MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

Engineering Post-Quantum Cryptography

May 4, 2022

A brief introduction

- 2001–2007: Aachen

A brief introduction

- 2001–2007: Aachen
- 2008–2011: Eindhoven

A brief introduction

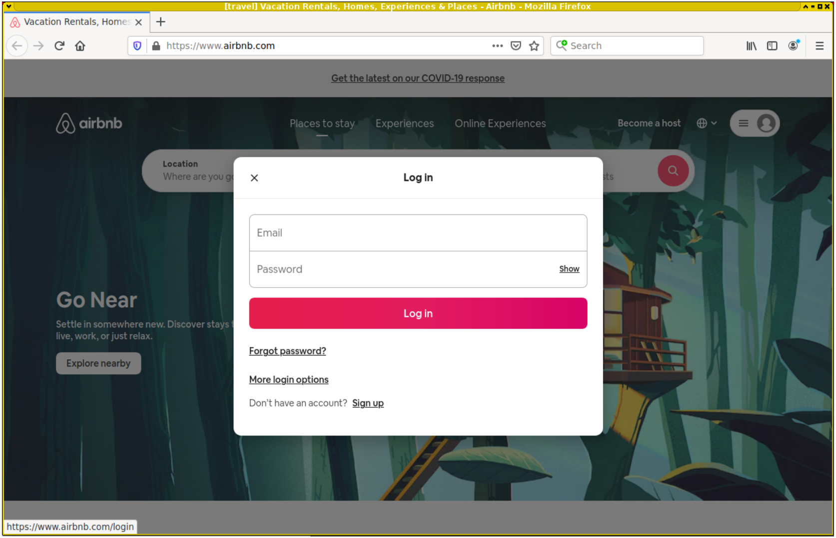
- 2001–2007: Aachen
- 2008–2011: Eindhoven
- 2011–2012: Taipei

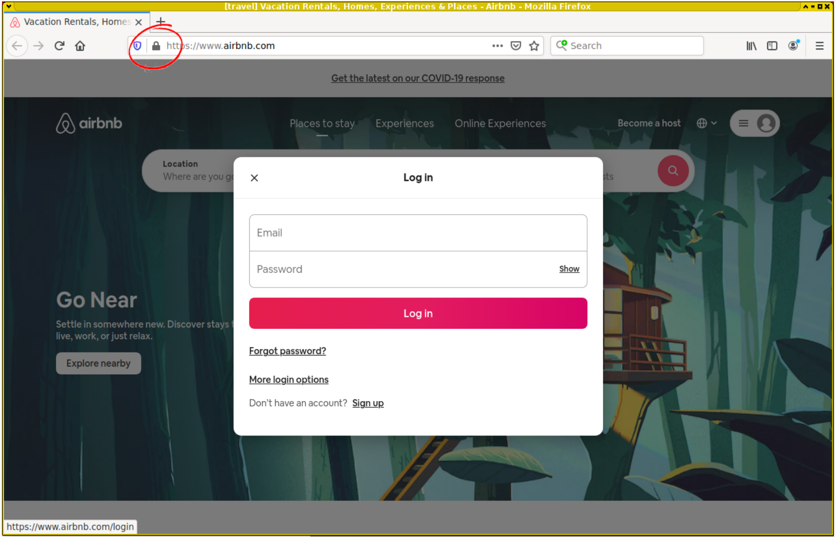
A brief introduction

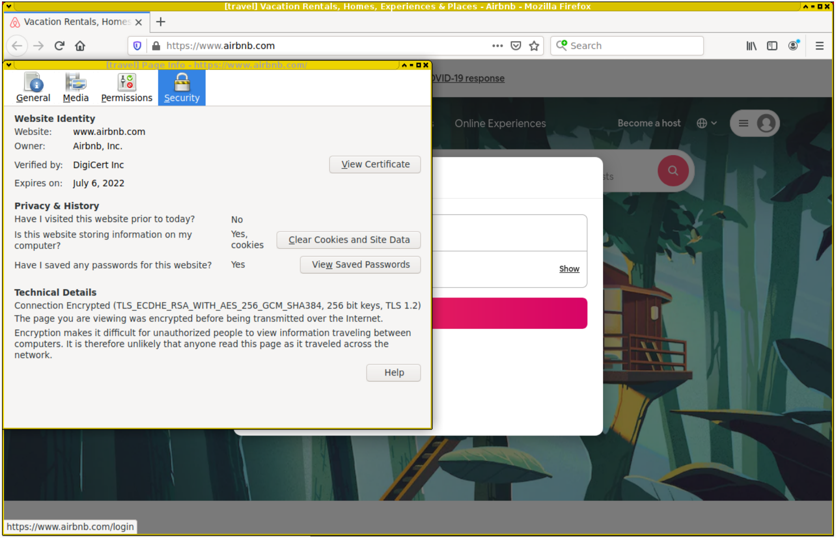
- 2001–2007: Aachen
- 2008–2011: Eindhoven
- 2011–2012: Taipei
- Since 2013: Nijmegen

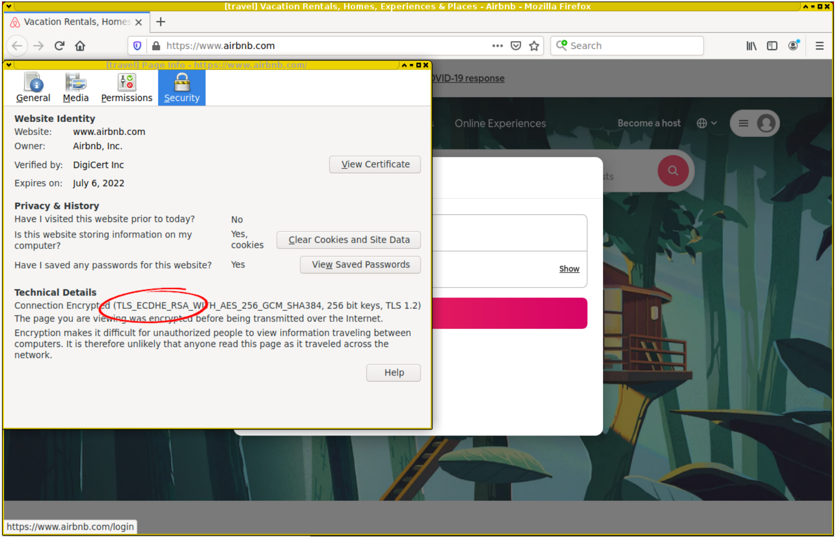
Since Sep. 2020: MPI-SP

Since Sep. 2020: MPI-SP


[travel] Vacation Rentals, Homes, Experiences & Places - Airbnb - Mozilla Firefox
Vacation Rentals, Homes
https://www.airbnb.com
Search
Get the latest on our COVID-19 response
airbnb
Places to stay
Experiences
Online Experiences
Become a host
Location
Where are you g
Log in
Email
Password
Show
Log in
Forgot password?
More login options
Don't have an account? Sign up
Go Near
Settle in somewhere new. Discover stays t
live, work, or just relax.
Explore nearby
https://www.airbnb.com/login


[travel] Vacation Rentals, Homes, Experiences & Places - Airbnb - Mozilla Firefox
Vacation Rentals, Homes
https://www.airbnb.com
Search
Get the latest on our COVID-19 response
airbnb
Places to stay
Experiences
Online Experiences
Become a host
Location
Where are you g
Log in
Email
Password
Show
Log in
Forgot password?
More login options
Don't have an account? Sign up
Go Near
Settle in somewhere new. Discover stays t
live, work, or just relax.
Explore nearby
https://www.airbnb.com/login


[travel] Vacation Rentals, Homes, Experiences & Places - Airbnb - Mozilla Firefox
Vacation Rentals, Homes
https://www.airbnb.com
Search
[travel] Page Info - https://www.airbnb.com/
General
Media
Permissions
Security
Website Identity
Website: www.airbnb.com
Owner: Airbnb, Inc.
Verified by: DigiCert Inc
View Certificate
Expires on: July 6, 2022
Privacy & History
Have I visited this website prior to today? No
Is this website storing information on my computer? Yes, cookies
Clear Cookies and Site Data
Have I saved any passwords for this website? Yes
View Saved Passwords
Technical Details
Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, 256 bit keys, TLS 1.2)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.
Help
VID-19 response
Online Experiences
Become a host
Show
https://www.airbnb.com/login


[travel] Vacation Rentals, Homes, Experiences & Places - Airbnb - Mozilla Firefox
Vacation Rentals, Homes
https://www.airbnb.com
Search
[travel] Page Info - https://www.airbnb.com/
General
Media
Permissions
Security
Website Identity
Website: www.airbnb.com
Owner: Airbnb, Inc.
Verified by: DigiCert Inc
View Certificate
Expires on: July 6, 2022
Privacy & History
Have I visited this website prior to today? No
Is this website storing information on my computer? Yes, cookies
Clear Cookies and Site Data
Have I saved any passwords for this website? Yes
View Saved Passwords
Technical Details
Connection Encrypted (TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384, 256 bit keys, TLS 1.2)
The page you are viewing was encrypted before being transmitted over the Internet.
Encryption makes it difficult for unauthorized people to view information traveling between computers. It is therefore unlikely that anyone read this page as it traveled across the network.
Help
VID-19 response
Online Experiences
Become a host
Show
https://www.airbnb.com/login

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.

“In the past, people have said, maybe it’s 50 years away, it’s a dream, maybe it’ll happen sometime. I used to think it was 50. Now I’m thinking like it’s 15 or a little more. It’s within reach. It’s within our lifetime. It’s going to happen.”

—Mark Ketchen (IBM), Feb. 2012, about quantum computers

Post-quantum crypto

Definition

Post-quantum crypto is (asymmetric) crypto that resists attacks using classical *and quantum* computers.

Post-quantum crypto

Definition

Post-quantum crypto is (asymmetric) crypto that resists attacks using classical *and quantum* computers.

5 main directions

- Lattice-based crypto (PKE and Sigs)
- Code-based crypto (mainly PKE)
- Multivariate-based crypto (mainly Sigs)
- Hash-based signatures (only Sigs)
- Isogeny-based crypto (so far, mainly PKE)

The NIST PQC “not-a-competition”

- Inspired by two earlier NIST crypto competitions:
  - AES, running from 1997 to 2000
  - SHA3, running from 2007 to 2012

The NIST PQC “not-a-competition”

- Inspired by two earlier NIST crypto competitions:
  - AES, running from 1997 to 2000
  - SHA3, running from 2007 to 2012
- Approach: NIST specifies criteria, everybody is welcome to submit proposals
- Selection through an open process and multiple rounds
- Actual decisions are being made by NIST

The NIST PQC "not-a-competition"

- Inspired by two earlier NIST crypto competitions:
  - AES, running from 1997 to 2000
  - SHA3, running from 2007 to 2012
- Approach: NIST specifies criteria, everybody is welcome to submit proposals
- Selection through an open process and multiple rounds
- Actual decisions are being made by NIST
- PQC project:
  - Announcement: Feb 2016
  - Call for proposals: Dec 2016 (based on community input)
  - Deadline for submissions: Nov 2017

The NIST competition: initial overview

Count of Problem Category	Column Labels		
Row Labels	Key Exchange	Signature	Grand Total
?	1		1
Braids	1	1	2
Chebychev	1		1
Codes	19	5	24
Finite Automata	1	1	2
Hash		4	4
Hypercomplex Numbers	1		1
Isogeny	1		1
Lattice	24	4	28
Mult. Var	6	7	13
Rand. walk	1		1
RSA	1	1	2
Grand Total	**57**	**23**	**80**

Overview tweeted by Jacob Alperin-Sheriff on Dec 4, 2017.

The NIST competition: Jan 2019

- Announcement planned at Real-World Crypto 2019

The NIST competition: Jan 2019

- Announcement planned at Real-World Crypto 2019
- Due to US government lockdown slightly later

The NIST competition: Jan 2019

- Announcement planned at Real-World Crypto 2019
- Due to US government lockdown slightly later

Encryption / Key agreement

- 9 lattice-based
- 7 code-based
- 1 isogeny-based

The NIST competition: Jan 2019

- Announcement planned at Real-World Crypto 2019
- Due to US government lockdown slightly later

Encryption / Key agreement

- 9 lattice-based
- 7 code-based
- 1 isogeny-based

Signature schemes

- 3 lattice-based
- 2 symmetric-crypto based
- 4 MQ-based

The NIST competition: Jul 2020

- Announcement planned for June 2020

The NIST competition: Jul 2020

- Announcement planned for June 2020
- Due to pandemic (?) slightly later

The NIST competition: Jul 2020

- Announcement planned for June 2020
- Due to pandemic (?) slightly later

Finalists

- 4 key-agreement schemes
  - 3 lattice-based
  - 1 code-based
- 3 signature schemes
  - 2 lattice-based
  - 1 MQ-based

The NIST competition: Jul 2020

- Announcement planned for June 2020
- Due to pandemic (?) slightly later

Finalists

- 4 key-agreement schemes
  - 3 lattice-based
  - 1 code-based
- 3 signature schemes
  - 2 lattice-based
  - 1 MQ-based

Alternate schemes

- 5 key-agreement schemes
  - 2 lattice-based
  - 2 code-based
  - 1 isogeny-based
- 3 signature schemes
  - 2 symmetric-crypto based
  - 1 MQ-based

The NIST competition right now

"Yes - the 3rd round will shortly be ending. NIST is actively writing the 3rd Round report which will explain our rationale for which algorithms we will standardize. We hope to be able to announce the results and report not later than the end of March."

—Dustin Moody (NIST), Feb 9, 2022

The NIST competition right now

"We wanted to give another quick update.

We've received a few inquiries about whether NIST's talk at the PKC conference next week will be where we announce the candidates we will select for standardization. The short answer is no - it will not be.

We are still hoping to make our announcement by the end of this month (March)."

—Dustin Moody (NIST), Mar 4, 2022

The NIST competition right now

"Another update:

We had been hoping to announce the results of our PQC standardization process by the end of March. We ask for a little bit more patience since we are not ready to make the announcement today. We still expect to make it very soon."

—Dustin Moody (NIST), Mar 31, 2022

The NIST competition right now

"Everybody,

We appreciate your patience. The announcement of the algorithms we will standardize is still coming very soon. This is a major milestone of our project, and the delay is not due to technical considerations but is due to some legal and procedural steps that are taking more time than we anticipated. Again, thank you for your patience."

—Dustin Moody (NIST), Apr 19, 2022

The NIST competition right now

Expectation

- NIST is expected to announce winners **very soon**
- $\approx$ one year later get standards

The NIST competition right now

Expectation

- NIST is expected to announce winners **very soon**
- $\approx$ one year later get standards
- Replace existing crypto with new crypto

The NIST competition right now

Expectation

- NIST is expected to announce winners **very soon**
- $\approx$ one year later get standards
- Replace existing crypto with new crypto

Mission accomplished – The world is safe again!

. . . or is it?

A bit of history: the case of MD5

- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place

A bit of history: the case of MD5

- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest

A bit of history: the case of MD5

- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest
- **1993**: Collisions in MD5 compression function (den Boer, Bosselaers)

A bit of history: the case of MD5

- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest
- **1993**: Collisions in MD5 compression function
  (den Boer, Bosselaers)
- **1996**: Dobbertin, Bosselaers, Preneel: concerns about MD5

A bit of history: the case of MD5

- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest
- **1993**: Collisions in MD5 compression function (den Boer, Bosselaers)
- **1996**: Dobbertin, Bosselaers, Preneel: concerns about MD5
- **2004**: Wang presents MD5 collisions

A bit of history: the case of MD5

- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- 1991: MD5 is proposed by Rivest
- 1993: Collisions in MD5 compression function
  (den Boer, Bosselaers)
- 1996: Dobbertin, Bosselaers, Preneel: concerns about MD5
- 2004: Wang presents MD5 collisions
- 2008: *Rogue CA certificate* using MD5
  (Sotirov, Stevens, Appelbaum, Lenstra, Molnar, Osvik, de Weger)

A bit of history: the case of MD5

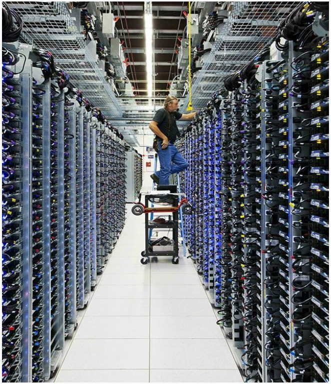
- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest
- **1993**: Collisions in MD5 compression function
  (den Boer, Bosselaers)
- **1996**: Dobbertin, Bosselaers, Preneel: concerns about MD5
- **2004**: Wang presents MD5 collisions
- **2008**: ***Rogue CA certificate*** using MD5
  (Sotirov, Stevens, Appelbaum, Lenstra, Molnar, Osvik, de Weger)
- **2012**: Flame malware exploits MD5 weaknesses

A bit of history: the case of MD5

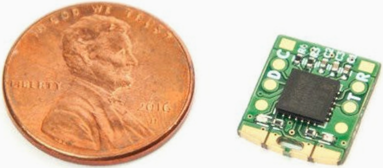
- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest
- **1993**: Collisions in MD5 compression function
  (den Boer, Bosselaers)
- **1996**: Dobbertin, Bosselaers, Preneel: concerns about MD5
- **2004**: Wang presents MD5 collisions
- **2008**: *Rogue CA certificate* using MD5
  (Sotirov, Stevens, Appelbaum, Lenstra, Molnar, Osvik, de Weger)
- **2012**: Flame malware exploits MD5 weaknesses

Replacing MD5 was "easy"!

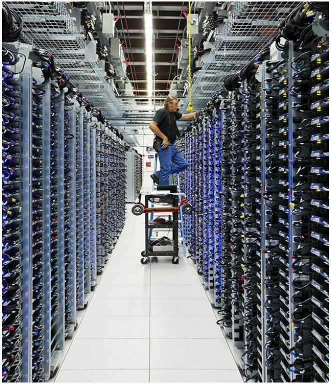
Challenge 1: Performance



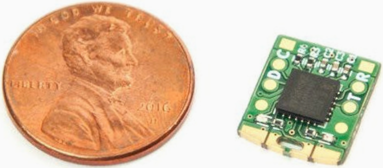
- 10% performance difference matters!
  - Reduce cost for busy servers
  - Fit into constrained devices



Challenge 1: Performance



- 10% performance difference matters!
  - Reduce cost for busy servers
  - Fit into constrained devices
- Small routines executed many times
- Often hand-optimized on assembly level



Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)

Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)
- Operations cost 50–200 kcycles (typical x64 CPU)

Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)
- Operations cost 50–200 kcycles (typical x64 CPU)
- Keys, signtures etc. are 32–64 bytes

Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)
- Operations cost 50–200 kcycles (typical x64 CPU)
- Keys, signtures etc. are 32–64 bytes

PQC performance examples

- McEliece public-key: $\approx$0.5 MB

Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)
- Operations cost 50–200 kcycles (typical x64 CPU)
- Keys, signtures etc. are 32–64 bytes

PQC performance examples

- McEliece public-key: $\approx$0.5 MB
- SPHINCS$^+$ signatures: $\approx$16 KB

Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)
- Operations cost 50–200 kcycles (typical x64 CPU)
- Keys, signtures etc. are 32–64 bytes

PQC performance examples

- McEliece public-key: $\approx$0.5 MB
- SPHINCS$^+$ signatures: $\approx$16 KB
- SPHINCS$^+$ signing: $\approx$3 billion cycles

Challenge 1: Performance (ctd.)

Elliptic-curve cryptography

- State of the art today (but broken by Shor)
- Operations cost 50–200 kcycles (typical x64 CPU)
- Keys, signtures etc. are 32–64 bytes

PQC performance examples

- McEliece public-key: $\approx$0.5 MB
- SPHINCS$^+$ signatures: $\approx$16 KB
- SPHINCS$^+$ signing: $\approx$3 billion cycles
- Kyber (all ops): $<$80 kcycles
- Kyber data sent: $<$1.2 KB

Challenge 2: Security

Security reductions

“An attacker who can break the security can also solve some hard mathematical problem”

Challenge 2: Security

Security reductions

"An attacker who can break the security can also solve some hard mathematical problem"

Great idea

Challenge 2: Security

Security reductions

"An attacker who can break the security can also solve some hard mathematical problem"

Great idea, but. . .

- reductions are often not *tight*

Challenge 2: Security

Security reductions

"An attacker who can break the security can also solve some hard mathematical problem"

Great idea, but. . .

- reductions are often not *tight*
- "hard problem" may turn out to be easier than expected

Challenge 2: Security

Security reductions

"An attacker who can break the security can also solve some hard mathematical problem"

Great idea, but. . .

- reductions are often not *tight*
- "hard problem" may turn out to be easier than expected
- proofs may be wrong

Challenge 2: Security (ctd.)

The case of OCB2

- 2004: Rogaway proposes OCB2
  - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

Challenge 2: Security (ctd.)

The case of OCB2

- 2004: Rogaway proposes OCB2
  - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

Some NIST PQC proof failures

Challenge 2: Security (ctd.)

The case of OCB2

- 2004: Rogaway proposes OCB2
    - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

Some NIST PQC proof failures

- Round-1 Kyber proof does not apply

Challenge 2: Security (ctd.)

The case of OCB2

- 2004: Rogaway proposes OCB2
  - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

Some NIST PQC proof failures

- Round-1 Kyber proof does not apply
- Round-1 SPHINCS$^+$ proof does not apply

Challenge 2: Security (ctd.)

The case of OCB2

- 2004: Rogaway proposes OCB2
  - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

Some NIST PQC proof failures

- Round-1 Kyber proof does not apply
- Round-1 SPHINCS$^+$ proof does not apply
- Round-2 MQDSS attack “hidden inside non-tightness”

Challenge 2: Security (ctd.)

The case of OCB2

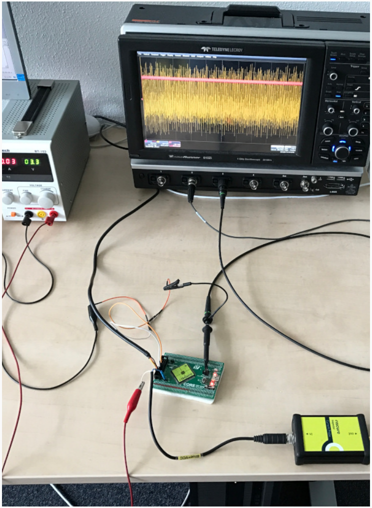
- 2004: Rogaway proposes OCB2
  - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

Some NIST PQC proof failures

- Round-1 Kyber proof does not apply
- Round-1 SPHINCS$^+$ proof does not apply
- Round-2 MQDSS attack "hidden inside non-tightness"
- Round-2 qTesla proof wrong (?) $\Rightarrow$ devastating attack

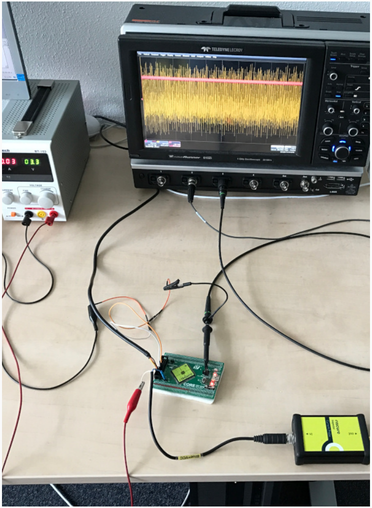
Challenge 3: Implementation Security

Challenge 3: Implementation Security



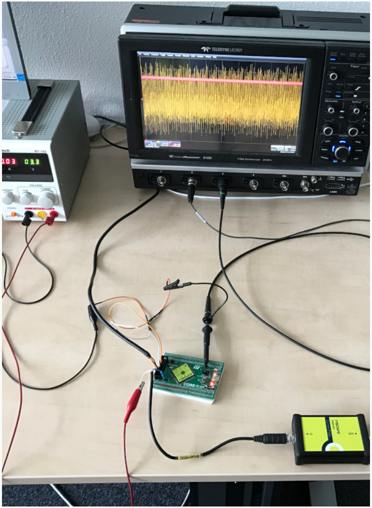
- Attackers see more than input/output:
  - Power consumption
  - Electromagnetic radiation
  - Timing

Challenge 3: Implementation Security



- Attackers see more than input/output:
  - Power consumption
  - Electromagnetic radiation
  - Timing
- Side-channel attacks:
  - Measure information
  - Use to obtain secret data

Challenge 3: Implementation Security



- Attackers see more than input/output:
  - Power consumption
  - Electromagnetic radiation
  - Timing
- **Side-channel attacks:**
  - Measure information
  - Use to obtain secret data
- Timing attacks can be done **remotely**
- Cost of countermeasures heavily depends on the scheme

Challenge 3: Implementation Security (ctd.)

“the implementation security aspect of lattice-based cryptography is still a vastly unexplored and open topic”

— Primas, Pessl, Mangard, 2017.

Challenge 3: Implementation Security (ctd.)

- Quite some work in the space since 2017
- Other “realms” of PQC less explored than lattices
- Even for lattices still very much an open topic

Challenge 3: Implementation Security (ctd.)

- Quite some work in the space since 2017
- Other “realms” of PQC less explored than lattices
- Even for lattices still very much an open topic
- Baseline: *“constant-time”* implementations
- Execution time does not depend on secret data

Challenge 3: Implementation Security (ctd.)

- Quite some work in the space since 2017
- Other “realms” of PQC less explored than lattices
- Even for lattices still very much an open topic
- Baseline: *“constant-time”* implementations
- Execution time does not depend on secret data
- Probably all round-3 schemes have constant-time implementations (?)

Challenge 3: Implementation Security (ctd.)

- Quite some work in the space since 2017
- Other “realms” of PQC less explored than lattices
- Even for lattices still very much an open topic
- Baseline: *“constant-time”* implementations
- Execution time does not depend on secret data
- Probably all round-3 schemes have constant-time implementations (?)
- Very few implementations with advanced countermeasures

Challenge 3: Implementation Security (ctd.)

- Quite some work in the space since 2017
- Other “realms” of PQC less explored than lattices
- Even for lattices still very much an open topic
- Baseline: *“constant-time”* implementations
- Execution time does not depend on secret data
- Probably all round-3 schemes have constant-time implementations (?)
- Very few implementations with advanced countermeasures
- Even worse if we look at *fault attacks*

Challenge 3: Implementation Security (ctd.)

- Quite some work in the space since 2017
- Other “realms” of PQC less explored than lattices
- Even for lattices still very much an open topic
- Baseline: *“constant-time”* implementations
- Execution time does not depend on secret data
- Probably all round-3 schemes have constant-time implementations (?)
- Very few implementations with advanced countermeasures
- Even worse if we look at *fault attacks*

For many applications, implementations are not ready.

Challenge 4: The curious case of Diffie-Hellman

Alice		Bob
$A \leftarrow g^a$		$B \leftarrow g^b$
	$\xrightarrow{\quad A \quad}$	
	$\xleftarrow{\quad B \quad}$	
$K \leftarrow B^a = (g^b)^a = g^{ab}$		$K \leftarrow A^b = (g^a)^b = g^{ab}$

Challenge 4: The curious case of Diffie-Hellman

Alice		Bob
$A \leftarrow g^a$		$B \leftarrow g^b$
	$\xleftarrow{\quad B \quad}$	
	$\xrightarrow{\quad A \quad}$	
$K \leftarrow B^a = (g^b)^a = g^{ab}$		$K \leftarrow A^b = (g^a)^b = g^{ab}$

KEMs: as close as you'll get to DH

Initiator		Responder
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KEM.Gen}$		
	$\xrightarrow{\mathsf{pk}}$	
		$(\mathsf{ct}, K) \leftarrow \mathsf{KEM.Enc}(\mathsf{pk})$
	$\xleftarrow{\mathsf{ct}}$	
$K \leftarrow \mathsf{KEM.Dec}(\mathsf{ct}, \mathsf{sk})$		

KEMs: as close as you'll get to DH*

Initiator		Responder
$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KEM.Gen}$		
	$\xrightarrow{\mathsf{pk}}$	
		$(\mathsf{ct}, K) \leftarrow \mathsf{KEM.Enc}(\mathsf{pk})$
	$\xleftarrow{\mathsf{ct}}$	
$K \leftarrow \mathsf{KEM.Dec}(\mathsf{ct}, \mathsf{sk})$		

*Except with CSIDH (Castryck, Lange, Martindale, Renes, Panny, 2018)

Challenge 5: “Huge foot cannons”

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS

Challenge 5: “Huge foot cannons”

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
  - Reasonable performance
  - Reasonable signature sizes
  - Small keys

Challenge 5: “Huge foot cannons”

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
    - Reasonable performance
    - Reasonable signature sizes
    - Small keys
    - Application-specific tradeoffs
    - Conservative security

Challenge 5: “Huge foot cannons”

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
    - Reasonable performance
    - Reasonable signature sizes
    - Small keys
    - Application-specific tradeoffs
    - Conservative security
- NIST fast-track standardization of XMSS and LMS

Challenge 5: “Huge foot cannons”

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
  - Reasonable performance
  - Reasonable signature sizes
  - Small keys
  - Application-specific tradeoffs
  - Conservative security
- NIST fast-track standardization of XMSS and LMS
- Caveat: **They are stateful**
  - Need to update the secret key for every signing
  - Updates are as easy as 1 – 2 – 3 . . .
  - Must never go back to earlier state!

Challenge 5: “Huge foot cannons”

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
  - Reasonable performance
  - Reasonable signature sizes
  - Small keys
  - Application-specific tradeoffs
  - Conservative security
- NIST fast-track standardization of XMSS and LMS
- Caveat: **They are stateful**
  - Need to update the secret key for every signing
  - Updates are as easy as 1 – 2 – 3 . . .
  - Must never go back to earlier state!
- Now combine this with, e.g., backups, VMs. . .

Challenge 5: "Huge foot cannons"

- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
  - Reasonable performance
  - Reasonable signature sizes
  - Small keys
  - Application-specific tradeoffs
  - Conservative security
- NIST fast-track standardization of XMSS and LMS
- Caveat: **They are stateful**
  - Need to update the secret key for every signing
  - Updates are as easy as 1 – 2 – 3 . . .
  - Must never go back to earlier state!
- Now combine this with, e.g., backups, VMs. . .

"It's a huge foot cannon" — Adam Langley

Is it already too late?

- Let’s assume that today’s crypto is broken in 15 years
- When do we need to start migrating?

Is it already too late?

- Let’s assume that today’s crypto is broken in 15 years
- When do we need to start migrating?
- Consider the following attack against confidentiality
  - Record encrypted message today
  - Decrypt in 15 years using quantum computer

Is it already too late?

- Let's assume that today's crypto is broken in 15 years
- When do we need to start migrating?
- Consider the following attack against confidentiality
  - Record encrypted message today
  - Decrypt in 15 years using quantum computer

How long do we need today's communication to be secure?

How long does it take us to migrate?

But for signatures we have time, right?

- Signatures provide authentication
- Cannot retroactively “decrypt” anything
- Stop accepting pre-quantum signatures once there is a quantum computer

But for signatures we have time, right?

- Signatures provide authentication
- Cannot retroactively “decrypt” anything
- Stop accepting pre-quantum signatures once there is a quantum computer
- Will we **know** when, say, NSA has a quantum computer?

But for signatures we have time, right?

- Signatures provide authentication
- Cannot retroactively “decrypt” anything
- Stop accepting pre-quantum signatures once there is a quantum computer
- Will we **know** when, say, NSA has a quantum computer?
- May need to prepare devices today!
- Signatures are used for, e.g., software updates
- What if I cannot update anymore in 15 years?
  - What’s the lifetime of a car?
  - What’s the lifetime of smart-home appliances?

Plan for today

Lattice-based KEMs

- Intro lecture
- Programming exercise

Plan for today

Lattice-based KEMs

- Intro lecture
- Programming exercise

Hash-based signatures

- Intro lecture
- Programming exercise