

The migration to post-quantum cryptography

Peter Schwabe

Max Planck Institute for Security and Privacy

July 11, 2025



- ▶ Located in **Bochum**
- ▶ Founded in 2019
- ▶ Currently 13 PIs
- ▶ Aim to have
 - ▶ 6 Departments
 - ▶ 12 Research Groups
 - ▶ Around 250 people total





[A small demo]



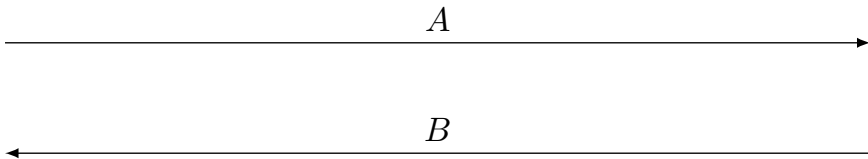
Let G be a finite cyclic group with generator g .

Alice

$$A \leftarrow g^a$$

Bob

$$B \leftarrow g^b$$

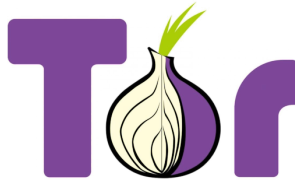


$$K \leftarrow B^a = (g^b)^a = g^{ab}$$

$$K \leftarrow A^b = (g^a)^b = g^{ab}$$



- ▶ Diffie, Hellman, 1976: Use $G = GF(q)^*$
- ▶ Miller, Koblitz (independently), 1985/86: Use group of points on an elliptic curve
- ▶ Bernstein, 2006: Use specific elliptic curve over $GF(2^{255} - 19)$





Definition

Given $P, Q \in G$ such that $Q \in \langle P \rangle$, find an integer k such that $P^k = Q$.



Definition

Given $P, Q \in G$ such that $Q \in \langle P \rangle$, find an integer k such that $kP = Q$.



Definition

Given $P, Q \in G$ such that $Q \in \langle P \rangle$, find an integer k such that $kP = Q$.

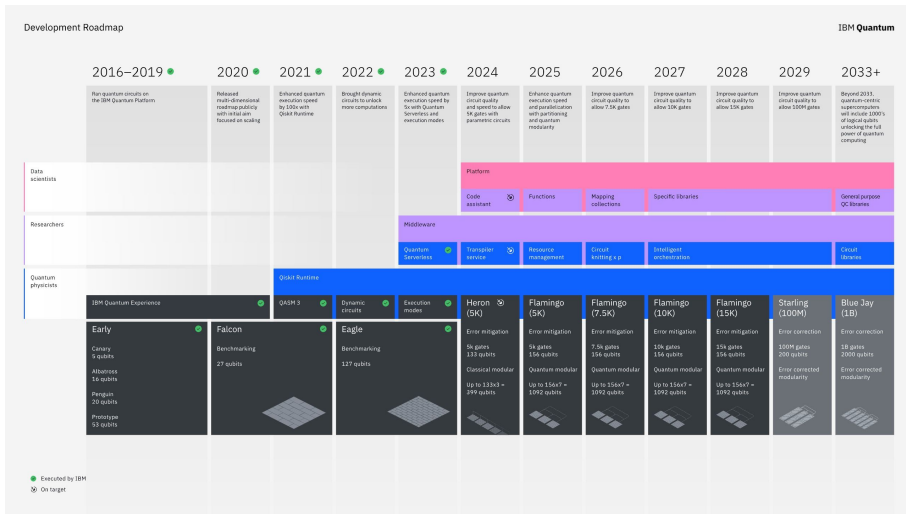
- ▶ DH needs group where DLP is hard
- ▶ (EC)DLP-based crypto also for signatures (DSA, ECDSA, EdDSA...)
- ▶ Prominent alternative: RSA (based on factoring)

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.





[Back to our demo]

POST-QUANTUM KEY EXCHANGE



A NEW HOPE

ERDEM ALKIM

LÉO DUCAS

THOMAS PÖPPELMANN

PETER SCHWABE

Key Encapsulation Mechanisms (KEMs)



Initiator

Responder

$(pk, sk) \leftarrow \text{KEM.Gen}$

pk

$(ct, K) \leftarrow \text{KEM.Enc}(pk)$

ct

$K \leftarrow \text{KEM.Dec}(ct, sk)$



- ▶ Given uniform $\mathbf{A} \in \mathbb{Z}_q^{k \times \ell}$
- ▶ Given “noise distribution” χ
- ▶ Given samples $\mathbf{A}\mathbf{s} + \mathbf{e}$, with $\mathbf{e} \leftarrow \chi$

- ▶ Given uniform $\mathbf{A} \in \mathbb{Z}_q^{k \times \ell}$
- ▶ Given “noise distribution” χ
- ▶ Given samples $\mathbf{A}\mathbf{s} + \mathbf{e}$, with $\mathbf{e} \leftarrow \chi$
- ▶ Search version: find $\mathbf{s}$
- ▶ Decision version: distinguish from uniform random

- ▶ Given uniform $\mathbf{a} \in \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- ▶ Given “noise distribution” χ
- ▶ Given samples $\mathbf{as} + \mathbf{e}$, with $\mathbf{e} \leftarrow \chi$

- ▶ Given uniform $\mathbf{a} \in \mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + 1)$
- ▶ Given “noise distribution” χ
- ▶ Given samples $\mathbf{a}\mathbf{s} + \mathbf{e}$, with $\mathbf{e} \leftarrow \chi$
- ▶ Search version: find $\mathbf{s}$
- ▶ Decision version: distinguish from uniform random

How to build a KEM?

The basic idea



Alice (server)		Bob (client)
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{\mathbf{b}}$	$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
	$\xleftarrow{\mathbf{u}}$	

Alice has $\mathbf{v} = \mathbf{u}\mathbf{s} = \mathbf{a}\mathbf{s}\mathbf{s}' + \mathbf{e}'\mathbf{s}$

Bob has $\mathbf{v}' = \mathbf{b}\mathbf{s}' = \mathbf{a}\mathbf{s}\mathbf{s}' + \mathbf{e}\mathbf{s}'$

- ▶ Secret and noise polynomials $\mathbf{s}, \mathbf{s}', \mathbf{e}, \mathbf{e}'$ are small
- ▶ $\mathbf{v}$ and $\mathbf{v}'$ are *approximately* the same

How to build a KEM, part 2



Alice		Bob
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$ $\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b})}$	$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$ $\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$ $\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}'$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$	$\xleftarrow{(\mathbf{u})}$	

How to build a KEM, part 2



Alice	Bob
$seed \xleftarrow{\$} \{0, 1\}^{256}$	
$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$	
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$	$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)} \mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$
	$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
	$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}'$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$	$\xleftarrow{(\mathbf{u})}$

How to build a KEM, part 2



14

Alice		Bob
$seed \xleftarrow{\$} \{0, 1\}^{256}$		
$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{as} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$
		$\mathbf{u} \leftarrow \mathbf{as}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{bs}'$
		$\mathbf{k} \xleftarrow{\$} \{0, 1\}^n$
		$\mathbf{k} \leftarrow \text{Encode}(\mathbf{k})$
$\mathbf{v}' \leftarrow \mathbf{us}$	$\xleftarrow{(\mathbf{u}, \mathbf{c})}$	$\mathbf{c} \leftarrow \mathbf{v} + \mathbf{k}$

How to build a KEM, part 2



14

Alice		Bob
$seed \xleftarrow{\$} \{0, 1\}^{256}$		
$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
		$k \xleftarrow{\$} \{0, 1\}^n$
		$\mathbf{k} \leftarrow \text{Encode}(k)$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$	$\xleftarrow{(\mathbf{u}, \mathbf{c})}$	$\mathbf{c} \leftarrow \mathbf{v} + \mathbf{k}$

How to build a KEM, part 2



14

Alice		Bob
$seed \xleftarrow{\$} \{0, 1\}^{256}$		
$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
		$k \xleftarrow{\$} \{0, 1\}^n$
		$\mathbf{k} \leftarrow \text{Encode}(k)$
	$\xleftarrow{(\mathbf{u}, \mathbf{c})}$	$\mathbf{c} \leftarrow \mathbf{v} + \mathbf{k}$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$		
$\mathbf{k}' \leftarrow \mathbf{c} - \mathbf{v}'$		

How to build a KEM, part 2



Alice		Bob
$seed \xleftarrow{\$} \{0, 1\}^{256}$		
$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
		$k \xleftarrow{\$} \{0, 1\}^n$
		$\mathbf{k} \leftarrow \text{Encode}(k)$
	$\xleftarrow{(\mathbf{u}, \mathbf{c})}$	$\mathbf{c} \leftarrow \mathbf{v} + \mathbf{k}$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$		$\mu \leftarrow \text{Extract}(\mathbf{k})$
$\mathbf{k}' \leftarrow \mathbf{c} - \mathbf{v}'$		
$\mu \leftarrow \text{Extract}(\mathbf{k}')$		

How to build a KEM, part 2



Alice		Bob
$seed \xleftarrow{\$} \{0, 1\}^{256}$		
$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$		
$\mathbf{s}, \mathbf{e} \xleftarrow{\$} \chi$		$\mathbf{s}', \mathbf{e}', \mathbf{e}'' \xleftarrow{\$} \chi$
$\mathbf{b} \leftarrow \mathbf{a}\mathbf{s} + \mathbf{e}$	$\xrightarrow{(\mathbf{b}, seed)}$	$\mathbf{a} \leftarrow \text{Parse}(\text{XOF}(seed))$
		$\mathbf{u} \leftarrow \mathbf{a}\mathbf{s}' + \mathbf{e}'$
		$\mathbf{v} \leftarrow \mathbf{b}\mathbf{s}' + \mathbf{e}''$
		$k \xleftarrow{\$} \{0, 1\}^n$
		$\mathbf{k} \leftarrow \text{Encode}(k)$
	$\xleftarrow{(\mathbf{u}, \mathbf{c})}$	$\mathbf{c} \leftarrow \mathbf{v} + \mathbf{k}$
$\mathbf{v}' \leftarrow \mathbf{u}\mathbf{s}$		$\mu \leftarrow \text{Extract}(\mathbf{k})$
$\mathbf{k}' \leftarrow \mathbf{c} - \mathbf{v}'$		
$\mu \leftarrow \text{Extract}(\mathbf{k}')$		

Encryption scheme by Lyubashevsky, Peikert, Regev. Eurocrypt 2010.

- ▶ Encoding in LPR encryption: map n bits to n coefficients:
 - ▶ A zero bit maps to 0
 - ▶ A one bit maps to $q/2$
- ▶ Idea: Noise affects low bits of coefficients, put data into high bits

- ▶ Encoding in LPR encryption: map n bits to n coefficients:
 - ▶ A zero bit maps to 0
 - ▶ A one bit maps to $q/2$
- ▶ Idea: Noise affects low bits of coefficients, put data into high bits
- ▶ Decode: map coefficient into $[-q/2, q/2]$
 - ▶ Closer to 0 (i.e., in $[-q/4, q/4]$): set bit to zero
 - ▶ Closer to $\pm q/2$: set bit to one



- ▶ Improve IEEE S&P 2015 results by Bos, Costello, Naehrig, Stebila (BCNS)
- ▶ Use reconciliation to go from approximate agreement to agreement
 - ▶ Originally proposed by Ding (2012)
 - ▶ Improvements by Peikert (2014)
 - ▶ More improvements in NEWHOPE

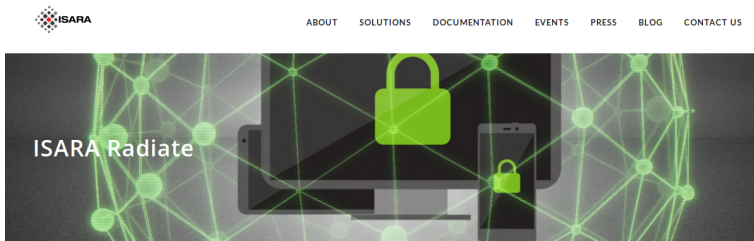
- ▶ Improve IEEE S&P 2015 results by Bos, Costello, Naehrig, Stebila (BCNS)
- ▶ Use reconciliation to go from approximate agreement to agreement
 - ▶ Originally proposed by Ding (2012)
 - ▶ Improvements by Peikert (2014)
 - ▶ More improvements in NEWHOPE
- ▶ NEWHOPE-Simple (2016): Simpler reconciliation (pay 6.25% increase in ciphertext size)

- ▶ Improve IEEE S&P 2015 results by Bos, Costello, Naehrig, Stebila (BCNS)
- ▶ Use reconciliation to go from approximate agreement to agreement
 - ▶ Originally proposed by Ding (2012)
 - ▶ Improvements by Peikert (2014)
 - ▶ More improvements in NEWHOPE
- ▶ NEWHOPE-Simple (2016): Simpler reconciliation (pay 6.25% increase in ciphertext size)
- ▶ Very conservative parameters ($n = 1024, q = 12289$)
- ▶ Parameters chosen to enable fast implementations (NTT)

- ▶ Improve IEEE S&P 2015 results by Bos, Costello, Naehrig, Stebila (BCNS)
- ▶ Use reconciliation to go from approximate agreement to agreement
 - ▶ Originally proposed by Ding (2012)
 - ▶ Improvements by Peikert (2014)
 - ▶ More improvements in NEWHOPE
- ▶ NEWHOPE-Simple (2016): Simpler reconciliation (pay 6.25% increase in ciphertext size)
- ▶ Very conservative parameters ($n = 1024, q = 12289$)
- ▶ Parameters chosen to enable fast implementations (NTT)
- ▶ Centered binomial noise ψ_k ($\text{HW}(a) - \text{HW}(b)$ for k -bit a, b)
- ▶ Achieve ≈ 256 bits of post-quantum security according to very conservative analysis
- ▶ Higher security, shorter messages, and $> 10\times$ speedup

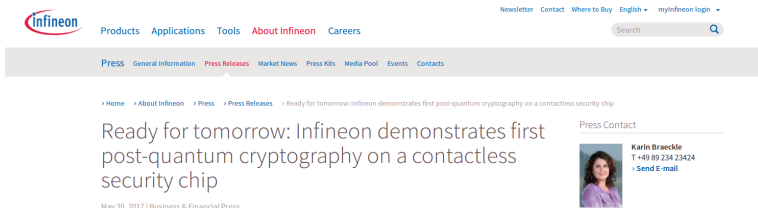
- ▶ Improve IEEE S&P 2015 results by Bos, Costello, Naehrig, Stebila (BCNS)
- ▶ Use reconciliation to go from approximate agreement to agreement
 - ▶ Originally proposed by Ding (2012)
 - ▶ Improvements by Peikert (2014)
 - ▶ More improvements in NEWHOPE
- ▶ NEWHOPE-Simple (2016): Simpler reconciliation (pay 6.25% increase in ciphertext size)
- ▶ Very conservative parameters ($n = 1024, q = 12289$)
- ▶ Parameters chosen to enable fast implementations (NTT)
- ▶ Centered binomial noise ψ_k ($\text{HW}(a) - \text{HW}(b)$ for k -bit a, b)
- ▶ Achieve ≈ 256 bits of post-quantum security according to very conservative analysis
- ▶ Higher security, shorter messages, and $> 10\times$ speedup
- ▶ Choose a fresh parameter $\mathbf{a}$ for every protocol run

- ▶ Improve IEEE S&P 2015 results by Bos, Costello, Naehrig, Stebila (BCNS)
- ▶ Use reconciliation to go from approximate agreement to agreement
 - ▶ Originally proposed by Ding (2012)
 - ▶ Improvements by Peikert (2014)
 - ▶ More improvements in NEWHOPE
- ▶ NEWHOPE-Simple (2016): Simpler reconciliation (pay 6.25% increase in ciphertext size)
- ▶ Very conservative parameters ($n = 1024, q = 12289$)
- ▶ Parameters chosen to enable fast implementations (NTT)
- ▶ Centered binomial noise ψ_k ($\text{HW}(a) - \text{HW}(b)$ for k -bit a, b)
- ▶ Achieve ≈ 256 bits of post-quantum security according to very conservative analysis
- ▶ Higher security, shorter messages, and $> 10\times$ speedup
- ▶ Choose a fresh parameter $\mathbf{a}$ for every protocol run
- ▶ Multiple implementations



ISARA Radiate is the first commercially available security solution offering quantum resistant algorithms that replace or augment classical algorithms, which will be weakened or broken by quantum computing threats.

"Key Agreement using the 'NewHope' lattice-based algorithm detailed in the New Hope paper, and LUKE (Lattice-based Unique Key Exchange), an ISARA speed-optimized version of the NewHope algorithm."



The screenshot shows the Infineon website's press release page. At the top, there is a navigation bar with the Infineon logo, links for Products, Applications, Tools, About Infineon, and Careers. A search bar is also present. Below this is a secondary navigation bar with links for Press, General Information, Press Releases (highlighted), Market News, Press Kits, Media Pool, Events, and Contacts. The main content area features the headline 'Ready for tomorrow: Infineon demonstrates first post-quantum cryptography on a contactless security chip' and a sub-headline 'May 30, 2017 | Business & Financial Press'. To the right, there is a 'Press Contact' section with a photo of Karin Braeckle, her name, title, phone number, and a 'Send E-mail' link.

Infineon

Products Applications Tools About Infineon Careers

Newsletter Contact Where to Buy English myinfineon login

Search

Press General Information Press Releases Market News Press Kits Media Pool Events Contacts

> Home > About Infineon > Press > Press Releases > Ready for tomorrow: Infineon demonstrates first post-quantum cryptography on a contactless security chip

Ready for tomorrow: Infineon demonstrates first post-quantum cryptography on a contactless security chip

May 30, 2017 | Business & Financial Press

Press Contact

 Karin Braeckle
T +49 89 234 23424
[Send E-mail](#)

"The deployed algorithm is a variant of "New Hope", a quantum-resistant cryptosystem"

<https://www.infineon.com/cms/en/about-infineon/press/press-releases/2017/INFCCS201705-056.html>

Google Security Blog

The latest news and insights from Google on security and safety on the Internet

Experimenting with Post-Quantum Cryptography

July 7, 2016

Posted by Matt Braithwaite, Software Engineer

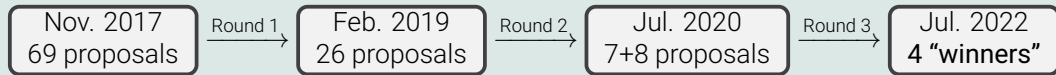
Archive

"We're indebted to Erdem Alkim, Léo Ducas, Thomas Pöppelmann and Peter Schwabe, the researchers who developed "New Hope", the post-quantum algorithm that we selected for this experiment."

- ▶ National Institute of Standards and Technology
- ▶ Public call for PQC proposals, aims at finding schemes for standardization
- ▶ Similar to earlier AES and SHA-3 efforts
- ▶ Process draft online in August 2016, comments by September 2016
- ▶ Call for proposals in December 2016, deadline November 2017

- ▶ National Institute of Standards and Technology
- ▶ Public call for PQC proposals, aims at finding schemes for standardization
- ▶ Similar to earlier AES and SHA-3 efforts
- ▶ Process draft online in August 2016, comments by September 2016
- ▶ Call for proposals in December 2016, deadline November 2017

How it went



- ▶ Second KEM selected for standardization in March 2025

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.



Roberto Avanzi
Léo Ducas
Vadim Lyubashevsky
Gregor Seiler

Joppe Bos
Eike Kiltz
John M. Schanck
Damien Stehlé

Jintai Ding
Tancrede Lepoint
Peter Schwabe

MLWE instead of RLWE

IND-CCA2 Security

MLWE instead of RLWE

- ▶ Easily scale security
- ▶ Optimized routines the same for all security levels

IND-CCA2 Security

MLWE instead of RLWE

- ▶ Easily scale security
- ▶ Optimized routines the same for all security levels

IND-CCA2 Security

- ▶ Support static (or cached) keys
- ▶ More robust
- ▶ Useful for authenticated key exchange
- ▶ Easy to construct PKE

- ▶ RLWE uses arithmetic on large degree polynomials
- ▶ For example, NEWHOPE uses $n = 1024$, $q = 12289$

- ▶ RLWE uses arithmetic on large degree polynomials
- ▶ For example, NEWHOPE uses $n = 1024$, $q = 12289$
- ▶ MLWE uses matrices and vectors of smaller polynomials of small dimension

- ▶ RLWE uses arithmetic on large degree polynomials
- ▶ For example, NEWHOPE uses $n = 1024, q = 12289$
- ▶ MLWE uses matrices and vectors of smaller polynomials of small dimension
- ▶ Kyber: $n = 256, q = 3329$
 - ▶ Security level 1 (AES-128): $d = 2$
 - ▶ Security level 3 (AES-192): $d = 3$
 - ▶ Security level 5 (AES-256): $d = 4$
- ▶ Core arithmetic is in $\mathbb{Z}_{3329}[X]/(X^{256} + 1)$ **for all security levels**

- ▶ RLWE uses arithmetic on large degree polynomials
- ▶ For example, NEWHOPE uses $n = 1024, q = 12289$
- ▶ MLWE uses matrices and vectors of smaller polynomials of small dimension
- ▶ Kyber: $n = 256, q = 3329$
 - ▶ Security level 1 (AES-128): $d = 2$
 - ▶ Security level 3 (AES-192): $d = 3$
 - ▶ Security level 5 (AES-256): $d = 4$
- ▶ Core arithmetic is in $\mathbb{Z}_{3329}[X]/(X^{256} + 1)$ **for all security levels**
- ▶ Noise is centered binomial $\text{HW}(x) - \text{HW}(y)$ for 2-bit x and y

- ▶ Decryption failures are a function of s, e, s', e'
- ▶ Attacker can choose larger secret/noise e' and s'
- ▶ Observe if decryption fails
- ▶ Learn something about s

- ▶ Decryption failures are a function of s, e, s', e'
- ▶ Attacker can choose larger secret/noise e' and s'
- ▶ Observe if decryption fails
- ▶ Learn something about s
- ▶ This is a chosen ciphertext attack (CCA)
- ▶ Learn full s after a few thousand queries

- ▶ Decryption failures are a function of s, e, s', e'
- ▶ Attacker can choose larger secret/noise e' and s'
- ▶ Observe if decryption fails
- ▶ Learn something about s
- ▶ This is a chosen ciphertext attack (CCA)
- ▶ Learn full s after a few thousand queries
- ▶ NEWHOPE never claimed CCA-security!
- ▶ This “attack” is completely expected
- ▶ Not a problem for ephemeral s

The Fujisaki-Okamoto Transform (idea)

- ▶ Build CCA-secure KEM from passively secure encryption scheme
- ▶ Make failure probability negligible for honest s', e', e''
- ▶ Force encapsulator to generate s', e', e'' honestly

The Fujisaki-Okamoto Transform

Alice (Server)

Gen():

$pk, sk \leftarrow \text{KeyGen}()$

Decaps((sk, pk), ct):

$x' \leftarrow \text{Decrypt}(sk, ct)$

$k', coins' \leftarrow \text{SHA3-512}(x')$

$ct' \leftarrow \text{Encrypt}(pk, x', coins')$

verify if $ct = ct'$

Bob (Client)

Encaps(pk):

$\xrightarrow{pk} x \leftarrow \{0, \dots, 255\}^{32}$

$k, coins \leftarrow \text{SHA3-512}(x)$

$\xleftarrow{ct} ct \leftarrow \text{Encrypt}(pk, x, coins)$

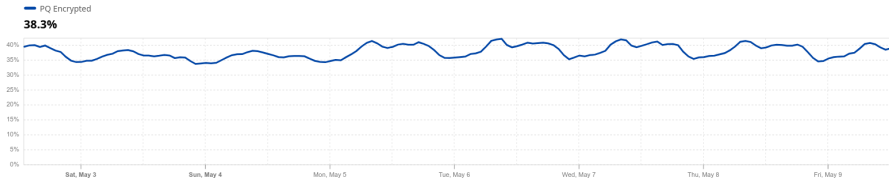
- ▶ Various tweaks through NIST PQC rounds
- ▶ Standardized in FIPS-203 as “ML-KEM” in 2024

- ▶ Various tweaks through NIST PQC rounds
- ▶ Standardized in FIPS-203 as “ML-KEM” in 2024
- ▶ Used in, e.g., Signal, iMessage, Firefox, Chrome, AWS. . .

- ▶ Various tweaks through NIST PQC rounds
- ▶ Standardized in FIPS-203 as “ML-KEM” in 2024
- ▶ Used in, e.g., Signal, iMessage, Firefox, Chrome, AWS. . .
- ▶ Used in TLS 1.3 by e.g. Cloudflare, Google

Post-quantum encryption adoption

Post-Quantum encrypted share of HTTPS request traffic ? @



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

- ▶ Several 100 billion connections per day

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>

Kyber website:

<https://pq-crystals.org/kyber>