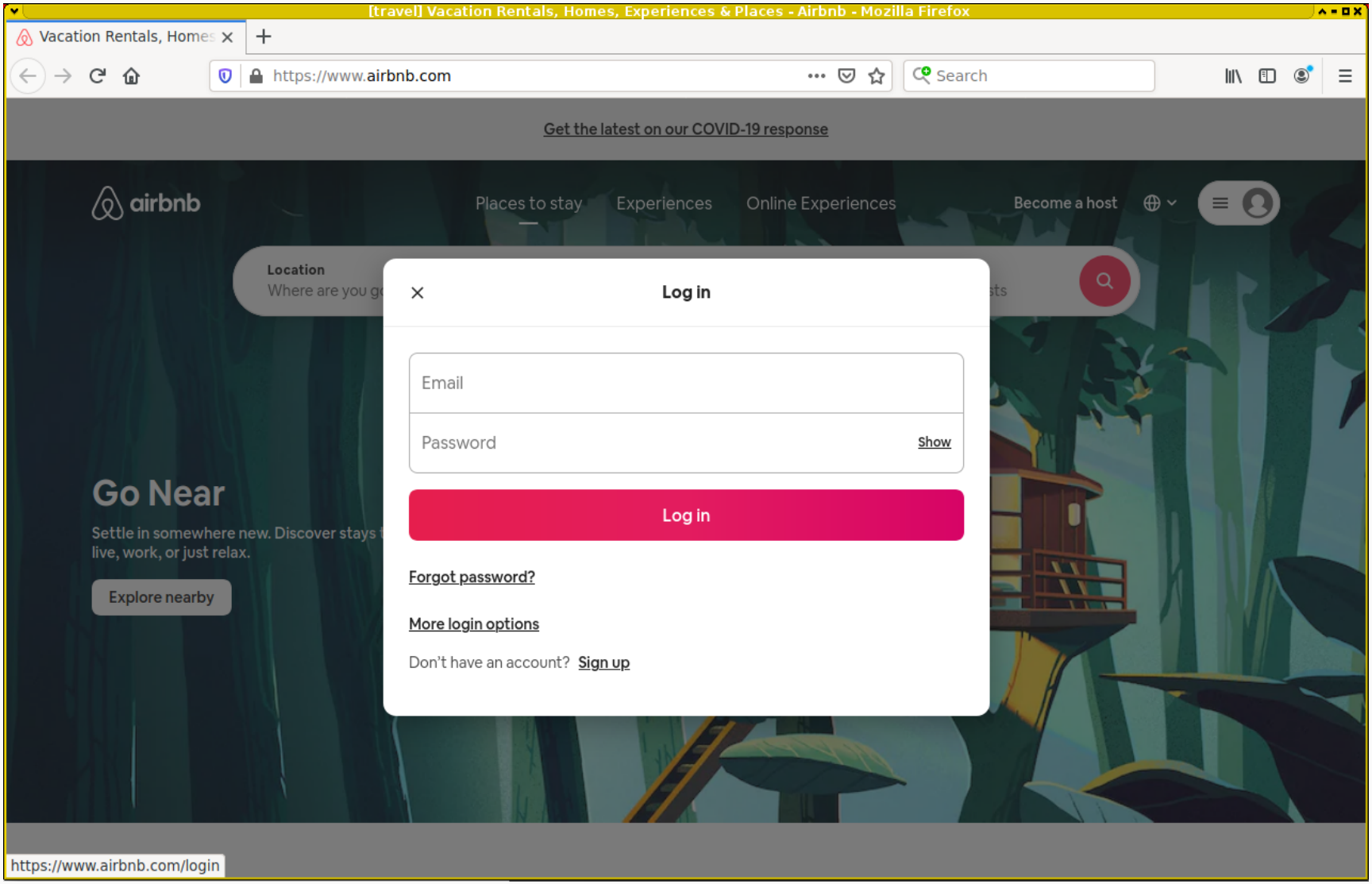
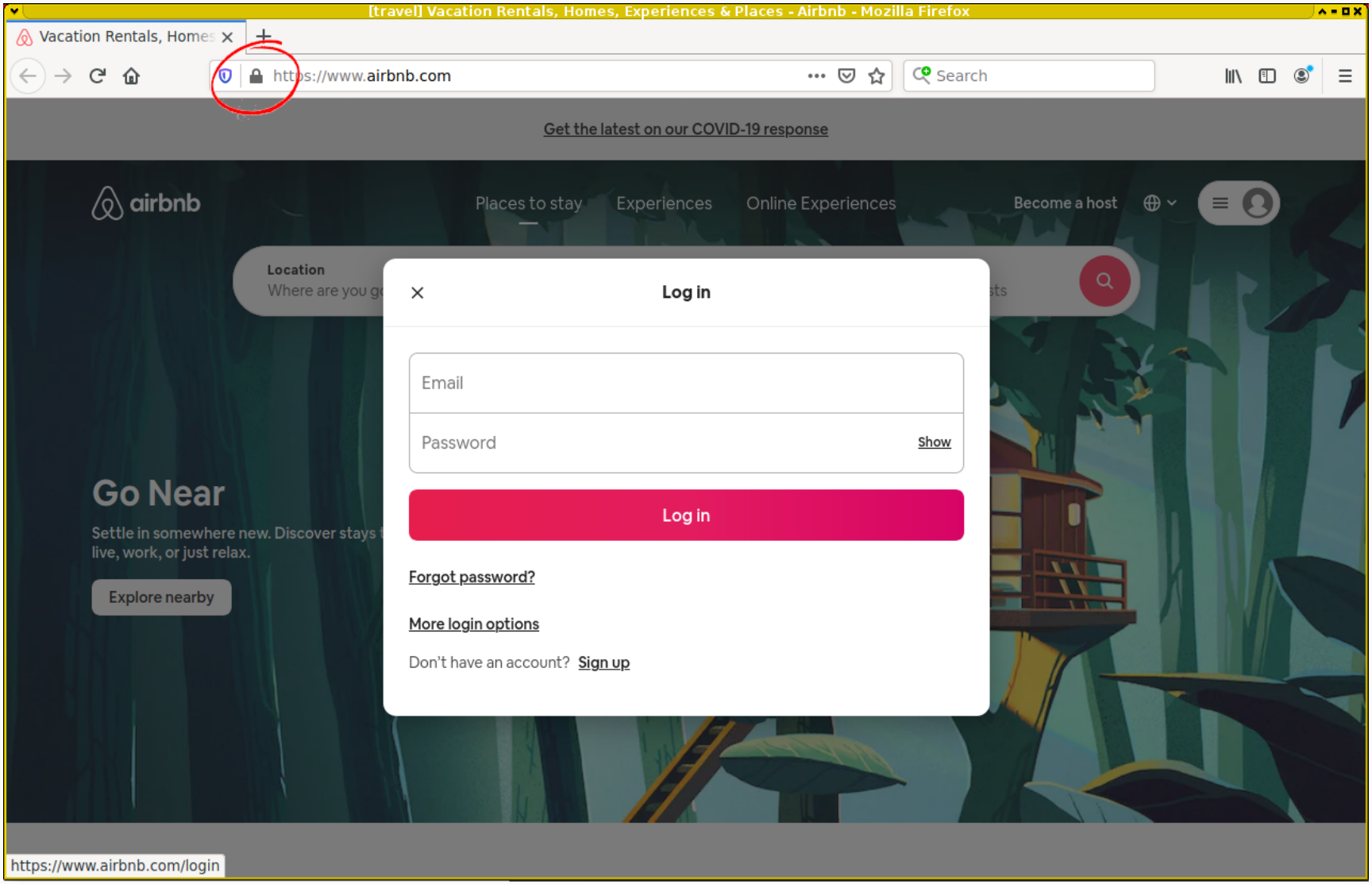
MAX PLANCK INSTITUTE
FOR SECURITY AND PRIVACY

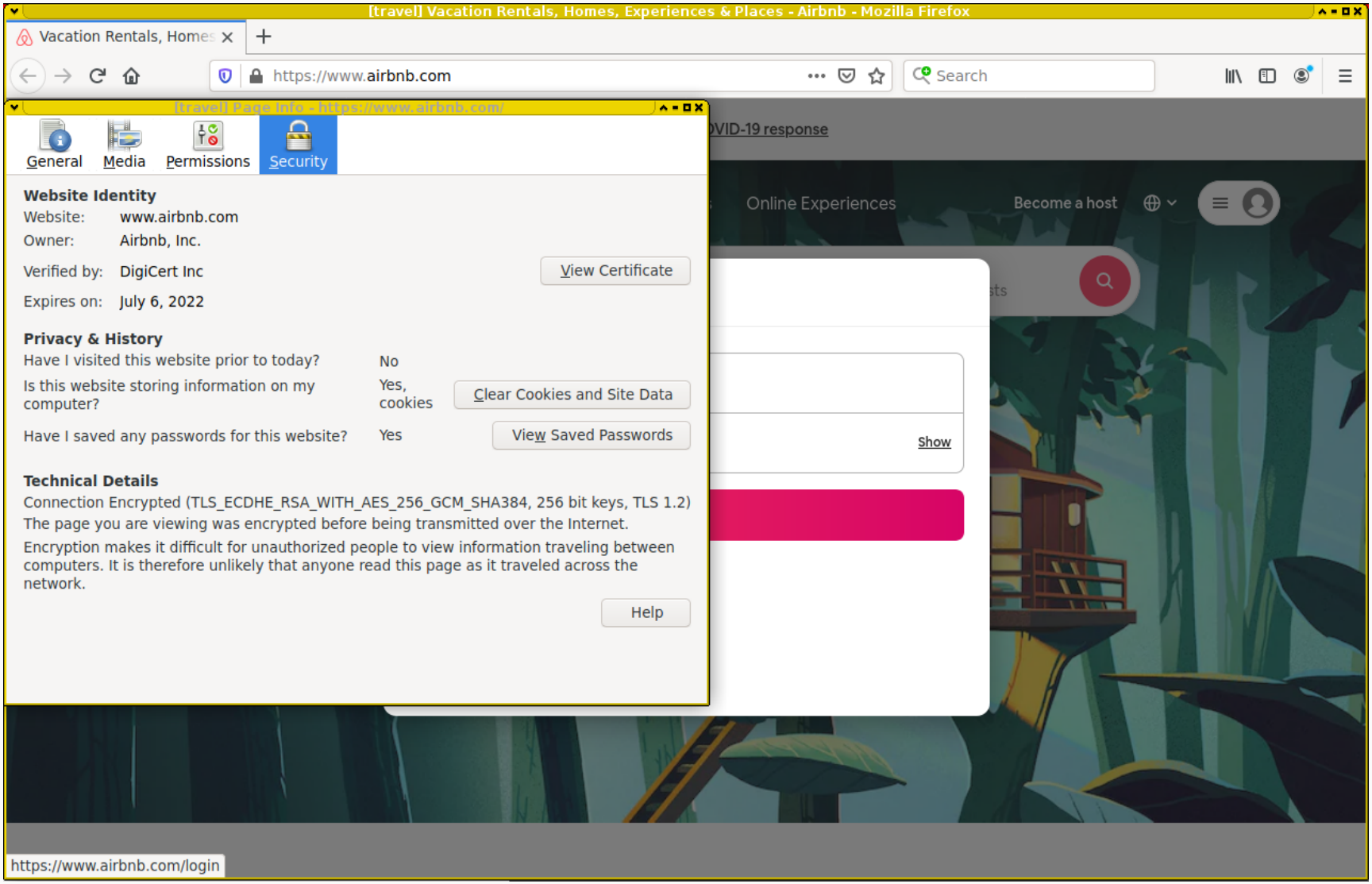
The transition to post-quantum cryptography: challenge and chance

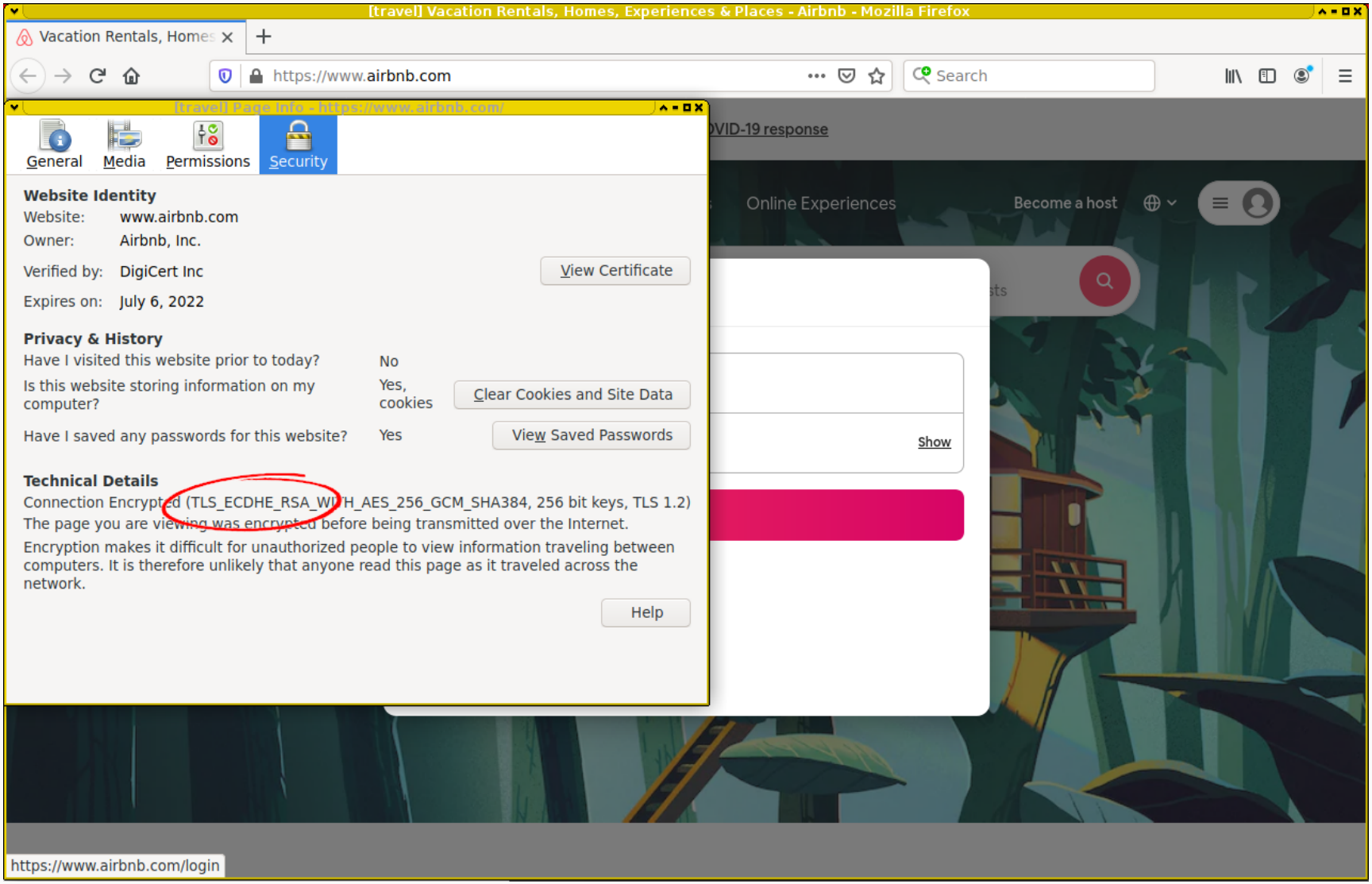
Peter Schwabe

November 14, 2020


[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 go
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
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 go
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
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**

4 31 27

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

What now?

- NIST is expected to announce winners in late 2021
- $\approx$ one year later get standards

What now?

- NIST is expected to announce winners in late 2021
- $\approx$ one year later get standards
- Replace existing crypto with new crypto

What now?

- NIST is expected to announce winners in late 2021
- $\approx$ one year later get standards
- Replace existing crypto with new crypto

Mission accomplished – The world is safe again!

What now?

- NIST is expected to announce winners in late 2021
- $\approx$ 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

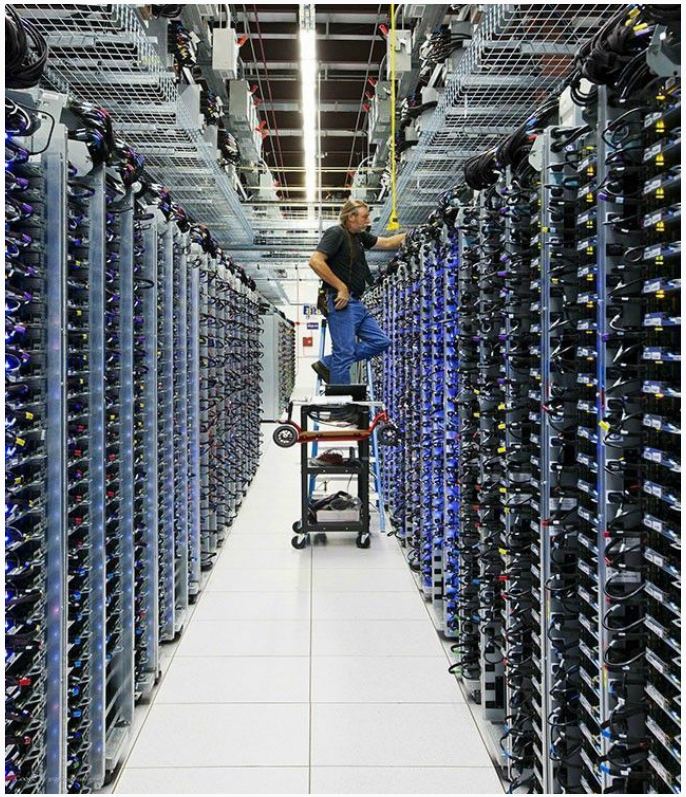
- MD5 is a cryptographic hash function
- Hash functions are used as building blocks all over the place
- **1991**: MD5 is proposed by Rivest
- **1993**: Collisions in MD5 compression function
  (den Boer, Bosselaers)
- **1996**: Dobbertin, Bosselaers, Preneel: concerns about MD5
- **2004**: Wang presents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

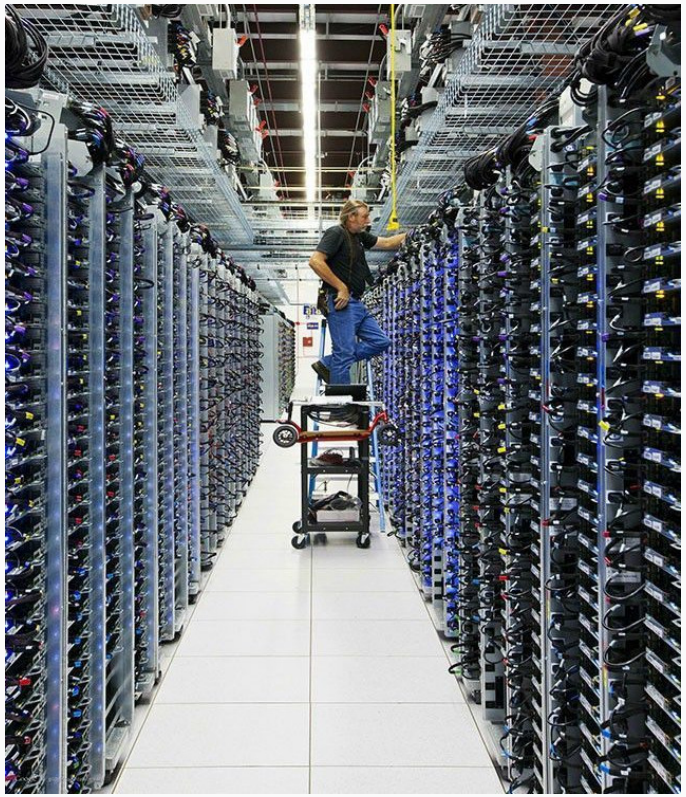
Replacing MD5 was “easy”!

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: ≈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: ≈0.5 MB
- SPHINCS$^+$ signatures: ≈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: ≈0.5 MB
- SPHINCS$^+$ signatures: ≈16 KB
- SPHINCS$^+$ signing: ≈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: ≈0.5 MB
- SPHINCS$^+$ signatures: ≈16 KB
- SPHINCS$^+$ signing: ≈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

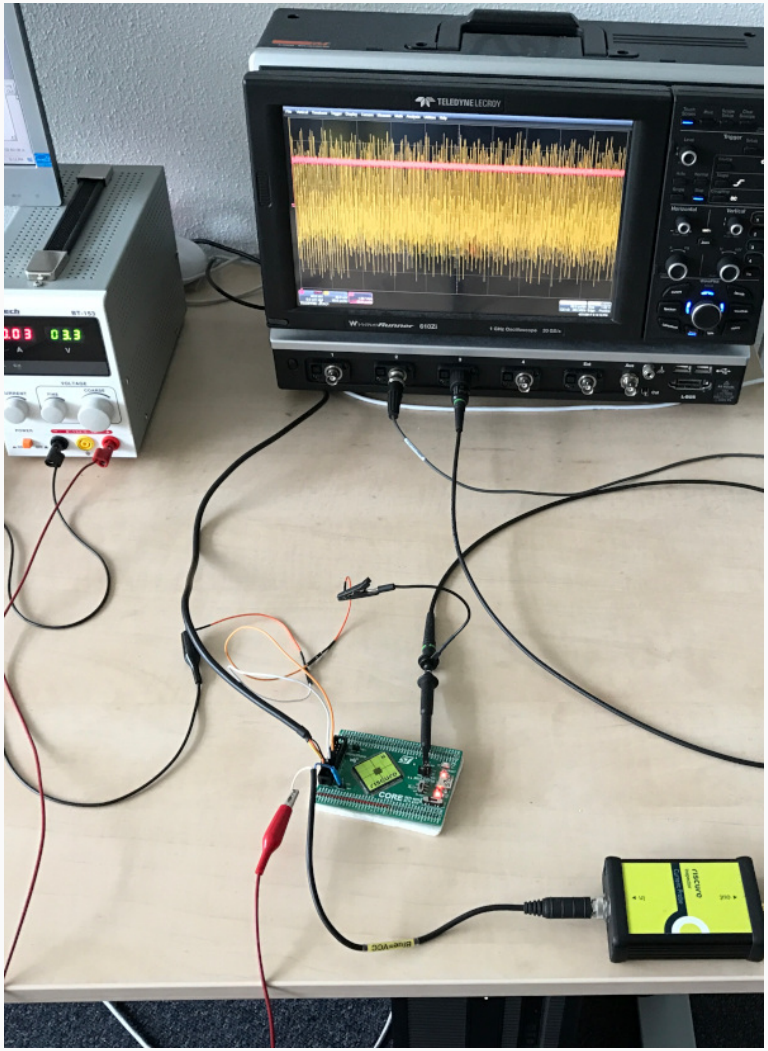
- 2004: Rogaway proposes OCB2
  - Security reduction guaranteeing confidentiality and authenticity
- 2009: OCB2 is standardized by ISO
- 26 Oct. 2018: Break of authenticity by Inoue and Minematsu
- 8/11 Nov. 2018: Break of confidentiality by Poettering / Iwata

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

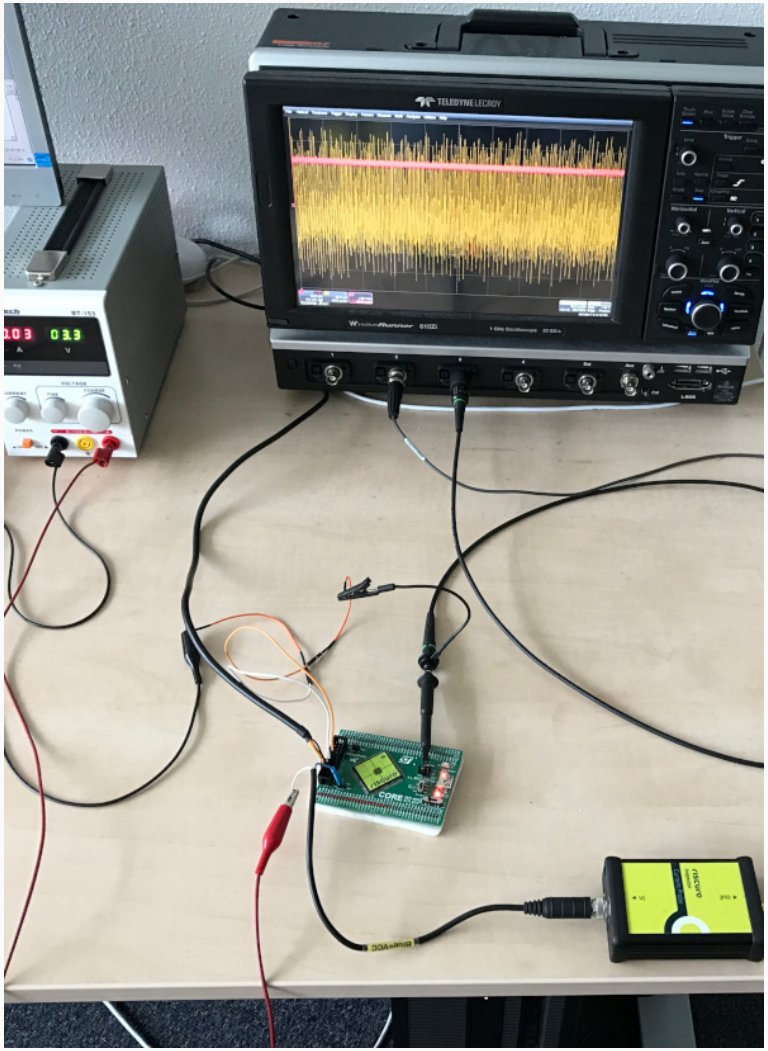
Challenge 3: Implementation Security

Challenge 3: Implementation Security



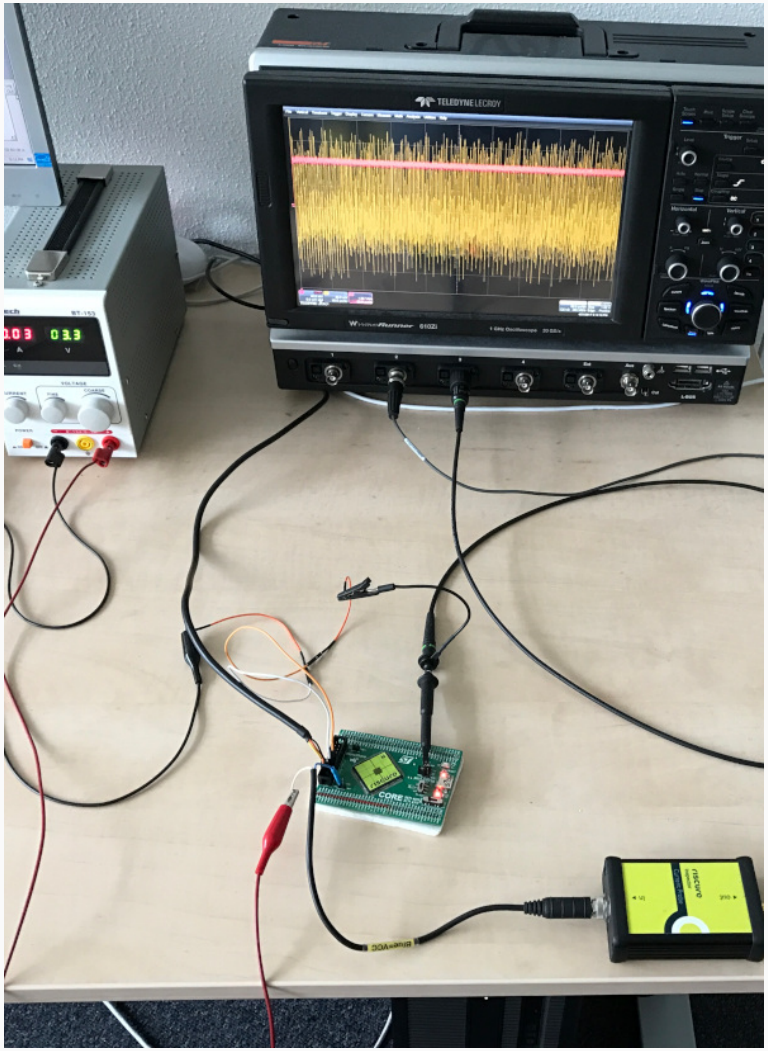
- Attackers see more than input/output:
  - Power consumption
  - Electromagnetic radiation
  - Timing

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.)

"…this isn't very different for any of the other areas of post-quantum crypto"

— Schwabe, 2020.

Challenge 3: Implementation Security (ctd.)

- Baseline: *“constant-time”* implementations
- Execution time does not depend on secret data

Challenge 3: Implementation Security (ctd.)

- Baseline: *"constant-time"* implementations
- Execution time does not depend on secret data
- Unclear if all round-3 schemes have constant-time implementations

Challenge 3: Implementation Security (ctd.)

- Baseline: *"constant-time"* implementations
- Execution time does not depend on secret data
- Unclear if all round-3 schemes have constant-time implementations
- Very few implementations with advanced countermeasures

Challenge 3: Implementation Security (ctd.)

- Baseline: *"constant-time"* implementations
- Execution time does not depend on secret data
- Unclear if 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.)

- Baseline: *"constant-time"* implementations
- Execution time does not depend on secret data
- Unclear if 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: “Huge foot cannons”

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

Challenge 4: “Huge foot cannons”

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

Challenge 4: “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 4: “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 4: “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 4: “Huge foot cannons”

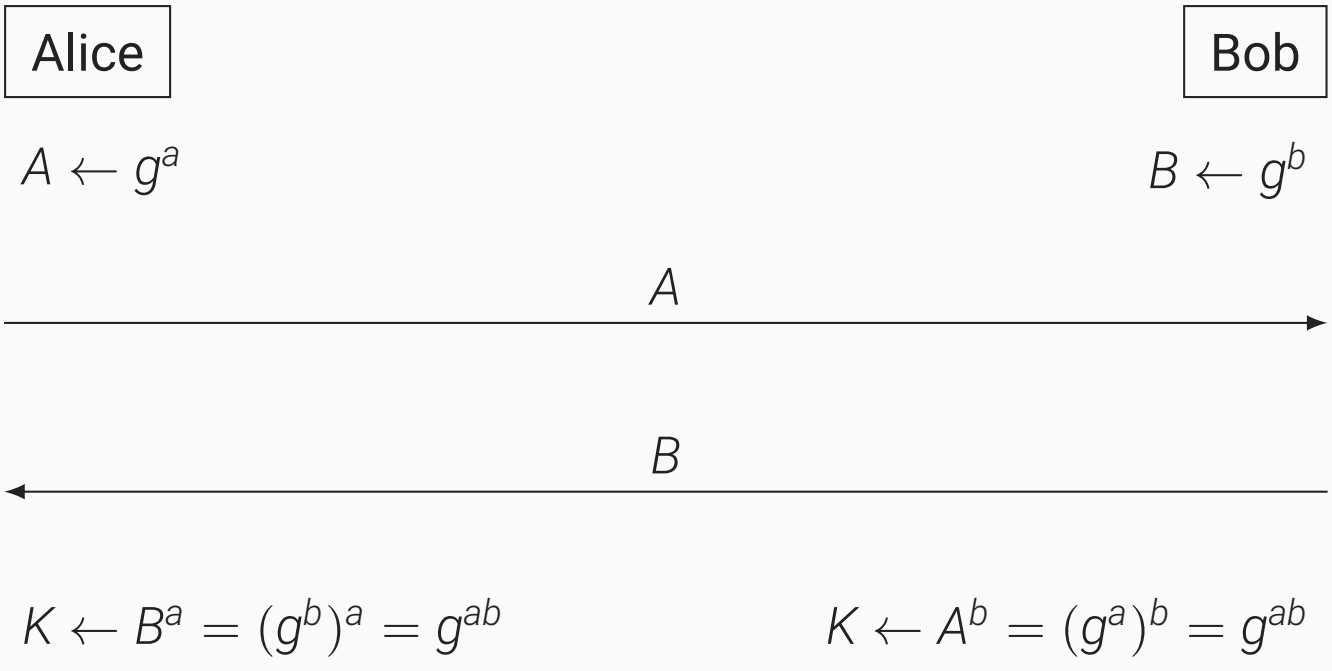
- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
  - Reasonable performance
  - Reasonable signature sizes
  - Small keys
  - Application-specific tradeoffs
  - Conservative security
- NIST fast-track standardization of XMSS and LMS
- Caveat: **They are stateful**
  - Need to 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 4: “Huge foot cannons”

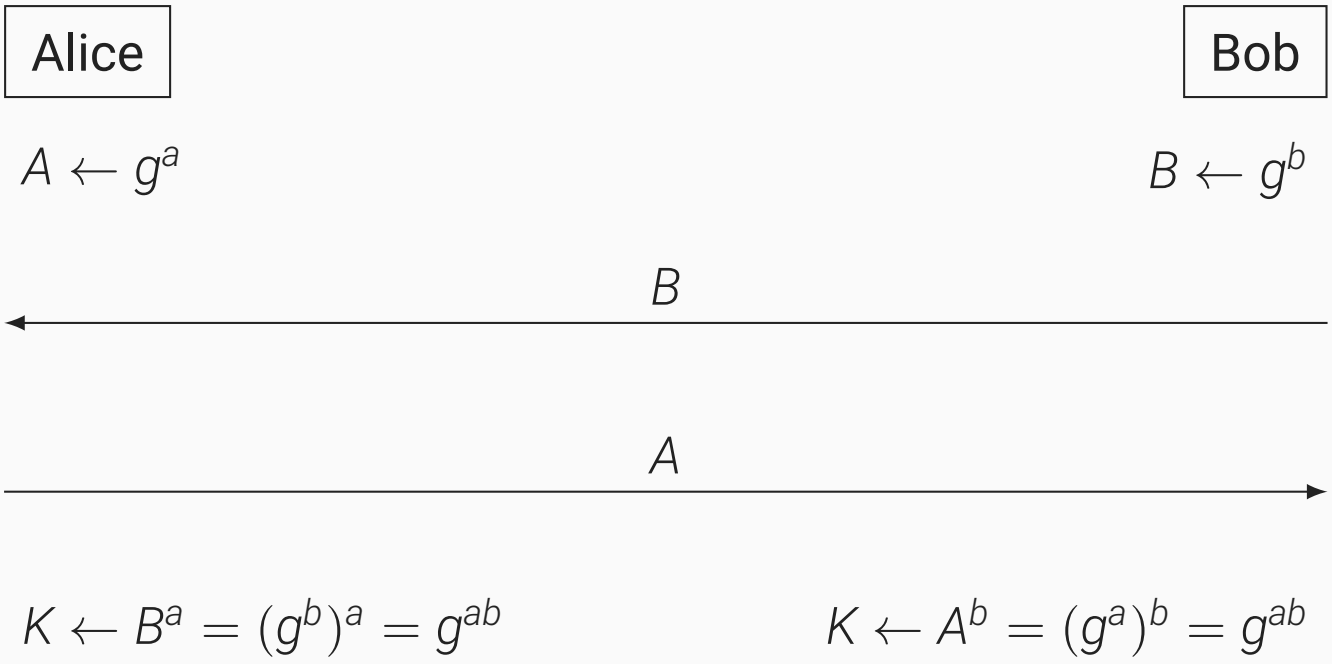
- We already have post-quantum RFCs
- Hash-based signatures XMSS and LMS
  - Reasonable performance
  - Reasonable signature sizes
  - Small keys
  - Application-specific tradeoffs
  - Conservative security
- NIST fast-track standardization of XMSS and LMS
- Caveat: **They are stateful**
  - Need to 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

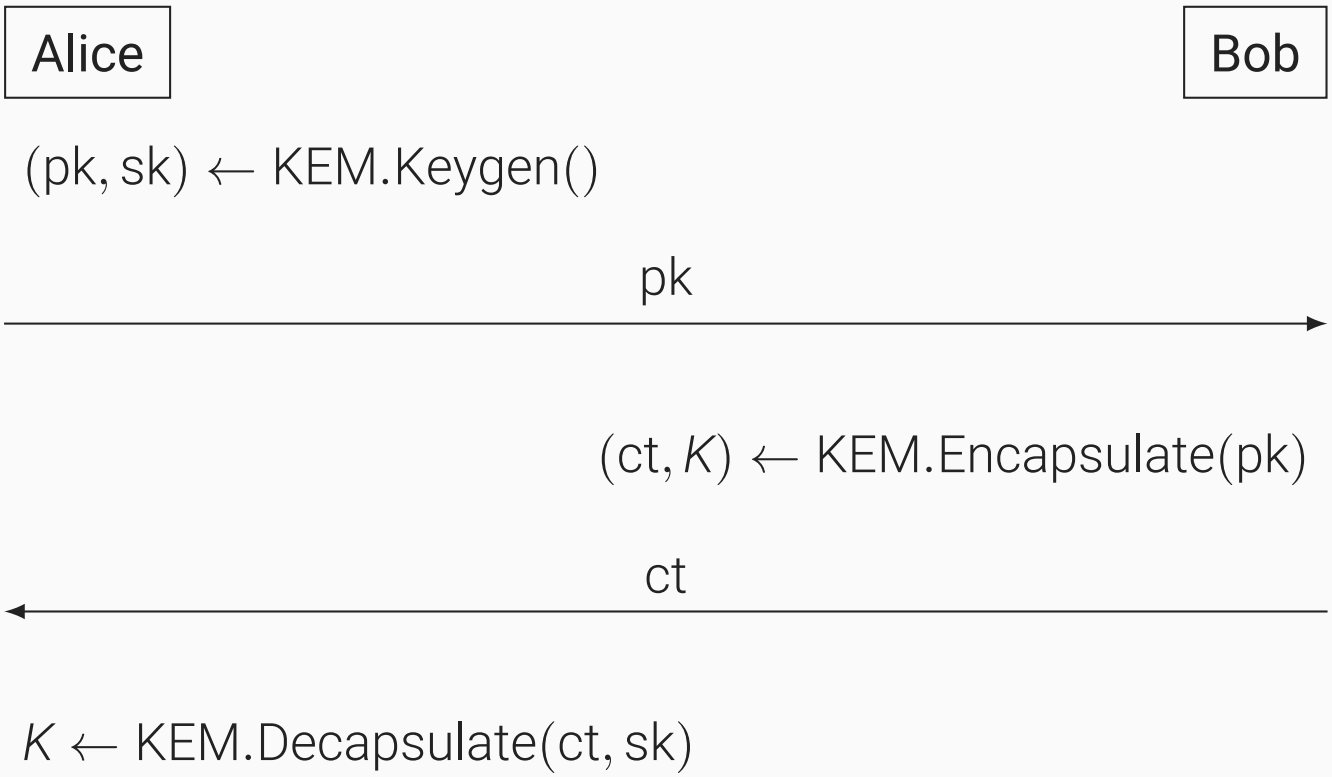
Challenge 5: The curious case of Diffie-Hellman



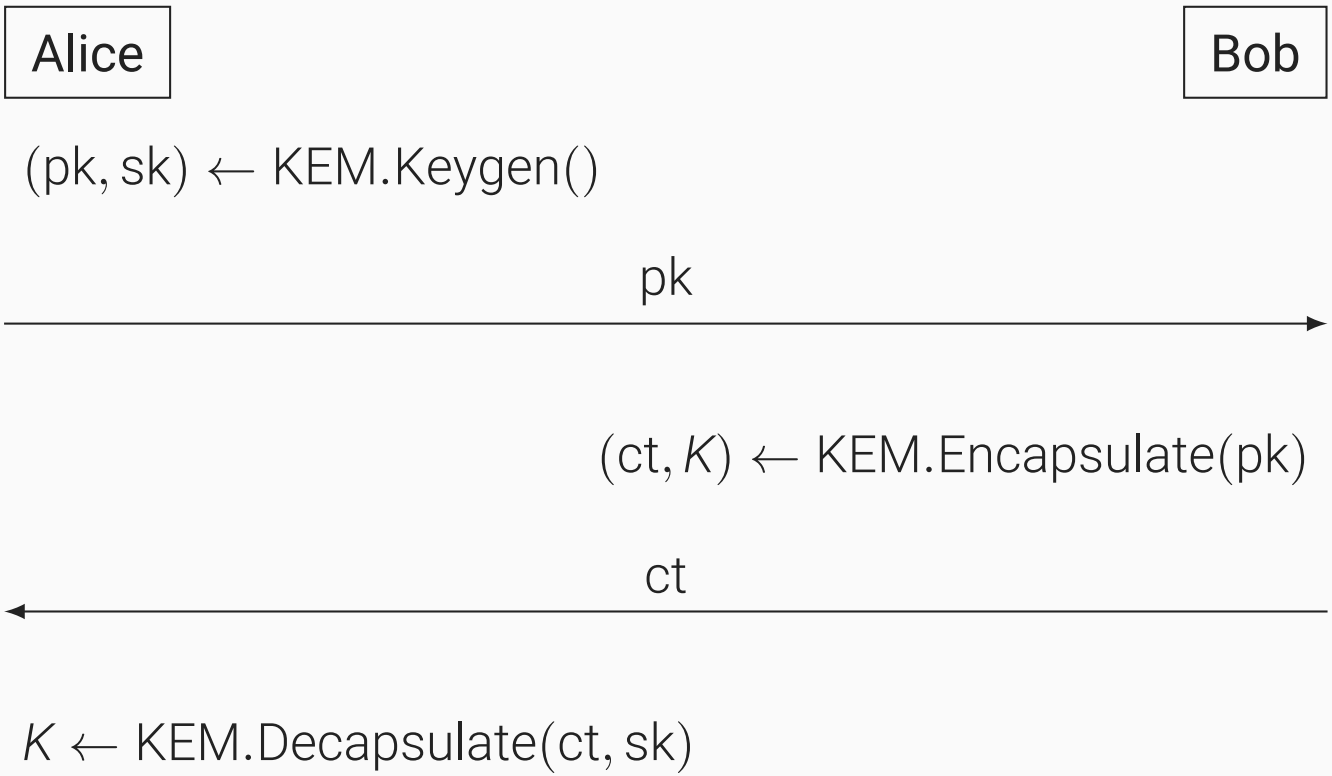
Challenge 5: The curious case of Diffie-Hellman



KEMs: as close as you'll get to DH



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



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

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
- 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?

A chance?

How can this migration be a *chance*?

Crypto (software) today

PlayStation 3 hack – how it happened and what it means

A group of coders claims the PS3 has been hacked, opening the doors to software piracy. We look into the implications

Crypto (software) today

Crypto (software) today

PlayStation 3 hack – how it happene
and what it means

A group of coders claims the PS3 has been hacked, opening the doors to software piracy. We look into the implications

18 OCT 2017 NEWS

ROCA Crypto Bug Compro

ANDY GREENBERG SECURITY 08.18.16 4:20 PM

A New Wireless Hack Can Unlock 100 Million Volkswagens

Crypto (software) today

PlayStation 3 hack – how it happene and what it means

A group of coders claims the PS3 has been hacked, opening the doors to software piracy. We look into the implications

18 OCT 2017 NEWS

ROCA

ANDY GREENBERG SECURITY 08.10.16 4:20 PM

A New Wireless Hack Can Unlock 100 Million Volkswagens

Malware and Vulnerabilities • October 04, 2019 • Cyware Hacker News

New 'Minerva attack' Can Recover Private Keys From Cryptographic Libraries

Crypto (software) today

High-assurance crypto software

- Use formal methods to improve crypto (software):

High-assurance crypto software

- Use formal methods to improve crypto (software):
  - Formal specification of primitives
  - Formal specification of security
  - Formal specification of implementation security

High-assurance crypto software

- Use formal methods to improve crypto (software):
  - Formal specification of primitives
  - Formal specification of security
  - Formal specification of implementation security

} Formal = machine readable

High-assurance crypto software

- Use formal methods to improve crypto (software):
  - Formal specification of primitives
  - Formal specification of security
  - Formal specification of implementation security

  } Formal = machine readable

  - Computer-verified correctness
  - Computer-verified security reduction
  - Computer-verified implementation security

High-assurance crypto software

- Use formal methods to improve crypto (software):
  - Formal specification of primitives
  - Formal specification of security
  - Formal specification of implementation security

  } Formal = machine readable

  - Computer-verified correctness
  - Computer-verified security reduction
  - Computer-verified implementation security

Careful: high-assurance does not mean “unbreakable”

HACS success stories

moz://a

Internet Health Technology

Mozilla Security Blog

FIREFOX SECURITY TLS

Verified cryptography for Firefox 57

Benjamin Beurdouche | September 13, 2017

Traditionally, software is produced in this way: write some code, maybe do some code review, run unit-tests, and then hope it is correct. Hard experience shows that it is very hard for programmers to write bug-free software. These bugs are sometimes caught in manual testing, but many bugs still are exposed to users, and then must be fixed in patches or subsequent versions. This works for most software, but it's not a great way to write cryptographic software; users expect and deserve assurances that the code providing security and privacy is well written and bug free.

HACS success stories

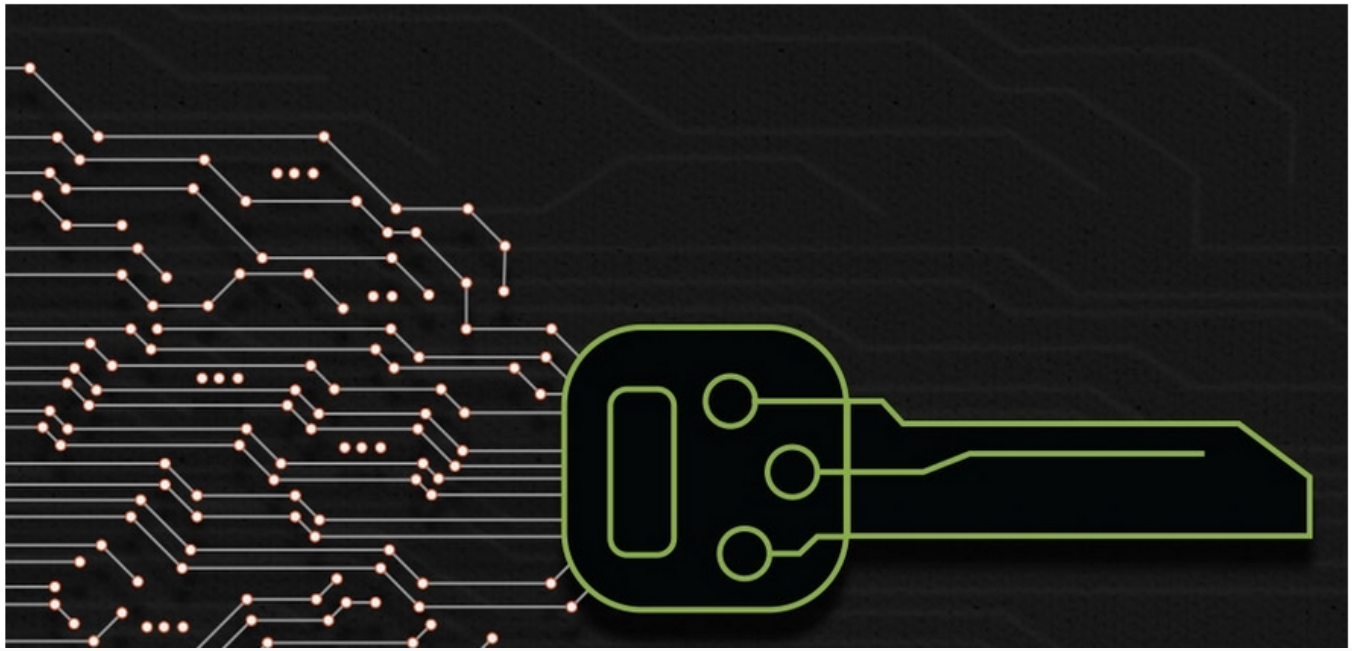
MIT News
ON CAMPUS AND AROUND THE WORLD

SUBSCRIBE

Automated cryptocode generator is helping secure the web

System automatically writes optimized algorithms to encrypt data in Google Chrome browsers and web applications.

Rob Matheson | MIT News Office
June 17, 2019



HACS success stories

CyLab Carnegie Mellon University Security and Privacy Institute

Research Education Partnerships News Events About Directory More Subscribe

HOME / NEWS / 2020 /

Provably-secure code incorporated into Linux kernel

Daniel Tkacik

APR 29, 2020

An overly optimistic outlook

- By end of 2021 years we'll have high-assurance software of all NIST PQC candidates
- All security reductions are computer verified
- All software is proven to be correct
- All software is proven to not leak through timing

An overly optimistic outlook

- By end of 2021 years we'll have high-assurance software of all NIST PQC candidates
- All security reductions are computer verified
- All software is proven to be correct
- All software is proven to not leak through timing
- Implementations with side-channel protection beyond timing
- Countermeasures also formally proven

An overly optimistic outlook

- By end of 2021 years we'll have high-assurance software of all NIST PQC candidates
- All security reductions are computer verified
- All software is proven to be correct
- All software is proven to not leak through timing
- Implementations with side-channel protection beyond timing
- Countermeasures also formally proven
- The whole world migrates to a better generation of crypto

An overly optimistic outlook

- By end of 2021 years we'll have high-assurance software of all NIST PQC candidates
- All security reductions are computer verified
- All software is proven to be correct
- All software is proven to not leak through timing
- Implementations with side-channel protection beyond timing
- Countermeasures also formally proven
- The whole world migrates to a better generation of crypto

Yes, this is overly optimistic.

An overly optimistic outlook

- By end of 2021 years we'll have high-assurance software of all NIST PQC candidates
- All security reductions are computer verified
- All software is proven to be correct
- All software is proven to not leak through timing
- Implementations with side-channel protection beyond timing
- Countermeasures also formally proven
- The whole world migrates to a better generation of crypto

Yes, this is overly optimistic.

. . . let's see it as an ambitious goal!

Some pointers

PQC resources

- NIST PQC website:
  `https://csrc.nist.gov/Projects/Post-Quantum-Cryptography`
- NIST mailing list:
  `https://csrc.nist.gov/projects/post-quantum-cryptography/email-list`
  `https://groups.google.com/a/list.nist.gov/g/pqc-forum`
- Open Quantum Safe `https://openquantumsafe.org/`
- PQC Wiki: `https://pqc-wiki.fau.edu`

HACS resources

- HACS workshop: `https://www.hacs-workshop.org/`