KCL
(Key Consensus from Lattice)

Yunlei Zhao
Fudan University, Shanghai, China
(on behalf of the KCL team)

April 11, 2018

Brief Summary

A General Framework

- A modular, systematic and versatile approach to key establishment, PKE, identity-concealed AKE based on LWE and its variants.
  - Applicable to almost all the variants of LWE with different mathematical structures.
  - We focus on implementations on LWE, LWR, RLWE and MLWE.
  - A unified framework for understanding and evaluating various KEM/PKE proposals from LWE and its variants.

Key Building Tool: Formulation, Upper-bounds, and Optimal Design

- Explicit formulation of key consensus (KC) and asymmetric key consensus (AKC), which are at the heart of KE and PKE from LWE and its variants.
    - Briefly speaking, KC corresponds to Diffie-Hellman, while AKC corresponds to ElGamal.
- Reveal inherent constraints on bandwidth, correctness and consensus range, for any KC and AKC.
- Design of optimal KC and AKC: OKCN and AKCN, guided by the proved inherent constraints on any KC/AKC.

KE/PKE From LWE and MLWE

- State-of-the-art of LWE-based key exchange (KE) and CPA-secure PKE.
  - The underlying KC mechanism of Frodo is not optimal, while our OKCN is.
  - The underlying AKC mechanism of FrodoKEM is a restricted (less versatile) version of our AKCN.
- State-of-the-art of MLWE-based KE and CPA-PKE.
  - The MLWE-based CPA-PKE from AKCN is essentially the same as Kyber.
  - Kyber focus on AKC-based implementations, while ours is for both AKCN and OKCN.

KE/PKE From LWR

- The first KE and CPA-PKE *merely* based on LWR, with a delicate analysis of error probability.
  - State-of-the-art KC-based KE from LWR.
  - Unified structure allowing for instantiations from both KC and AKC, with the technique of randomness lifting.
  - Related proposals: the subsequent works of Saber and Round-2 show that for LWR-based KE from AKC, randomness lifting is not necessary. But the structure of Saber and Round-2 does not apply to KC-based KE.

KE/PKE From RLWE (I)

- When applied to RLWE-based cryptosystems, we make a key observation by proving that the errors in different positions in the shared-key are essentially independent.
  - It is just heuristically claimed in existing works, without any arguments or justifications.
- AKCN4:1, apply AKCN with lattice-code in D_4 of NewHope
  - It is the first AKC-based variant of NewHope: publicly available earlier than NewHope-Simple (the proposal to NIST).
  - Almost as simple as NewHope-Simple, but relatively more efficient in bandwidth.
- We want to do better: new error-correction mechanisms.

KE/PKE From RLWE (II)

- single-error correction: we propose an extremely simple and fast code, referred to as *single-error correction* (SEC) code, to correct at least one bit error.
    - By equipping OKCN/AKCN with the SEC code, we achieve the simplest RLWE-based key exchange (up to now) *with negligible error rate* for much longer shared-key size.
- To further improve the bandwidth, error rate and post-quantum security simultaneously, we develop new lattice code in E_8.
    - Packing in E_8 is optimal.
    - Packing in D_4 as in NewHope is not optimal.
    - Packing in Leech lattice is also optimal in 24 dimensions, but is more complicated.

CCA-PKE Transformation

- All the OKCN-based KE and AKCN-based CPA-PKE can be transformed into CCA-secure using the FO-transformation. We use a variant of [BDKLLSSS17].
- Being different from the variant of FO-transformation proposed in [BDKLLSSS17], in our CCA-PKE construction one part derived from PK and randomness seed is not sent in plain but encrypted with AEAD.
  - It is more conservative and prudent for security in practice.
  - Well compatible with identity-concealed AKE.

Concealed Non-malleable Key Establishment (CNKE)

CNKE does not use signatures, and is carefully designed to enjoy the following advantages simultaneously:

- Computational efficiency: replacing CCA-secure KEM in existing constructions with an ephemeral KE protocol.
- Robust resistance to MIM malleating attacks, to secrecy exposure, and to side-channel attacks.
- Privacy protection: identity information, as well as the components of the underlying ephemeral KE, is encrypted.
  - The only AKE of this feature in proposals to NIST.
  - Identity privacy is mandated by some prominent standards like TLS1.3, EMV, etc.
- Well compatibility with TLS1.3: uses AE (mandated by TLS1.3), and uses the Finish mechanism of TLS1.3.

Definitions, Properties, and Constructions of OKCN and AKCN

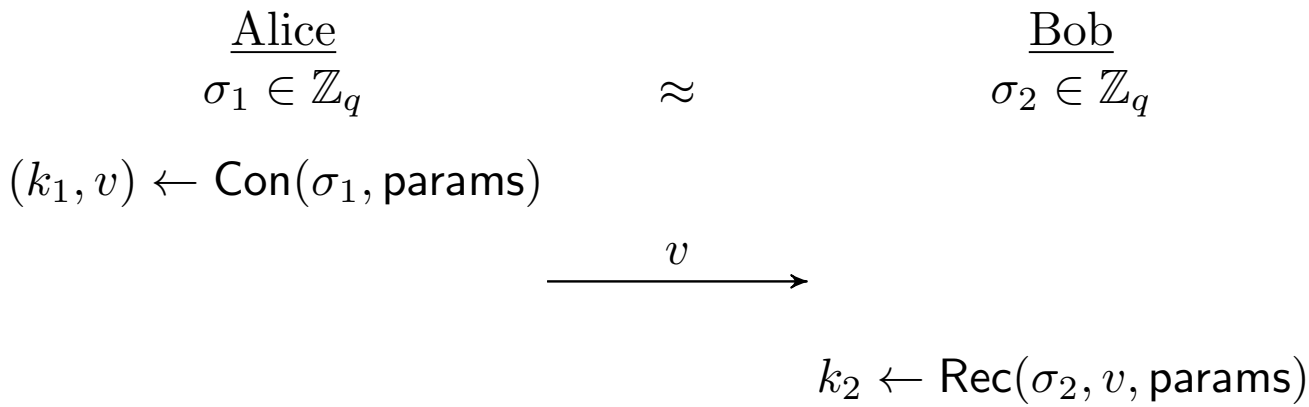


Figure: Depiction of KC

Completeness: $k_1 = k_2$, whenever $|\sigma_1 - \sigma_2|_q < d$.

Security: k_1 is independent of v, and is distributed uniformly at random over Z_m, whenever $\sigma_1 \leftarrow \mathbb{Z}_q$.

Constraint: $2md \leq q\left(1 - \frac{1}{g}\right)$.

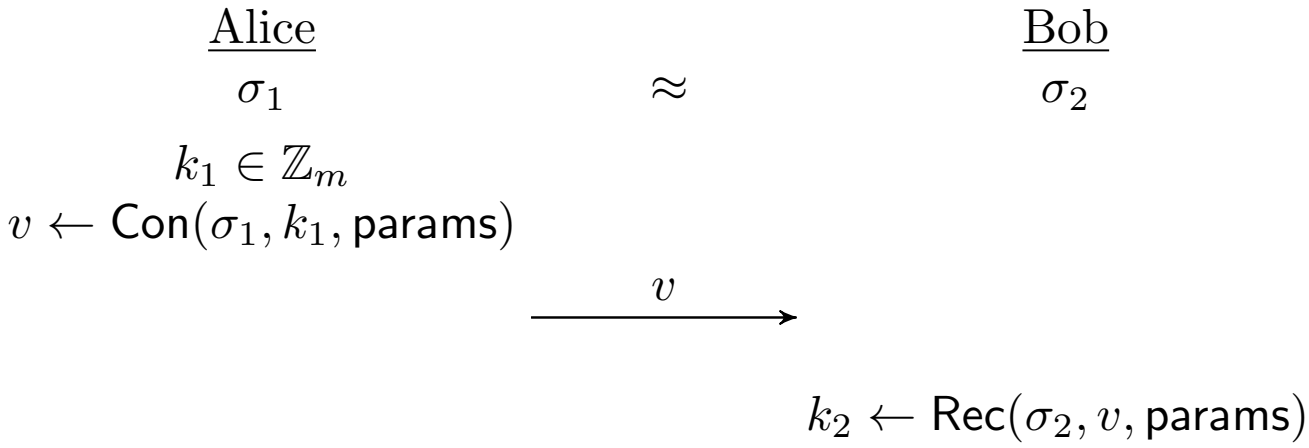


Figure: Depiction of Asymmetric KC (AKC)

Completeness: $k_1 = k_2$, whenever $|\sigma_1 - \sigma_2|_q < d$.

Security: k_1 is independent of v, whenever $\sigma_1 \leftarrow \mathbb{Z}_q$.

Constraint: $2md \leq q\left(1 - \frac{m}{g}\right)$.

Algorithm 1 OKCN

params $= (q, m, g, d, aux)$, $aux = \{q' = \text{lcm}(q, m), \alpha = q'/q, \beta = q'/m\}$

procedure Con($(\sigma_1, \text{params})$) ▷ $\sigma_1 \in [0, q-1]$

  $e \leftarrow [-\lfloor(\alpha-1)/2\rfloor, \lfloor\alpha/2\rfloor]$

  $\sigma_A = (\alpha\sigma_1 + e) \bmod q'$

  $k_1 = \lfloor\sigma_A/\beta\rfloor \in \mathbb{Z}_m$

  $v' = \sigma_A \bmod \beta$

  $v = \lfloor v'g/\beta\rfloor$ ▷ $v \in \mathbb{Z}_g$

  return (k_1, v)

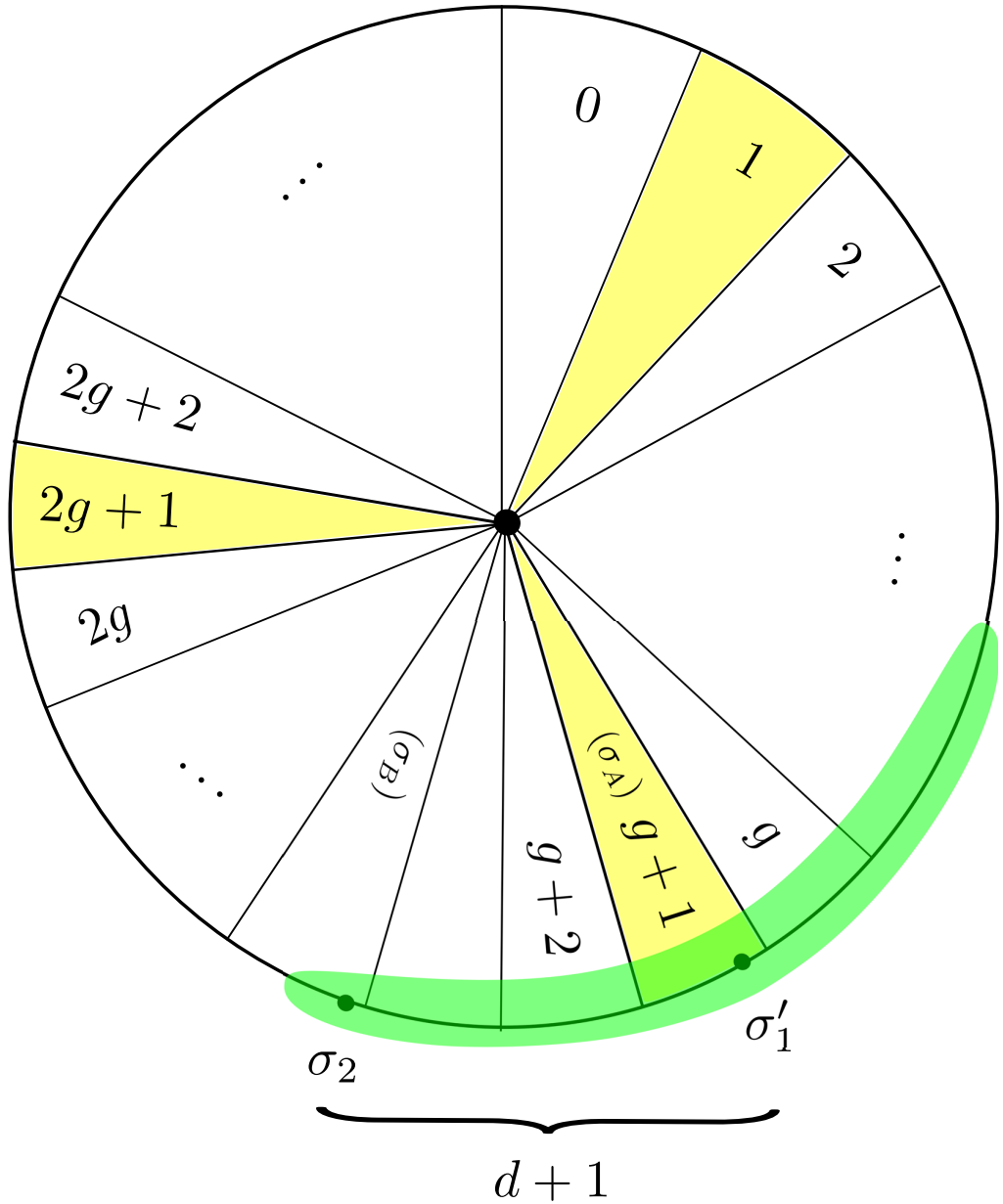
end procedure

procedure Rec($\sigma_2, v, \text{params}$) ▷ $\sigma_2 \in [0, q-1]$

  $k_2 = \lfloor\alpha\sigma_2/\beta - (v + 1/2)/g\rceil \bmod m$

  return k_2

end procedure


0
1
2
$\cdots$
$2g+2$
$2g+1$
$2g$
$\cdots$
(σ_B)
$g+2$
(σ_A) $g+1$
g
$\cdots$
σ_2
σ_1'
$d+1$

Algorithm 2 OKCN simple

params : $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}, d$, where $\bar{g} + \bar{m} = \bar{q}$

procedure Con(σ_1, params)

    $k_1 = \left\lfloor \frac{\sigma_1}{g} \right\rfloor$

    $v = \sigma_1 \bmod g$

    return (k_1, v)

end procedure

procedure Rec(σ_2, v, params)

    $k_2 = \left\lfloor \frac{\sigma_2 - v}{g} \right\rceil \bmod m$

    return k_2

end procedure

Algorithm 3 AKCN

params $= (q, m, g, d, aux)$, where $aux = \emptyset$.
procedure $\textsc{Con}((\sigma_1, k_1, \text{params}))$ ▷ $\sigma_1 \in [0, q-1]$
    $v = \lfloor g\,(\sigma_1 + \lfloor k_1 q/m \rceil)\,/q \rceil \bmod g$
    return v
end procedure
procedure $\textsc{Rec}((\sigma_2, v, \text{params}))$ ▷ $\sigma_2 \in [0, q-1]$
    $k_2 = \lfloor m(v/g - \sigma_2/q) \rceil \bmod m$
    return k_2
end procedure

Algorithm 4 AKCN power 2

1: params : $q = g = 2^{\bar{q}}, m = 2^{\bar{m}}, aux = \{G = q/m\}$
2: **procedure** $\textsc{Con}(\sigma_1, k_1, \text{params})$
3: $v = (\sigma_1 + k_1 \cdot G) \bmod q$, where $k_1 \cdot G$ can be offline computed
return v
4: **end procedure**
5: **procedure** $\textsc{Rec}(\sigma_2, v, \text{params})$
6: $k_2 = \lfloor (v - \sigma_2)/G \rceil \bmod m$
return k_2
7: **end procedure**

A Note on KC/AKC

- KC and AKC were there in pioneering works, but were not formally formulated, nor the upper-bounds on various parameters were studied.
- Inherent upper-bounds: allow us to understand what can or cannot be achieved with any KC/AKC, and guide our actual protocol design.
  - These upper-bounds also guide parameter choosing for various trade-offs, and are insightful in comparing the performance of KC vs. AKC.
- Optimality and Flexibility of OKCN and AKCN.
- Much simplify future design and analysis of cryptosystems from LWE and its variants.

LWR-Based Key Exchange from OKCN/AKCN

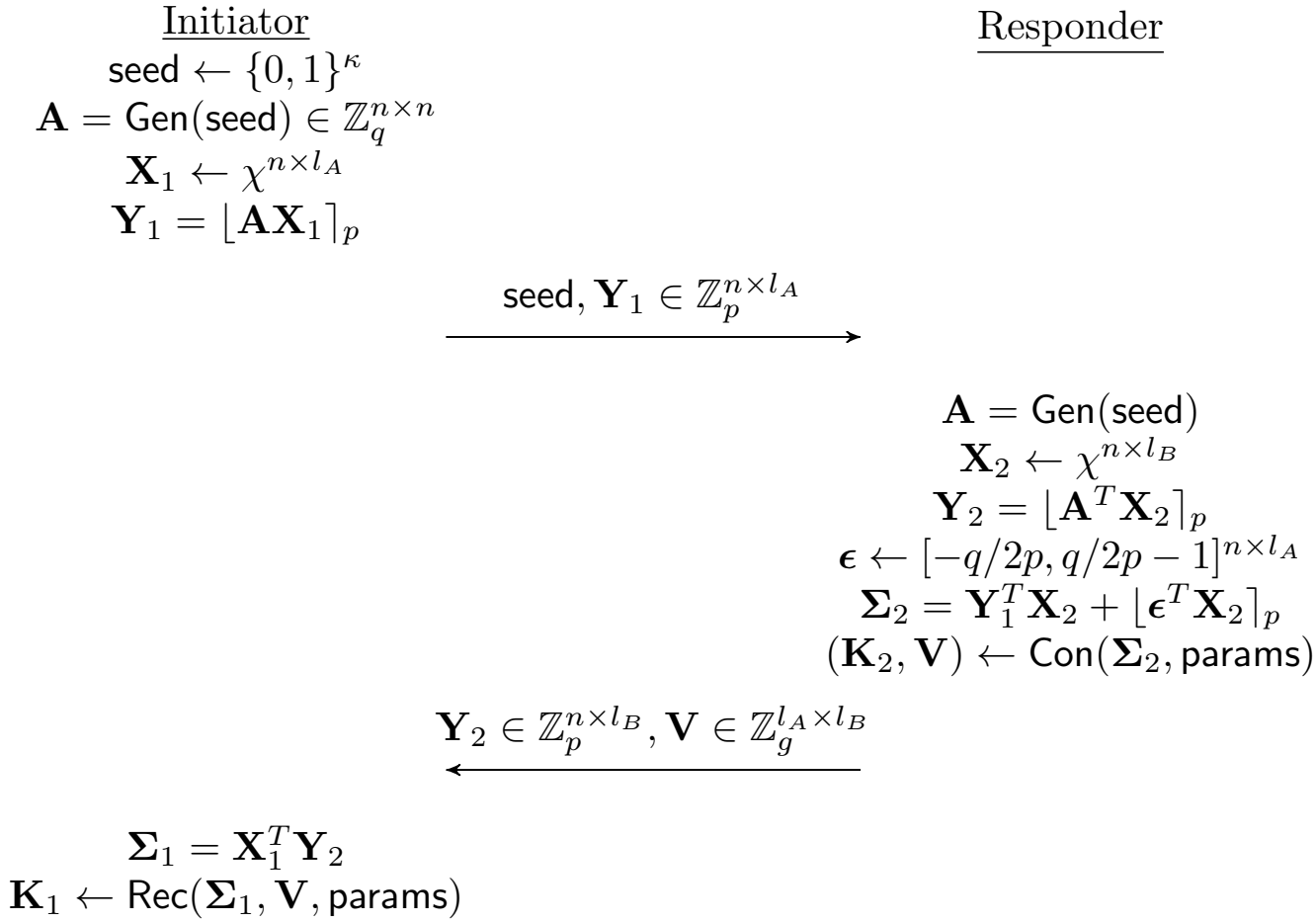


Figure: LWR-based KE from KC/AKC

Brief Comparison with Saber and Round-2

- To our knowledge, this is the first KE-protocol *merely* based on LWR, with a delicate analysis of error rate.
  - Publicly available from arXiv since Feb 16 2017.
  - Unified protocol structure, with randomness lifting, which supports implementations based on both KC and AKC.
    - For recommended parameters, randomness lifting corresponds to $\epsilon \leftarrow [-2^k, 2^k)^{n \times l_A}$, which is highly efficient.
- The subsequent works of Saber and Round-2 show that for LWR-based KE from AKC, randomness lifting is not necessary. But the structure of Saber and Round-2 does not apply to KC-based KE.

LWE-Based Key Exchange From OKCN/AKCN

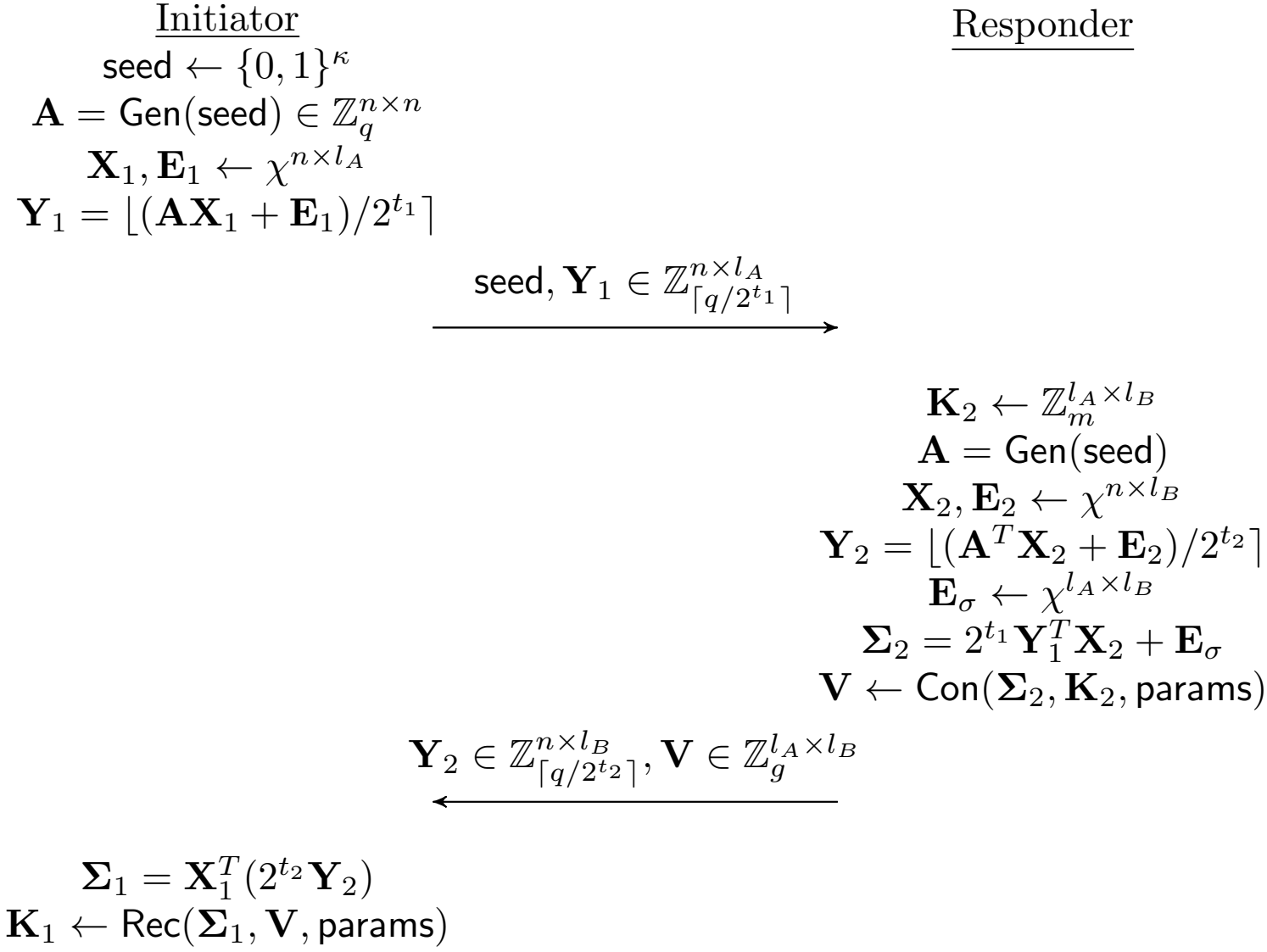


Figure: LWE-based key exchange from AKC (recommended: $t_1 = 0$ but $t_2 \neq 0$

AKCN-LWE vs. FrodoKEM

FrodoKEM (the actual proposal to NIST) can be viewed as a restricted (less versatile) version of AKCN-LWE:

- In AKCN-LWE, we recommend that $t_2 \neq 0$ for reducing bandwidth, while $t_2 = 0$ in FrodoKEM.
- In AKCN-LWE, q (the security parameter) and g (the bandwidth parameter) are not necessarily are identical, and it is recommended for $g < q$ for bandwidth reduction, while $q = g$ in FrodoKEM.

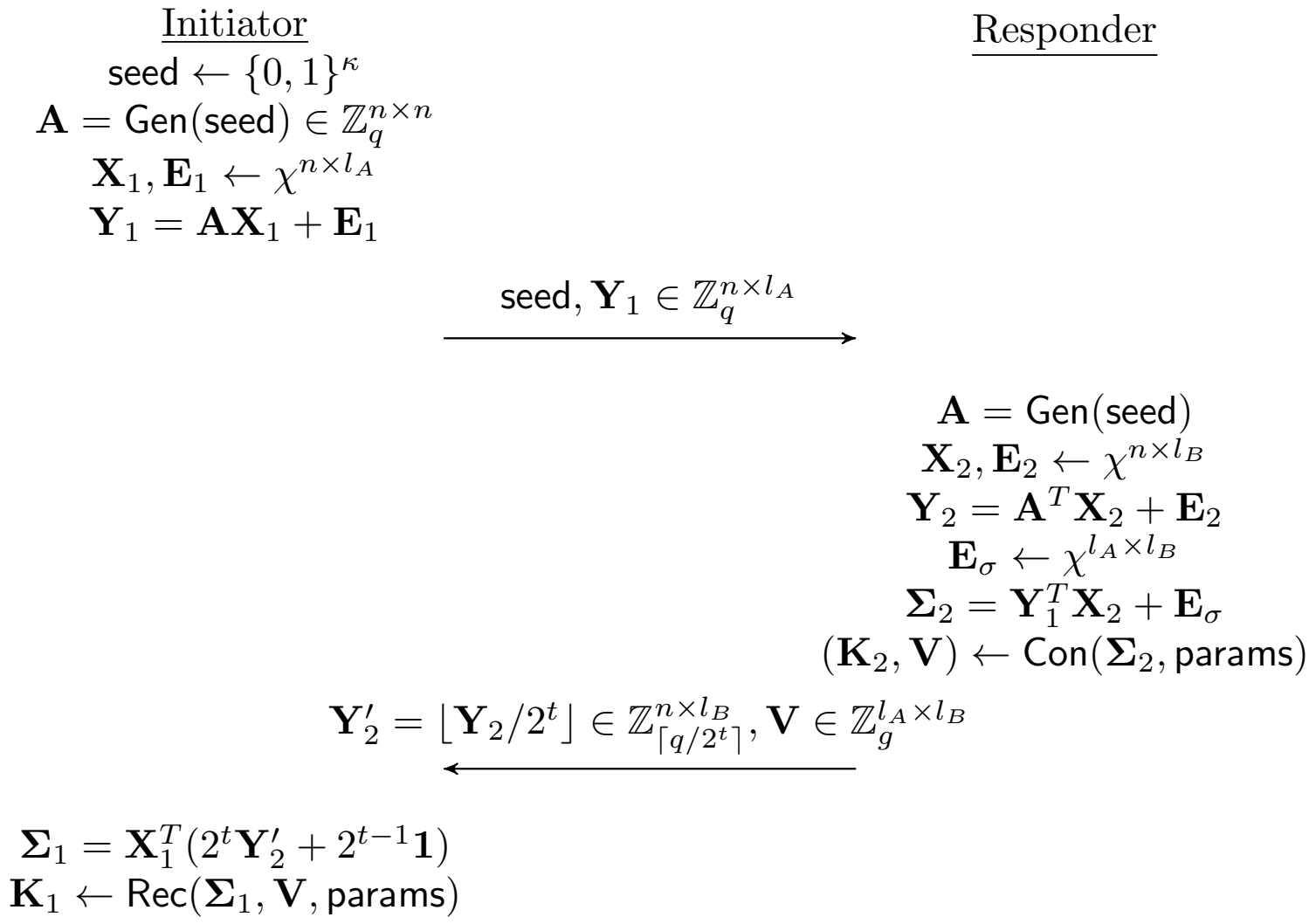


Figure: LWE-based KE from KC

Algorithm 5 Key consensus scheme in Frodo

procedure $\textsc{Con}(\sigma_1, \text{params})$ ▷ $\sigma_1 \in [0, q)$

  $v = \left\lfloor 2^{-\bar{B}+1}\sigma_1 \right\rfloor \bmod 2$

  $k_1 = \left\lfloor 2^{-\bar{B}}\sigma_1 \right\rceil \bmod 2^B$ **return** (k_1, v)

end procedure

procedure $\textsc{Rec}(\sigma_2, v, \text{params})$ ▷ $\sigma_2 \in [0, q)$

  find $x \in \mathbb{Z}_q$ closest to σ_2 s.t. $\left\lfloor 2^{-\bar{B}+1}x \right\rfloor \bmod 2 = v$

  $k_2 = \left\lfloor 2^{-\bar{B}}x \right\rceil \bmod 2^B$ **return** k_2

end procedure

Constraint: $4md < q$: not optimal.

Algorithm 6 OKCN simple

1: params : $q = 2^{\bar{q}}, g = 2^{\bar{g}}, m = 2^{\bar{m}}, d$, where $\bar{g} + \bar{m} = \bar{q}$
2: **procedure** Con(σ_1, params)
3: $k_1 = \left\lfloor \frac{\sigma_1}{g} \right\rfloor$
4: $v = \sigma_1 \bmod g$
return (k_1, v)
5: **end procedure**
6: **procedure** Rec(σ_2, v, params)
7: $k_2 = \left\lfloor \frac{\sigma_2 - v}{g} \right\rceil \bmod m$
return k_2
8: **end procedure**

Constraint: $2md < q$, optimal.

Discrete distributions and their Rényi divergences

dist.	bits	var.	probability of 0	±1	±2	±3	±4	±5	order	divergence
D_1	8	1.10	94	62	17	2			15.0	1.0015832
D_2	12	0.90	1646	992	216	17			75.0	1.0003146
D_3	12	1.66	1238	929	393	94	12	1	30.0	1.0002034
D_4	16	1.66	19794	14865	6292	1499	200	15	500.0	1.0000274

dist.	bits	var.	probability of 0	±1	±2	±3	±4	±5	order	divergence
D_5	16	1.30	22218	15490	5242	858	67	2	500.0	1.0000337

dist.	bits	var.	probability of 0	±1	±2	±3	±4	±5	±6	order	divergence
$\bar{D}_1$	8	1.25	88	61	20	3				25.0	1.0021674
$\bar{D}_2$	12	1.00	1570	990	248	24	1			40.0	1.0001925
$\bar{D}_3$	12	1.75	1206	919	406	104	15	1		100.0	1.0003011
$\bar{D}_4$	16	1.75	19304	14700	6490	1659	245	21	1	500.0	1.0000146

OKCN vs. Frodo

	q	n	l	m	g		d		dist.	error rates		bw. (kB)	$\lvert A\rvert$ (kB)	$\lvert K\rvert$
					OKCN	Frodo	OKCN	Frodo		OKCN	Frodo			
Challenge	2^{10}	334	8	2^1	2^9	2	255	127	D_1	$2^{-47.9}$	$2^{-14.9}$	6.75	139.45	64
Classical	2^{11}	554	8	2^2	2^9	2	255	127	D_2	$2^{-39.4}$	$2^{-11.5}$	12.26	422.01	128
Recommended	2^{14}	718	8	2^4	2^{10}	2	511	255	D_3	$2^{-37.9}$	$2^{-10.2}$	20.18	902.17	256
Paranoid	2^{14}	818	8	2^4	2^{10}	2	511	255	D_4	$2^{-32.6}$	$2^{-8.6}$	22.98	1170.97	256
Paranoid-512	2^{12}	700	16	2^2	2^{10}	2	511	255	$\bar{D}_4$	$2^{-33.6}$	$2^{-8.3}$	33.92	735.00	512

	q	n	l	m	g		d		dist.	error rates		bw. (kB)		$\lvert A\rvert$ (kB)	$\lvert K\rvert$
					OKCN	Frodo	OKCN	Frodo		OKCN	Frodo	OKCN	Frodo		
Challenge	2^{11}	352	8	2^1	2^2	2	383	255	$\bar{D}_1$	$2^{-80.1}$	$2^{-41.8}$	7.76	7.75	170.37	64
Classical	2^{12}	592	8	2^2	2^2	2	383	255	$\bar{D}_2$	$2^{-70.3}$	$2^{-36.2}$	14.22	14.22	525.70	128
Recommended	2^{15}	752	8	2^4	2^3	2	895	511	$\bar{D}_3$	$2^{-105.9}$	$2^{-38.9}$	22.58	22.57	1060.32	256
Paranoid	2^{15}	864	8	2^4	2^3	2	895	511	$\bar{D}_4$	$2^{-91.9}$	$2^{-33.8}$	25.94	25.93	1399.68	256

	q	n	l	m	g	t	d	dist.	err.	bw. (kB)	$\lvert A\rvert$ (kB)	$\lvert K\rvert$
Recommended	2^{14}	712	8	2^4	2^8	2	509	D_5	$2^{-39.0}$	18.58	887.15	256
Recommended-Enc	2^{14}	712	8	2^4	2^8	1	509	D_5	$2^{-52.3}$	19.29	887.15	256

OKCN-LWE/LWR vs. KC-Based Frodo

	$\lvert\mathbf{K}\rvert$	bw.(kB)	err.	pq-sec
OKCN-LWR	265	16.19	2^{-30}	130
OKCN-LWE	265	18.58	2^{-39}	134
Frodo	256	22.57	$2^{-38.9}$	130

Table: Comparison between OKCN-LWE/LWR and Frodo.

RLWE-Based Key Exchange from OKCN/AKCN

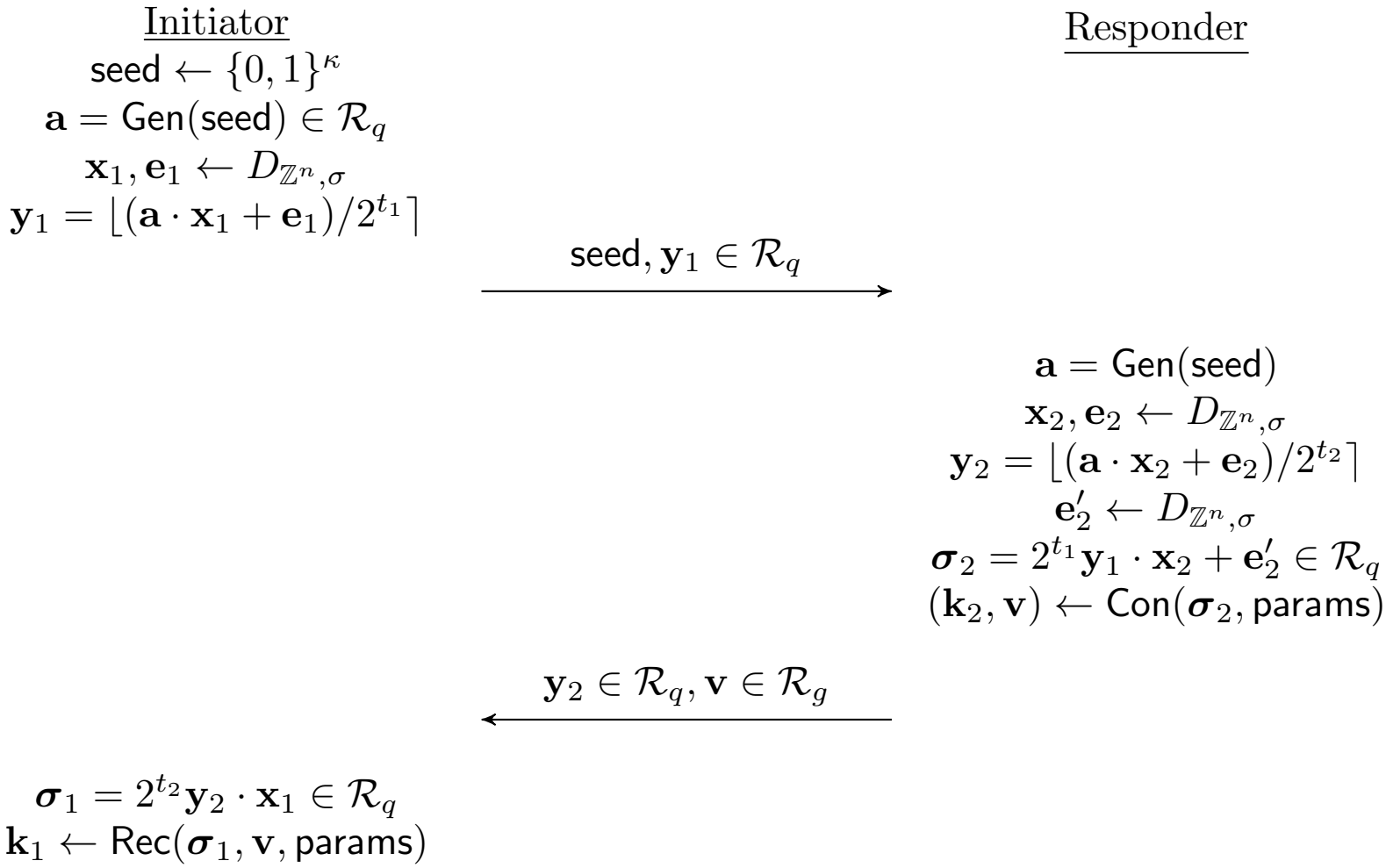


Figure: RLWE-based KE from KC/AKC

Algorithm 7 NewHope Consensus Mechanism

procedure $\textsc{Decode}(\mathbf{x} \in \mathbb{R}^4/\mathbb{Z}^4)$ ▷ Return a bit k such that $k\mathbf{g}$ is closest to $\mathbf{x} + \mathbb{Z}^4$

  $\mathbf{v} = \mathbf{x} - \lfloor \mathbf{x} \rceil$ **return** $k = 0$ if $\|\mathbf{v}\|_1 \leq 1$, and 1 otherwise

end procedure

$\mathrm{HelpRec}(\mathbf{x}, b) = \mathrm{CVP}_{\tilde{D}_4}\left(\frac{2^r}{q}(\mathbf{x} + b\mathbf{g})\right) \bmod 2^r$ ▷ b corresponds to the dbl trick[P14]

$rec\left(\mathbf{x} \in \mathbb{Z}_q^4, \mathbf{v} \in \mathbb{Z}_{2^r}^4\right) = \mathrm{Decode}\left(\frac{1}{q}\mathbf{x} - \frac{1}{2^r}\mathbf{B}\mathbf{v}\right)$

procedure $\textsc{Con}(\boldsymbol{\sigma}_1 \in \mathbb{Z}_q^4, \mathrm{params})$

  $b \leftarrow \{0, 1\}$

  $\mathbf{v} \leftarrow \mathrm{HelpRec}(\boldsymbol{\sigma}_1, b)$

  $k_1 \leftarrow rec(\boldsymbol{\sigma}_1, \mathbf{v})$

  return $(k_1, \mathbf{v})$

end procedure

procedure $\textsc{Rec}(\boldsymbol{\sigma}_2 \in \mathbb{Z}_q^4, \mathbf{v} \in \mathbb{Z}_{2^r}^4, \mathrm{params})$

  $k_2 \leftarrow rec(\boldsymbol{\sigma}_2, \mathbf{v})$

end procedure

Combining AKCN with Lattice Code in D_4

Algorithm 8 AKCN-4:1

1: **procedure** $\textsc{Con}(\boldsymbol{\sigma}_1 \in \mathbb{Z}_q^4, k_1 \in \{0,1\}, \text{params})$

2: $\quad \mathbf{v} = \text{CVP}_{\tilde{D}_4}(g(\boldsymbol{\sigma}_1 + k_1(q+1)\mathbf{g})/q) \bmod (g,g,g,2g)^T$

$\quad$ **return** $\mathbf{v}$

3: **end procedure**

4: **procedure** $\textsc{Rec}(\boldsymbol{\sigma}_2 \in \mathbb{Z}_q^4, \boldsymbol{v} \in \mathbb{Z}_g^3 \times \mathbb{Z}_{2g}, \text{params})$

5: $\quad \mathbf{x} = \mathbf{B}\mathbf{v}/g - \boldsymbol{\sigma}_2/q$

$\quad$ **return** $k_2 = 0$ if $\|\mathbf{x} - \lfloor \mathbf{x} \rceil\|_1 < 1$, 1 otherwise.

6: **end procedure**

- For simplicity, we focus on mathematical structure. By simple programming trick, it can be implemented with only integer arithmetic (without operating floating numbers).

AKC of NewHope: The Mathematical Structure

Algorithm 9 AKC Mechanism of NEWHOPE

1: **procedure** $\textsc{Con}(\boldsymbol{\sigma}_1 \in \mathbb{Z}_q^4, k_1 \in \{0,1\}, \text{params})$

2: $\quad \mathbf{v} = \text{CVP}_{\mathbb{Z}_4}(g(\boldsymbol{\sigma}_1 + k_1(q-1)\mathbf{g})/q) \bmod (g,g,g,g)^T$
return $\mathbf{v}$

3: **end procedure**

4: **procedure** $\textsc{Rec}(\boldsymbol{\sigma}_2 \in \mathbb{Z}_q^4, \boldsymbol{v} \in \mathbb{Z}_g^4, \text{params})$

5: $\quad \mathbf{x} = \lfloor q\mathbf{v}/g \rceil - \boldsymbol{\sigma}_2 - (q-1)\mathbf{g}$ **return** $k_2 = 1$ if $\|\mathbf{x}\|_1 < q$, 0 otherwise.

6: **end procedure**

- Note: This is the equivalent mathematical structure of NewHope-Simple (the actual proposal to NIST).
- In the actual proposal, some programming trick is used to explicitly avoid floating number operations.

AKCN4:1 vs NewHope-Simple

- AKCN4:1 is the *first* AKC-based variant of NewHope.
  - AKCN4:1 was publicly available from arXiv already since Nov 2016!, much earlier than NewHope-simple.
- When both AKCN4:1 and NewHope-Simple are presented in their mathematical structures, it is obvious that they are close.
- The difference is that: we did a bit further to reduce bandwidth expansion.
  - With the natural implementation, the bandwidth expansion of AKCN4:1 is 256 bits, while that of NewHope-Simple is 1024 bits.
- We want to do better...
  - New error correction mechanisms, joint with OKCN/AKCN.

Single-Error Correction Code

Algorithm 10 $\text{Encode}_{\mathcal{C}}(\mathbf{x} = (x_1, \ldots, x_{N_H-1}))$

1: $x_0 = \oplus_{i=1}^{N_H-1} x_i$
2: $\mathbf{p}^T = \mathbf{H}\mathbf{x}^T$
3: $\mathbf{c} = (x_0, \mathbf{x}, \mathbf{p})$ $\mathbf{c}$

Algorithm 11 $\text{Decode}_{\mathcal{C}}(x_0, \mathbf{x} = (x_1, \ldots, x_{N_H-1}), \mathbf{p})$

1: $p = \oplus_{i=0}^{N_H-1} x_i$
2: **if** $(\text{p} = 1)$
3: $\quad i = \overline{\mathbf{H}\mathbf{x}^T} \oplus \overline{\mathbf{p}}$
4: $x_i = x_i \oplus 1$ $\mathbf{x}$

AKCN Equipped with SEC

$$\underline{\text{Alice}} \qquad \approx \qquad \underline{\text{Bob}}$$

$$\boldsymbol{\sigma}_A \in \mathbb{Z}_q^{N_H+n_H} \qquad\qquad \boldsymbol{\sigma}_B \in \mathbb{Z}_q^{N_H+n_H}$$

$$\mathbf{k}_A = \mathsf{Encode}_{\mathcal{C}}(\mathbf{x})$$

$$\mathbf{v} \leftarrow \mathsf{Con}(\boldsymbol{\sigma}_A, \mathbf{k}_A, \mathsf{params})$$

$$\xrightarrow{\quad\mathbf{v}\quad}$$

$$\mathbf{k}_B \leftarrow \mathsf{Rec}(\boldsymbol{\sigma}_B, \mathbf{v}, \mathsf{params})$$

$$\mathbf{x}' = \mathsf{Decode}_{\mathcal{C}}(\mathbf{k}_B)$$

Lattice Code in E_8

- To further improve the bandwidth, error rate and post-quantum security simultaneously, we develop new lattice code in E_8.
- Note that packing in E_8 is optimal.
  - Packing in D_4 as in NewHope is not optimal.
  - Packing in Leech lattice is also optimal in 24 dimensions, but is more complicated.
- The construction and implementation is relatively complicated. Please refer to the paper for details.

Algorithm 12 AKCN-E8

1: **procedure** $\textsc{Con}(\sigma_1 \in \mathbb{Z}_q^8, \mathbf{k}_1 \in \mathbb{Z}_2^4)$

2: $\quad \mathbf{v} = \left\lfloor \frac{g}{q} \left(\boldsymbol{\sigma}_1 + \frac{q-1}{2} (\mathbf{k}_1 \mathbf{H} \bmod 2) \right) \right\rceil \bmod g$ **return** $\mathbf{v}$

3: **end procedure**

4: **procedure** $\mathrm{Rec}(\boldsymbol{\sigma}_2 \in \mathbb{Z}_q^8, \mathbf{v} \in \mathbb{Z}_g^8)$

5: $\quad \mathbf{k}_2 = \mathrm{Decode}_{\mathrm{E}_8} \left(\left\lfloor \frac{q}{g} \mathbf{v} \right\rceil - \boldsymbol{\sigma}_2 \right)$ **return** $\mathbf{k}_2$

6: **end procedure**

Algorithm 1 Decoding in E_8 and C

procedure $\mathsf{Decode}_{E_8}(\mathbf{x} \in \mathbb{Z}_q^8)$
  for $i = 0 \dots 7$ **do**
    $\mathsf{cost}_{i,0} = |x_i|_q^2$
    $\mathsf{cost}_{i,1} = |x_i + \frac{q-1}{2}|_q^2$
  end for
  $(\mathbf{k}^{00}, \mathsf{TotalCost}^{00}) \leftarrow \mathsf{Decode}_C^{00}(\mathsf{cost}_{i\in 0\dots 7, b\in\{0,1\}})$
  $(\mathbf{k}^{01}, \mathsf{TotalCost}^{01}) \leftarrow \mathsf{Decode}_C^{01}(\mathsf{cost}_{i\in 0\dots 7, b\in\{0,1\}})$
  if $\mathsf{TotalCost}^{00} < \mathsf{TotalCost}^{01}$ **then**
    $b = 0$
  else
    $b = 1$
  end if
  $(k_0, k_1, k_2, k_3) \leftarrow \mathbf{k}^{0b}$
  $\mathbf{k}_2 = (k_0, k_1 \oplus k_0, k_3, b)$
  return $\mathbf{k}_2$
end procedure
procedure $\mathsf{Decode}_C^{b_0 b_1}(\mathsf{cost}_{i\in 0\dots 7, b\in\{0,1\}} \in \mathbb{Z}^{8\times 2})$
  $min_d = +\infty$
  $min_i = 0$
  $\mathsf{TotalCost} = 0$
  for $j = 0 \dots 3$ **do**
    $c_0 \leftarrow \mathsf{cost}_{2j,b_0} + \mathsf{cost}_{2j+1,b_1}$
    $c_1 \leftarrow \mathsf{cost}_{2j,1-b_0} + \mathsf{cost}_{2j+1,1-b_1}$
    if $c_0 < c_1$ **then**
      $k_i \leftarrow 0$
    else
      $k_i \leftarrow 1$
    end if
    $\mathsf{TotalCost} \leftarrow \mathsf{TotalCost} + c_{k_i}$
    if $c_{1-k_i} - c_{k_i} < min_d$ **then**
      $min_d \leftarrow c_{1-k_i} - c_{k_i}$
      $min_i \leftarrow i$
    end if
  end for
  if $k_0 + k_1 + k_2 + k_3 \bmod 2 = 1$ **then**
    $k_{min_i} \leftarrow 1 - k_{min_i}$
    $\mathsf{TotalCost} \leftarrow \mathsf{TotalCost} + min_d$
  end if
  $\mathbf{k} = (k_0, k_1, k_2, k_3)$
  return $(\mathbf{k}, \mathsf{TotalCost})$
end procedure

OKCN/AKCN-RLWE vs. NewHope

	$\lvert\mathbf{K}\rvert$	bw.(B)	err.	pq-sec
OKCN-RLWE-SEC-1	765	3136	$2^{-68.4}$	250
OKCN-RLWE-SEC-2	765	3392	2^{-61}	258
NewHope	256	3872	2^{-61}	255
AKCN-RLWE-SEC-1	765	3264	$2^{-68.4}$	250
AKCN-RLWE-SEC-2	765	3520	2^{-61}	258
AKCN-RLWE-E8	512	3360	$2^{-63.3}$	262
NewHope-Simple	256	4000	2^{-61}	255

Table: Comparison between OKCN/AKCN-RLWE and NewHope. The actual NewHope proposal uses some smaller parameters, but with lowered security level.

On the desirability of OKCN/AKCN-SEC and OKCN/AKCN-E8

- OKCN/AKCN-SEC schemes are the simplest RLWE-based KE protocols *with error probability that can be viewed negligible in practice*, which are better suitable for hardware or software implementations.
- OKCN/AKCN-SEC and OKCN/AKCN-E8 are more versatile and flexible, allowing various trade-offs among performances and parameters.
- It is more desirable to have KE protocols that directly share or transport keys of larger size.
  - Shared key of size 256 bits can only provide 128 post-quantum security in reality. In this sense, 255-bit ps-sec of NewHope is overshot in reality.
  - Note that for NewHope to achieve 512-bit shared key, it needs a polynomial of 2048 degrees, which is significantly inefficient and less flexible.

A Note on Ring Choices

In the proposal, for RLWE-based protocols we recommended the popular power-of-two cyclotomic rings. In practice, we also suggest to use the Safe-Prime rings $\mathcal{R}_q = \mathbb{Z}_q[X]/(X^n + X^{n-1} + \cdots + 1)$ proposed by LIMA:

- Safe-Prime-1: Let m be a safe prime such that $m = 2m' + 1$, e denote the smallest integer satisfying $2^e > 2m$, q be a prime such that $q \equiv 1 \pmod{2^e \cdot m}$. Let $n = \varphi(m) = 2m'$, $\Phi_m(X) = X^{m-1} + X^{m-2} + \cdots + 1 = X^n + X^{n-1} + \cdots + 1$.
- Safe-Prime-2: Let m be a safe prime such that $m = n + 1$. Then $\varphi(m) = n$, and $\Phi_m(X) = X^n + X^{n-1} + \cdots + 1$.

MLWE-Based Key Exchange from OKCN/AKCN

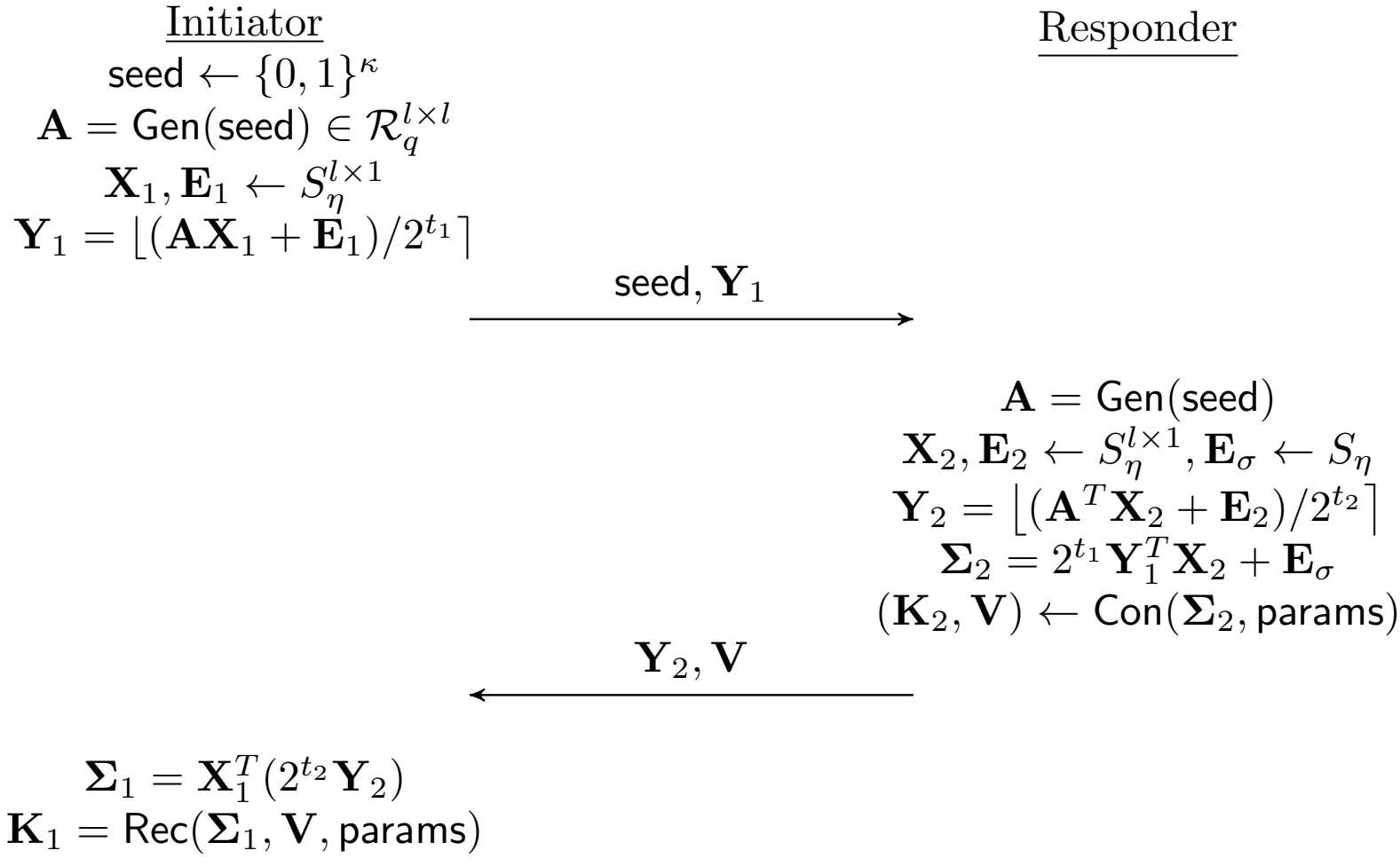


Figure: OKCN-MLWE: Just Kyber!

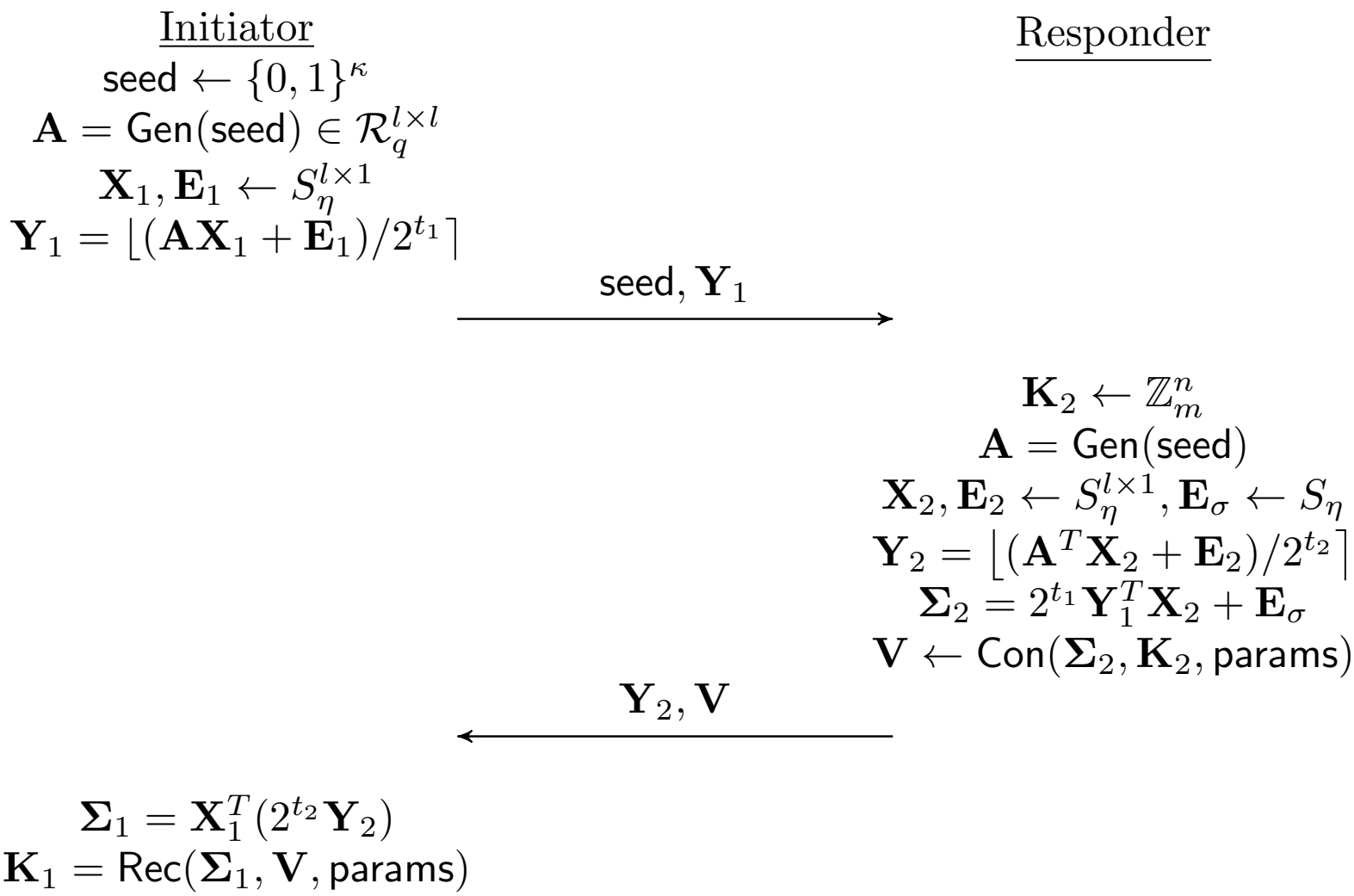


Figure: AKCN-MLWE: Just Kyber! Kyber only focus on AKCN-based, while ours is for both OKCN and AKCN

	$\|K\|$	n	q	η (η')	g	t	l	pq-sec (t-sec)	err	pk (B)	cipher (B)	bw. (B)
OKCN-MLWE-KE-light	256	256	7681	5 (13)	2^3	4	2	102 (116)	$2^{-36.2}$	608	704	1312
OKCN-MLWE-KE	256	256	7681	2 (10)	2^2	4	3	147 (183)	$2^{-50.1}$	896	960	1856
OKCN-MLWE-PKE-light	256	256	7681	5 (9)	2^3	3	2	102 (111)	$2^{-105.5}$	672	768	1440
OKCN-MLWE-PKE-1	256	256	7681	2 (10)	2^5	4	3	147 (183)	$2^{-80.3}$	896	1056	1952
OKCN-MLWE-PKE-2	256	256	7681	2 (6)	2^2	3	3	147 (171)	$2^{-166.4}$	992	1056	2048
AKCN-MLWE-PKE-light	256	256	7681	5 (9)	2^3	3	2	102 (111)	$2^{-105.5}$	672	800	1472
AKCN-MLWE-PKE-1	256	256	7681	2 (10)	2^6	4	3	147 (183)	$2^{-80.3}$	896	1088	1984
AKCN-MLWE-PKE-2	256	256	7681	2 (6)	2^3	3	3	147 (171)	$2^{-166.4}$	992	1088	2080
OKCN-MLWE-Alt1	256	256	7681	4	2^2	2	3	161	$2^{-142.7}$	1088	1152	2240
AKCN-MLWE-Alt1(Kyber)	256	256	7681	4	2^3	2	3	161	$2^{-142.7}$	1088	1184	2272
OKCN-MLWE-Alt2	256	256	7681	4	2^2	3	3	161	$2^{-71.9}$	992	1056	2048
OKCN-MLWE-Alt3	256	256	7681	4	2^4	3	3	161	2^{-109}	992	1120	2112

Table: Parameters for OKCN/AKCN-MLWE.

KC vs AKC

Most KEM proposals to NIST (from LWE and its variants) only provide AKC-based protocol, while our proposal focus on both KC-based and AKC-based. Moreover, our OKCN-based protocols have optimal performance.

- KC-based corresponds to Diffie-Hellman, while AKC-based to El Gamal (there are some asymmetric unfair issues).
- KC-based is more versatile (for both KE and PKE), and is more appropriate for incorporating into the existing DH-based standards like TLS, IKE, EMV,...
- On the same parameters (q, m, g) (which implies the same bandwidth), OKCN-based KE has lower error rate than AKCN-based KE. Or, on the same parameters (q, m, d) (which implies the same error rate), OKCN-based KE has smaller bandwidth than AKCN-based
    - This comparison is enabled by the proved upper-bounds on these parameters.

Shortcoming of Our Proposal

- More than 40 protocols can be instantiated from our general framework, but we were only able to implement a few of them.
- Some more implementations are available from `http://github.com/OKCN`
  - These implementations use codes from Frodo, NewHope, and are not appropriate for submission to NIST.
- We aim for implementing more of them in the future.

Conculusion

- Above all, with OKCN and AKCN, we provide a general framework for achieving key exchange and public-key encryption from lattice (specifically, LWE and its variants: LWR, RLWE, MLWE), *in a systemized and modular way*.
- We provide a set of practical yet powerful tools for dealing with noise: OKCN, AKCN, single-error correction code and lattice code in E_8,
    - which we suggest may play a basic role in the future design and analysis of cryptographic schemes from LWE and its variants.

- OKCN-based KE can be viewed as the equivalent of Diffie-Hellman in the lattice world.
- OKCN-based KE is more versatile, and is more appropriate for incorporating into the existing standards like IKE and TLS that are based on the SIGMA mechanism.

- For KE of 256-bit shared-key, OKCN/AKCN-MLWE is the most efficient. But for KE with shared-key of size 512 bits or more (which is necessary for ensuring 256-bit post-quantum security in reality), OKCN/AKCN-RLWE is the most efficient.
- Compared to RLWE and MLWE, the LWE and LWR problems have fewer algebraic structures that can be exploited by attacks. As noise sampling is relatively cumbersome for lattice-based cryptography, LWR-based KE may be more desirable in this sense.

References

- Zhengzhong Jin and Yunlei Zhao. Optimal Key Consensus in Presence of Noise. https://arxiv.org/abs/1611.06150
  - It was also introduced at Asia PQC Forum, Korean, in Nov 2016.
- Zhengzhong Jin and Yunlei Zhao. Optimal Key Consensus in Presence of Noise. https://eprint.iacr.org/2017/1058
- Leixiao Cheng and Yunlei Zhao. A Comparative Study of Lattice-Based KEM Proposals to NIST PQC Standardization. To be posted at ePrint shortly.

Thanks!

- Questions and comments please kindly send to

ylzhao@fudan.edu.cn