Quantum Cryptography standardization project. The new attacks outperform previously known attacks for all the parameter sets submitted to NIST and make a key-recovery practical for the SL 1 parameters. Concretely, given a Rainbow public key for the SL 1 parameters of the second-round submission, our attack returns the corresponding secret key after on average 53 hours (one weekend) of computation time on a standard laptop.

https://eprint.iacr.org/2022/214.pdf

#1 goal: Verify security



AN EFFICIENT KEY RECOVERY ATTACK ON SIDH (PRELIMINARY VERSION)

WOUTER CASTRYCK AND THOMAS DECRU

imec-COSIC, KU Leuven

Abstract. We present an efficient key recovery attack on the Supersingular Isogeny Diffie–Hellman protocol (SIDH), based on a "glue-and-split" theorem due to Kani. Our attack exploits the existence of a small non-scalar endomorphism on the starting curve, and it also relies on the auxiliary torsion point information that Alice and Bob share during the protocol. Our Magma implementation breaks the instantiation `SIKEp434`, which aims at security level 1 of the Post-Quantum Cryptography standardization process currently ran by NIST, in about one hour on a single core. This is a preliminary version of a longer article in preparation.

https://eprint.iacr.org/2022/975.pdf

Security

- Rainbow & GeMSS came from a class of MQ-based ad-hoc constructions with a troubled history
  - Attacks unexpected but not totally surprising
- Isogenies were the latest family added to the zoo.
  - However, they did accumulate trust!
    - Many smart people tried to break schemes.
  - Break at this point surprising
- Lattice crypto probably received most attention
  - No break
  - Slight movements in parameters

Further criteria

- Performance:
  - Size
  - Speed (on different platforms)
  - Main reason for schemes to be rejected after round 2
- Ease of (side-channel resistant) implementation

Round 3 - KEM

Lattice

- Kyber (finalist): MLWE, random error
- Saber (finalist): MLWR, rounding
- NTRU (finalist): NTRU assumption, random error
- Frodo (alternate): LWE, random error
- NTRUPrime (alternate): NTRU over different ring with less structure, random error & rounded version

Codes

- Classic McEliece (finalist): goppa codes (close to random codes)
- BIKE (alternate): structured codes
- HQC (alternate): structured codes

SIKE (R.I.P.)

Round 3 - Signatures

Lattice

- Dilithium (finalist): SelfTargetMSIS + MLWE, “Fiat-Shamir with aborts”
- Falcon (finalist): NTRU, “Full domain hash”

MQ

- Rainbow (finalist):R.I.P.
- GeMSS (alternate): R.I.P

Picnic (alternate): LowMC, MPCitH

SPHINCS+ (alternate): Hash, SPHINCS

And the winner is...

The signature schemes

Scheme	Assumption	Sign speed	Verify speed	Sig size	PK size	Impl. Difficulty	Maturity
Dilithium	Lattice						
Falcon	Lattice						
SPHINCS+	Hash						

good	fair	poor	bad

Based on slide by Maran Heesch (TNO) that was made in consultation with Leo Ducas (Dilithium), Thomas Prest (Falcon) and me.

The KEM: Christals-Kyber

Scheme	Assumption	Encaps speed	Decaps speed	ct size	PK size	Impl. Difficulty	Maturity
Kyber	Lattice						

Reasoning?

- Lattice schemes have best overall performance
- NIST wants diversification
- Signatures:
  - MQ broken, LowMC (Picnic) security questionable
  - Leaves SPHINCS+
- KEM:
  - SIKE gone,
  - Frodo, NTRU & NTRUPrime rejected based on performance,
  - Saber rejected due to rounding
  - No decision for code-based yet (waiting for more analysis of structured codes?)

Open topic is still IPR

- NIST is in negotiation with different parties
- NTRU as fallback
- E.g., Google still uses NTRU, not Kyber for internal security (https://cloud.google.com/blog/products/identity-security/why-google-now-uses-post-quantum-cryptography-for-internal-comms)

Next steps

Signature call

- NIST is requesting new submissions for signature schemes
- Interest in general purpose signatures not based on lattices
- Also schemes with very small sigs and fast verification interesting

NIST only gives us KEM + SIG

- Sad news:
  - No post-quantum DH
  - PQC performance != ECC performance
  - Many agencies require hybrid
- Need to redesign protocol:
  - Exploit that KEM more efficient than SIG
- Basic communication security well under way:
  - See PQWireGuard, KEM-TLS, PQNoise, PQConnect, …
- More advanced things still open:
  - PAKE, Deniable authenticated key exchange,...

Integration work begins

- Define key & object formats
- Specify PQC versions of protocols
- Decide how to do hybrid right
- How to smoothly transition?

Security proofs

Proofs for PQC turn out to be even more complex than security proofs of traditional schemes.

There are a lot of ways to fail:

- Wrong proof
- Wrong statement proven
- Too loose bound
- …

Way out?

- Good news:
  - Issues only slightly lowered security / could be easily fixed
- Bad news:
  - Look at OCB2
- Fix: Machine-checked proofs
  - Catch wrong proofs
  - When linked to implementation, make at least all assumptions clear
  - Work in progress (Kyber, Dilithium, SPHINCS+)

Thank you! Questions?

- Post-Quantum Cryptography – Integration study.
  ENISA report.
  https://www.enisa.europa.eu/publications/post-quantum-cryptography-integration-study
- Post-Quantum Cryptography: Current state and quantum mitigation.
  ENISA report.
  https://www.enisa.europa.eu/publications/post-quantum-cryptography-current-state-and-quantum-mitigation/at_download/fullReport
- PQC Summer School material (2019).
  https://www.pqcschool.org/
- Machine checked proofs (for PQC and more):
  https://formosa-crypto.org/