Algorithmic of LWE-based submissions to NIST Post-Quantum Standardization Effort

Tancrède Lepoint

March 21, 2018

SRI International

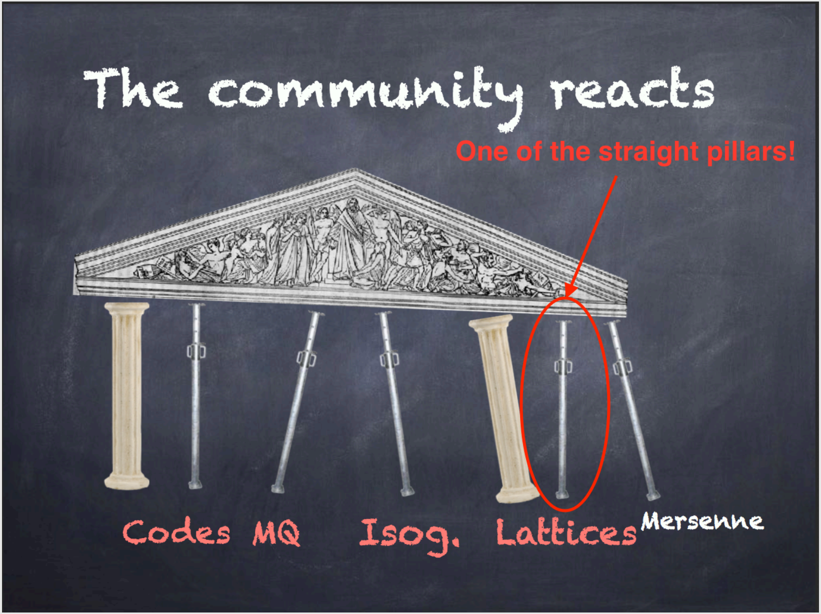
Outline

1. Introduction
2. SIS and LWE
3. Public-Key Encryption and Signatures
4. Adding Structure
5. A Few Submissions to NIST
6. Some Implementation Considerations
7. Conclusion

Small break :)

Introduction

Lattices in Antoine’s Landscape



Lattices in NIST Post-Quantum Standardization Effort

Almacrypt Workshop | Post-Quantum Crypto Lounge – SAF | Estimate all the {LWE, NTRU} schem

https://www.safecrypto.eu/pqclounge/

Home | About SAFEcrypto | More Information | Outcomes | News and Events | Post-Quantum Crypto Lounge

Ramstake Zip file	Alan Szepieniec	Lattice	Standard	KEM	Round 1		CCA	
Odd Manhattan Zip file	Thomas Plantard	Lattice	Standard	Encryption	Round 1		CPA	Not CCA secure-*patched*
NTRU Prime Zip file	Daniel J. Bernstein /Chitchanok Chuengsatiansup /Tanja Lange /Christine van Vredendaal	Lattice	Ring	KEM	Round 1		CCA2	
Three Bears Zip file	Mike Hamburg	Lattice	Module	KEM	Round 1		CCA	
CRYSTALS-KYBER Zip file	Peter Schwabe /Roberto Avanzi /Joppe Bos /Leo Ducas /Eike Kiltz /Tancrede Lepoint /Vadim Lyubashevsky /John M. Schanck /Gregor Seiler /Damien Stehle	Lattice	Module	KEM	Round 1		CCA2	Concerns surrounding proof of IND-CPA security
LOTUS Zip file	Le Trieu Phong /Takuya Hayashi /Yoshinori Aono /Shiho Moriai	Lattice	Standard	KEM Encryption	Round 1		CCA2	CCA attack-*patched*
NTRUEncrypt Zip file	Zhenfei Zhang /Cong Chen /Jeffrey Hoffstein /William Whyte	Lattice	Ring	KEM Encryption	Round 1		CCA2	
pqNTRUsign Zip file	Zhenfei Zhang /Cong Chen /Jeffrey Hoffstein /William Whyte	Lattice	Ring Module	Signature	Round 1		EUF-CMA	Vulnerable to CMA attack - *patched*
SABER Zip file	Jan-Pieter D'Anvers /Angshuman Karmakar /Sujoy Sinha Roy /Frederik Vercauteren	Lattice	Module	KEM	Round 1		CCA	

https://www.safecrypto.eu/pqclounge/

Lattices in NIST Post-Quantum Standardization Effort

Compact LWE Zip file	Dongxi Liu /Nan Li Jongkil Kim /Surya Nepa	Lattice	Standard	Encryption	Round 1	ATTACKED	CCA2	Secret key can be recovered from ciphertext
Ding Key Exchange Zip file	Jintai Ding /Tsuyoshi Takagi /Xinwei Gao /Yuntao Wang	Lattice	Ring	KEM	Round 1		CPA	
KINDI Zip file	Rachid El Bansarkhani	Lattice	Ring	KEM Encryption	Round 1		CCA	
Lizard Zip file	Jung Hee Cheon /Sangjoon Park /Joohee Lee /Duhyeong Kim /Yongsoo Song /Seungwan Hong /Dongwoo Kim /Jinsu Kim /Seong-Min Hong /Aaram Yun /Jeongsu Kim Haeryong Park /Eunyoung Choi /Kimoon kim /Jun-Sub Kim /Jieun Lee	Lattice	Standard, Ring	KEM Encryption	Round 1		CCA2	
Round2 Zip file	Oscar Garcia-Morchon /Zhenfei Zhang /Sauvik Bhattacharya /Ronald Rietman /Ludo Tolhuizen /Jose-Luis Torre-Arce	Lattice	Standard, Ring	KEM Encryption	Round 1		CCA	Concerns surrounding proof of the IND-CPA security
LIMA Zip file	Nigel P. Smart /Martin R. Albrecht /Yehuda Lindell /Emmanuela Orsini /Valery Osheter /Kenny Paterson /Guy Peer	Lattice	Ring	KEM Encryption	Round 1		CCA	Concerns surrounding rejection sampling analysis - patch proposed

https://www.safecrypto.eu/pqclounge/

Lattices in NIST Post-Quantum Standardization Effort

Home | About SAFEcrypto | More Information | Outcomes | News and Events | Post-Quantum Crypto Lounge

EMBLEM and R.EMBLEM Zip file	Minhye Seo /Jong Hwan Park /Dong Hoon Lee /Suhri Kim /Seung-Joon Lee	Lattice	Standard, Ring	Encryption	Round 1		CPA	
NewHope Zip file	Thomas Poppelmann /Erdem Alkim /Roberto Avanzi /Joppe Bos /Leo Ducas /Antonio de la Piedra /Peter Schwabe /Douglas Stebila	Lattice	Ring	KEM	Round 1		CCA	
Titanium Zip file	Ron Steinfeld /Amin Sakzad /Raymond K. Zhao	Lattice	Poly	KEM Encryption	Round 1		CCA CPA	
HILA5 Zip file	Markku-Juhani O. Saarinen	Lattice	Ring	KEM	Round 1		CPA	
qTESLA Zip file	Nina Bindel /Sedat Akleylek /Erdem Alkim /Paulo S.L.M. Barreto /Johannes Buchmann /Edward Eaton /Gus Gutoski /Juliane Kramer/ Patrick Longa /Harun Polat / Jefferson E. Ricardini /Gustavo Zanon	Lattice	Ring	Signature	Round 1		EUF-CMA	
CRYSTALS-DILITHIUM Zip file	Vadim Lyubashevsky/ Leo Ducas / Eike Kiltz /Tancrede Lepoint/ Peter Schwabe /Gregor Seiler /Damien Stehle	Lattice	Module	Signature	Round 1		SUF-CMA	
KCL (OKCN/AKCN /CNKE)	Yunlei Zhao /Zhengzhong jin /Boru Gong /Guangye Sui	Lattice	Standard, Ring	KEM Encryption	Round 1		CCA	

https://www.safecrypto.eu/pqclounge/

Lattices in NIST Post-Quantum Standardization Effort

Almacrypt Workshop | Post-Quantum Crypto Lounge – SAF | Estimate all the {LWE, NTRU} schen

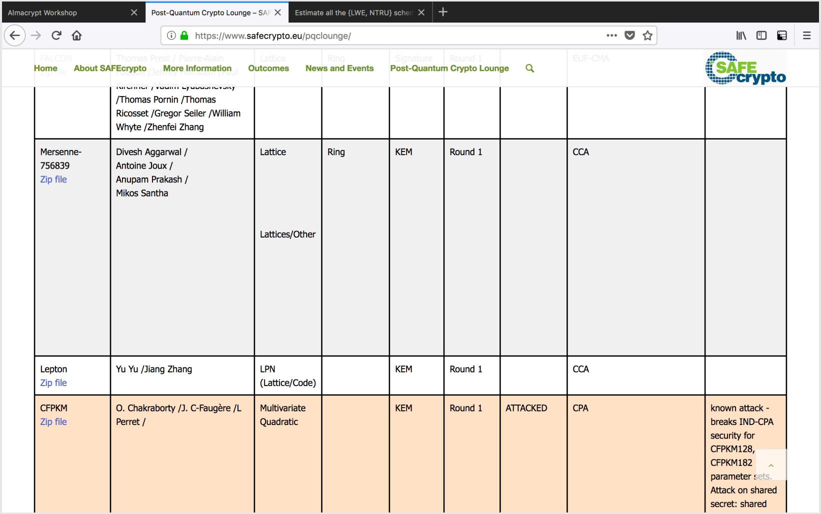
https://www.safecrypto.eu/pqclounge/

Home | About SAFEcrypto | More Information | Outcomes | News and Events | Post-Quantum Crypto Lounge

LAC Zip file	Xianhui Lu /Yamin Liu /Dingding Jia /Haiyang Xue /Jingnan He /Zhenfei Zhang	Lattice	Poly	KEM Encryption	Round 1		CCA	
DRS Zip file	Thomas Plantard/ Arnaud Sipasseuth/ Cedric Dumondelle/ Willy Susilo	Lattice	Standard	Signature	Round 1		EUF-CMA	
FrodoKEM Zip file	Michael Naehrig /Erdem Alkim /Joppe Bos /Leo Ducas /Karen Easterbrook /Brian LaMacchia /Patrick Longa /Ilya Mironov /Valeria Nikolaenko /Christopher Peikert /Ananth Raghunathan /Douglas Stebila	Lattice	Standard	KEM	Round 1		CCA	
Giophantus Zip file	Koichiro Akiyama /Yasuhiro Goto /Shinya Okumura /Tsuyoshi Takagi /Koji Nuida /Goichiro Hanaoka /Hideo Shimizu /Yasuhiko Ikematsu	Lattice	Standard	Encryption	Round 1	ATTACKED	CPA	Distinguishing attack that breaks the claimed IND-CPA security, can be avoided by switching the base ring
NTRU-HRSS-KEM Zip file	John M. Schanck /Andreas Hulsing /Joost Rijneveld /Peter Schwabe	Lattice	Ring	KEM	Round 1		CCA2	
FALCON Zip file	Thomas Prest / Pierre-Alain Fouque /Jeffrey Hoffstein /Paul	Lattice	Ring	Signature	Round 1		EUF-CMA	

https://www.safecrypto.eu/pqclounge/

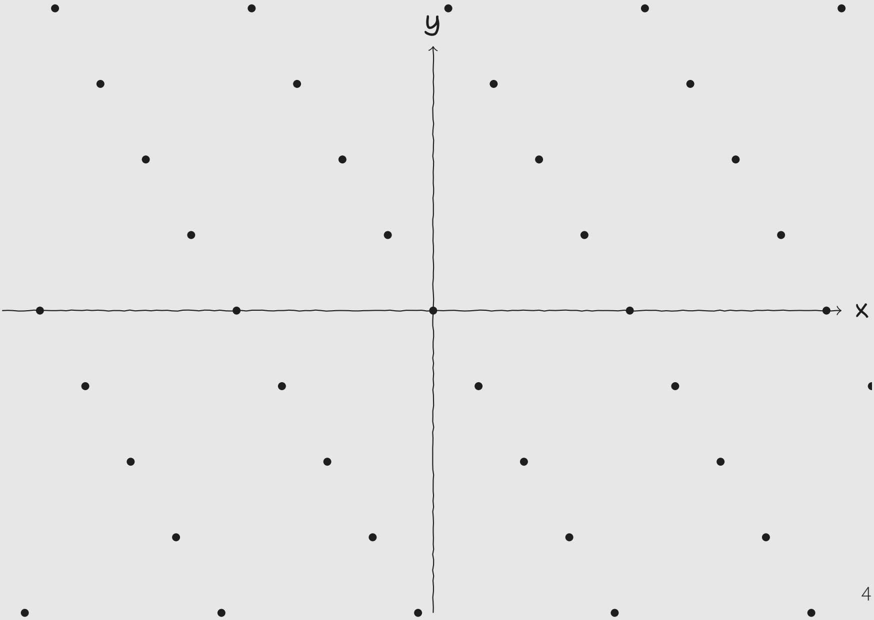
Lattices in NIST Post-Quantum Standardization Effort



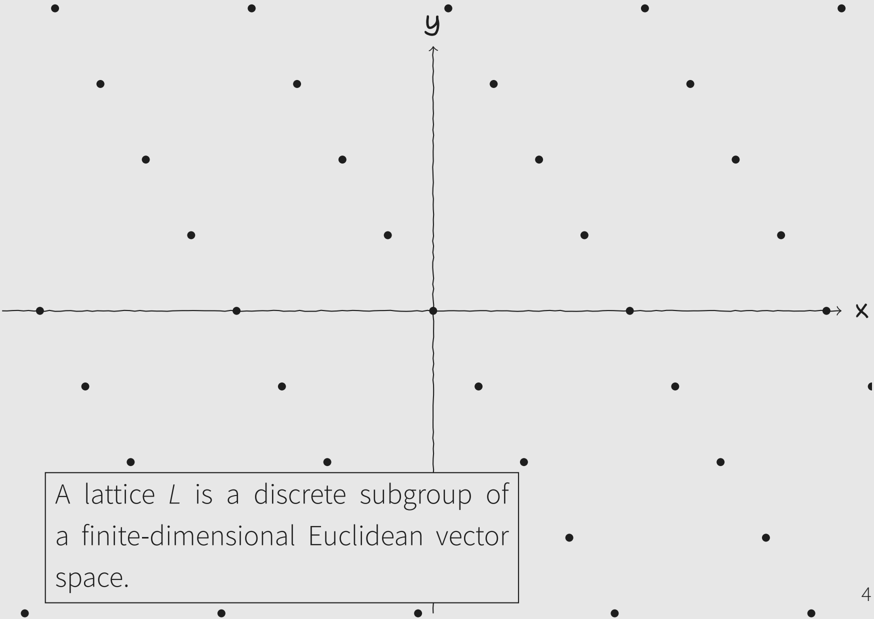
	Kirchner / Vadim Lyubashevsky /Thomas Pornin /Thomas Ricosset /Gregor Seiler /William Whyte /Zhenfei Zhang							
Mersenne-756839 Zip file	Divesh Aggarwal / Antoine Joux / Anupam Prakash / Mikos Santha	Lattice Lattices/Other	Ring	KEM	Round 1		CCA	
Lepton Zip file	Yu Yu /Jiang Zhang	LPN (Lattice/Code)		KEM	Round 1		CCA	
CFPKM Zip file	O. Chakraborty /J. C-Faugère /L Perret /	Multivariate Quadratic		KEM	Round 1	ATTACKED	CPA	known attack - breaks IND-CPA security for CFPKM128, CFPKM182 parameter sets. Attack on shared secret: shared

https://www.safecrypto.eu/pqclounge/

Lattices (Quick Reminders)

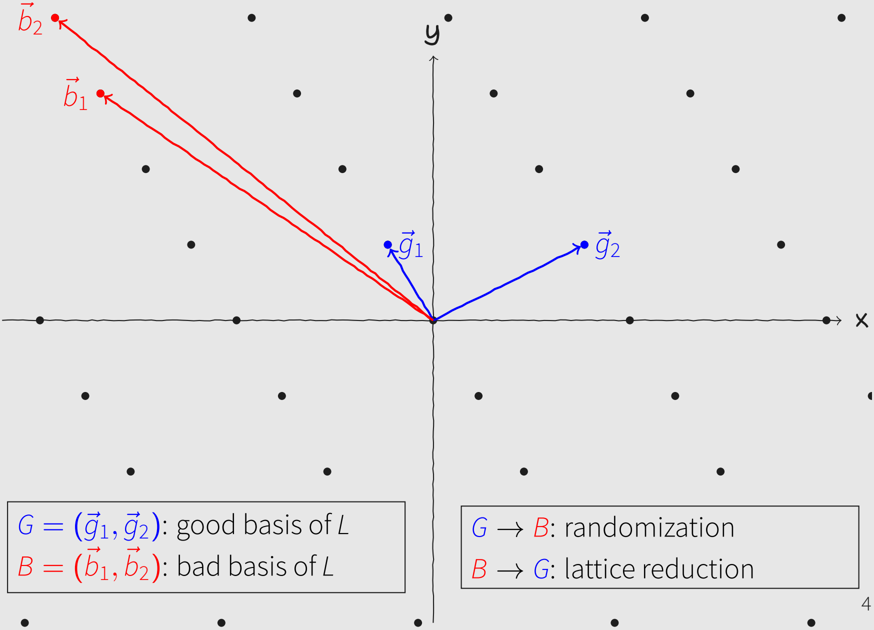


Lattices (Quick Reminders)



A lattice L is a discrete subgroup of a finite-dimensional Euclidean vector space.

Lattices (Quick Reminders)



$G = (\vec{g}_1, \vec{g}_2)$: good basis of L
$B = (\vec{b}_1, \vec{b}_2)$: bad basis of L

$G \rightarrow B$: randomization
$B \rightarrow G$: lattice reduction

Worst-Case to Average-Case Reduction

Generating Hard Instances of Lattice Problems

Extended abstract

M. Ajtai
IBM Almaden Research Center
650 Harry Road, San Jose, CA, 95120
e-mail: ajtai@almaden.ibm.com

ABSTRACT. We give a random class of lattices in $\mathbf{Z}^n$ so that, if there is a probabilistic polynomial time algorithm which finds a short vector in a random lattice with a probability of at least $\frac{1}{2}$ then there is also a probabilistic polynomial time algorithm which solves the following three lattice problems in *every* lattice in $\mathbf{Z}^n$ with a probability exponentially close to one. (1) Find the length of a shortest nonzero vector in an n-dimensional lattice, approximately, up to a polynomial factor. (2) Find the shortest nonzero vector in an n-dimensional lattice L where the shortest vector v is unique in the sense that any other vector whose length is at most $n^c\|v\|$ is parallel to v, where c is a sufficiently large absolute constant. (3) Find a basis $b_1, ..., b_n$ in the n-dimensional lattice L whose length, defined as $\max_{i=1}^{n} \|b_i\|$, is the smallest possible up to a polynomial factor.

Worst-Case to Average-Case Reduction (Credit: D. Micciancio)

Quotes from Ajtai's paper [Ajtai'96]

- "cryptography […] generation of a specific instance of a problem in NP which is thought to be difficult"
  - "NP-hard problems"
  - "very famous questions (e.g., factorization)"

  "Unfortunately, 'difficult to solve' means […] in the worst case"
- "no guidance how to create [a hard instance]"
- "possible solution"
  1. "find a set of randomly generated problems", and
  2. "show that if there is an algorithm which [works] with a positive probability, there is also an algorithm that solves the famous problem in the worst case"
- "In this paper we give such a class of random problems"

An Example: Discrete Logarithm Self-Reducibility

We use a similar property in cryptanalysis: discrete logarithm self-reducibility

To avoid this pitfall, we propose a new, faster, technique to find good representations. The main idea is that, since we are searching for a smooth representation (involving good primes only), it seems natural to try a sieving algorithm. Thus, we use some kind of sieving to write x (the number whose logarithm is wanted) as A/B, where both A and B are smooth. Since not all values of x admit a good representation, we also use the now classical trick (also in [24]) of replacing x by $z = s^i x$, where s is the largest small prime whose logarithms can be computed from the factor bases, and where i is incremented whenever we need a new value for z. Since the logarithm of s is known at this step, computing the logarithm of z clearly gives the logarithm of x.

[Joux-Lercier'2002] (citing [LaMacchia-Odlyzko'91])

An Example: Discrete Logarithm Self-Reducibility

We use a similar property in cryptanalysis: discrete logarithm self-reducibility

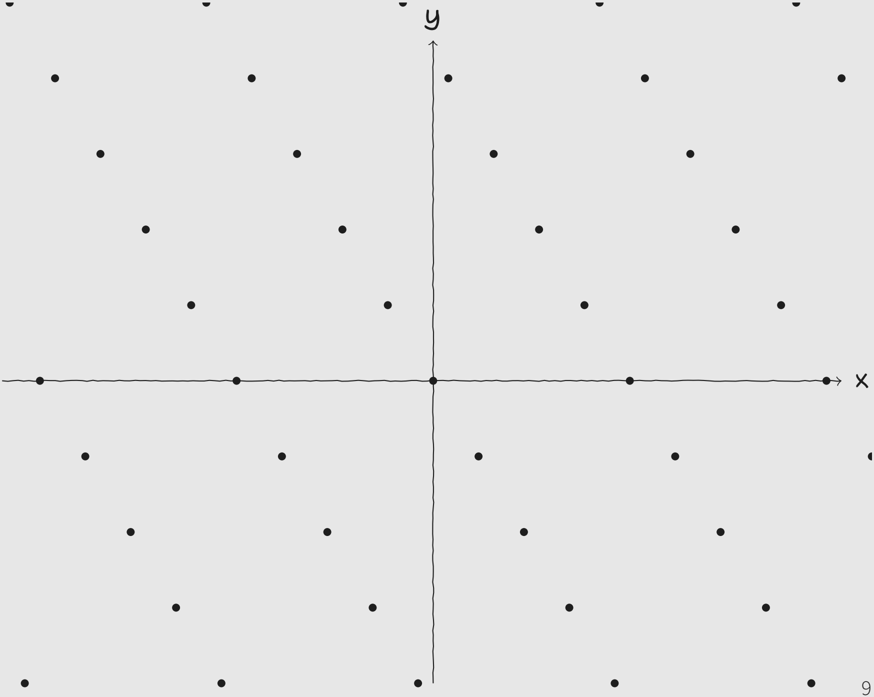
- **Goal.** Let p be a prime, $g \in \mathbf{Z}_p^\times$ generator of (prime order sub-)group $G = \{g^i \mid i \in \mathbf{Z}\}$, input $h = g^i$. Find $i \bmod |G|$.
- **Key idea.** Given $g, h \in G$, compute $g' = g^a$ and $h' = h^{ab}$ for random $a, b \in \mathbf{Z}_q^\times$.
  - g', h' almost uniformly random
  - $h' = h^{ab} = (g^i)^{ab} = (g')^{ib}$

  Finding discrete logarithm of h' wrt base point g' allows to find that of h wrt g.

This Talk

- SIS and LWE: the building blocks for lattice-based cryptography
- Regev's Encryption Scheme and Key Encapsulation Mechanism
- Ring and Module-LWE
- Specifications of real NIST candidates
- Some comments on noise errors and implementations choices

SIS and LWE


y
x

How did I draw this picture?

Matrix representation

```
\foreach \x in {-10,-9,...,10} {
  \foreach \y in {-10,-9,...,10} {
      \node[fill,inner sep=0pt, minimum size=3pt,
      circle] at ($(2*\x+1.4*\y,2*\y+\x)$) {};
    }
}
```

Matrix representation

```
\foreach \x in {-10,-9,...,10} {
  \foreach \y in {-10,-9,...,10} {
      \node[fill,inner sep=0pt, minimum size=3pt,
      circle] at ($(2*\x+1.4*\y,2*\y+\x)$) {};
    }
}
```

$$L = \mathrm{Im}\begin{pmatrix} 2 & 1.4 \\ 1 & 2 \end{pmatrix}$$

Construction of q-ary lattice (Primal)

Let q be a prime integer and $n < m$ two integer parameters.

The matrix $A \in \mathbb{Z}_q^{m \times n}$ spans the q-ary lattice:

$$\begin{aligned} \Lambda_q(A) &= \{\vec{x} \in \mathbb{Z}^m \mid \exists \vec{y} \in \mathbb{Z}_q^n, \vec{x} = A\vec{y} \bmod q\} \\ &= A \cdot \mathbb{Z}_q^n + q\mathbb{Z}^m \end{aligned}$$

Assuming A is full-rank:

- $\dim(\Lambda_q(A)) = m$
- $\mathrm{vol}(\Lambda_q(A)) = q^{m-n}$

Construction of q-ary lattice (Dual)

Let q be a prime integer and $n < m$ two integer parameters.

The matrix $A^t \in \mathbf{Z}_q^{n \times m}$ is the parity-check of the lattice:

$$\begin{aligned}\Lambda_q^{\perp}(A^t) &= \{\vec{x} \in \mathbf{Z}^m \mid A^t\vec{x} \equiv 0 \textbf{ mod } q\} \\ &= \ker(\vec{x} \mapsto A^t\vec{x} \textbf{ mod } q)\end{aligned}$$

Assuming A is full-rank:

- $\dim(\Lambda_q^{\perp}(A)) = m$
- $\mathrm{vol}(\Lambda_q^{\perp}(A)) = q^n$

The Short Integer Solution Problem (SIS)

Definition (SIS Assumption)

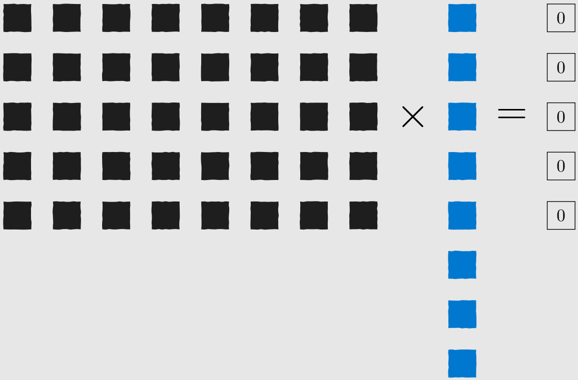
Given a random matrix A, finding a small non-zero $\vec{x} \in \mathbf{Z}_q^n$ such that $A\vec{x} \equiv 0 \bmod q$ is **hard**.

Lattice formulation

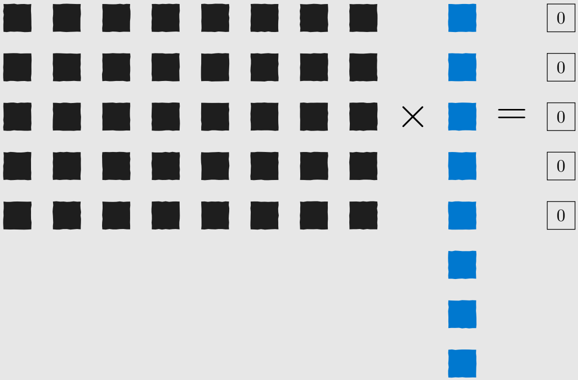
Solving Approx-SVP in $\Lambda_q^\perp(A)$ is hard.

Worst-case to average-case connection due in [Ajtai'96].

Graphical Representation of SIS

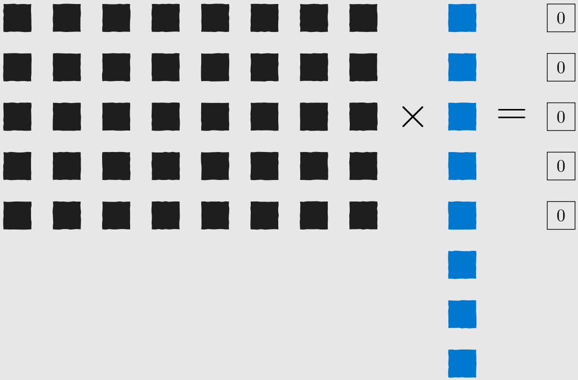


Graphical Representation of SIS



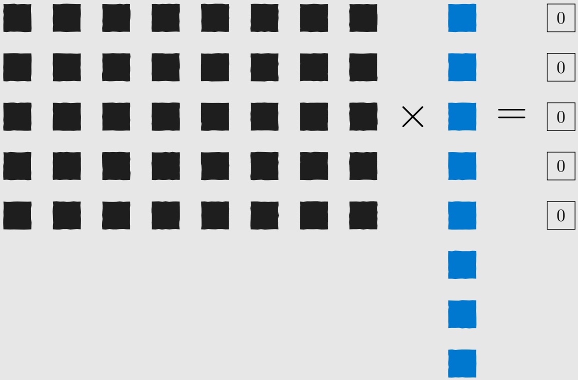
- Finding a solution $\vec{x}$ is easy

Graphical Representation of SIS



- Finding a solution $\vec{x}$ is easy
- Finding a non-zero solution $\vec{x}$ is easy

Graphical Representation of SIS



- Finding a solution $\vec{x}$ is easy
- Finding a non-zero solution $\vec{x}$ is easy
- Finding a *small non-zero* solution $\vec{x}$ can be hard

Collision Resistant Hash Function from SIS

Let $m \gg n \log q$.

$$\begin{aligned} f_A : \quad \{0,1\}^m &\to \mathbb{Z}_q^n \\ \vec{x} &\mapsto A\vec{x} \bmod q \end{aligned} .$$

SIS $\Rightarrow$ Collision Resistant Hashing (and One-Way Function)

- Collision must exist when $m > n \log q$
- Finding collision is as hard as SIS

The Learning With Error problem (LWE)

[Regev'05]

European Association for Theoretical Computer Science

2018 Gödel Prize

The 2018 Gödel Prize is awarded to Professor Oded Regev for his paper:

- On lattices, learning with errors, random linear codes, and cryptography Journal of the ACM, volume 56, issue 6, 2009 (preliminary version in the 37th annual Symposium on Theory of Computing, STOC 2005.)

This year the prize will be awarded at the 45th International Colloquium on Automata, Languages, and Programming to be held during July 9-13, 2018 in Prague, Czech Republic.

Regev's paper introduced the Learning With Errors (LWE) problem, and proved its average-case hardness assuming the worst-case (quantum) hardness of various well-studied problems on point lattices in Rn. It also gave an LWE-based public-key encryption scheme that is much simpler and more efficient than prior ones having similar worst-case hardness guarantees; this system has served as the foundation for countless subsequent works. Lastly, the paper introduced elegant and powerful techniques, including a beautiful quantum algorithm, for the study of lattice problems in cryptography and computational complexity. Regev's work has ushered in a revolution in cryptography, in both theory and practice. On the theoretical side, LWE has served as a simple and yet amazingly versatile foundation for nearly every kind of cryptographic object imaginable—along with many that were unimaginable until recently, and which still have no known constructions without LWE. Toward the practical end, LWE and its direct descendants are at the heart of several efficient real-world cryptosystems.

The Gödel Prize includes an award of USD 5,000, and is named in honor of Kurt Gödel, who was born in Austria-Hungary (now the Czech Republic) in 1906. Gödel's work has had immense impact upon scientific and philosophical thinking in the 20th century. The award recognizes his major contributions to mathematical logic and the foundations of computer science.

The Learning With Error problem (LWE) [Regev'05]

Let χ be a distribution of small errors $\ll q$.

Definition (Decisional LWE)

For $A \leftarrow \mathsf{Z}_q^{m\times n}, \vec{s} \leftarrow \mathsf{Z}_q^n$, and $\vec{e} \leftarrow \chi^m$,
distinguishing $(A, A\vec{s} + \vec{e})$ from uniform is hard.

The Learning With Error problem (LWE) [Regev'05]

Let χ be a distribution of small errors $\ll q$.

Definition (Decisional LWE)

For $A \leftarrow \mathbb{Z}_q^{m \times n}$, $\vec{s} \leftarrow \mathbb{Z}_q^n$, and $\vec{e} \leftarrow \chi^m$,
distinguishing $(A, A\vec{s} + \vec{e})$ from uniform is hard.

Definition (Search LWE)

For $A \leftarrow \mathbb{Z}_q^{m \times n}$, $\vec{s} \leftarrow \mathbb{Z}_q^n$, and $\vec{e} \leftarrow \chi^m$,
given $(A, A\vec{s} + \vec{e})$, finding $\vec{s}$ is hard.

Both problems are easily proved equivalent.

The Learning With Error problem (LWE) [Regev'05]

Let χ be a distribution of small errors $\ll q$.

Definition (Decisional LWE)

For $A \leftarrow \mathbb{Z}_q^{m\times n}$, $\vec{s} \leftarrow \mathbb{Z}_q^n$, and $\vec{e} \leftarrow \chi^m$,
distinguishing $(A, A\vec{s} + \vec{e})$ from uniform is hard.

Definition (Search LWE)

For $A \leftarrow \mathbb{Z}_q^{m\times n}$, $\vec{s} \leftarrow \mathbb{Z}_q^n$, and $\vec{e} \leftarrow \chi^m$,
given $(A, A\vec{s} + \vec{e})$, finding $\vec{s}$ is hard.

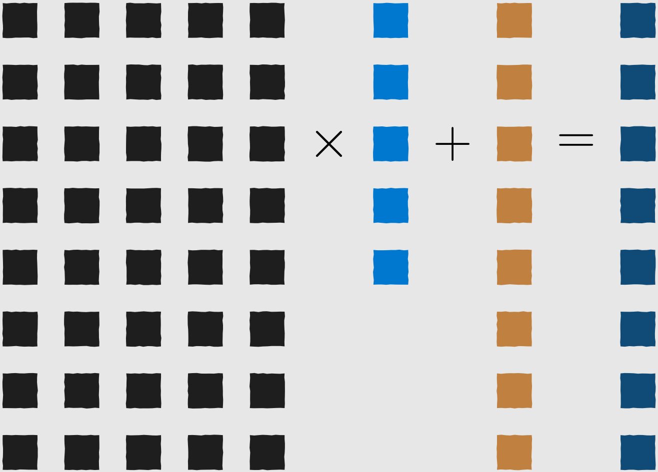
Both problems are easily proved equivalent.

Lattice formulation

Solving BDD in $\Lambda_q(A)$ is hard.

Worst-case to average-case connection due to [Regev'05].

Graphical Representation of LWE



(One-bit) Secret-Key Encryption from LWE

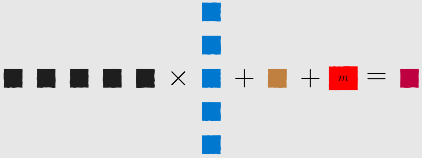
Keygen: $\mathsf{sk} = \vec{s} \leftarrow \$$

(One-bit) Secret-Key Encryption from LWE

Keygen:

$\mathsf{sk} = \vec{s} \leftarrow \$$

Encrypt:



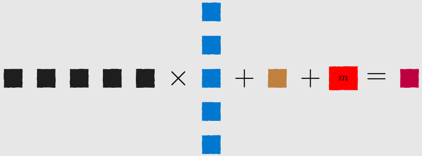
$\vec{a} \leftarrow \mathbb{Z}_q^n$
$\vec{e} \leftarrow \chi$
$ct = (\vec{a}, b = \vec{a}^t \cdot \vec{s} + \vec{e} + \lfloor q/2 \rfloor \cdot m)$

(One-bit) Secret-Key Encryption from LWE

Keygen:

$\mathsf{sk} = \vec{s} \leftarrow \$$

Encrypt:

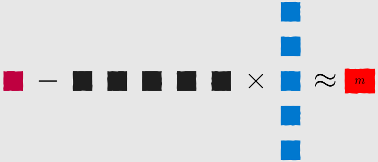


$\vec{a} \leftarrow \mathbf{Z}_q^n$

$\vec{e} \leftarrow \chi$

$ct = (\vec{a}, b = \vec{a}^t \cdot \vec{s} + \vec{e} + \lfloor q/2 \rfloor \cdot m)$

Decrypt:



$w = b - \vec{a}^t \vec{s}$

$\mu = \lfloor \frac{2}{q} w \rceil \bmod 2$

What’s “small”?

- Usually a discrete Gaussian distribution of width $s = \alpha q$ for error rate $\alpha < 1$

- Define the Gaussian function

$$\rho_s(\vec{x}) = \exp(-\pi \|\vec{x}\|^2 / s^2)$$

- The continuous Gaussian distribution has probability density function

$$f(\vec{x}) = \rho_s(\vec{x}) / \int_{\mathbb{R}^n} \rho_s(\vec{z}) d\vec{z} = \rho_s(\vec{x}) / s^n \, .$$

Reductions

- Parameters: integer n, integer modulus q, error 'rate' α ($s = q\alpha$)

Worst-case SIVP $\leq$ Search-LWE [Regev'05]

One reduction for best known parameters: any $q \geq \sqrt{n}/\alpha$

Search-LWE $\leq$ Decision-LWE

- Messy, many reductions for different q's
  - Any prime $q = \mathrm{poly}(n)$ [Regev'05]
  - Any "somewhat smooth" $q = p_1 \cdots p_t$, large p_i's [Peikert'09]
  - Any $q = p^e$ [ACPS'09,MM'11,MP'12]
  - Any q via 'modulus switching' (increases α) [BLPRS'13]
- Increasing q, α yields a weaker ultimate hardness guarantee

Public-Key Encryption

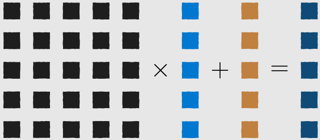
Let's encrypt (one bit)! [Regev'05,Lindner-Peikert'11]

Using a systemic-normal form, one can assume that $\vec{s} \leftarrow \chi^n$ is small as well. Take $m = n$.

Let’s encrypt (one bit)! [Regev’05,Lindner-Peikert’11]

Using a systemic-normal form, one can assume that $\vec{s} \leftarrow \chi^n$ is small as well. Take $m = n$.

Keygen:



$\vec{e} \leftarrow \chi^m$

$\mathrm{sk} = \vec{s} \leftarrow \chi^n$

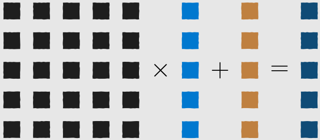
$\mathrm{pk} = (A, A\vec{s} + \vec{e})$

Let's encrypt (one bit)!

[Regev'05,Lindner-Peikert'11]

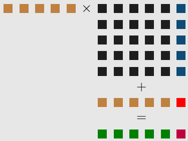
Using a systemic-normal form, one can assume that $\vec{s} \leftarrow \chi^n$ is small as well. Take $m = n$.

Keygen:



$$\vec{e} \leftarrow \chi^m$$
$$\text{sk} = \vec{s} \leftarrow \chi^n$$
$$\text{pk} = (A, A\vec{s} + \vec{e})$$

Encrypt:

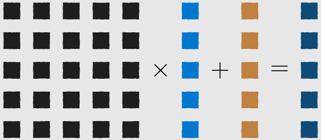


$$\vec{t} \leftarrow \chi^n$$
$$e, f \leftarrow \chi^n \times \chi$$
$$\vec{u} = \vec{t}^t A + e$$
$$v = \vec{t}^t \vec{b} + f + \lfloor q/2 \rfloor m$$

Let's encrypt (one bit)! [Regev'05,Lindner-Peikert'11]

Using a systemic-normal form, one can assume that $\vec{s} \leftarrow \chi^n$ is small as well. Take $m = n$.

Keygen:

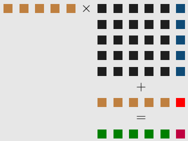


$\vec{e} \leftarrow \chi^m$

$\mathrm{sk} = \vec{s} \leftarrow \chi^n$

$\mathrm{pk} = (A, A\vec{s} + \vec{e})$

Encrypt:



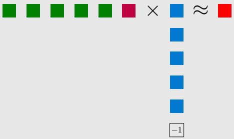
$\vec{t} \leftarrow \chi^n$

$e, f \leftarrow \chi^n \times \chi$

$\vec{u} = \vec{t}^t A + e$

$v = \vec{t}^t \vec{b} + f + \lfloor q/2 \rfloor m$

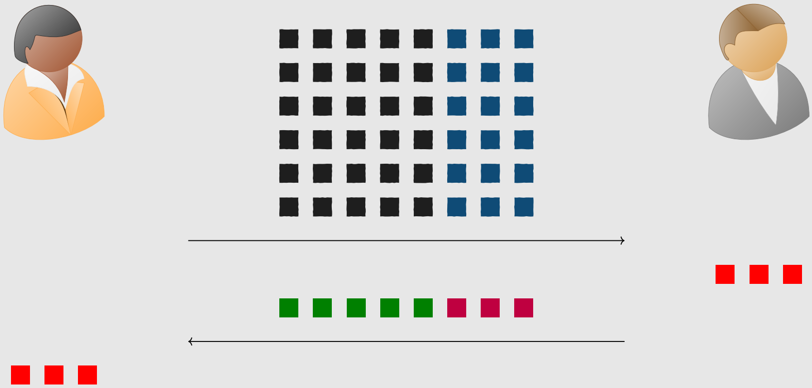
Decrypt:



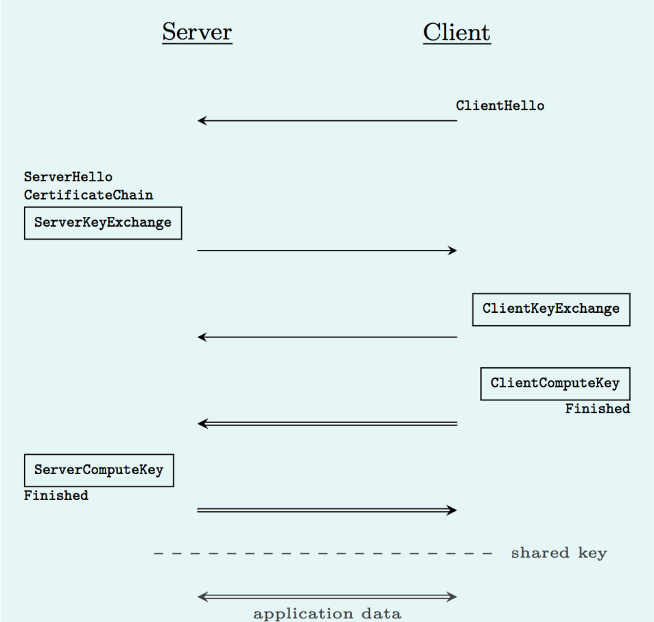
$w = v - \vec{u}^t \vec{s}$

$\mu = \left\lfloor \frac{2}{q} w \right\rfloor \bmod 2$

Key Encapsulation Mechanism



Motivation: TLS



Frodo-PKE

Algorithm 9 FrodoPKE.KeyGen.

Input: None.
Output: Key pair $(pk, sk) \in (\{0,1\}^{\mathsf{len}_\mathbf{A}} \times \mathbb{Z}_q^{n\times\overline{n}}) \times \mathbb{Z}_q^{n\times\overline{n}}$.

Choose a uniformly random seed $\mathsf{seed}_\mathbf{A} \leftarrow\!\!\$\, U(\{0,1\}^{\mathsf{len}_\mathbf{A}})$
Generate the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ via $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_\mathbf{A})$
Choose a uniformly random seed $\mathsf{seed}_\mathbf{E} \leftarrow\!\!\$\, U(\{0,1\}^{\mathsf{len}_\mathbf{E}})$
Sample error matrix $\mathbf{S} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 1)$
Sample error matrix $\mathbf{E} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 2)$
Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E}$
return public key $pk \leftarrow (\mathsf{seed}_\mathbf{A}, \mathbf{B})$ and secret key $sk \leftarrow \mathbf{S}$

Algorithm 10 FrodoPKE.Enc.

Input: Message $\mu \in \mathcal{M}$ and public key $pk = (\mathsf{seed}_\mathbf{A}, \mathbf{B}) \in \{0,1\}^{\mathsf{len}_\mathbf{A}} \times \mathbb{Z}_q^{n\times\overline{n}}$.
Output: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m}\times n} \times \mathbb{Z}_q^{\overline{m}\times\overline{n}}$.

Generate $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_\mathbf{A})$
Choose a uniformly random seed $\mathsf{seed}_\mathbf{E} \leftarrow\!\!\$\, U(\{0,1\}^{\mathsf{len}_\mathbf{E}})$
Sample error matrix $\mathbf{S}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, \overline{m}, n, T_\chi, 4)$
Sample error matrix $\mathbf{E}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, \overline{m}, n, T_\chi, 5)$
Sample error matrix $\mathbf{E}'' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, \overline{m}, \overline{n}, T_\chi, 6)$
Compute $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$ and $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
return ciphertext $c \leftarrow (\mathbf{C}_1, \mathbf{C}_2) = (\mathbf{B}', \mathbf{V} + \mathsf{Frodo.Encode}(\mu))$

Algorithm 11 FrodoPKE.Dec.

Input: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m}\times n} \times \mathbb{Z}_q^{\overline{m}\times\overline{n}}$ and secret key $sk = \mathbf{S} \in \mathbb{Z}_q^{n\times\overline{n}}$.
Output: Decrypted message $\mu' \in \mathcal{M}$.

Compute $\mathbf{M} = \mathbf{C}_2 - \mathbf{C}_1\mathbf{S}$
return message $\mu' \leftarrow \mathsf{Frodo.Decode}(\mathbf{M})$

FrodoKEM Sizes

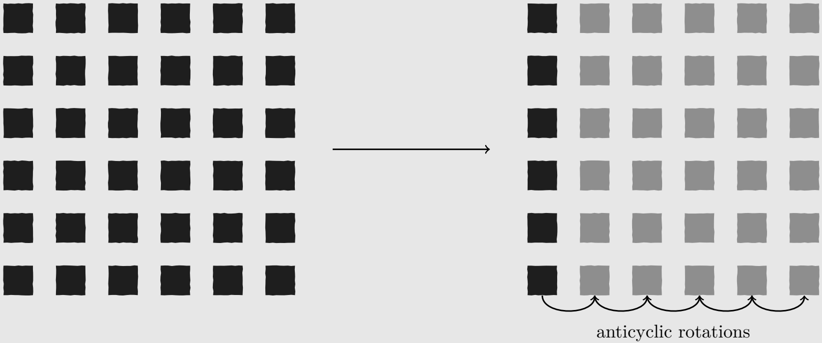
- FrodoKEM-640: $n = 640$, $q = 32768$, $\bar{m} = \bar{n} = 8$
- FrodoKEM-946: $n = 946$, $q = 65536$, $\bar{m} = \bar{n} = 8$

Table 4: **Size (in bytes) of inputs and outputs of FrodoKEM.** Secret key size is the sum of the sizes of the actual secret value and of the public key (the NIST API does not include the public key as explicit input to decapsulation).

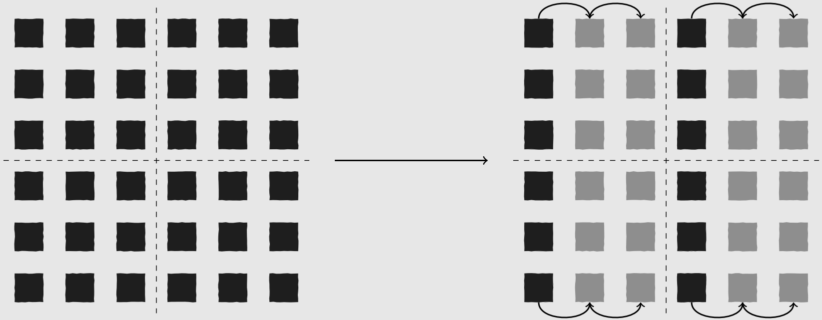
Scheme	secret key *sk*	public key *pk*	ciphertext *c*	shared secret ss
FrodoKEM-640	19,872 (10,256 + 9,616)	9,616	9,736	16
FrodoKEM-976	31,272 (15,640 + 15,632)	15,632	15,768	24

Adding Structure

Key Ideas: Ring-LWE and Module-LWE



Key Ideas: Ring-LWE and Module-LWE



Wishful thinking

(Credit slide: C. Peikert)

$$\begin{pmatrix} \vdots \\ \mathsf{a}_i \\ \vdots \end{pmatrix} \star \begin{pmatrix} \vdots \\ \mathsf{s} \\ \vdots \end{pmatrix} + \begin{pmatrix} \vdots \\ \mathsf{e}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \mathsf{b}_i \\ \vdots \end{pmatrix}$$

- Get n pseudorandom scalars from just one cheap $\star$ operation?

Wishful thinking

(Credit slide: C. Peikert)

$$\begin{pmatrix} \vdots \\ a_i \\ \vdots \end{pmatrix} \star \begin{pmatrix} \vdots \\ s \\ \vdots \end{pmatrix} + \begin{pmatrix} \vdots \\ e_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ b_i \\ \vdots \end{pmatrix}$$

- Get n pseudorandom scalars from just one cheap $\star$ operation?

Question

- How to define the product $\star$ so that (a_i, b_i) is pseudorandom?
- Careful! With small error, coordinate-wise multiplication is insecure!

Wishful thinking

(Credit slide: C. Peikert)

$$\begin{pmatrix} \vdots \\ \mathsf{a}_i \\ \vdots \end{pmatrix} \star \begin{pmatrix} \vdots \\ \mathsf{s} \\ \vdots \end{pmatrix} + \begin{pmatrix} \vdots \\ \mathsf{e}_i \\ \vdots \end{pmatrix} = \begin{pmatrix} \vdots \\ \mathsf{b}_i \\ \vdots \end{pmatrix}$$

- Get n pseudorandom scalars from just one cheap $\star$ operation?

Question

- How to define the product $\star$ so that $(\mathsf{a}_i, \mathsf{b}_i)$ is pseudorandom?
- Careful! With small error, coordinate-wise multiplication is insecure!

Answer

- $\star$ = multiplication in a polynomial ring: e.g., $\mathbb{Z}_q[x]/(x^n + 1)$. Fast and practical with FFT: $n \log n$ operations mod q.
- Same ring structures used in NTRU cryptosystem [HPS'98], & in compact one-way / CR hash functions [Mic'02,PR'06,LM'06,...]

- Let R be a ring, often $R = \mathbb{Z}[x]/(f(x))$ for irreducible f of degree n (or $R = \mathcal{O}_K$).
  Has a 'dual ideal' $R^\vee$ (w.r.t. 'canonical' geometry).

Ring Learning With Errors

[LPR'10] (Slide credit: C. Peikert)

- Let R be a ring, often $R = \mathbb{Z}[x]/(f(x))$ for irreducible f of degree n (or $R = \mathcal{O}_K$).
  Has a 'dual ideal' $R^\vee$ (w.r.t. 'canonical' geometry).
- Integer modulus q defining $R_q = R/qR$ and $R_q^\vee = R^\vee/qR^\vee$

Ring Learning With Errors

- Let R be a ring, often $R = \mathbb{Z}[x]/(f(x))$ for irreducible f of degree n (or $R = \mathcal{O}_K$).
  Has a 'dual ideal' $R^\vee$ (w.r.t. 'canonical' geometry).
- Integer modulus q defining $R_q = R/qR$ and $R_q^\vee = R^\vee/qR^\vee$
- Gaussian error of width $\approx \alpha q$ over $R^\vee$

- Let R be a ring, often $R = \mathbb{Z}[x]/(f(x))$ for irreducible f of degree n (or $R = \mathcal{O}_K$).
  Has a 'dual ideal' $R^\vee$ (w.r.t. 'canonical' geometry).
- Integer modulus q defining $R_q = R/qR$ and $R_q^\vee = R^\vee/qR^\vee$
- Gaussian error of width $\approx \alpha q$ over $R^\vee$

Definition (Search R-LWE)

Find secret ring element $s \in R_q^\vee$, given m independent samples $(\mathsf{a}_i, \mathsf{b}_i = \mathsf{a}_i \cdot \mathsf{s} + \mathsf{e}_i)$

Ring Learning With Errors

[LPR'10] (Slide credit: C. Peikert)

- Let R be a ring, often $R = \mathbb{Z}[x]/(f(x))$ for irreducible f of degree n (or $R = \mathcal{O}_K$).
  Has a 'dual ideal' $R^\vee$ (w.r.t. 'canonical' geometry).
- Integer modulus q defining $R_q = R/qR$ and $R_q^\vee = R^\vee/qR^\vee$
- Gaussian error of width $\approx \alpha q$ over $R^\vee$

Definition (Search R-LWE)

Find secret ring element $s \in R_q^\vee$, given m independent samples $(\mathsf{a}_i, \mathsf{b}_i = \mathsf{a}_i \cdot \mathsf{s} + \mathsf{e}_i)$

Definition (Decisional R-LWE)

Distinguish $(\mathsf{a}_i, \mathsf{b}_i)$ from uniform $(\mathsf{a}_i, \mathsf{b}_i) \in R_q \times R_q^\vee$

Hardness of Ring-LWE

[PeiRegSte'17]

Reduction

worst-case (n^c/α)-SIVP $\leq$ worst-case Ring-LWE$_{q,\alpha}$

on ideal lattices in R

(quantum, any $R = \mathcal{O}_K$, any $q \geq n^{c-1/2}/\alpha$)

Hardness of Ring-LWE

[PeiRegSte'17]

Reduction

worst-case (n^c/α)-SIVP $\leq$ worst-case Ring-LWE$_{q,\alpha}$
on ideal lattices in R

(quantum, any $R = \mathcal{O}_K$, any $q \geq n^{c-1/2}/\alpha$)

Which ring to use?

- Previous result gives no guidance
- There exists no nontrivial relation between lattice problems over different rings
- Progress on Ideal-SIVP
  - Quantum poly-time $\exp(\tilde{O}(\sqrt{n}))$-Ideal-SIVP in prime power cyclotomics [Ber14,CGS14,BS16,CDPR16,CDW17]
  - Classical quasi-poly-time in multiquadratic fields [Ber14,BBdVLvV'17]

Rings in Literature

What are the typical options for *R* throughout the literature?

Rings in Literature

What are the typical options for R throughout the literature?

- $R = \mathbb{Z}[x]/(x^n - 1)$ [HPS'96, Mic'02]

Rings in Literature

What are the typical options for R throughout the literature?

- $R = \mathbb{Z}[x]/(x^n - 1)$ [HPS'96, Mic'02]
- $R = \mathbb{Z}[x]/(x^n + 1)$ & cyclotomics [LR'06,ML'06,LPR'10]

Rings in Literature

What are the typical options for R throughout the literature?

- $R = \mathsf{Z}[x]/(x^n - 1)$ [HPS'96, Mic'02]
- $R = \mathsf{Z}[x]/(x^n + 1)$ & cyclotomics [LR'06,ML'06,LPR'10]
- Alternate rings with even less structure like $\mathsf{Z}[x]/(x^p - x - 1)$ [BCLvV'16]
    - Complex (but still fast!) multiplication

Rings in Literature

What are the typical options for R throughout the literature?

- $R = \mathsf{Z}[x]/(x^n - 1)$ [HPS'96, Mic'02]
- $R = \mathsf{Z}[x]/(x^n + 1)$ & cyclotomics [LR'06,ML'06,LPR'10]
- Alternate rings with even less structure like $\mathsf{Z}[x]/(x^p - x - 1)$ [BCLvV'16]
    - Complex (but still fast!) multiplication
- Module-LWE [LS'15]
    - MLWE bridges a gap between LWE and RLWE;
    - $R = R_1 \times \cdots \times R_\ell$;
    - where each R_i can have a different structure;
    - LWE: $R_i = \mathsf{Z}$; RLWE: $\ell = 1$;
    - Used in the CRYSTALS crypto suite (Kyber; Dilithium) to be submitted to NIST

Power-of-2 cyclotomic

$$R = \mathbb{Z}[x]/(x^n + 1)$$

- Choosing small errors in the **polynomial embedding** is equivalent to selecting small errors in the canonical embedding (where the actual Ring-LWE problem lies)
- **Fast polynomial multiplication** using the Fast Fourier Transform ($n \log n$ operations over coefficients mod q)
- No indication that these rings would be insecure; most **widely studied**, and best understood, rings (along with other cyclotomic rings) in algebraic number theory

Module-LWE

Let R be a ring, q be an integer, and $R_q = R/qR$. Let χ be a distribution of small errors. Let k, ℓ be parameter integers.

Module-LWE

Let R be a ring, q be an integer, and $R_q = R/qR$. Let χ be a distribution of small errors. Let k, ℓ be parameter integers.

Decision Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
distinguish $(A, A\vec{s} + \vec{e})$ from uniform is hard.

Module-LWE

Let R be a ring, q be an integer, and $R_q = R/qR$. Let χ be a distribution of small errors. Let k, ℓ be parameter integers.

Decision Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
distinguish $(A, A\vec{s} + \vec{e})$ from uniform is hard.

Search Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
given $(A, A\vec{s} + \vec{e})$, finding $\vec{s}$ is hard.

Module-LWE

Let R be a ring, q be an integer, and $R_q = R/qR$. Let χ be a distribution of small errors. Let k, ℓ be parameter integers.

Decision Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
distinguish $(A, A\vec{s} + \vec{e})$ from uniform is hard.

Search Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
given $(A, A\vec{s} + \vec{e})$, finding $\vec{s}$ is hard.

- For $R = \mathbb{Z}[x]/(x^n + 1)$ and $\ell = 1$, we get Ring-LWE

Module-LWE

Let R be a ring, q be an integer, and $R_q = R/qR$. Let χ be a distribution of small errors. Let k, ℓ be parameter integers.

Decision Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
distinguish $(A, A\vec{s} + \vec{e})$ from uniform is hard.

Search Module-LWE

For $A \leftarrow R_q^{k\times\ell}, \vec{s} \leftarrow R_q^{\ell}$, and $\vec{e} \leftarrow (\chi^n)^k$,
given $(A, A\vec{s} + \vec{e})$, finding $\vec{s}$ is hard.

- For $R = \mathbb{Z}[x]/(x^n + 1)$ and $\ell = 1$, we get Ring-LWE
- For $R = \mathbb{Z}$ and $\ell = n$, we get LWE

A Few Submissions to NIST

Let’s review some submissions

- FrodoKEM: Key Encapsulation Mechanism based on LWE
- NewHope: Key Encapsulation Mechanism based on Ring-LWE
- CRYSTALS-Kyber: Key Encapsulation Mechanism based on Module-LWE

- Special mentions of ThreeBears, OddManhattan, Titanium

Frodo-PKE

Algorithm 9 FrodoPKE.KeyGen.

Input: None.
Output: Key pair $(pk, sk) \in (\{0,1\}^{\mathsf{len}_\mathbf{A}} \times \mathbb{Z}_q^{n\times\overline{n}}) \times \mathbb{Z}_q^{n\times\overline{n}}$.

- Choose a uniformly random seed $\mathsf{seed}_\mathbf{A} \leftarrow_\$ U(\{0,1\}^{\mathsf{len}_\mathbf{A}})$
- Generate the matrix $\mathbf{A} \in \mathbb{Z}_q^{n\times n}$ via $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_\mathbf{A})$
- Choose a uniformly random seed $\mathsf{seed}_\mathbf{E} \leftarrow_\$ U(\{0,1\}^{\mathsf{len}_\mathbf{E}})$
- Sample error matrix $\mathbf{S} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 1)$
- Sample error matrix $\mathbf{E} \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, n, \overline{n}, T_\chi, 2)$
- Compute $\mathbf{B} = \mathbf{AS} + \mathbf{E}$
- **return** public key $pk \leftarrow (\mathsf{seed}_\mathbf{A}, \mathbf{B})$ and secret key $sk \leftarrow \mathbf{S}$

Algorithm 10 FrodoPKE.Enc.

Input: Message $\mu \in \mathcal{M}$ and public key $pk = (\mathsf{seed}_\mathbf{A}, \mathbf{B}) \in \{0,1\}^{\mathsf{len}_\mathbf{A}} \times \mathbb{Z}_q^{n\times\overline{n}}$.
Output: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m}\times n} \times \mathbb{Z}_q^{\overline{m}\times\overline{n}}$.

- Generate $\mathbf{A} \leftarrow \mathsf{Frodo.Gen}(\mathsf{seed}_\mathbf{A})$
- Choose a uniformly random seed $\mathsf{seed}_\mathbf{E} \leftarrow_\$ U(\{0,1\}^{\mathsf{len}_\mathbf{E}})$
- Sample error matrix $\mathbf{S}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, \overline{m}, n, T_\chi, 4)$
- Sample error matrix $\mathbf{E}' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, \overline{m}, n, T_\chi, 5)$
- Sample error matrix $\mathbf{E}'' \leftarrow \mathsf{Frodo.SampleMatrix}(\mathsf{seed}_\mathbf{E}, \overline{m}, \overline{n}, T_\chi, 6)$
- Compute $\mathbf{B}' = \mathbf{S}'\mathbf{A} + \mathbf{E}'$ and $\mathbf{V} = \mathbf{S}'\mathbf{B} + \mathbf{E}''$
- **return** ciphertext $c \leftarrow (\mathbf{C}_1, \mathbf{C}_2) = (\mathbf{B}', \mathbf{V} + \mathsf{Frodo.Encode}(\mu))$

Algorithm 11 FrodoPKE.Dec.

Input: Ciphertext $c = (\mathbf{C}_1, \mathbf{C}_2) \in \mathbb{Z}_q^{\overline{m}\times n} \times \mathbb{Z}_q^{\overline{m}\times\overline{n}}$ and secret key $sk = \mathbf{S} \in \mathbb{Z}_q^{n\times\overline{n}}$.
Output: Decrypted message $\mu' \in \mathcal{M}$.

- Compute $\mathbf{M} = \mathbf{C}_2 - \mathbf{C}_1\mathbf{S}$
- **return** message $\mu' \leftarrow \mathsf{Frodo.Decode}(\mathbf{M})$

NewHope-PKE

Algorithm 1 NewHope-CPA-PKE Key Generation

function NewHope-CPA-PKE.Gen()
  $seed \xleftarrow{\$} \{0, \ldots, 255\}^{32}$
  $z \leftarrow \mathsf{SHAKE256}(64, seed)$
  $publicseed \leftarrow z[0{:}31]$
  $noiseseed \leftarrow z[32{:}63]$
  $\hat{\mathbf{a}} \leftarrow \mathsf{GenA}(publicseed)$
  $\mathbf{s} \leftarrow \mathsf{PolyBitRev}(\mathsf{Sample}(noiseseed, 0))$
  $\hat{\mathbf{s}} \leftarrow \mathsf{NTT}(\mathbf{s})$
  $\mathbf{e} \leftarrow \mathsf{PolyBitRev}(\mathsf{Sample}(noiseseed, 1))$
  $\hat{\mathbf{e}} \leftarrow \mathsf{NTT}(\mathbf{e})$
  $\hat{\mathbf{b}} \leftarrow \hat{\mathbf{a}} \circ \hat{\mathbf{s}} + \hat{\mathbf{e}}$
  return $(pk = \mathsf{EncodePK}(\hat{\mathbf{b}}, publicseed), sk = \mathsf{EncodePolynomial}(\mathbf{s}))$

Algorithm 2 NewHope-CPA-PKE Encryption

function NewHope-CPA-PKE.Encrypt($pk \in \{0, \ldots, 255\}^{7 \cdot n/4+32}$, $\mu \in \{0, \ldots, 255\}^{32}$, $coin \in \{0, \ldots, 255\}^{32}$)
  $(\hat{\mathbf{b}}, publicseed) \leftarrow \mathsf{DecodePk}(pk)$
  $\hat{\mathbf{a}} \leftarrow \mathsf{GenA}(publicseed)$
  $\mathbf{s}' \leftarrow \mathsf{PolyBitRev}(\mathsf{Sample}(coin, 0))$
  $\mathbf{e}' \leftarrow \mathsf{PolyBitRev}(\mathsf{Sample}(coin, 1))$
  $\mathbf{e}'' \leftarrow \mathsf{Sample}(coin, 2)$
  $\hat{\mathbf{t}} \leftarrow \mathsf{NTT}(\mathbf{s}')$
  $\hat{\mathbf{u}} \leftarrow \hat{\mathbf{a}} \circ \hat{\mathbf{t}} + \mathsf{NTT}(\mathbf{e}')$
  $\mathbf{v} \leftarrow \mathsf{Encode}(\mu)$
  $\mathbf{v}' \leftarrow \mathsf{NTT}^{-1}(\hat{\mathbf{b}} \circ \hat{\mathbf{t}}) + \mathbf{e}'' + \mathbf{v}$
  $h \leftarrow \mathsf{Compress}(\mathbf{v}')$
  return $c = \mathsf{EncodeC}(\hat{\mathbf{u}}, h)$

Algorithm 3 NewHope-CPA-PKE Decryption

function NewHope-CPA-PKE.Decrypt($c \in \{0, \ldots, 255\}^{7\frac{n}{4}+3\frac{n}{8}}$, $sk \in \{0, \ldots, 255\}^{7 \cdot n/4}$)
  $(\hat{\mathbf{u}}, h) \leftarrow \mathsf{DecodeC}(c)$
  $\hat{\mathbf{s}} \leftarrow \mathsf{DecodePolynomial}(sk)$
  $\mathbf{v}' \leftarrow \mathsf{Decompress}(h)$
  $\mu \leftarrow \mathsf{Decode}(\mathbf{v}' - \mathsf{NTT}^{-1}(\hat{\mathbf{u}} \circ \hat{\mathbf{s}}))$
  return μ

CRYSTALS-Kyber-PKE

Algorithm 4 $\textsc{Kyber}.\mathsf{CPAPKE.KeyGen}()$: key generation

Output: Secret key $sk \in \mathcal{B}^{13 \cdot k \cdot n/8}$
Output: Public key $pk \in \mathcal{B}^{d_t \cdot k \cdot n/8+32}$

$d \leftarrow \mathcal{B}^{32}$
$(\rho, \sigma) := \mathrm{G}(d)$
$N := 0$
for i from 0 to $k-1$ **do** $\quad\triangleright$ Generate matrix $\hat{\mathbf{A}} \in R_q^{k \times k}$ in NTT domain
  for j from 0 to $k-1$ **do**
    $\hat{\mathbf{A}}[i][j] := \mathsf{Parse}(\mathsf{XOF}(\rho \| j \| i))$
  end for
end for
for i from 0 to $k-1$ **do** $\quad\triangleright$ Sample $\mathbf{s} \in R_q^k$ from B_η
  $\mathbf{s}[i] := \mathsf{CBD}_\eta(\mathsf{PRF}(\sigma, N))$
  $N := N + 1$
end for
for i from 0 to $k-1$ **do** $\quad\triangleright$ Sample $\mathbf{e} \in R_q^k$ from B_η
  $\mathbf{e}[i] := \mathsf{CBD}_\eta(\mathsf{PRF}(\sigma, N))$
  $N := N + 1$
end for
$\hat{\mathbf{s}} := \mathsf{NTT}(\mathbf{s})$
$\mathbf{t} := \mathsf{NTT}^{-1}(\hat{\mathbf{A}} \circ \hat{\mathbf{s}}) + \mathbf{e}$
$pk := (\mathsf{Encode}_{d_t}(\mathsf{Compress}_q(\mathbf{t}, d_t)) \| \rho)$ $\quad\triangleright$ $pk := \mathbf{As} + \mathbf{e}$
$sk := \mathsf{Encode}_{13}(\hat{\mathbf{s}} \mod {}^{+}q)$ $\quad\triangleright$ $sk := \mathbf{s}$
return (pk, sk)

CRYSTALS-Kyber-PKE

Algorithm 5 KYBER.CPAPKE.Enc(pk, m, r): encryption

Input: Public key $pk \in \mathcal{B}^{d_t \cdot k \cdot n/8+32}$
Input: Message $m \in \mathcal{B}^{32}$
Input: Random coins $r \in \mathcal{B}^{32}$
Output: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$

```
N := 0
t := Decompress_q(Decode_{d_t}(pk), d_t)
ρ := pk + d_t · k · n/8
for i from 0 to k − 1 do                          ▷ Generate matrix Â ∈ R_q^{k×k} in NTT domain
    for j from 0 to k − 1 do
        Â^T[i][j] := Parse(XOF(ρ‖i‖j))
    end for
end for
for i from 0 to k − 1 do                          ▷ Sample r ∈ R_q^k from B_η
    r[i] := CBD_η(PRF(r, N))
    N := N + 1
end for
for i from 0 to k − 1 do                          ▷ Sample e_1 ∈ R_q^k from B_η
    e_1[i] := CBD_η(PRF(r, N))
    N := N + 1
end for
e_2 := CBD_η(PRF(r, N))                           ▷ Sample e_2 ∈ R_q from B_η
r̂ := NTT(r)
u := NTT^{-1}(Â^T ∘ r̂) + e_1                      ▷ u := A^T r + e_1
v := NTT^{-1}(NTT(t)^T ∘ r̂) + e_2 + Decode_1(Decompress_q(m, 1))   ▷ v := t^T r + e_2 + Decompress_q(m, 1)
c_1 := Encode_{d_u}(Compress_q(u, d_u))
c_2 := Encode_{d_v}(Compress_q(v, d_v))
return c = (c_1‖c_2)                              ▷ c := (Compress_q(u, d_u), Compress_q(v, d_v))
```

CRYSTALS-Kyber-PKE

Algorithm 6 KYBER.CPAPKE.Dec(sk, c): decryption

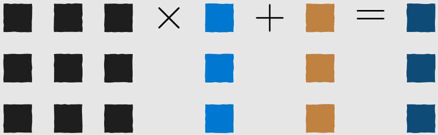
Input: Secret key $sk \in \mathcal{B}^{13 \cdot k \cdot n/8}$
Input: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Output: Message $m \in \mathcal{B}^{32}$

```
u := Decompress_q(Decode_{d_u}(c), d_u)
v := Decompress_q(Decode_{d_v}(c + d_u · k · n/8), d_v)
ŝ := Decode_13(sk)
m := Encode_1(Compress_q(v − NTT^{-1}(ŝ^T ∘ NTT(u)), 1))    ▷ m := Compress_q(v − s^T u, 1)
return m
```

$\mathbf{u} := \mathsf{Decompress}_q(\mathsf{Decode}_{d_u}(c), d_u)$
$v := \mathsf{Decompress}_q(\mathsf{Decode}_{d_v}(c + d_u \cdot k \cdot n/8), d_v)$
$\hat{\mathbf{s}} := \mathsf{Decode}_{13}(sk)$
$m := \mathsf{Encode}_1(\mathsf{Compress}_q(v - \mathsf{NTT}^{-1}(\hat{\mathbf{s}}^T \circ \mathsf{NTT}(\mathbf{u})), 1))$ $\quad \triangleright\ m := \mathsf{Compress}_q(v - \mathbf{s}^T \mathbf{u}, 1)$
return m

CRYSTALS-Kyber-PKE Graphically

Keygen:



Notation:

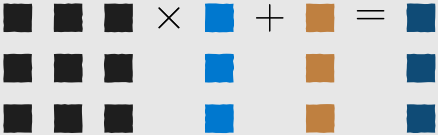
$\leftarrow \mathbb{Z}_{7681}[X]/(X^{256}+1)$

$\leftarrow \left(\sum_{i=1}^{4}(a_i - b_i)\right)^{256}$

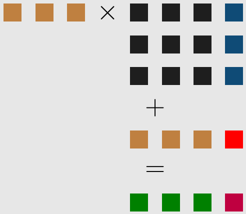
$\leftarrow \left(\sum_{i=1}^{4}(a_i - b_i)\right)^{256}$

$\in \{0, \lfloor q/2 \rfloor\}^{256}$

Keygen:



Encrypt:



Notation:

$\leftarrow Z_{7681}[X]/(X^{256}+1)$

$\leftarrow (\sum_{i=1}^{4}(a_i - b_i))^{256}$

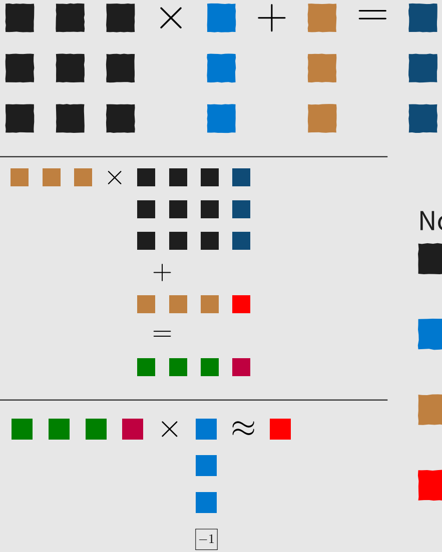
$\leftarrow (\sum_{i=1}^{4}(a_i - b_i))^{256}$

$\in \{0, \lfloor q/2 \rfloor\}^{256}$

Keygen:

Encrypt:

Decrypt:



Notation:

$\leftarrow \mathbb{Z}_{7681}[x]/(x^{256}+1)$

$\leftarrow (\sum_{i=1}^{4}(a_i - b_i))^{256}$

$\leftarrow (\sum_{i=1}^{4}(a_i - b_i))^{256}$

$\in \{0, \lfloor q/2 \rfloor\}^{256}$

CRYSTALS-Kyber-PKE on 1 slide

KeyGen()

$\rho \leftarrow \{0,1\}^{256}$

$A \leftarrow \mathrm{XOF}(\rho)$

$\vec{s}, \vec{e} \leftarrow (\chi^{256})^{\ell}$

$\vec{t} = \mathrm{Compress}(A\vec{s} + \vec{e}, dt)$

$pk = (\vec{t}, \rho), sk = \vec{s}$

Enc$(pk, \vec{m} \in \{0,1\}^{256})$

$\vec{t} = \mathrm{Decompress}(t, d_t)$

$\vec{A} \leftarrow \mathrm{XOF}(\rho)$

$\vec{r}, \vec{e}_1, e_2 \leftarrow (\chi^{256})^{\ell} \times (\chi^{256})^{\ell} \times \chi^{256}$

$\vec{u} = \mathrm{Compress}(\vec{r}^T A + \vec{e}_1^T, d_u)$

$v = \mathrm{Compress}(\vec{r}^T \vec{t} + e_2 + \lfloor q/2 \rfloor \vec{m}, d_v)$

$ct = (\vec{u}, v)$

Dec(sk, ct)

$\vec{u} \leftarrow \mathrm{Decompress}(\vec{u}, d_u)$

$v \leftarrow \mathrm{Decompress}(v, d_v)$

$m = \mathrm{Compress}(v - \vec{s}^t \cdot \vec{u}, 1)$

The Fujisaki-Okamoto Transform

- Constructs an IND-CCA2-secure public key encryption scheme from a one-way-secure public key encryption scheme in the classical random oracle model
- Variant by Targhi and Unruh against a **quantum adversary** in the **quantum random oracle model**

The Fujisaki-Okamoto Transform

- Constructs an IND-CCA2-secure public key encryption scheme from a one-way-secure public key encryption scheme in the classical random oracle model
- Variant by Targhi and Unruh against a **quantum adversary** in the **quantum random oracle model**

Key ideas

- Encapsulate: hash a seed *s* to get (1) the key and (2) the randomness seed *r*, and encrypt *s* using *r*.
- Decapsulate: decrypt to recover *s*, and **re-encrypt**. If the ciphertext is the same, then use the key.

- Dennis Hofheinz, Kathrin Hövelmanns, Eike Kiltz: A Modular Analysis of the Fujisaki-Okamoto Transformation. TCC (1) 2017.

CRYSTALS-Kyber

Algorithm 7 $\text{KYBER.CCAKEM.KeyGen}()$

Output: Public key $pk \in \mathcal{B}^{d_t \cdot k \cdot n/8+32}$
Output: Secret key $sk \in \mathcal{B}^{(13+d_t)\cdot k \cdot n/8+96}$
$z \leftarrow \mathcal{B}^{32}$
$(pk, sk') := \text{KYBER.CPAPKE.KeyGen}()$
$sk := (sk' \| pk \| \mathrm{H}(pk) \| z)$
return (pk, sk)

Algorithm 8 $\text{KYBER.CCAKEM.Enc}(pk)$

Input: Public key $pk \in \mathcal{B}^{d_t \cdot k \cdot n/8+32}$
Output: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Output: Shared key $K \in \mathcal{B}^{32}$
$m \leftarrow \mathcal{B}^{32}$
$m \leftarrow \mathrm{H}(m)$ ▷ Do not send output of system RNG
$(\bar{K}, r) := \mathrm{G}(m \| \mathrm{H}(pk))$
$c := \text{KYBER.CPAPKE.Enc}(pk, m; r)$
$K := \mathrm{H}(\bar{K} \| \mathrm{H}(c))$
return (c, K)

Algorithm 9 $\text{KYBER.CCAKEM.Dec}(c, sk)$

Input: Ciphertext $c \in \mathcal{B}^{d_u \cdot k \cdot n/8 + d_v \cdot n/8}$
Input: Secret key $sk \in \mathcal{B}^{(13+d_t)\cdot k \cdot n/8+96}$
Output: Shared key $K \in \mathcal{B}^{32}$
$pk := sk + 13 \cdot k \cdot n/8$
$h := sk + (13 + d_t) \cdot k \cdot n/8 + 32 \in \mathcal{B}^{32}$
$z := sk + (13 + d_t) \cdot k \cdot n/8 + 64$
$m' := \text{KYBER.CPAPKE.Dec}(\mathbf{s}, (\mathbf{u}, v))$
$(\bar{K}', r') := \mathrm{G}(m' \| h)$
$c' := \text{KYBER.CPAPKE.Enc}(pk, m', r')$
if $c = c'$ **then**
  return $K := \mathrm{H}(\bar{K}' \| \mathrm{H}(c))$
else
  return $K := \mathrm{H}(z \| \mathrm{H}(c))$

CRYSTALS-Kyber Security

(Credit Slide: P. Schwabe)

- Tight reduction from MLWE in the ROM
  (if we don’t compress the public key)
- Non-tight reduction in the QROM
- Tight reduction in the QROM with non-standard assumption

CRYSTALS-Kyber Security

(Credit Slide: P. Schwabe)

- Tight reduction from MLWE in the ROM
  (if we don't compress the public key)
- Non-tight reduction in the QROM
- Tight reduction in the QROM with non-standard assumption
- Failure probability of $< 2^{-140}$
- Interesting questions:
  - How much of a problem are a few failures?
  - How much can an attacker exploit Groven to produce failures?

CRYSTALS-Kyber Security

(Credit Slide: P. Schwabe)

- Tight reduction from MLWE in the ROM
  (if we don't compress the public key)
- Non-tight reduction in the QROM
- Tight reduction in the QROM with non-standard assumption
- Failure probability of $< 2^{-140}$
- Interesting questions:
  - How much of a problem are a few failures?
  - How much can an attacker exploit Groven to produce failures?
- Three different parameter set submitted:
  - **Kyber512**: 102 bit of post-quantum security
  - **Kyber768**: 161 bit of post-quantum security
  - **Kyber1024**: 218 bit of post-quantum security

CRYSTAL-Kyber Performances

Kyber512

Sizes (in bytes)		Haswell cycles (ref)		Haswell cycles (AVX2)	
sk:	1632	gen:	141872	gen:	55160
pk:	736	enc:	205468	enc:	75680
ct:	800	dec:	246040	dec:	74428

- Cycles counts on one core, without TurboBoost and HyperThreading
- Comparison: X25519 gen: 90668 cycles, enc/dec: 138963 cycles.
- However, only 32-bytes for X25519 pk and ct

CRYSTAL-Kyber Performances

Kyber768

Sizes (in bytes)		Haswell cycles (ref)		Haswell cycles (AVX2)	
sk:	2400	gen:	243004	gen:	85472
pk:	1088	enc:	332616	enc:	112600
ct:	1152	dec:	394424	dec:	108904

- Cycles counts on one core, without TurboBoost and HyperThreading
- Comparison: X25519 gen: 90668 cycles, enc/dec: 138963 cycles.
- However, only 32-bytes for X25519 pk and ct

CRYSTAL-Kyber Performances

Kyber1024

Sizes (in bytes)		Haswell cycles (ref)		Haswell cycles (AVX2)	
sk:	3168	gen:	368564	gen:	121056
pk:	1440	enc:	481042	enc:	157964
ct:	1504	dec:	558740	dec:	154952

- Cycles counts on one core, without TurboBoost and HyperThreading
- Comparison: X25519 gen: 90668 cycles, enc/dec: 138963 cycles.
- However, only 32-bytes for X25519 pk and ct

Error Distributions

- Binomial distribution
  - Sample $a_1, \dots, a_\eta, b_1, \dots, b_\eta \leftarrow \{0,1\}$ and output $\sum_i (a_i - b_i)$
  - Used by NewHope with $\eta = 8$, CRYSTALS-Kyber with parameter $\eta = 5, 4, 3$, LIMA with parameter $\eta = 20$,
- Approximation Gaussian sampling (FrodoKEM)

	σ	**Probability of (in multiples of 2^{-15})**												**Rényi**	
		0	±1	±2	±3	±4	±5	±6	±7	±8	±9	±10	±11	**order**	**divergence**
$\chi_{\text{Frodo-640}}$	2.75	9456	8857	7280	5249	3321	1844	898	384	144	47	13	3	500.0	0.72×10^{-4}
$\chi_{\text{Frodo-976}}$	2.3	11278	10277	7774	4882	2545	1101	396	118	29	6	1		500.0	0.14×10^{-4}

- Bounded Discrete Gaussian sampling (Ding Key Exchange, LOTUS)
- Often Rényi-divergence-based justification based on [BLRSSS'18]

Other Submissions

Almacrypt Workshop | Post-Quantum Crypto Lounge – SAF | Estimate all the {LWE, NTRU} schen

https://www.safecrypto.eu/pqclounge/

Home | About SAFEcrypto | More Information | Outcomes | News and Events | Post-Quantum Crypto Lounge

SAFEcrypto

Ramstake Zip file	Alan Szepieniec	Lattice	Standard	KEM	Round 1		CCA	
Odd Manhattan Zip file	Thomas Plantard	Lattice	Standard	Encryption	Round 1		CPA	Not CCA secure-*patched*
NTRU Prime Zip file	Daniel J. Bernstein /Chitchanok Chuengsatiansup /Tanja Lange /Christine van Vredendaal	Lattice	Ring	KEM	Round 1		CCA2	
Three Bears Zip file	Mike Hamburg	Lattice	Module	KEM	Round 1		CCA	
CRYSTALS-KYBER Zip file	Peter Schwabe /Roberto Avanzi /Joppe Bos /Leo Ducas /Eike Kiltz /Tancrede Lepoint /Vadim Lyubashevsky /John M. Schanck /Gregor Seiler /Damien Stehle	Lattice	Module	KEM	Round 1		CCA2	Concerns surrounding proof of IND-CPA security
LOTUS Zip file	Le Trieu Phong /Takuya Hayashi /Yoshinori Aono /Shiho Moriai	Lattice	Standard	KEM Encryption	Round 1		CCA2	CCA attack-*patched*
NTRUEncrypt Zip file	Zhenfei Zhang /Cong Chen /Jeffrey Hoffstein /William Whyte	Lattice	Ring	KEM Encryption	Round 1		CCA2	
pqNTRUsign Zip file	Zhenfei Zhang /Cong Chen /Jeffrey Hoffstein /William Whyte	Lattice	Ring Module	Signature	Round 1		EUF-CMA	Vulnerable to CMA attack - *patched*
SABER Zip file	Jan-Pieter D'Anvers /Angshuman Karmakar /Sujoy Sinha Roy /Frederik Vercauteren	Lattice	Module	KEM	Round 1		CCA	

Other Submissions

OddManhattan:

- Keygen

$$pk = \left(\frac{w_i}{w_1} \bmod p\right)_i ; sk = w_1$$

with $w_i \bmod 2 = 1$ and $\|w\|_1 \leq (p-1)/(2b)$

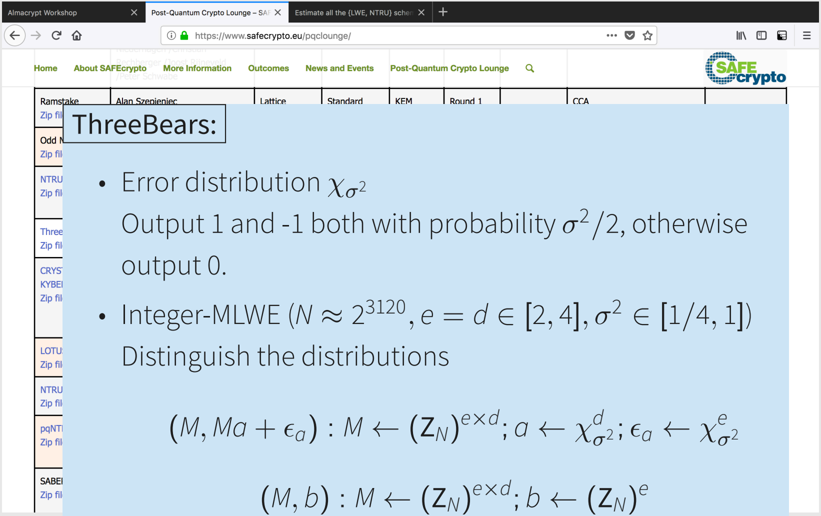
- Encrypt

$$ct = \langle u, h \rangle \bmod p$$

where

$$\sum_i u_i = m \bmod 2 \text{ and } \|u\|_\infty \leq b.$$

Other Submissions



ThreeBears:

- Error distribution χ_{σ^2}
  Output 1 and -1 both with probability $\sigma^2/2$, otherwise output 0.
- Integer-MLWE ($N \approx 2^{3120}, e = d \in [2,4], \sigma^2 \in [1/4, 1]$)
  Distinguish the distributions

$$(M, Ma + \epsilon_a) : M \leftarrow (\mathbb{Z}_N)^{e \times d}; a \leftarrow \chi_{\sigma^2}^d; \epsilon_a \leftarrow \chi_{\sigma^2}^e$$

$$(M, b) : M \leftarrow (\mathbb{Z}_N)^{e \times d}; b \leftarrow (\mathbb{Z}_N)^e$$

- “We expect the difficulty of this problem to be similar to the traditional problem over cyclotomic rings.”

Other Submissions

Home | About SAFEcrypto | More Information | Outcomes | News and Events | Post-Quantum Crypto Lounge

Compact LWE Zip file	Dongxi Liu /Nan Li Jongkil Kim /Surya Nepa	Lattice	Standard	Encryption	Round 1	ATTACKED	CCA2	Secret key can be recovered from ciphertext
Ding Key Exchange Zip file	Jintai Ding /Tsuyoshi Takagi /Xinwei Gao /Yuntao Wang	Lattice	Ring	KEM	Round 1		CPA	
KINDI Zip file	Rachid El Bansarkhani	Lattice	Ring	KEM Encryption	Round 1		CCA	
Lizard Zip file	Jung Hee Cheon /Sangjoon Park /Joohee Lee /Duhyeong Kim /Yongsoo Song /Seungwan Hong /Dongwoo Kim /Jinsu Kim /Seong-Min Hong /Aaram Yun /Jeongsu Kim Haeryong Park /Eunyoung Choi /Kimoon kim /Jun-Sub Kim /Jieun Lee	Lattice	Standard, Ring	KEM Encryption	Round 1		CCA2	
Round2 Zip file	Oscar Garcia-Morchon /Zhenfei Zhang /Sauvik Bhattacharya /Ronald Rietman /Ludo Tolhuizen /Jose-Luis Torre-Arce	Lattice	Standard, Ring	KEM Encryption	Round 1		CCA	Concerns surrounding proof of the IND-CPA security
LIMA Zip file	Nigel P. Smart /Martin R. Albrecht /Yehuda Lindell /Emmanuela Orsini /Valery Osheter /Kenny Paterson /Guy Peer	Lattice	Ring	KEM Encryption	Round 1		CCA	Concerns surrounding rejection sampling analysis - patch proposed

Other Submissions

Almacrypt Workshop | Post-Quantum Crypto Lounge – SAF | Estimate all the {LWE, NTRU} schem

https://www.safecrypto.eu/pqclounge/

Home About SAFEcrypto More Information Outcomes News and Events Post-Quantum Crypto Lounge

SAFEcrypto

EMBLEM and R.EMBLEM Zip file	Minhye Seo /Jong Hwan Park /Dong Hoon Lee /Suhri Kim /Seung-Joon Lee	Lattice	Standard, Ring	Encryption	Round 1		CPA	
NewHope Zip file	Thomas Poppelmann /Erdem Alkim /Roberto Avanzi /Joppe Bos /Leo Ducas /Antonio de la Piedra /Peter Schwabe /Douglas Stebila	Lattice	Ring	KEM	Round 1		CCA	
Titanium Zip file	Ron Steinfeld /Amin Sakzad /Raymond K. Zhao	Lattice	Poly	KEM Encryption	Round 1		CCA CPA	
HILA5 Zip file	Markku-Juhani O. Saarinen	Lattice	Ring	KEM	Round 1		CPA	
qTESLA Zip file	Nina Bindel /Sedat Akleylek /Erdem Alkim /Paulo S.L.M. Barreto /Johannes Buchmann /Edward Eaton /Gus Gutoski /Juliane Kramer/ Patrick Longa /Harun Polat / Jefferson E. Ricardini /Gustavo Zanon	Lattice	Ring	Signature	Round 1		EUF-CMA	
CRYSTALS-DILITHIUM Zip file	Vadim Lyubashevsky/ Leo Ducas / Eike Kiltz /Tancrede Lepoint/ Peter Schwabe /Gregor Seiler /Damien Stehle	Lattice	Module	Signature	Round 1		SUF-CMA	
KCL (OKCN/AKCN /CNKE)	Yunlei Zhao /Zhengzhong jin /Boru Gong /Guangye Sui	Lattice	Standard, Ring	KEM Encryption	Round 1		CCA	

Other Submissions

Titanium:

- Middle Product (MP)

$$(a \odot_n s) = \lfloor (a \cdot s \bmod x^{2n-1} / (x^n - 1)) \rfloor \in \mathbb{Z}_q^{<n}[x]$$

- MP-LWE: LWE with middle product
- Reduction from (decision) $PLWE^f$ to (decision) MP-LWE of parameter n, for every monic f of degree n whose constant coefficient is coprime with q

Other Submissions

https://www.safecrypto.eu/pqclounge/

Home | About SAFEcrypto | More Information | Outcomes | News and Events | Post-Quantum Crypto Lounge

LAC Zip file	Xianhui Lu /Yamin Liu /Dingding Jia /Haiyang Xue /Jingnan He /Zhenfei Zhang	Lattice	Poly	KEM Encryption	Round 1		CCA	
DRS Zip file	Thomas Plantard/ Arnaud Sipasseuth/ Cedric Dumondelle/ Willy Susilo	Lattice	Standard	Signature	Round 1		EUF-CMA	
FrodoKEM Zip file	Michael Naehrig /Erdem Alkim /Joppe Bos /Leo Ducas /Karen Easterbrook /Brian LaMacchia /Patrick Longa /Ilya Mironov /Valeria Nikolaenko /Christopher Peikert /Ananth Raghunathan /Douglas Stebila	Lattice	Standard	KEM	Round 1		CCA	
Giophantus Zip file	Koichiro Akiyama /Yasuhiro Goto /Shinya Okumura /Tsuyoshi Takagi /Koji Nuida /Goichiro Hanaoka /Hideo Shimizu /Yasuhiko Ikematsu	Lattice	Standard	Encryption	Round 1	ATTACKED	CPA	Distinguishing attack that breaks the claimed IND-CPA security, can be avoided by switching the base ring
NTRU-HRSS-KEM Zip file	John M. Schanck /Andreas Hulsing /Joost Rijneveld /Peter Schwabe	Lattice	Ring	KEM	Round 1		CCA2	
FALCON Zip file	Thomas Prest / Pierre-Alain Fouque /Jeffrey Hoffstein /Paul	Lattice	Ring	Signature	Round 1		EUF-CMA	

Other Submissions

Almacrypt Workshop | Post-Quantum Crypto Lounge – SAF | Estimate all the {LWE, NTRU} schem

https://www.safecrypto.eu/pqclounge/

Home About SAFEcrypto More Information Outcomes News and Events Post-Quantum Crypto Lounge

SAFEcrypto

FALCON Zip file	Thomas Prest / Pierre-Alain Fouque /Jeffrey Hoffstein /Paul Kirchner /Vadim Lyubashevsky /Thomas Pornin /Thomas Ricosset /Gregor Seiler /William Whyte /Zhenfei Zhang	Lattice	Ring	Signature	Round 1		EUF-CMA	
Mersenne-756839 Zip file	Divesh Aggarwal / Antoine Joux / Anupam Prakash / Mikos Santha	Lattice Lattices/Other	Ring	KEM	Round 1		CCA	
Lepton Zip file	Yu Yu /Jiang Zhang	LPN (Lattice/Code)		KEM	Round 1		CCA	
CFPKM Zip file	O. Chakraborty /J. C-Faugère /L Perret /	Multivariate Quadratic		KEM	Round 1	ATTACKED	CPA	known attack - breaks IND-CPA security for CFPKM128, CFPKM182 parameter sets. Attack on shared secret: shared

Some Implementation Considerations

Increasing Security in CRYSTALS-Kyber

```c
#ifndef KYBER_K
#define KYBER_K 3 /* Change this for different security strengths */
#endif

/* Don't change parameters below this line */

#define KYBER_N 256
#define KYBER_Q 7681

#if   (KYBER_K == 2) /* Kyber512 */
#define KYBER_ETA 5
#elif (KYBER_K == 3) /* Kyber768 */
#define KYBER_ETA 4
#elif (KYBER_K == 4) /*KYBER1024 */
#define KYBER_ETA 3
#else
#error "KYBER_K must be in {2,3,4}"
#endif
```

```c
/*************************************************
* Name:        polyvec_ntt
*
* Description: Apply forward NTT to all elements of a vector of polynomials
*
* Arguments:   - polyvec *r: pointer to in/output vector of polynomials
**************************************************/
void polyvec_ntt(polyvec *r)
{
  int i;
  for(i=0;i<KYBER_K;i++)
    poly_ntt(&r->vec[i]);
}
```

Side-Channel Countermeasures

- Most implementations submitted to NIST are **constant-time**
- Need more research against **fault attacks** and **DCA**-like attacks

Loop-Abort Faults on Lattice-Based Fiat–Shamir and Hash-and-Sign Signatures

Thomas Espitau[4], Pierre-Alain Fouque[2],
Benoît Gérard[1], and Mehdi Tibouchi[3]

Side-Channel Attacks on BLISS Lattice-Based Signatures

Exploiting Branch Tracing Against strongSwan and Electromagnetic Emanations in Microcontrollers

Thomas Espitau
UPMC
France
thomas.espitau@lip6.fr

Pierre-Alain Fouque
Université de Rennes I
France
pierre-alain.fouque@univ-rennes1.fr

Benoît Gérard
DGA.MI
France
benoit.gerard@irisa.fr

Mehdi Tibouchi
NTT Corporation
Japan
tibouchi.mehdi@lab.ntt.co.jp

NIST API for KEM:

```
int crypto_kem_dec(unsigned char *ss, const
unsigned char *ct, const unsigned char *sk)
```

Question

What happens to `ss` is decryption fails?

Cleaning ss

- NTRUPrime

```c
#endif

  hide(checkcstr,maybek,sk,r);
  result = verify(cstr,checkcstr);

  for (i = 0;i < 32;++i) k[i] = maybek[i] & ~result;
  return result;
```

Cleaning ss

- NTRUPrime

```
#endif

  hide(checkcstr,maybek,sk,r);
  result = verify(cstr,checkcstr);

  for (i = 0;i < 32;++i) k[i] = maybek[i] & ~result;
  return result;
```

- CRYSTALS-Kyber

```
  fail = verify(ct, cmp, KYBER_CIPHERTEXTBYTES);

  sha3_256(kr+KYBER_SYMBYTES, ct, KYBER_CIPHERTEXTBYTES);                     /* overwrite coins in kr with H(c)  */

  cmov(kr, sk+KYBER_SECRETKEYBYTES-KYBER_SYMBYTES, KYBER_SYMBYTES, fail);     /* Overwrite pre-k with z on
  re-encryption failure */
```

Cleaning ss

- NTRUPrime

```
#endif

  hide(checkcstr,maybek,sk,r);
  result = verify(cstr,checkcstr);

  for (i = 0;i < 32;++i) k[i] = maybek[i] & ~result;
  return result;
```

- CRYSTALS-Kyber

```
  fail = verify(ct, cmp, KYBER_CIPHERTEXTBYTES);

  sha3_256(kr+KYBER_SYMBYTES, ct, KYBER_CIPHERTEXTBYTES);              /* overwrite coins in kr with H(c)  */

  cmov(kr, sk+KYBER_SECRETKEYBYTES-KYBER_SYMBYTES, KYBER_SYMBYTES, fail);     /* Overwrite pre-k with z on
  re-encryption failure */
```

- OddManhattan and LOTUS were not cleaning the buffer
    - We essentially get a decryption oracle…
    - Secret key can be recovered in polynomial time!
    - Demo.

Conclusion

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

Encryption and KEMs:

- Combine with pre-quantum crypto, e.g., X25519

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

Encryption and KEMs:

- Combine with pre-quantum crypto, e.g., X25519
- CRYSTALS-Kyber768 is a great candidate: stable theory, great performance and security

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

Encryption and KEMs:

- Combine with pre-quantum crypto, e.g., X25519
- CRYSTALS-Kyber768 is a great candidate: stable theory, great performance and security
- If you don't care about public-key size, you can also use McEliece

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

Encryption and KEMs:

- Combine with pre-quantum crypto, e.g., X25519
- CRYSTALS-Kyber768 is a great candidate: stable theory, great performance and security
- If you don’t care about public-key size, you can also use McEliece

Signatures:

- If you (really) know what you are doing and can handle a state, use forward-secure stateful hash-based signatures

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

Encryption and KEMs:

- Combine with pre-quantum crypto, e.g., X25519
- CRYSTALS-Kyber768 is a great candidate: stable theory, great performance and security
- If you don't care about public-key size, you can also use McEliece

Signatures:

- If you (really) know what you are doing and can handle a state, use forward-secure stateful hash-based signatures
- If you can be slow, large, and can handle complex implementation, you can use stateless hash-based signatures

My Personal View on Post-Quantum Crypto

- Worst-case to average-case reduc.: *soundness* of constructions
- More work needed on security estimations

Encryption and KEMs:

- Combine with pre-quantum crypto, e.g., X25519
- CRYSTALS-Kyber768 is a great candidate: stable theory, great performance and security
- If you don't care about public-key size, you can also use McEliece

Signatures:

- If you (really) know what you are doing and can handle a state, use forward-secure stateful hash-based signatures
- If you can be slow, large, and can handle complex implementation, you can use stateless hash-based signatures
- Otherwise, combine Ed25519 with CRYSTALS-Dilithium

Some avenues of work

- Module-LWE have only been used in public key encryption and signatures so far. It could be interesting to look at **new applications** (e.g., attribute-based encryption, fully homomorphic encryption)?
- Failures and quantum adversaries?
- Quantum speed-ups for enumeration or sieving
- Are there ideas proposed in code-based submissions or multivariate-based submissions we can use?
- Attack Ring-LWE challenges! `http://web.eecs.umich.edu/~cpeikert/rlwe-challenges/`

Implementations

- Module-LWE makes possible to create highly optimized **SW or HW multiplier** that works for many security levels

- Need work on impact of side-channel countermeasures (fault, masking, etc.)

- Systematic implementation of the Fujisaki–Okamoto transform

Help Verify Parameter Estimator

Challenges for Learning With E… | Almacrypt Workshop | Estimate all the {LWE, NTRU} schem…

https://estimate-all-the-lwe-ntru-schemes.github.io/docs/

Estimate all the {LWE, NTRU} schemes!

Complexity estimates for running the primal-uSVP and dual attacks against all LWE-based, and the primal-uSVP attack against all NTRU-based, Round 1 schemes proposed as part of the PQC process run by NIST. We make use of the [APS15] estimator. The code for generating this table is available on Github, as well as the paper. Clicking on a particular estimate cell in the table will provide with stand-alone Sagemath code for reproducing the estimate.

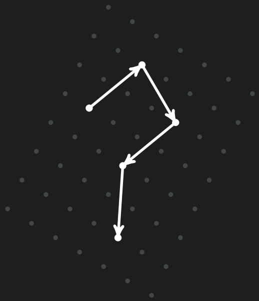
Below, we provide LWE-equivalent parameters, where n = LWE secret dimension, k = MLWE rank (if any), q = modulo, σ = standard deviation of the error, $\mathbb{Z}_q/(\phi)$ is the ring (if any). For NTRU schemes we provide ‖f‖, ‖g‖ = lengths of the short polynomials. If you spot a mistake in a parameter set or cost model, please feel free to open a ticket or to make a pull-request.

LWE n samples | LWE 2n samples | NTRU | 14 selected | Search:

Scheme	Assumption	Primitive	Parameters	Claimed security	NIST Category	Attack	Q-Core-Sieve	Q-Core-Sieve + O(1)	Q-Core-Sieve (min space)	Q-β-Sieve	Q-8c + O(
AKCN-MLWE	MLWE	KEM	n = 768, k = 3,	147	4	primal	148.67	164.67	166.90	157.80	178.
AKCN-MLWE	MLWE	KEM	n = 768, k = 3,	147	4	dual	179.42	194.28	197.26	189.34	205.
AKCN-MLWE	MLWE	KEM	n = 768, k = 3,	183	4	primal	184.44	200.44	207.06	193.88	214.
AKCN-MLWE	MLWE	KEM	n = 768, k = 3,	183	4	dual	226.31	242.31	249.31	236.05	249.
AKCN-RLWE	RLWE	KEM	n = 1024, q =	255	5	primal	257.31	273.31	288.87	267.24	287.
AKCN-RLWE	RLWE	KEM	n = 1024, q =	255	5	dual	315.62	320.81	338.79	318.59	334.
BabyBear	ILWE	KEM	n = 624, k = 2,	152	2	primal	152.91	168.91	171.66	162.08	182.
BabyBear	ILWE	KEM	n = 624, k = 2	152	2	dual	192.40	205.02	210.36	201.89	217.

Martin R. Albrecht, Benjamin R. Curtis, Amit Deo, Alex Davidson, Rachel Player, Eamonn Postlethwaite, Fernando Virdia, Thomas Wunderer.

Thank you. Any questions?



https://tlepoint.github.io

Interesting Links

- NIST Post-Quantum Cryptography
  `https://nist.gov/pqcrypto`
- Post-Quantum Cryptography Lounge
  `https://www.safecrypto.eu/pqclounge/`
- libpqcrypto
  `https://libpqcrypto.org`
- Open Quantum Safe
  `https://openquantumsafe.org`
- Estimate all the {LWE, NTRU} schemes!
  `https://estimate-all-the-lwe-ntru-schemes.github.io/`
- CRYSTALS website
  `https://pq-crystals.org`