Lepton: LPN-based KEMs with Post-Quantum Security

Yu Yu and Jiang Zhang

April 11, 2018

1st PQC Standardization conference

Outline

- Post-quantum Cryptography
- Learning Parity with Noise (over rings)
- Ring-LPN-based Key Encapsulation Mechanism
- Parameters and Performance

Post-Quantum Cryptography (PQC)

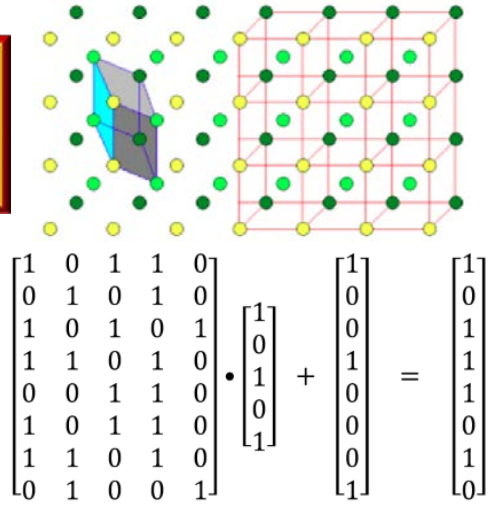
THE LIMITS OF Quantum

∃ problems not succumb to quantum power

- **(worst-case): NP-Complete, lattice problems, etc.**
- **(average-case): decoding problems, hard learning problems, etc.**

- Lattice-based
- Code-based
- Hash-based
- Multivariate

Learning with Errors (LWE)
Learning Parity with Noise (LPN)



Quantum hard problem → quantum resistant crypto-systems

LWE (Learning with Errors)
LPN (Learning Parity with Noise)

- $\boldsymbol{A} \xleftarrow{\$} \mathrm{Z}_q^{m\times n}, \boldsymbol{x} \xleftarrow{\$} \mathrm{Z}_q^{n}, \boldsymbol{e} \leftarrow \chi_q^{\mathrm{m}}$, compute $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$

$$\underset{\boldsymbol{A}}{\begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} & a_{15} \\ a_{21} & a_{22} & a_{23} & a_{24} & a_{25} \\ a_{31} & a_{32} & a_{33} & a_{34} & a_{35} \\ a_{41} & a_{42} & a_{43} & a_{44} & a_{45} \\ a_{51} & a_{52} & a_{53} & a_{54} & a_{55} \\ a_{61} & a_{62} & a_{63} & a_{64} & a_{65} \\ a_{71} & a_{72} & a_{73} & a_{74} & a_{75} \\ a_{81} & a_{82} & a_{83} & a_{84} & a_{85} \end{bmatrix}} \cdot \underset{\boldsymbol{x}}{\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{bmatrix}} + \underset{\boldsymbol{e}}{\begin{bmatrix} e_1 \\ e_2 \\ e_3 \\ e_4 \\ e_5 \\ e_6 \\ e_7 \\ e_8 \end{bmatrix}} = \underset{\mathrm{y}}{\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix}} \pmod q$$

LPN: q=2, $\chi_q^{u,\mathrm{m}}$~ Bernoulli
Noise rate: u

LWE: q≥$poly(n)$,
χ_q^{m}~ discrete Gaussian

- Search LPN/LWE: find out $\boldsymbol{x}$ given $\boldsymbol{A}, \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e}$
- Decisional LPN/LWE: distinguish $(\boldsymbol{A}, \boldsymbol{y} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{e})$ and $(\boldsymbol{A}, \boldsymbol{z})$, $z \xleftarrow{\$} \mathrm{Z}_q^m$
- Search and decisional are (polynomially) equivalent
- Solutions?: Gaussian elimination, least square, etc. (nothing efficient!)

On the hardness of LPN

Worst-case hardness [Brakerski et al., ePrint/2018/279]

- LPN (for certain params) at least as hard as a worst-case decoding problem

Average-case hardness

1. Constant-noise (μ independent of n)
   - BKW [Blum, Kalai, Wasserman, JACM 2003]: time $2^{O(n/\log n)}$
2. Low-noise $\mu \leq 1/\sqrt{n}$
   - Time $poly(n) \cdot e^{\mu n}$ with unbound many samples

Quantum hardness [Esser et al., Crypto 2017]

- Speedup (based on Grover search) $\leq$ quadratic over classic algorithms

Learning Parity with Noise over Rings (Ring-LPN)

$$\boldsymbol{a}\begin{bmatrix}a_1\\a_2\\a_3\\a_4\\a_5\\a_6\\a_7\\a_8\end{bmatrix}\boldsymbol{x}\begin{bmatrix}x_1\\x_2\\x_3\\x_4\\x_5\\x_6\\x_7\\x_8\end{bmatrix}+\boldsymbol{e}\begin{bmatrix}e_1\\e_2\\e_3\\e_4\\e_5\\e_6\\e_7\\e_8\end{bmatrix}=\mathrm{y}\begin{bmatrix}y_1\\y_2\\y_3\\y_4\\y_5\\y_6\\y_7\\y_8\end{bmatrix}$$

R= $\mathbb{F}_2[X]/(g)$ for an irreducible trinomial g over $\mathbb{F}_2$
- • : multiplications over R
- \+ : additions over R (bit-wise XOR)

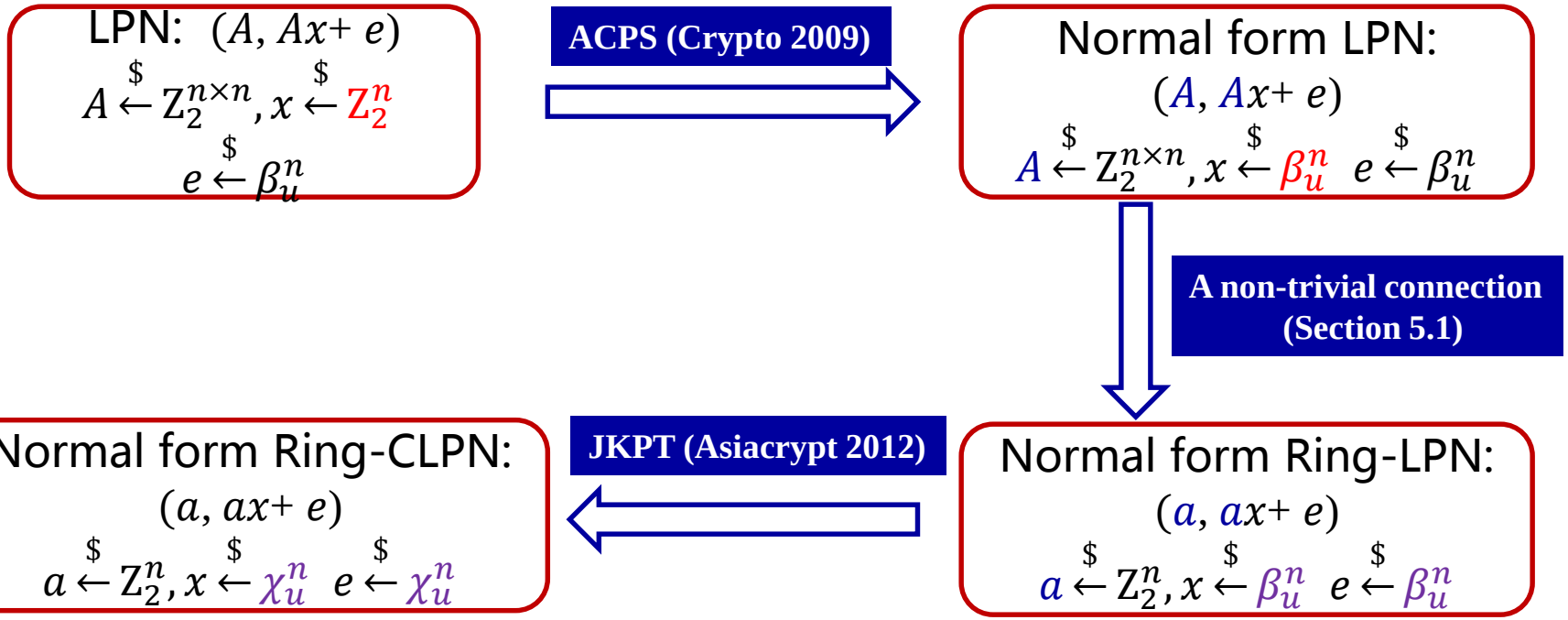
- Search Ring-LPN: find out $\boldsymbol{x}$ given $a, \boldsymbol{y} = a\boldsymbol{x} + \boldsymbol{e}$
- Decisional Ring-LPN: distinguish $(a, \boldsymbol{y} = a\boldsymbol{x} + \boldsymbol{e})$ and $(a, \boldsymbol{z})$, $z \xleftarrow{\$} \mathrm{Z}_2^m$
- Why irreducible g : conservative security

  preventing known attacking utilizing the factors of the underlying polynomial
- Why trinomial g: lightness-preserving, $|x \cdot y| \leq 2|x| \cdot |y|$,

  improving upon $|x \cdot y| \leq 7|x| \cdot |y|$ by Damgård & Park (ePrint/2012/699)

Efficiency optimizations and security arguments

- Ring R= $\mathbb{F}_2$[X]/(g) for an irreducible trinomial g over $\mathbb{F}_2$
- β_u^n (weight expected un) : n -fold Bernoulli with noise rate u
- χ_u^n (weight exact un) : random $(un\ 1's)$-out-of-n-bit distribution



Key-Encapsulation Mechanism (KEM)

- A PKE Π=(KeyGen,Enc,Dec) :
  - Key Generation: $(pk,sk) \xleftarrow{\$} KeyGen(1^n)$
  - Encapsulation: $(c,k) \xleftarrow{\$} Enc(pk)$
  - Decapsulation: $k \text{ or} \perp \xleftarrow{\$} Dec(sk,c)$
- Correctness

$$Pr[(pk,sk) \xleftarrow{\$} KeyGen(1^n), (c,k) \xleftarrow{\$} Enc(pk): Dec(sk,c)=k] \geq 1 - 2^{-large}$$

- Security

Chosen-Plaintext Attack (CPA), Chosen-Ciphertext Attack (CCA)

CPA KEM

Algorithm 1: Lepton.CPA.KeyGen(1^κ)

$\eta \xleftarrow{\$} \mathbb{F}_2^\kappa$;
$(\rho, \sigma) := G(\eta, 2)$;
$\mathbf{a} := \mathsf{Samp}(\rho)$;
$(\mathbf{s}, \mathbf{e}) := F(\sigma, 2)$;
$\mathbf{b} := \mathbf{as} + \mathbf{e}$;
return $(pk, sk) = \big((\rho, \mathbf{b}), \mathbf{s}\big)$;

Algorithm 2: Lepton.CPA.Encaps$(pk = (\rho, \mathbf{b}))$

$\eta \xleftarrow{\$} \mathbb{F}_2^\kappa$;
$(m, r) := G(pk \| \eta, 2)$;
$\mathbf{a} := \mathsf{Samp}(\rho)$;
$(\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2) \sim (\chi_\mu^n)^3 := F(r, 3)$;
$\mathbf{c}_1 := \mathbf{ax} + \mathbf{e}_1$;
$\mathbf{c}_2 := \mathsf{Trunc}(\mathbf{bx} + \mathbf{e}_2, \ell) + \mathsf{ECC}(m)$;
$K := G(\mathbf{c}_1 \| \mathbf{c}_2 \| m)$;
return $\big(K, \mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2)\big)$;

Algorithm 3: Lepton.CPA.Decaps$\big(sk = \mathbf{s}, \mathbf{C} = (\mathbf{c}_1, \mathbf{c}_2)\big)$

$\tilde{\mathbf{c}}_0 := \mathbf{c}_2 - \mathsf{Trunc}(\mathbf{c}_1\mathbf{s}, \ell)$;
$m' := \mathsf{ECC}^{-1}(\tilde{\mathbf{c}}_0)$;
return $K := G(\mathbf{c}_1 \| \mathbf{c}_2 \| m')$;

ECC and ECC^{-1}: BCH code (can even achieve perfect decryption)

Proof. CPA security follows from the Ring-CLPN assumption.

CCA KEM

Algorithm 4: Lepton.CCA.KeyGen(1^κ)

$\eta \xleftarrow{\$} \mathbb{F}_2^\kappa$;
$(\rho, \sigma) := G(\eta, 2)$;
$(\mathbf{s}, \mathbf{e}) \sim (\chi_\mu^n)^2 := F(\rho, 2)$;
$\mathbf{a} := \mathsf{Samp}(\sigma) \in \mathbb{F}_2^n$;
$\mathbf{b} := \mathbf{a} \cdot \mathbf{s} + \mathbf{e}$;
return $(pk, sk) = \big((\sigma, \mathbf{b}), \mathbf{s}\big)$;

Algorithm 5: Lepton.CCA.Encaps$(pk = (\sigma, \mathbf{b}))$

$\mathbf{a} := \mathsf{Samp}(\sigma) \in \mathbb{F}_2^n$;
$m \xleftarrow{\$} \mathbb{F}_2^\kappa$;
$(K', r, d) := G(m \| pk, 3) \in (\mathbb{F}_2^\kappa)^3$;
$(\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2) \sim (\chi_\mu^n)^3 := F(r, 3)$;
$\mathbf{c}_1 := \mathbf{a}\mathbf{x} + \mathbf{e}_1$;
$\mathbf{c}_2 := \mathsf{Trunc}(\mathbf{b}\mathbf{x} + \mathbf{e}_2, \ell) + \mathsf{ECC}(m)$;
$\mathbf{C} := (\mathbf{c}_1, \mathbf{c}_2, d) \in \mathbb{F}_2^{n+\ell+\kappa}$;
$K := G(K' \| \mathbf{C}) \in \mathbb{F}_2^\kappa$;
return $\big(\mathbf{C}, K\big)$;

Algorithm 6: Lepton.CCA.Decaps$\big(sk = \mathbf{s}, \mathbf{C}\big)$

$\tilde{\mathbf{c}}_0 := \mathbf{c}_2 - \mathsf{Trunc}(\mathbf{c}_1\mathbf{s}, \ell)$;
$m := \mathsf{ECC}^{-1}(\tilde{\mathbf{c}}_0)$;
$(K', \mathbf{C}') = \mathsf{Lepton.CCA.Encaps}(pk; m)$;
if $\mathbf{C}' = \mathbf{C}$ **then**
  $K := K'$;
else
  $K := \perp$;
return K ;

Proof. Apply the Fujisaki-Okamoto transform to CPA.KEM in the quantum random oracle.

Parameters and Performance

LPN-based KEMs/PKEs enjoy simplicity and parallelizability.

- Reference software implementation of Lepton.CCA.KEM:

			bytes			CPU cycles			err. prob
	n	μn	$\|pk\|$	$\|sk\|$	$\|c\|$	Keygen	Enc	Dec	
Light I	8100	16	1045	1077	1617	34308	79152	87043	2^{-87}
Light II	8100	20	1045	1085	1998	34536	86584	100141	2^{-121}
Moderate I	16153	19	2052	2090	2497	49943	121564	132708	2^{-114}
Moderate II	16153	24	2052	2100	2751	51658	124426	141988	2^{-123}
Moderate III	16153	28	2052	2108	3005	52699	130631	151185	2^{-120}
Moderate IV	16153	37	2052	2126	4021	59450	154473	179520	2^{-129}
Paranoid I	32767	35	4128	4198	5335	94454	234441	264881	2^{-122}
Paranoid II	32767	40	4128	4208	5589	97569	244706	282199	2^{-148}

- Hardware implementations expected more desirable
  - Additions and multiplications over GF(2) are XOR and AND logical gates respectively.
  - Suitable for low-power devices and even RFID tags .

Adjusted Parameters and Security

Asymptotic security: we proved $2^{O(\mu n)}$ but we claim the aggressive $n^{\mu n/2}$

more precisely $O(n^3)2^{3\log(\frac{3}{2})\,\mu n}$ thanks to Dr. Philippe Gaborit

Concrete security: evaluated using Stern-Dumer ISD

(similar results by May-Meurer-Thomae ISD, Becker-Joux-May-Meurer ISD)

						Adjusted	
	n	m	μn	C-Sec	Q-Sec	C-Sec	Q-Sec
Light I	8100	9	16	103	51	56	28
Light II	8100	9	20	128	64	63	31
Moderate I	16153	10	19	132	66	63	31
Moderate II	16153	10	24	167	83	72	36
Moderate III	16153	10	28	195	97	79	39
Moderate IV	16153	10	37	258	129	95	47
Paranoid I	32767	1	35	262	131	93	46
Paranoid II	32767	1	40	299	149	102	51

1. The last three (Moderate IV and Paranoid I/II) remain of 80+ bits of classic security.

2. May need to double/quadruple the sizes of Paranoid I/II to get 128+/256+ bits of classic security.

Conclusions

- Lepton: KEMs based on the (quantum) hardness of Ring-LPN
- Efficient software and even faster hardware implementations.
- Adjusted security estimates:
  - The original proposal supports up to 100 bits of classic security.
  - Double/quadruple the sizes for 128/256-bit (resp. 64/128-bit) classic (resp. quantum) security.
  - The adaption can be done efficiently by tuning the parameters, e.g., tabulated below

Adjusted	n	μn	C-Sec	Q-Sec	PK size	CT size	Err Prob.
Light I	$32767 \approx 2^{15}$	40	90	45	4KB	5.5KB	2^{-148}
Moderate I	$32767 \approx 2^{15}$	62	128	64	4KB	8KB	2^{-102}
Moderate II	2^{16}	90	192	96	8KB	13KB	2^{-110}

Thank you

for your attention!

E-mail: yuyu@yuyu.hk