CRYSTALS

Cryptographic Suite for Algebraic Lattices

Shi Bai
Eike Kiltz
John M. Schanck
Joppe Bos
Tancrède Lepoint
Peter Schwabe
Léo Ducas
Vadim Lyubashevsky
Damien Stehlé

SRI International®

Jan 4, 2017 - Real World Crypto

Outline

Outline

Previous talk: NIST

http://nist.gov/pqcrypto

This talk is about LATTICE-BASED CRYPTOGRAPHY

Lattice crypto in strongSwan

OpenSource IPsec-based VPN Solution

- Early adopter of lattice-based crypto:
  - NTRUEncrypt[1] since Feb 2014
  - BLISS signature[2] since Jan 2015
  - NewHope[3] key exchange since Oct 2016

---

[1] John Hoffstein, Jill Pipher, and Joseph E. Silverman. "NTRU: A New High Speed Public Key Cryptosystem". In: *ANTS III*. vol. 1423. LNCS. Springer, 1998.

[2] Léo Ducas et al. "Lattice Signatures and Bimodal Gaussians". In: *CRYPTO (1)*. Vol. 8042. LNCS. Springer, 2013.

[3] Erdem Alkim et al. "Post-quantum Key Exchange - A New Hope". In: *USENIX Security Symposium*. USENIX Association, 2016.

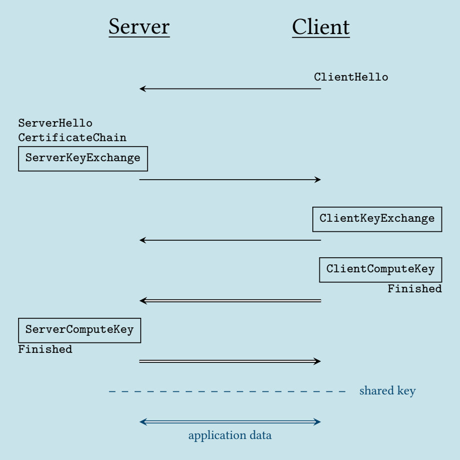
Google's experimentation with PQCrypto

Impact assessment

- Combination of NewHope with ECDH (X25519) in TLS.
- Result: "**we did not find any unexpected impediment to deploying something like NewHope**"[4]

[4] `https://www.imperialviolet.org/2016/11/28/cecpq1.html`

Primary focus: KEM



`ServerKeyExchange` = KEM.Setup()	Key generation Send public key pk
`ClientKeyExchange` = KEM.Encaps()	Sample random value Encrypt value using pk Send ciphertext c
`ClientComputeKey`	key = KDF(value)
`ServerComputeKey` = KEM.Decaps()	Decrypt c to recover value key = KDF(value)

The question is what post-quantum encryption scheme to use?

Current lattice-based key exchanges (learn more next talk)

	Reconciliation[5]	Encryption
LWE-based	Frodo[6] $\lvert\mathrm{comm}\rvert = 22.6$KiB	$\lvert\mathrm{comm}\rvert > 22.6$ KiB
RLWE-based	BCNS15[7] $\lvert\mathrm{comm}\rvert = 8.2$KiB NewHope[8] $\lvert\mathrm{comm}\rvert = 3.9$KiB	NewHope-Simple[9] $\lvert\mathrm{comm}\rvert = 4$KiB

[5] More complicated to implement (randomized doubling, lattice-quantizers, etc.) - cf. Jintai Ding. "A Simple Provably Secure Key Exchange Scheme Based on the Learning with Errors Problem". In: *IACR Cryptology ePrint Archive* 2012/688 (2012) and Chris Peikert. "Lattice Cryptography for the Internet". In: *PQCrypto*. Vol. 8772. LNCS. Springer, 2014

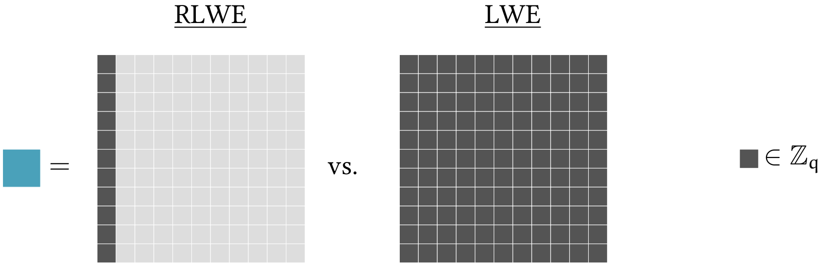
[6] Joppe W. Bos et al. "Frodo: Take off the Ring! Practical, Quantum-Secure Key Exchange from LWE". . In: *ACM Conference on Computer and Communications Security*. ACM, 2016.

[7] Joppe W. Bos et al. "Post-Quantum Key Exchange for the TLS Protocol from the Ring Learning with Errors Problem". In: *IEEE Symposium on Security and Privacy*. IEEE Computer Society, 2015, pp. 553–570.

[8] Erdem Alkim et al. "Post-quantum Key Exchange - A New Hope". In: *USENIX Security Symposium*. USENIX Association, 2016.

[9] Erdem Alkim et al. "NewHope without reconciliation". In: *IACR Cryptology ePrint Archive* 2016/1157 (2016).

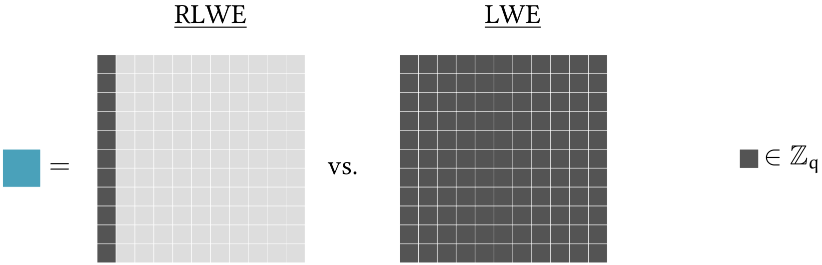
Why do people use a ring?



[10] John Hoffstein, Jill Pipher, and Joseph E. Silverman. “NTRU: A New High Speed Public Key Cryptosystem”. In: (1996). Preliminary Draft.

[11] Daniel J. Bernstein et al. “NTRU Prime”. In: *IACR Cryptology ePrint Archive* 2016/461 (2016).

Why do people use a ring?



- usual ring $\mathbb{Z}_q[x]/(x^n+1)$
- other possibilities[10][11] $x^n - 1$ or $x^p - x - 1$

[10] John Hoffstein, Jill Pipher, and Joseph E. Silverman. “NTRU: A New High Speed Public Key Cryptosystem”. In: (1996). Preliminary Draft.

[11] Daniel J. Bernstein et al. “NTRU Prime”. In: *IACR Cryptology ePrint Archive* 2016/461 (2016).

Crystals: our cryptographic suite

CRYSTALS

CRYPTOGRAPHIC SUITE FOR ALGEBRAIC LATTICES

Simplicity:

- no reconciliation
- no Gaussian sampling
- CCA-secure KEM
- no NTRU assumption

← **Module lattices**[12] →

Modularity:

- easy to increase security
- KEM can be used for encryption (KEM-DEM), key exchange, AKE

[12] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptography* 75.3 (2015).

Kyber and Dilithium

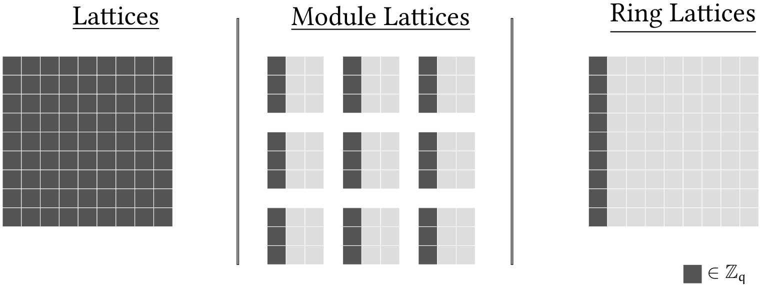
- Module lattices : d-dimensional matrices of elements in $\mathbb{Z}_q[x]/(x^{256}+1)$
  - 256 is the number of bits we want to encrypt
  - Allow to reach dimensions $256 \cdot d$'s
  - Increase d to increase security
- Kyber [13] the KEM
  - CCA security
  - Encryption-based KEM
- Dilithium the digital signature *(Not today)*
  - No Gaussian distribution (à la GLP12[14])

[13] Thanks !

[14] Tim Güneysu, Vadim Lyubashevsky, and Thomas Pöppelmann. "Practical Lattice-Based Cryptography: A Signature Scheme for Embedded Systems". In: *CHES*. vol. 7428. LNCS. Springer, 2012.

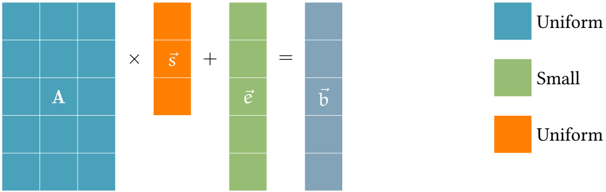
Outline

Module lattices



- Module lattices are "more general" than Ring lattices (finitely generated modules over the ring of integers of a number field), and less structured
- Example: d-dimensional matrices of polynomials in $\mathbb{Z}_q[x]/(x^{256}+1)$
  - allows to reach all dimensions $256 \cdot d$
  - allows to reduce modulus q w.r.t. to ring lattices for same security
  - more flexible

Module learning with errors[15][16][17][18] over $R = \mathbb{Z}_q[x]/(x^n + 1)$



[15] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *STOC*. ACM, 2005.

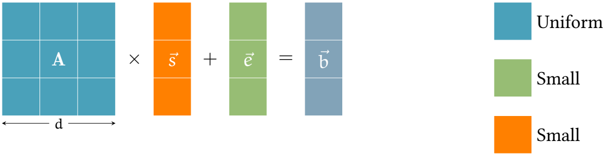
[16] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO*. vol. 5677. LNCS. Springer, 2009.

[17] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT*. vol. 6110. LNCS. Springer, 2010.

[18] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices".

Module learning with errors[15][16][17][18] over $R = \mathbb{Z}_q[x]/(x^n + 1)$

with small secret and square matrices



[15] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *STOC*. ACM, 2005.

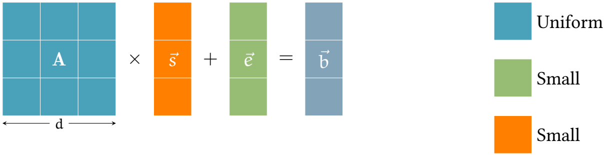
[16] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO*. vol. 5677. LNCS. Springer, 2009.

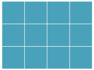
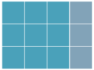
[17] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT*. vol. 6110. LNCS. Springer, 2010.

[18] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices".

Module learning with errors[15][16][17][18] over $R = \mathbb{Z}_q[x]/(x^n+1)$

with small secret and square matrices



<u>Decision MLWE:</u> Distinguish  and 

---

[15] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *STOC*. ACM, 2005.

[16] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO*. vol. 5677. LNCS. Springer, 2009.

[17] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT*. vol. 6110. LNCS. Springer, 2010.

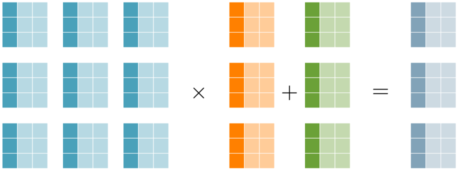
[18] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices".

Why Module-LWE is not less efficient than Ring-LWE?

- The matrix $\mathbf{A} = (a_{ij})_{1 \leqslant i,j \leqslant 3} \in (\mathbb{Z}_q[x]/(x^{256}+1))^{3\times 3}$ can be represented as one seed
  - expanded 3 times more bits, but no need to store it even during computation

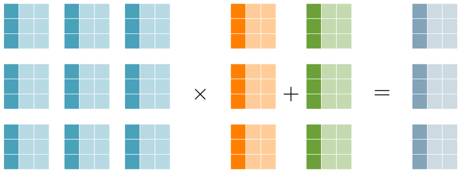
Why Module-LWE is not less efficient than Ring-LWE?

- The matrix $\mathbf{A} = (a_{ij})_{1 \leqslant i,j \leqslant 3} \in (\mathbb{Z}_q[x]/(x^{256}+1))^{3 \times 3}$ can be represented as one seed
  - expanded 3 times more bits, but no need to store it even during computation
- **Key point:**



Why Module-LWE is not less efficient than Ring-LWE?

- The matrix $\mathbf{A} = (a_{ij})_{1 \leqslant i,j \leqslant 3} \in (\mathbb{Z}_q[x]/(x^{256}+1))^{3\times 3}$ can be represented as one seed
  - expanded 3 times more bits, but no need to store it even during computation
- **Key point:**



  - $d \times d$ multiplications of polynomials
  - resulting element has **same size as RLWE element** of dimension $256 \cdot d$
  - In general, Module-LWE is less efficient than Ring-LWE… but not if we need to only encrypt 256 bits

Easiness of implementation

1. Efficient multiplications using a single NTT in dim. 256

```
void polyvec_ntt(polyvec *r)
{
  int i;
  for(i=0; i<KYBER_D; i++) {
    poly_ntt(&r->vec[i]);
  }
}
```

Easiness of implementation

1. Efficient multiplications using a single NTT in dim. 256

```
void polyvec_ntt(polyvec *r)
{
  int i;
  for(i=0; i<KYBER_D; i++) {
    poly_ntt(&r->vec[i]);
  }
}
```

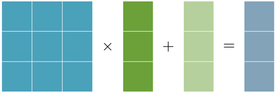
2. Easy to increase security *with very little reimplementation*: increase d (and reduce noise), e.g. by setting $\mathrm{KYBER_D} = 4$ instead of $\mathrm{KYBER_D} = 3$

$\mathrm{KYBER_D}$	2	3	4
Security level	98	161	227

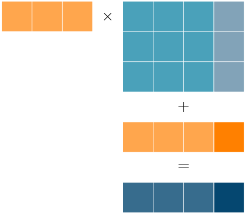
KYBER

THE KEM

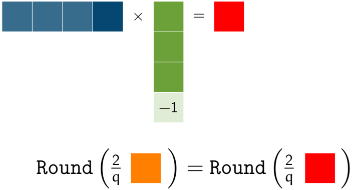
KEM from an MLWE (over R) encryption scheme[19][20][21][22]



Public key / Secret key Generation



Encapsulation



Decapsulation

[19] Oded Regev. "On lattices, learning with errors, random linear codes, and cryptography". In: *STOC*. ACM, 2005.

[20] Benny Applebaum et al. "Fast Cryptographic Primitives and Circular-Secure Encryption Based on Hard Learning Problems". In: *CRYPTO*. vol. 5677. LNCS. Springer, 2009.

[21] Vadim Lyubashevsky, Chris Peikert, and Oded Regev. "On Ideal Lattices and Learning with Errors over Rings". In: *EUROCRYPT*. vol. 6110. LNCS. Springer, 2010.

[22] Adeline Langlois and Damien Stehlé. "Worst-case to average-case reductions for module lattices". In: *Des. Codes Cryptography* 75.3 (2015).

Kyber's encryption scheme

$q = 7681, n = 256, d = 3$

We work with matrices of polynomials in $\mathbb{Z}_{7681}[x]/(x^{256}+1)$ of dim. $d = 3$ and a distribution of poly with binomial coeffs. Ψ_4

KeyGen():

- seed $\leftarrow \{0, \ldots, 255\}^{32}$
- $\mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \leftarrow \mathsf{SHAKE}(\text{seed})$
- $\vec{s}, \vec{e} \leftarrow \Psi_4^d$
- $\vec{b} = \mathbf{A} \cdot \vec{s} + \vec{e}$
- Define pk $= (\text{seed}, \vec{b})$ and sk $= \vec{s}$

Kyber's encryption scheme

$q = 7681, n = 256, d = 3$

We work with matrices of polynomials in $\mathbb{Z}_{7681}[x]/(x^{256}+1)$ of dim. $d = 3$ and a distribution of poly with binomial coeffs. Ψ_4

Encrypt$(\mathsf{pk}, m \in \{0,1\}^{256}, \mathsf{coins})$:

- $\mathsf{seed}, \vec{b} \leftarrow \mathsf{pk}$
- $\mathbf{A} = \mathsf{SHAKE}(\mathsf{seed})$
- $\vec{s}' \leftarrow \Psi_4^d(\mathsf{coins}, 1)$
- $\vec{e}' \leftarrow \Psi_4^d(\mathsf{coins}, 2)$
- $e'' \leftarrow \Psi_4(\mathsf{coins}, 3)$
- $\vec{u} = (\vec{s}')^t \cdot \mathbf{A} + \vec{e}'$
- $v = \langle \vec{b}, \vec{s}' \rangle + e'' + \lfloor q/2 \rfloor \cdot \sum_i m_i x^i$
- Output $(\vec{u}, v)$

Decrypt$(\mathsf{sk}, (\vec{u}, v))$:

- $w = v - \langle \vec{u}, \vec{s} \rangle$
- for $i \in \{0, \ldots, 255\}$, $m_i \leftarrow \begin{cases} 1 & \text{if } w_i \in (\frac{q}{4}, \frac{3\cdot q}{4}) \\ 0 & \text{otherwise} \end{cases}$
- Output $(m_0, \ldots, m_{255})$

CRYSTALS-KYBER: the KEM

- $q = 7681$ and $n = 256$: poly in $\mathbb{Z}_{7681}[x]/(x^{256}+1)$
- Matrices of dim. $d = 3$, distribution of poly with binomial coeffs. Ψ_4

Alice (Server)		Bob (Client)
$\underline{\mathsf{Gen}()}$:		$\underline{\mathsf{Encaps}(\mathsf{seed}, \vec{b})}$:
$\mathsf{pk}, \mathsf{sk} \leftarrow \mathsf{KeyGen}()$		$x \leftarrow \{0, \ldots, 255\}^{32}$
$\mathsf{seed}, \vec{b} \leftarrow \mathsf{pk}$	$\xrightarrow{\mathsf{seed}, \vec{b}}$	$x \leftarrow \mathsf{SHA3\text{-}256}(x)$ $k, \mathsf{coins} \leftarrow \mathsf{SHA3\text{-}512}(x)$
	$\xleftarrow{\vec{u}, v}$	$\vec{u}, v \leftarrow \mathsf{Encrypt}((\mathsf{seed}, \vec{b}), x, \mathsf{coins})$ $c = v + x \cdot \lfloor q/2 \rfloor$
$\underline{\mathsf{Decaps}(\vec{s}, (\vec{u}, v))}$: $x' \leftarrow \mathsf{Decrypt}(\vec{s}, (\vec{u}, v))$ $k', coins' \leftarrow \mathsf{SHA3\text{-}512}(x')$ $\vec{u}', v' \leftarrow \mathsf{Encrypt}((\mathsf{seed}, \vec{b}), x', \mathsf{coins}')$ **verify if** $(\vec{u}', v') = (\vec{u}, v)$		

Implementation aspects

- NTT in dimension 256 (Barrett & Montgomery)
- Primitives used: SHAKE128 as XOF, SHA3-256 and SHA3-512
- Binomial error distribution (smaller than in NewHope, same code)
- Compression: rounding c, but also $\vec{u}$
  - during decryption, we compute $\langle \vec{u}, \vec{s} \rangle$: we can round the coefficients of $\vec{u}$ ($\approx$ 1500 bits of saving)
- Similar to NewHope and NewHope-Simple (therefore easy to integrate), but *much* more general because of CCA security
  - can be used like NewHope (+ no problem of key reuse)
  - can be used in KEM-DEM
  - or in AKE

Can I see the code?

Soon (i.e., this month).

We still have a couple of things to figure out with respect to the QROM, and we didn't want to rush and change the code next week. We might revisit the CCA transformation and are expecting very similar performance to current version.

Will be on GitHub, public domain under the CC0 deed.

https://github.com/pq-crystals/kyber

Outline

1. Motivation
2. Module Lattices
3. The KEM
4. **Open Quantum Safe & Performances**
5. Conclusion

Open Quantum Safe

https://openquantumsafe.org

Open-source C library: common interface, prototype integration into application level protocols

Project leaders: Michele Mosca (U. of Waterloo) and Douglas Stebila (McMaster U.)

./openssl speed

AWS c4.large (Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz)

Scheme	**Alice 0**	**Bob** (ms)	**Alice 1**	**Communication** (bytes) $A \rightarrow B$	$B \rightarrow A$	**Security** (bits) Class.	PQ.
SIDH	15.836	35.144	14.967	564	564	192	128
McBits	69.918	0.039	0.147	311,736	109	157	157
BCNS15 (RLWE)	0.721	1.170	0.160	4,096	4,224	86	78
NewHope (RLWE)	0.052	0.079	0.018	1,824	2,048	281	255
NewHope-Simple				1,824	2,176		
Frodo (LWE)	0.905	1.327	0.162	11,377	11,296	144	130
Kyber (MLWE)							

./openssl speed

AWS c4.large (Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz)

Scheme	Alice 0	Bob (ms)	Alice 1	Communication (bytes) $A \to B$	Communication (bytes) $B \to A$	Security (bits) Class.	Security (bits) PQ.
SIDH	15.836	35.144	14.967	564	564	192	128
McBits	69.918	0.039	0.147	311,736	109	157	157
BCNS15 (RLWE)	0.721	1.170	0.160	4,096	4,224	86	78
NewHope (RLWE)	0.052	0.079	0.018	1,824	2,048	281	255
NewHope-Simple				1,824	2,176		
Frodo (LWE)	0.905	1.327	0.162	11,377	11,296	144	130
Kyber (MLWE)	**0.061**	**0.075**	**0.088**	**1,088**	**1,152**	178	161

./openssl speed

AWS c4.large (Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz)

Scheme	Alice 0	Bob (ms)	Alice 1	Communication (bytes) $A \to B$	$B \to A$	Security (bits) Class.	PQ.
SIDH	15.836	35.144	14.967	564	564	192	128
McBits	69.918	0.039	0.147	311,736	109	157	157
BCNS15 (RLWE)	0.721	1.170	0.160	4,096	4,224	86	78
NewHope (RLWE)	0.052	0.079	0.018	1,824	2,048	281	255
NewHope-Simple				1,824	2,176		
Frodo (LWE)	0.905	1.327	0.162	11,377	11,296	144	130
Kyber (MLWE)	**0.061**	**0.075**	**0.088**	**1,088**	**1,152**	178	161

- Security estimates: known classical and known quantum attacks that correspond to the core SVP hardness, that is the cost of *one call to an SVP oracle in dimension* b, (*pessimistic* estimation from defender's point of view)

`./openssl speed`

AWS `c4.large` (Intel(R) Xeon(R) CPU E5-2666 v3 @ 2.90GHz)

Scheme	Alice 0	Bob (ms)	Alice 1	Communication (bytes) $A \to B$	Communication (bytes) $B \to A$	Security (bits) Class.	Security (bits) PQ.
SIDH	15.836	35.144	14.967	564	564	192	128
McBits	69.918	0.039	0.147	311,736	109	157	157
BCNS15 (RLWE)	0.721	1.170	0.160	4,096	4,224	86	78
NewHope (RLWE)	0.052	0.079	0.018	1,824	2,048	281	255
NewHope-Simple				1,824	2,176		
Frodo (LWE)	0.905	1.327	0.162	11,377	11,296	144	130
Kyber (MLWE)	**0.061**	**0.075**	**0.088**	**1,088**	**1,152**	178	161

- Security estimates: known classical and known quantum attacks that correspond to the core SVP hardness, that is the cost of *one call to an SVP oracle in dimension* b, (*pessimistic* estimation from defender's point of view)

Available soon as PRs on `https://github.com/open-quantum-safe/`

Outline

1. Motivation
2. Module Lattices
3. The KEM
4. Open Quantum Safe & Performances
5. **Conclusion**

Conclusion

https://pq-crystals.org

- **Module lattices**: modularity and easiness of implementating different security params
- **Kyber**: KEM with almost halving of message sizes compared to NewHope(-Simple)
  - CCA security by default allowing Kyber to be used in AKE constructions, in KEM-DEM constructions, and making it safe to use long-term (or cached) keys
- **Dilithium** (soon): we also base the signature on module lattices (larger matrices, larger modulus) for **simplicity** and **modularity**

Internships

Side-channel protection aspects of post-quantum cryptography
Anytime 2017, 12 weeks — Belgium — *Joppe Bos*

Post-quantum Internet-of-Things
Anytime 2017, ≈ 12 weeks — NY or CA — *Tancrède Lepoint*

Post-quantum signatures for V2V communication and secure post-quantum implementations
Summer 2017, ≈ 12 weeks — MA — `wwhyte@securityinnovation.com`