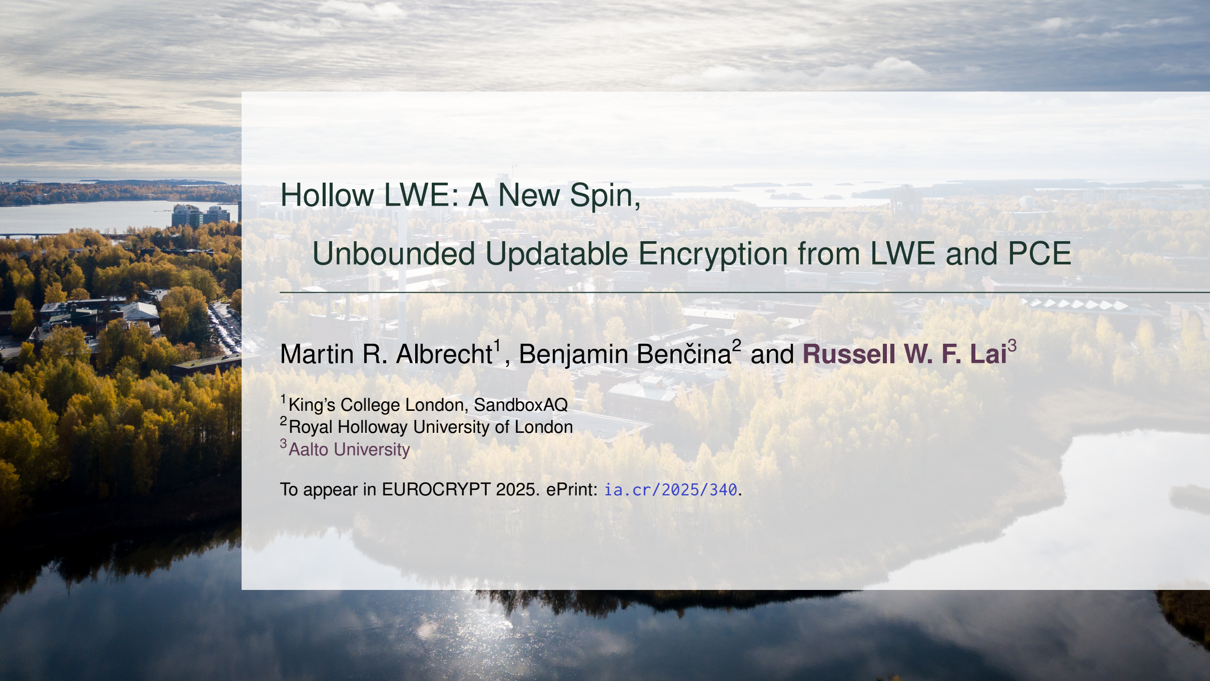


Hollow LWE: A New Spin, Unbounded Updatable Encryption from LWE and PCE

Martin R. Albrecht[1], Benjamin Benčina[2] and **Russell W. F. Lai**[3]

[1] King's College London, SandboxAQ
[2] Royal Holloway University of London
[3] Aalto University

To appear in EUROCRYPT 2025. ePrint: `ia.cr/2025/340`.

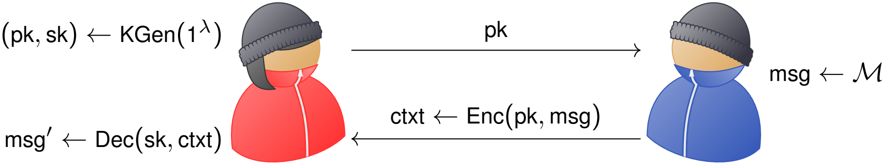
Unbounded Updatable Encryption from LWE and PCE

Overview:

- Updatable public-key encryption (UPKE)
- PKE from learning with errors (LWE)
- Prior key-update mechanism
- Lattice isomorphism problem (LIP)
- Linear codes and permutation code equivalence (PCE)
- PCE-based key-update mechanism
- Summary and open problems

Public-Key Encryption (PKE)

Encrypt and decrypt



$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$

pk

$\mathsf{msg} \leftarrow \mathcal{M}$

$\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{msg})$

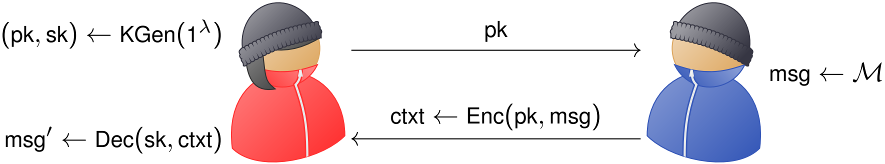
$\mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$

Properties:

- Decryption Correctness: $\mathsf{msg}' = \mathsf{msg}$.
- IND-CPA Security: $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_0)) \approx_c (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_1))$.
  - IND-CPA = indistinguishability under chosen plaintext attack

Public-Key Encryption (PKE)

Encrypt and decrypt



$(\mathsf{pk}, \mathsf{sk}) \leftarrow \mathsf{KGen}(1^\lambda)$

pk

$\mathsf{msg} \leftarrow \mathcal{M}$

$\mathsf{ctxt} \leftarrow \mathsf{Enc}(\mathsf{pk}, \mathsf{msg})$

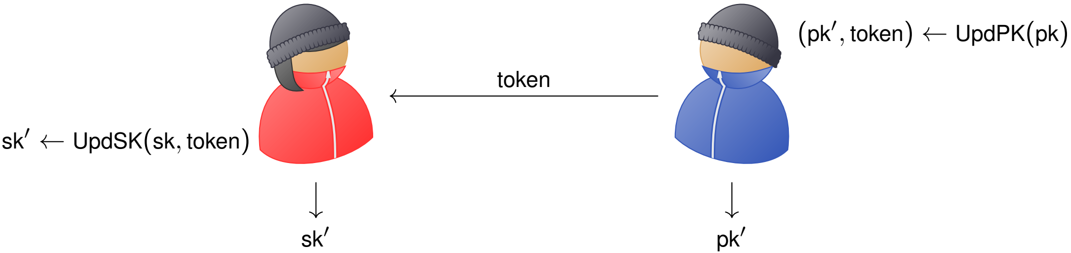
$\mathsf{msg}' \leftarrow \mathsf{Dec}(\mathsf{sk}, \mathsf{ctxt})$

Properties:

- Decryption Correctness: $\mathsf{msg}' = \mathsf{msg}$.
- IND-CPA Security: $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_0)) \approx_c (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_1))$.
    - IND-CPA = indistinguishability under chosen plaintext attack

Updatable Public-Key Encryption (UPKE)

Key update



$(\mathsf{pk}', \mathsf{token}) \leftarrow \mathsf{UpdPK}(\mathsf{pk})$

token

$\mathsf{sk}' \leftarrow \mathsf{UpdSK}(\mathsf{sk}, \mathsf{token})$

sk'

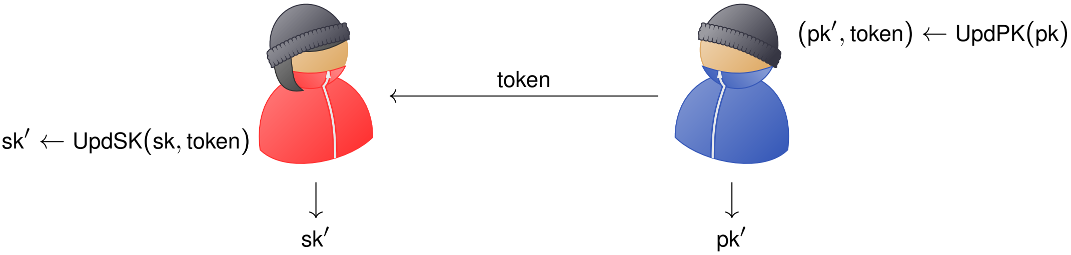
pk'

Additional property:

- Update correctness: Decryption correctness holds for updated keys $(\mathsf{pk}', \mathsf{sk}')$.
- IND-CR-CPA Security: $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_0), \mathsf{sk}') \approx_c (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_1), \mathsf{sk}')$, i.e. security of old ciphertexts still holds even if updated secret key is leaked. “Forward secrecy”.
  - IND-CR-CPA = indistinguishability under chosen randomness and chosen plaintext attack

Updatable Public-Key Encryption (UPKE)

Key update



$(\mathsf{pk}', \mathsf{token}) \leftarrow \mathsf{UpdPK}(\mathsf{pk})$

token

$\mathsf{sk}' \leftarrow \mathsf{UpdSK}(\mathsf{sk}, \mathsf{token})$

sk'

pk'

Additional property:

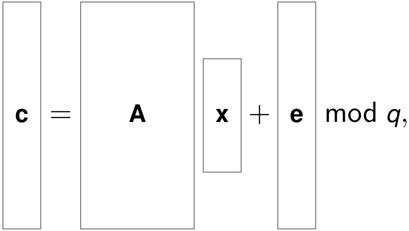
- Update correctness: Decryption correctness holds for updated keys $(\mathsf{pk}', \mathsf{sk}')$.
- IND-CR-CPA Security: $(\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_0), \mathsf{sk}') \approx_c (\mathsf{pk}, \mathsf{Enc}(\mathsf{pk}, \mathsf{msg}_1), \mathsf{sk}')$, i.e. security of old ciphertexts still holds even if updated secret key is leaked. “Forward secrecy”.
  - IND-CR-CPA = indistinguishability under chosen randomness and chosen plaintext attack

How to construct post-quantum updatable PKE?

Learning with errors (LWE)

Setting: $\mathbb{Z}_q := \mathbb{Z}/q\mathbb{Z}$, q prime, dimensions $n > k$.

LWE assumption: For $\mathbf{A} \leftarrow\!\$\ \mathbb{Z}_q^{n \times k}$, $\mathbf{x} \leftarrow\!\$\ \mathbb{Z}_q^{k}$, short noise $\mathbf{e} \leftarrow\!\$\ \chi^n$,



$$\mathbf{c} = \mathbf{A}\ \mathbf{x} + \mathbf{e} \bmod q,$$

it holds that

$$(\mathbf{A}, \mathbf{c}) \approx_c (\mathbf{A}, \$).$$

Typically, $\chi =$ discrete Gaussian distribution or bounded uniform distribution with $\|\chi\| \ll q$.

Dual-Regev encryption

Encrypt and decrypt

$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n \times k}$

$\mathrm{sk} := \mathbf{u} \xleftarrow{\$} \chi^n$

$\mathbf{v}^\top := \mathbf{u}^\top \cdot \mathbf{A} \bmod q$

$\xrightarrow{\mathrm{pk} = (\mathbf{A}, \mathbf{v})}$

$\xleftarrow{\mathrm{ctxt} = (\mathbf{c}, d)}$

$\mathrm{msg} \leftarrow \{0, 1\}$

$\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^k; \mathbf{e} \xleftarrow{\$} \chi^n; f \xleftarrow{\$} \chi$

$\mathbf{c} := \mathbf{A} \cdot \mathbf{x} + \mathbf{e} \bmod q$

$d = \mathbf{v}^\top \cdot \mathbf{x} + f + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathrm{msg} \bmod q$

$\downarrow$

$\mathrm{msg}' \leftarrow (|d - \mathbf{u}^\top \cdot \mathbf{c} \bmod q| < q/4)$

† Correctness: $\mathbf{u}$, $\mathbf{e}$, f are short enough $\Longrightarrow$ Dual-Regev has decryption correctness.

† Security: LWE assumption $\Longrightarrow$ Dual-Regev is IND-CPA secure.

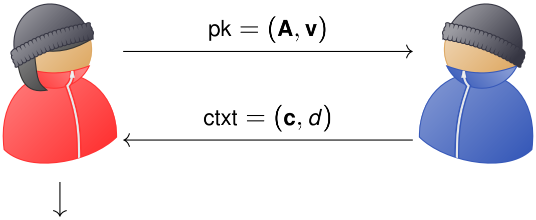
Dual-Regev encryption

Encrypt and decrypt

$\mathbf{A} \xleftarrow{\$} \mathbb{Z}_q^{n\times k}$

$\mathsf{sk} := \mathbf{u} \xleftarrow{\$} \chi^n$

$\mathbf{v}^\top := \mathbf{u}^\top \cdot \mathbf{A} \bmod q$



$\mathsf{pk} = (\mathbf{A}, \mathbf{v})$

$\mathsf{ctxt} = (\mathbf{c}, d)$

$\mathsf{msg} \leftarrow \{0,1\}$

$\mathbf{x} \xleftarrow{\$} \mathbb{Z}_q^k$; $\mathbf{e} \xleftarrow{\$} \chi^n$; $f \xleftarrow{\$} \chi$

$\mathbf{c} := \mathbf{A} \cdot \mathbf{x} + \mathbf{e} \bmod q$

$d = \mathbf{v}^\top \cdot \mathbf{x} + f + \left\lfloor \frac{q}{2} \right\rceil \cdot \mathsf{msg} \bmod q$

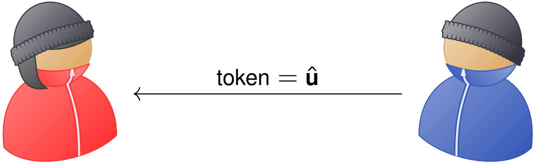
$\mathsf{msg}' \leftarrow \left(\left|d - \mathbf{u}^\top \cdot \mathbf{c} \bmod q\right| < q/4\right)$

- † Correctness: **u**, **e**, *f* are short enough $\Longrightarrow$ Dual-Regev has decryption correctness.
- † Security: LWE assumption $\Longrightarrow$ Dual-Regev is IND-CPA secure.

Prior key-update mechanism

Recall: $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$ and $\mathsf{sk} = \mathbf{u} \leftarrow\$ \chi^n$ with $\mathbf{v}^\intercal = \mathbf{u}^\intercal \cdot \mathbf{A} \bmod q$.

Key update



token $= \hat{\mathbf{u}}$

$\hat{\mathbf{u}} \leftarrow\$ \chi^n$

$\hat{\mathbf{v}}^\intercal := \hat{\mathbf{u}}^\intercal \cdot \mathbf{A} \bmod q$

$\mathsf{sk}' := \mathbf{u}' := \mathbf{u} + \hat{\mathbf{u}}$

$\mathsf{pk}' := (\mathbf{A}, \mathbf{v}')$ where $\mathbf{v}' := \mathbf{v} + \hat{\mathbf{v}} \bmod q$

Note: $\mathbf{A}$ does not change.

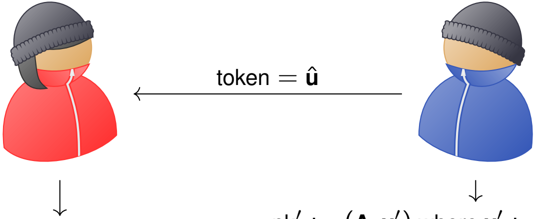
Issue:

- Updated secret key $\mathsf{sk}' = \mathbf{u}'$ has increased norm.
- To maintain correctness with many updates, either
  1. restrict number of updates to be fixed a-priori, or
  2. to supper arbitrary $\mathsf{poly}(\lambda)$ many updates, set super-polynomial modulus $q > \lambda^{\omega(1)} \implies$ large ctxt.

Prior key-update mechanism

Recall: $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$ and $\mathsf{sk} = \mathbf{u} \leftarrow\!\!\$\ \chi^n$ with $\mathbf{v}^\intercal = \mathbf{u}^\intercal \cdot \mathbf{A} \bmod q$.

Key update



$\hat{\mathbf{u}} \leftarrow\!\!\$\ \chi^n$

$\hat{\mathbf{v}}^\intercal := \hat{\mathbf{u}}^\intercal \cdot \mathbf{A} \bmod q$

token $= \hat{\mathbf{u}}$

$\mathsf{sk}' := \mathbf{u}' := \mathbf{u} + \hat{\mathbf{u}}$

$\mathsf{pk}' := (\mathbf{A}, \mathbf{v}')$ where $\mathbf{v}' := \mathbf{v} + \hat{\mathbf{v}} \bmod q$

Note: $\mathbf{A}$ does not change.

Issue:

- † Updated secret key $\mathsf{sk}' = \mathbf{u}'$ has increased norm.
- † To maintain correctness with many updates, either
  1. restrict number of updates to be fixed a-priori, or
  2. to supper arbitrary $\mathsf{poly}(\lambda)$ many updates, set super-polynomial modulus $q > \lambda^{\omega(1)} \implies$ large ctxt.

The question

How to support unbounded $\text{poly}(\lambda)$ *many key updates with a* $\text{poly}(\lambda)$ *size modulus* q*?*

q-ary Lattices

- A lattice $\Lambda \subseteq \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$, i.e.

  $$\Lambda = \mathbf{B} \cdot \mathbb{Z}^r$$

  for some basis $\mathbf{B} \in \mathbb{R}^{n \times r}$ where $r \leq n$.
- All bases $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{n \times r}$ are related by unimodular $\mathbf{U} \in \mathbb{Z}^{r \times r}$ via $\mathbf{B}' = \mathbf{B} \cdot \mathbf{U}$.
- Define the "primal lattice" a.k.a. the "Construction A" lattice of $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$:

  $$\Lambda_q(\mathbf{A}) = \mathbf{A} \cdot \mathbb{Z}^k + q \cdot \mathbb{Z}^n.$$

- Note that $\Lambda_q(\mathbf{A})$ is "q-ary", i.e.

  $$q \cdot \mathbb{Z}^n \subseteq \Lambda_q(\mathbf{A}) \subseteq \mathbb{Z}^n.$$

q-ary Lattices

- A lattice $\Lambda \subseteq \mathbb{R}^n$ is a discrete additive subgroup of $\mathbb{R}^n$, i.e.

$$\Lambda = \mathbf{B} \cdot \mathbb{Z}^r$$

  for some basis $\mathbf{B} \in \mathbb{R}^{n \times r}$ where $r \leq n$.

- All bases $\mathbf{B}, \mathbf{B}' \in \mathbb{R}^{n \times r}$ are related by unimodular $\mathbf{U} \in \mathbb{Z}^{r \times r}$ via $\mathbf{B}' = \mathbf{B} \cdot \mathbf{U}$.
- Define the “primal lattice” a.k.a. the “Construction A” lattice of $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$:

$$\Lambda_q(\mathbf{A}) = \mathbf{A} \cdot \mathbb{Z}^k + q \cdot \mathbb{Z}^n.$$

- Note that $\Lambda_q(\mathbf{A})$ is “q-ary”, i.e.

$$q \cdot \mathbb{Z}^n \subseteq \Lambda_q(\mathbf{A}) \subseteq \mathbb{Z}^n.$$

LWE and Dual-Regev: Lattice point of view

- LWE assumption: $(\mathbf{A}, \mathbf{A} \cdot \mathbf{x} + \mathbf{e} \bmod q) \approx_c (\mathbf{A}, \$)$.
- Lattice point of view:

$$(\mathbf{A}, \mathcal{U}(\Lambda_q(\mathbf{A})) + \chi^n) \approx_c (\mathbf{A}, \mathcal{U}(\mathbb{Z}_q^n)).$$

- A Dual-Regev secret key is a short vector

$$\mathbf{u} \in \Lambda_q^{\mathbf{v}}(\mathbf{A}) := \left\{\mathbf{w} \in \mathbb{Z}^n : \mathbf{w}^{\mathsf{T}} \cdot \mathbf{A} = \mathbf{v}^{\mathsf{T}} \bmod q\right\}$$

which is a random lattice coset of the “kernel lattice”

$$\Lambda_q^{\perp}(\mathbf{A}) := \left\{\mathbf{w} \in \mathbb{Z}^n : \mathbf{w}^{\mathsf{T}} \cdot \mathbf{A} = \mathbf{0}^{\mathsf{T}} \bmod q\right\}.$$

Lattice isomorphism problem (LIP), decision version

Lattice isomorphism

Lattices Λ, Λ' are isomorphic, denoted $\Lambda \sim \Lambda'$, if there exists orthogonal matrix $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, i.e.

$$\mathbf{O} \in \mathbb{R}^{n \times n} \text{ with } \mathbf{O}^{\mathsf{T}} \cdot \mathbf{O} = \mathbf{I}_n,$$

such that

$$\Lambda' = \mathbf{O} \cdot \Lambda,$$

i.e. Λ' can be obtained by rotating and reflecting Λ.
If $\mathbf{B}$ and $\mathbf{B}'$ are bases of Λ and Λ', then it means $\mathbf{B}' = \mathbf{O} \cdot \mathbf{B} \cdot \mathbf{U}$ for some unimodular $\mathbf{U} \in \mathbb{Z}^{r \times r}$.

Lattice isomorphism problem (LIP)

Given lattices $\Lambda, \Lambda_0, \Lambda_1 \subseteq \mathbb{R}^n$, decide if

$$\Lambda \sim \Lambda_0 \text{ or } \Lambda \sim \Lambda_1.$$

Lattice isomorphism problem (LIP), decision version

Lattice isomorphism

Lattices Λ, Λ' are isomorphic, denoted $\Lambda \sim \Lambda'$, if there exists orthogonal matrix $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$, i.e.

$$\mathbf{O} \in \mathbb{R}^{n\times n} \text{ with } \mathbf{O}^\top \cdot \mathbf{O} = \mathbf{I}_n,$$

such that

$$\Lambda' = \mathbf{O} \cdot \Lambda,$$

i.e. Λ' can be obtained by rotating and reflecting Λ.
If $\mathbf{B}$ and $\mathbf{B}'$ are bases of Λ and Λ', then it means $\mathbf{B}' = \mathbf{O} \cdot \mathbf{B} \cdot \mathbf{U}$ for some unimodular $\mathbf{U} \in \mathbb{Z}^{r\times r}$.

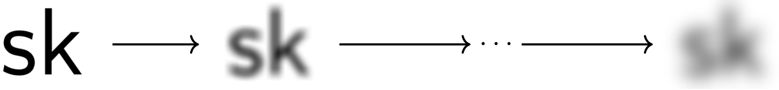
Lattice isomorphism problem (LIP)

Given lattices $\Lambda, \Lambda_0, \Lambda_1 \subseteq \mathbb{R}^n$, decide if

$$\Lambda \sim \Lambda_0 \text{ or } \Lambda \sim \Lambda_1.$$

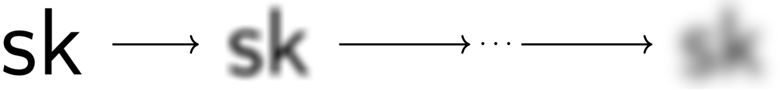
Rotate keys with LIP?

Prior method: Adding noise

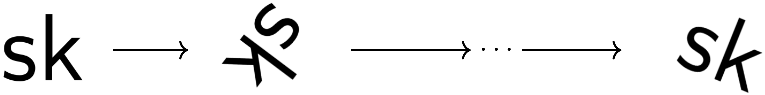


Rotate keys with LIP?

Prior method: Adding noise



New method: Rotating keys



Rotate keys with LIP?

The idea, more concretely

- † Rotate $\mathbf{A}$ to $\mathbf{A}' := \mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U} \bmod q$.
- † Rotate $\mathbf{u}$ to $\mathbf{u}' := \mathbf{O} \cdot \mathbf{u} \bmod q$.
- † Update $\mathbf{v}$ to $\mathbf{v}' := \mathbf{U}^\top \cdot \mathbf{v} \bmod q$.

Issue

Orthogonal matrices $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ are real-valued.
$\implies$ $\mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U}$ and $\mathbf{O} \cdot \mathbf{u}$ may not be integral.

Rotate keys with LIP?

The idea, more concretely

- Rotate $\mathbf{A}$ to $\mathbf{A}' := \mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U} \bmod q$.
- Rotate $\mathbf{u}$ to $\mathbf{u}' := \mathbf{O} \cdot \mathbf{u} \bmod q$.
- Update $\mathbf{v}$ to $\mathbf{v}' := \mathbf{U}^\mathsf{T} \cdot \mathbf{v} \bmod q$.

Issue

Orthogonal matrices $\mathbf{O} \in \mathcal{O}_n(\mathbb{R})$ are real-valued.
$\implies$ $\mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U}$ and $\mathbf{O} \cdot \mathbf{u}$ may not be integral.

Lattice automorphism of $\mathbb{Z}^n$

- The automorphism group $\mathrm{Aut}(\Lambda)$ of a lattice Λ is the group of all isomorphisms from Λ to itself.
- It is well-known that $\mathrm{Aut}(\mathbb{Z}^n) = \mathcal{O}_n(\mathbb{Z})$, i.e. the group of signed permutation matrices

$$\mathcal{O}_n(\mathbb{Z}) = \left\{\mathbf{D} \cdot \mathbf{P} \in \{-1, 0, 1\}^{n \times n} : \mathbf{D} \in \mathrm{diag}(\{\pm 1\}^n),\ \mathbf{P} \text{ permutation matrix}\right\}.$$

- Since

$$q \cdot \mathbb{Z}^n \subseteq \Lambda_q(\mathbf{A}) \subseteq \mathbb{Z}^n,$$

we have

$$q \cdot \mathbb{Z}^n \subseteq \mathbf{O} \cdot \Lambda_q(\mathbf{A}) \subseteq \mathbb{Z}^n,$$

i.e. rotating $\Lambda_q(\mathbf{A})$ by $\mathbf{O} \in \mathcal{O}_n(\mathbb{Z})$ gives another q-ary lattice.

Coding theory point of view

- † The “primal lattice” a.k.a. the “Construction A” lattice of $\mathbf{A} \in \mathbb{Z}_q^{n \times k}$

$$\Lambda_q(\mathbf{A}) = \mathbf{A} \cdot \mathbb{Z}^k + q \cdot \mathbb{Z}^n$$

  is isomorphic to the $\mathbb{Z}_q$-linear code $\mathcal{C} = \mathbf{A} \cdot \mathbb{Z}_q^k$ generated by $\mathbf{A}$.

- † The (signed) permutation code equivalence ((S)PCE) problem is to decide if two codes $\mathcal{C}$ and $\mathcal{C}'$ are equivalent by a (signed) permutation matrix, i.e. whether

$$\mathcal{C}' = \mathbf{O} \cdot \mathcal{C}$$

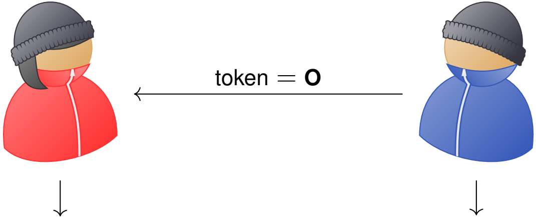
  for some (signed) permutation matrix $\mathbf{O} \in \mathcal{O}_n(\mathbb{Z})$.

- † SPCE is essentially LIP with Λ restricted to q-ary lattices and $\mathbf{O}$ restricted to signed permutations.

PCE-based key-update mechanism

Recall: $\mathsf{pk} = (\mathbf{A}, \mathbf{v})$ and $\mathsf{sk} = \mathbf{u} \leftarrow\$ \chi^n$ with $\mathbf{v}^\mathsf{T} = \mathbf{u}^\mathsf{T} \cdot \mathbf{A} \bmod q$.

Key update



$\mathbf{O} \leftarrow\$ \mathcal{O}_n(\mathbb{Z})$

$\mathbf{U} : \mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U} = \mathsf{NormalForm}(\mathbf{O} \cdot \mathbf{A}) \bmod q$

$\mathsf{token} = \mathbf{O}$

$\mathsf{sk}' := \mathbf{u}' := \mathbf{O} \cdot \mathbf{u}$

$\mathsf{pk}' := (\mathbf{A}', \mathbf{v}') := (\mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U}, \mathbf{U}^\mathsf{T} \cdot \mathbf{v}) \bmod q$

Update correctness:

$$(\mathbf{u}')^\mathsf{T} \cdot \mathbf{A}' = \mathbf{u}^\mathsf{T} \cdot \mathbf{O}^\mathsf{T} \cdot \mathbf{O} \cdot \mathbf{A} \cdot \mathbf{U} = \mathbf{u}^\mathsf{T} \cdot \mathbf{A} \cdot \mathbf{U} \bmod q = \mathbf{v}^\mathsf{T} \cdot \mathbf{U} = (\mathbf{v}')^\mathsf{T} \bmod q.$$

Caution

To make the idea provably secure from reasonable assumptions, we need to be cautious:

- † For the hardness of (S)PCE, the hull of the code $\mathcal{C} = \mathbf{A} \cdot \mathbb{Z}_q^k$ is important.
- † Hull $\mathcal{H}(\mathbf{A}) := \mathcal{C} \cap \mathcal{C}^\perp$, where $\mathcal{C}^\perp = \left\{\mathbf{b} \in \mathbb{Z}_q^n : \mathbf{b}^\mathsf{T} \cdot \mathcal{C} = \mathbf{0}^\mathsf{T}\right\}$.
- † Random $\mathbf{A}$ has trivial hull dimension, i.e. $\mathcal{H}(\mathbf{A}) = \emptyset$ or $h = \dim(\mathcal{H}(\mathbf{A})) = 0$, w.h.p.
- † Existing attacks against PCE run in time $O(q^h)$ or $O(n^h)$, i.e. efficient when h is small.

Solution:

1. Sample $\mathbf{A}$ such that $h = \dim(\mathcal{H}(\mathbf{A}))$ is sufficiently large. We call these "h-hollow matrices".
2. Prove that LWE w.r.t. h-hollow matrices is as hard as LWE w.r.t. random matrices (i.e. $h = 0$).
3. Prove that the leftover hash lemma holds for h-hollow matrices.
4. Prove the the UPKE is IND-CR-CPA secure under the h-hollow LWE assumption and the PCE assumption for h-hollow matrices (in the random oracle model).

Caution

To make the idea provably secure from reasonable assumptions, we need to be cautious:

- † For the hardness of (S)PCE, the hull of the code $\mathcal{C} = \mathbf{A} \cdot \mathbb{Z}_q^k$ is important.
- † Hull $\mathcal{H}(\mathbf{A}) := \mathcal{C} \cap \mathcal{C}^\perp$, where $\mathcal{C}^\perp = \left\{\mathbf{b} \in \mathbb{Z}_q^n : \mathbf{b}^\mathsf{T} \cdot \mathcal{C} = \mathbf{0}^\mathsf{T}\right\}$.
- † Random $\mathbf{A}$ has trivial hull dimension, i.e. $\mathcal{H}(\mathbf{A}) = \emptyset$ or $h = \dim(\mathcal{H}(\mathbf{A})) = 0$, w.h.p.
- † Existing attacks against PCE run in time $O(q^h)$ or $O(n^h)$, i.e. efficient when h is small.

Solution:

1. Sample $\mathbf{A}$ such that $h = \dim(\mathcal{H}(\mathbf{A}))$ is sufficiently large. We call these "*h*-hollow matrices".
2. Prove that LWE w.r.t. *h*-hollow matrices is as hard as LWE w.r.t. random matrices (i.e. $h = 0$).
3. Prove that the leftover hash lemma holds for *h*-hollow matrices.
4. Prove the the UPKE is IND-CR-CPA secure under the *h*-hollow LWE assumption and the PCE assumption for *h*-hollow matrices (in the random oracle model).

Summary and open problems

Summary

- New unbounded key-update mechanism for lattice-based cryptosystems
- Applied to PKE $\Longrightarrow$ Updatable PKE

Open Problems

- Application to other primitives? Other existing techniques compatible with *h*-hollow matrices?
- Ring/module setting for efficiency? Related to re-using the same rotation more than once.
- More choices of rotation?
  E.g. characterise all rotations from a *q*-ary lattice to another *q*-ary lattice?

Russell W. F. Lai

Aalto University, Finland

✉ russell.lai@aalto.fi
🌐 russell-lai.hk
🌐 research.cs.aalto.fi/crypto

Thank You!