Practical Post-Quantum Public-Key Encryptions

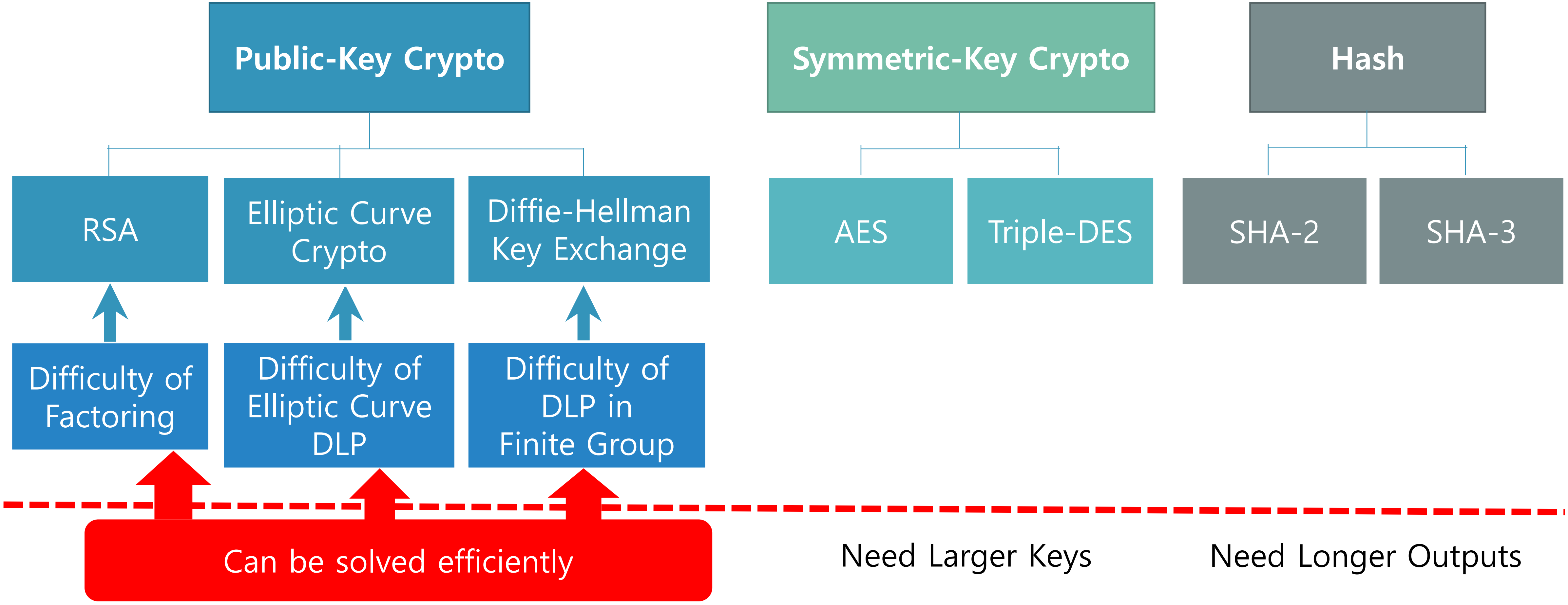
2017.03.24

Yongsoo Song

Contents

Motivation

Contemporary Cryptography



< Quantum Computing Era >

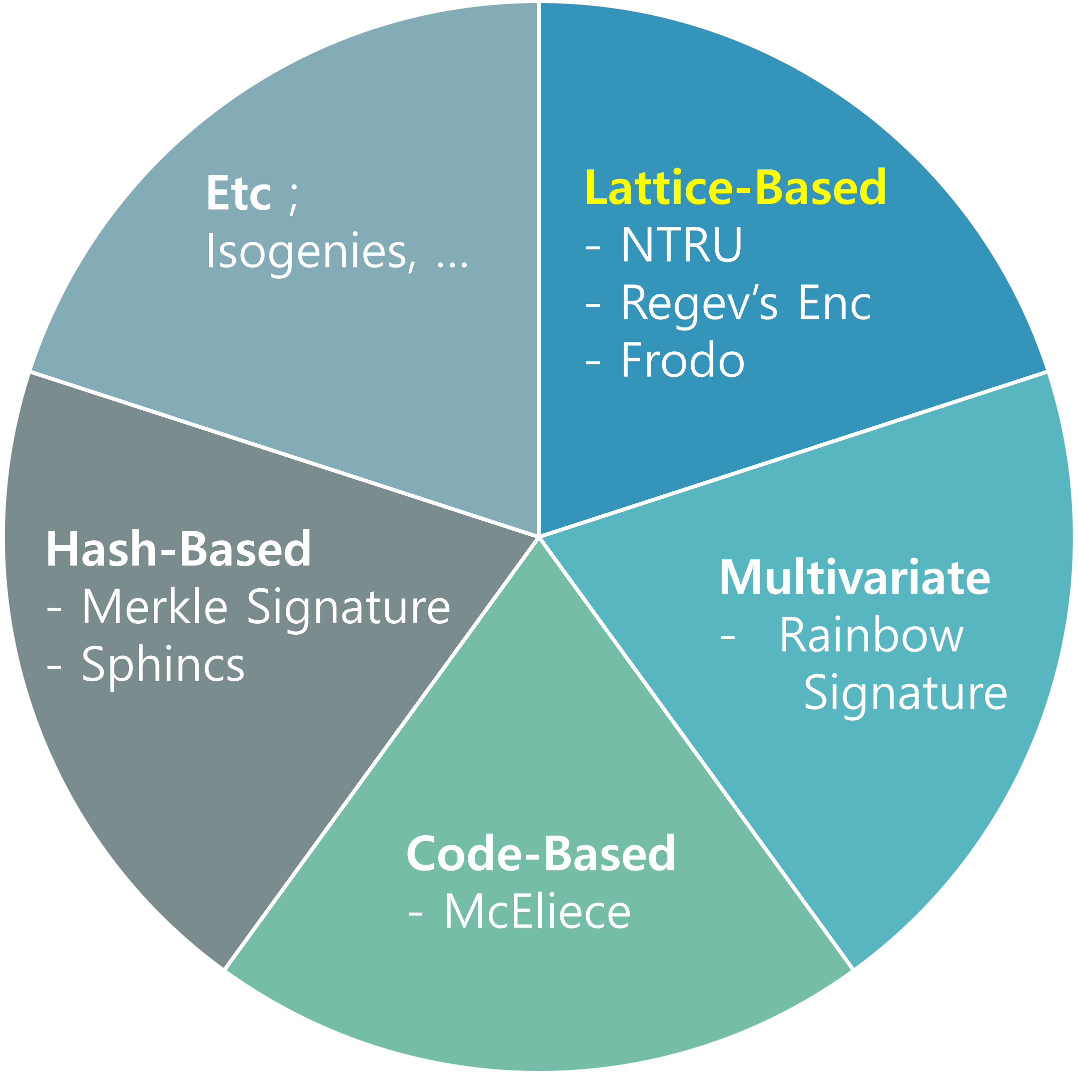
Post-Quantum Cryptography

- **NSA is transitioning to post-quantum crypto in the "not too distant" future;** http://www.iad.gov/iad/programs/iad-initiatives/cnsa-suite.cfm

- **NIST launched Post-Quantum Crypto Project on Aug. 2, 2016;** http://csrc.nist.gov/groups/ST/post-quantum-crypto
  - ➢ To standardize Post-Quantum public-key crypto : Encryption / Signature / Key Exchange
  - ➢ Timeline

Fall 2016	Formal Call for Proposals
Nov 2017	Deadline for Submissions

Post-Quantum Crypto



- **Lattice-based crypto gains increasing attentions;**
  - Security based on the NP-hard worst-case lattice problems
  - Fast implementation
  - Versatility in many applications: HE, IBE, …
- **We focus on LWE-based Encryption**

Learning with Errors (LWE) Problem

Solving a linear equation system

- Q.

$$\underbrace{\begin{pmatrix} 1 & 3 & 7 \\ 4 & 5 & 7 \\ 6 & 6 & 9 \\ 2 & 7 & 3 \\ 3 & 8 & 7 \\ 5 & 4 & 2 \\ 1 & 0 & 5 \\ 4 & 5 & 3 \end{pmatrix}}_{\in\, \mathbb{Z}_{10}^{8\times 3}} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 7 \\ 9 \\ 2 \\ 9 \\ 6 \\ 8 \\ 2 \\ 7 \end{pmatrix} \pmod{10} \Rightarrow \text{Find } \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}!$$

; Easy!

(We can solve it by using Gaussian elimination)

Learning with Errors Problem (LWE)

- Q.

$$\underbrace{\begin{bmatrix} 1 & 3 & 7 \\ 4 & 5 & 7 \\ 6 & 6 & 9 \\ 2 & 7 & 3 \\ 3 & 8 & 7 \\ 5 & 4 & 2 \\ 1 & 0 & 5 \\ 4 & 5 & 3 \end{bmatrix}}_{\in \mathbb{Z}_{10}^{8\times 3}} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \underbrace{\begin{bmatrix} 0 \\ 2 \\ 9 \\ 1 \\ 0 \\ 1 \\ 0 \\ 8 \end{bmatrix}}_{\text{Small Error (unknown)}} = \begin{bmatrix} 7 \\ 1 \\ 1 \\ 0 \\ 6 \\ 0 \\ 2 \\ 5 \end{bmatrix} \pmod{10} \Rightarrow \text{Find } \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} !$$

; Hard!

Decision-LWE Problem

- Q. Distinguish

1	3	7
4	5	7
6	6	9
2	7	3
3	8	7
5	4	2
1	0	5
4	5	3

,

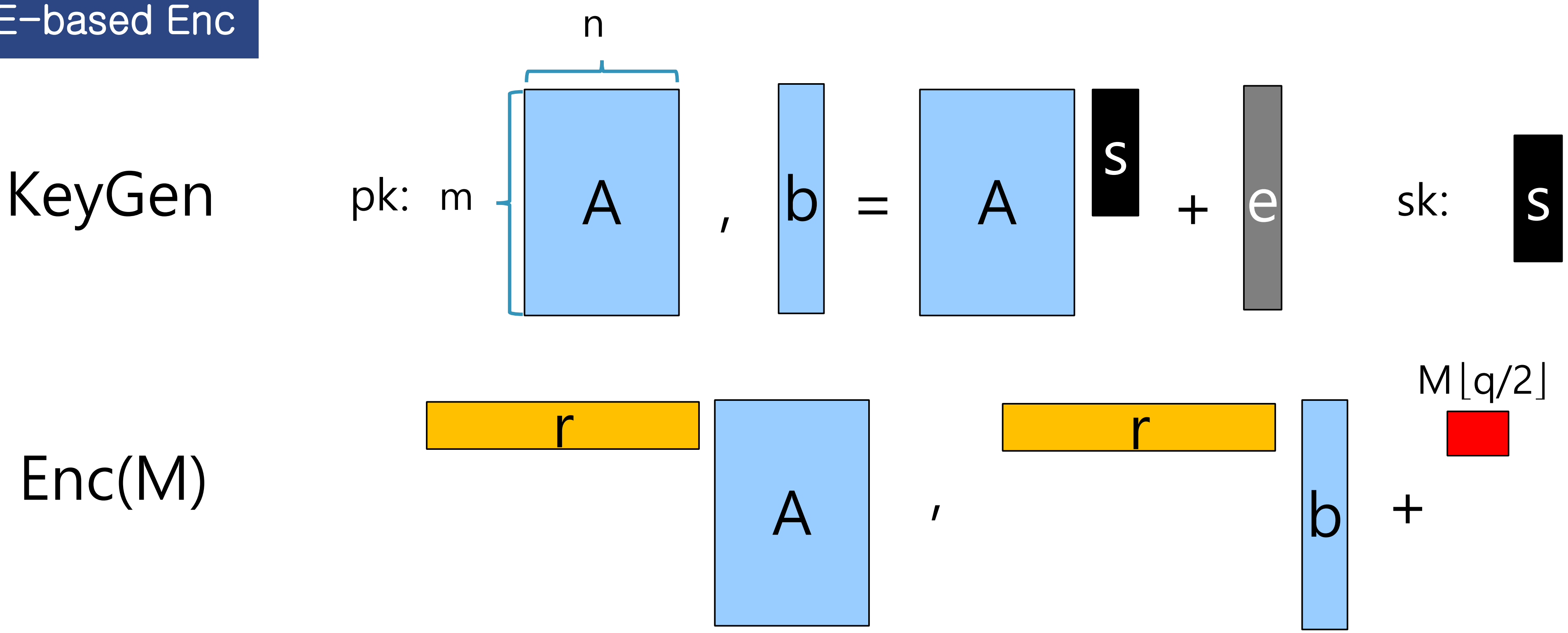
7
1
1
0
6
0
2
5

from a uniform random sample in $\mathbb{Z}_{10}^{8\times 4}$!

; Hard!

LWE-based Encryptions

LWE + LHL [Reg05]



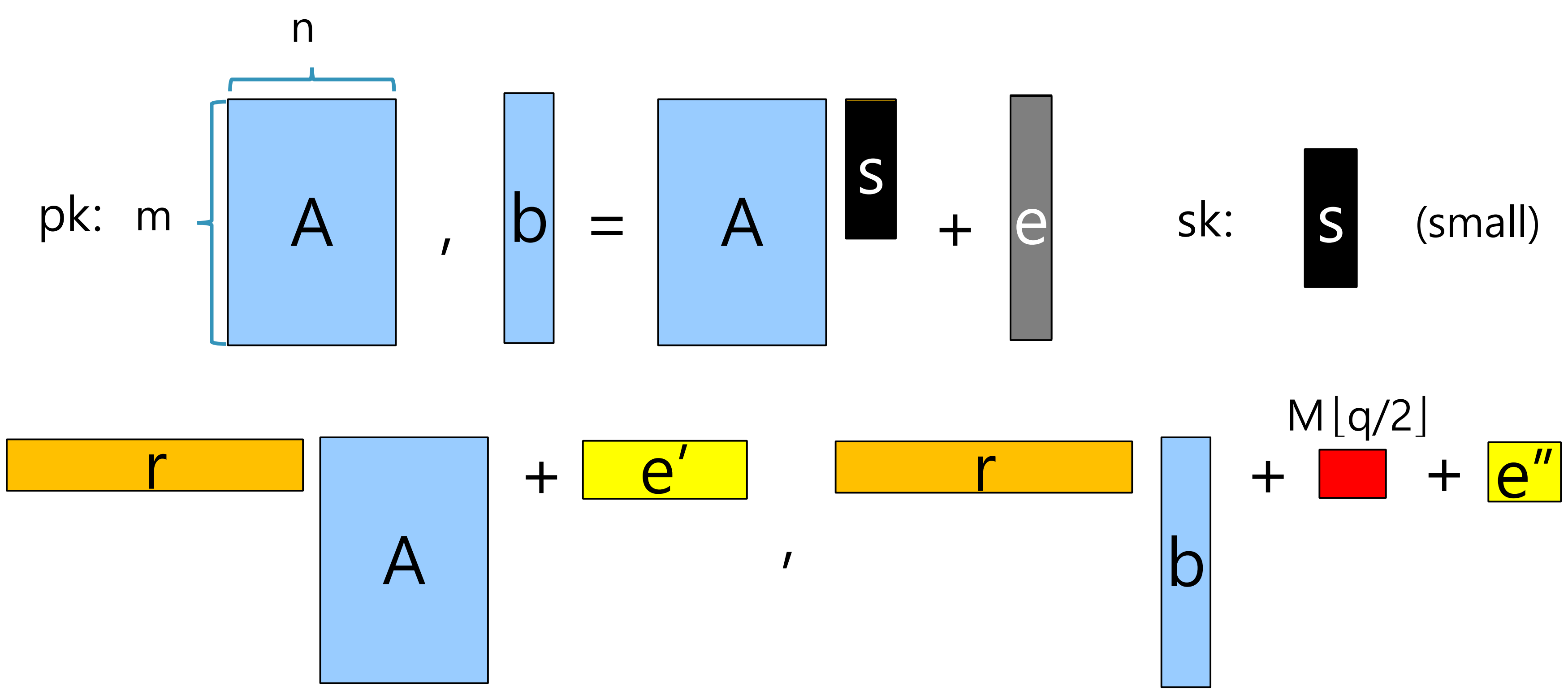
- **Require a large m to randomize LWE samples in Encryption**
  - ➢ Leftover Hash Lemma

➢Can We Reduce m?

LWE + LWE [LP11]

KeyGen

Enc(M)



- **Pros: smaller m by replacing LHL with LWE**
- **Cons:** ***Discrete Gaussian samplings***

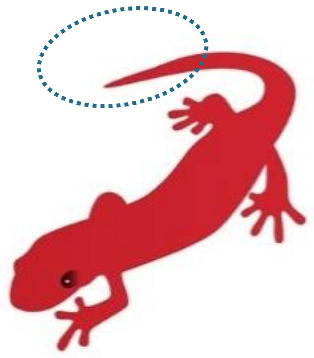
LWE + LWR [CKLS16]

KeyGen

pk: m $\underbrace{A}_{n}$, $b = A s + e$ sk: s (small)

$\mathbf{d}$ = Enc*(M) $r A$, $r b + M\lfloor q/2 \rfloor$

$c = \left\lfloor \frac{p}{q} \cdot \mathbf{d} \right\rceil$ (cf. $\mathbf{d} = \begin{bmatrix} 10110110 \\ 01101011 \\ 11010100 \\ \vdots \\ 01001001 \end{bmatrix} \rightarrow \begin{bmatrix} 10110110 \\ 01101011 \\ 11010100 \\ \vdots \\ 01001001 \end{bmatrix}$, if $p = 2^7, q = 2^9$.)



LWE + LWR [CKLS16]

KeyGen pk: $\underbrace{\left.A\right\}_{m}}_{n}$, $b = A\,s + e$ sk: s

Setup Choose moduli q, p. Integers m, n.

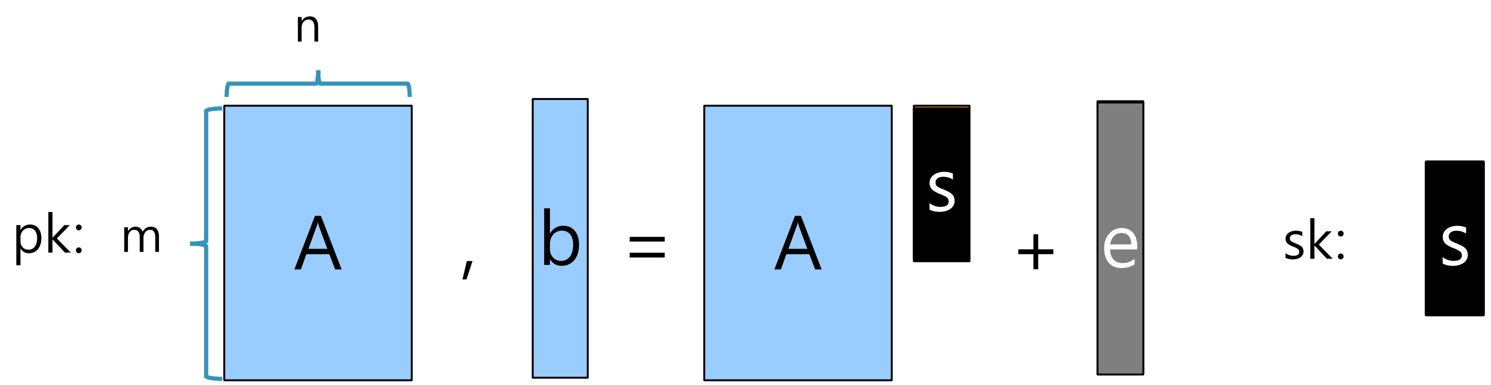
s Sampled from a small distribution,
e.g. Binary (with small Hamming weight), Gaussian

A Uniformly sampled from $Z_q^{m \times n}$

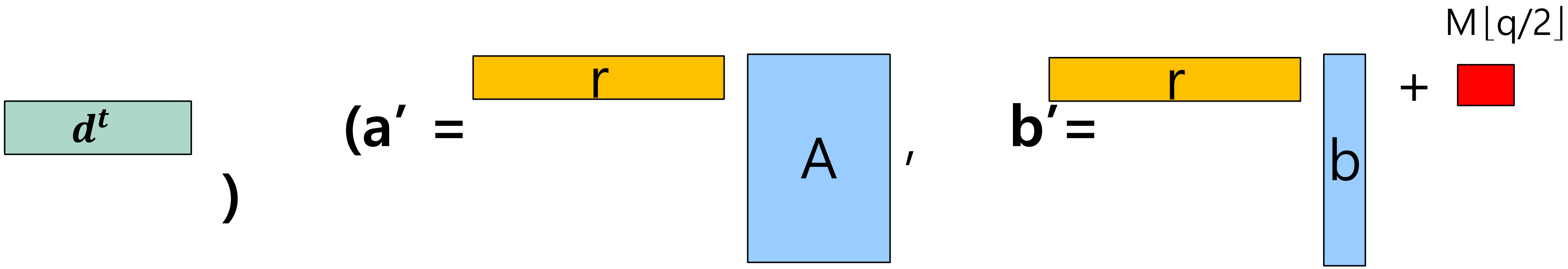
e Sampled from Gaussian distribution

LWE + LWR [CKLS16]

KeyGen



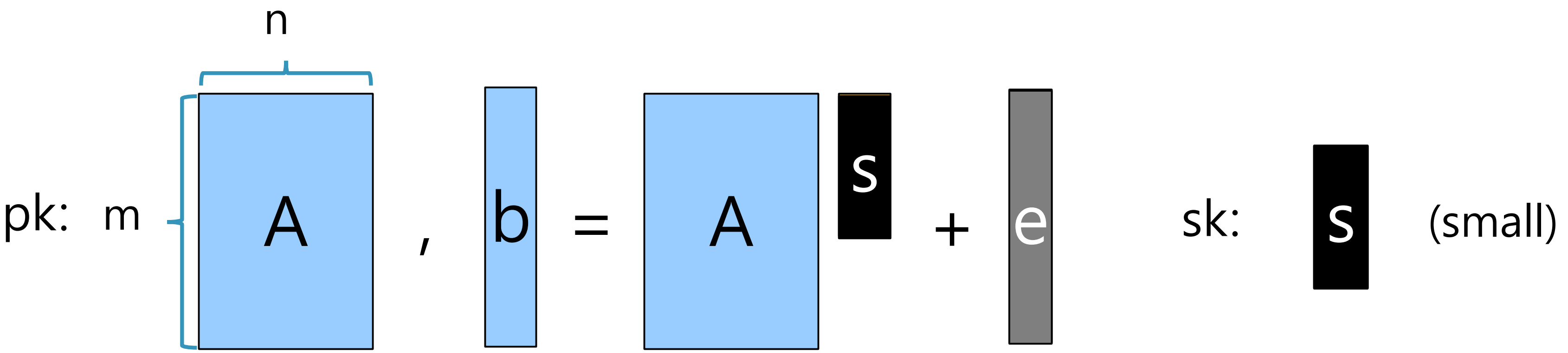
Sampled from a small distribution,
e.g. Binary (with small Hamming weight), Gaussian

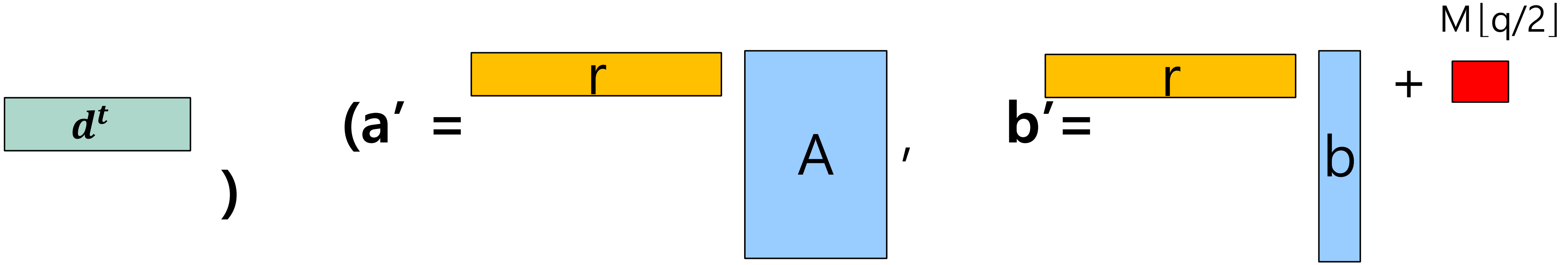


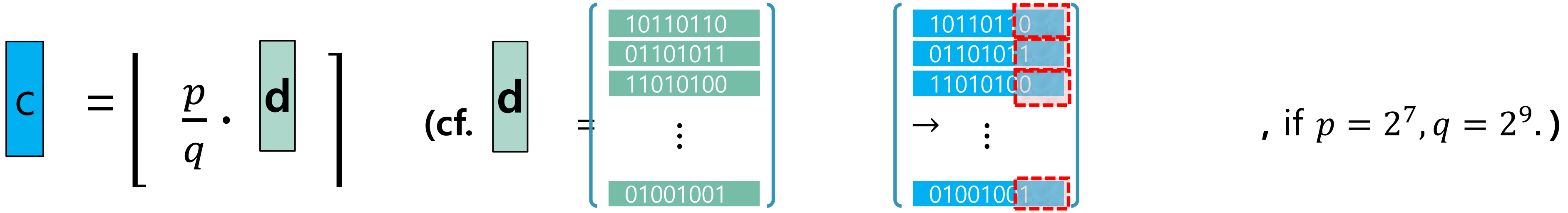
$$d = (a', b') \;\Rightarrow\; b' \approx \langle a', s \rangle + M(q/2) \pmod{q}$$

LWE + LWR [CKLS16]

KeyGen







$$c = (a, b) \Rightarrow b \approx \langle a', s \rangle + M(p/2) \pmod{p}$$

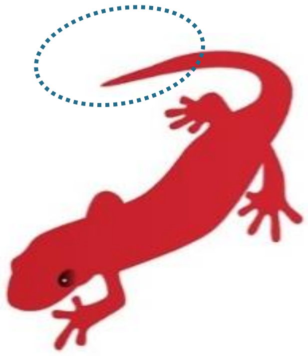
Learning with Rounding (LWR) Problem

- **Surprisingly,** it is secure under LWR assumption

- LWR: Distinguish any m pairs of type

$$\left(\overbrace{a_i}^{n} , b_i = \left\lfloor \frac{p}{q} a_i \; s \right\rceil \right) \in Z_q^n \times Z_p$$ from uniform

Discard the least significant bits of $<a_i,s>$
instead of adding small errors

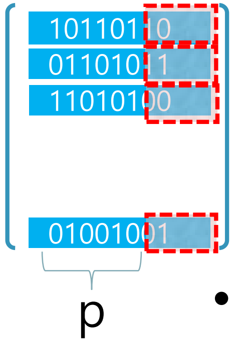


- **Have reduction from LWE:** q is large or **m is small**

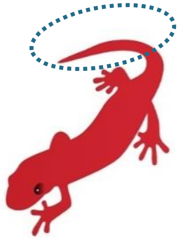
The Hardness of LWR Problem

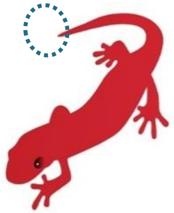
(q: LWR modulus, p: rounding modulus, n: LWR dimension.)

- Before 2016, security reduction <u>only when the modulus is somewhat large.</u>
  - ➢ Banergee, Peikert, Rosen [BPR12] introduced LWR, and showed *LWR ≥ LWE when q is sufficiently large.* ($q \geq p \cdot B \cdot n^{\omega(1)}$, B: LWE noise support bound)
  - ➢ Alwen et al. [AKPW13] showed *LWR ≥ LWE when the modulus and modulus-to-error ratio are super-poly.*
- Bogdanov et al. [BGM+16] in TCC 2016 showed LWR ≥ LWE when the number of samples is no larger than $O(q/Bp)$. (B: LWE noise support bound)
- Cryptanalytic hardness against best known lattice attacks: LWR = LWE when the variance of LWE noise is $12q^2/p^2$. (size of noise vectors are the same)



Caution! - How many LSBs can be discarded?

- **(Correctness)** If we cut a large proportion;  , the correctness will not hold. ☹

- **(Security)** We can not remove noise addition ☹ if we cut very small; 

→ Since **the number of samples of LWR** in the Enc procedure is restricted to be **small**, we can choose a proper rounding modulus "p" to satisfy both security and correctness. ☺

<Bogdanov et al.> If the # of samples(m) is no larger than $O(q/Bp)$,
we cannot distinguish either one from uniform;

$$\left(m\underbrace{\left[\,A\,\right]}_{n} ,\ \left\lfloor \frac{p}{q}\cdot\left(A\,s + e \right) \right\rceil \right) \leftrightarrow \left(m\underbrace{\left[\,A\,\right]}_{n} ,\ \left\lfloor \frac{p}{q}\cdot A\,s \right\rceil \right)$$

Advantage of LWR assumption

pk: A , $b = As + e$ sk = (-s, 1)

LP11.Enc(M) $rA + e_1$, $rb + M\lfloor q/2 \rceil + e_2$ $Var(e_i) = \sigma^2$

Lizard.Enc(M) $\left\lfloor \frac{p}{q} \cdot \left(rA ,\ rb + M\lfloor q/2 \rceil \right) \right\rceil$

Rounding error (e_1, e_2):
(uniform over $[\pm q/2p]$)
Variance $\sigma^2 = q^2/12p^2$

Encryption noise: $\langle r, e\rangle + \langle (e_1, e_2),\ sk\rangle$

Set the parameter $\sigma^2 = q^2/12p^2$: *Preserve cryptanalytic hardness* LWE(m,q,σ) = LWR(m,q,p) and functionality (encryption noise)

- **Smaller CTXT**
- **No Gaussian sampling in Encryption**

Performance of IND-CPA scheme

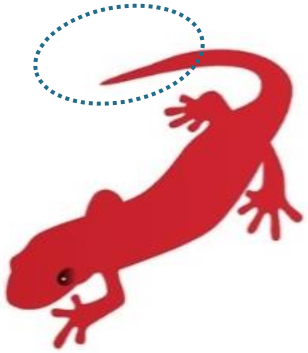
- Enc/Dec speeds; encrypting 256 bits with 128-bit post-quantum security

Scheme	Enc	Dec
RSA-3072	0.035 (116,894)	2.673 (8,776,864)
NTRU EES593EP1	0.024 (80,558)	0.025 (82,078)
Our Scheme	0.024 (80,558)	0.020 (62,813)

[Table] Performance of our Enc/Dec procedures in miliseconds (nb of cycles)

- Our scheme: measured on a PC with Intel dual-core i5 running at 2.6 GHz w/o parallelization.
- RSA, NTRU: measured on a PC with Intel quad-core i5-6600 running at 3.3 GHz processor, drawn from ECRYPT Benchmarking of Crypto Systems.
- RSA does not achieve post-quantum security.

Security



- **Asymptotic hardness;**
  - LWE with small secrets (e.g. Discrete Gaussian, Binary, Sparse binary)
  - Thanks to reduction from LWE to LWR
- **Concrete hardness;**
  - Follow the framework of Frodo / NewHope in parameter selection
  - Extension to LWR problem (OLA)
  - Current Combinatorial Attack on Sparse Secret LWE [Alb17]
- **Quantum Security;**
  - IND-CCA in Quantum ROM using modified FO conversion [TU16] ➔ Optimal?

Questions?

Any comments,
Implementation tips,
applications,
and even attacks would be appreciated!

PQ Lizard: Cut off the Tail! Practical Post-Quantum Public-Key Encryption from LWE and LWR
Jung Hee Cheon, Duhyeong Kim, Joohee Lee, and Yongsoo Song, **ePrint 2016 / 1126**


Thank You !