The lattice isomorphism problem and its applications in cryptography

Gustavo de Castro Biage

Feb 2024

Summary

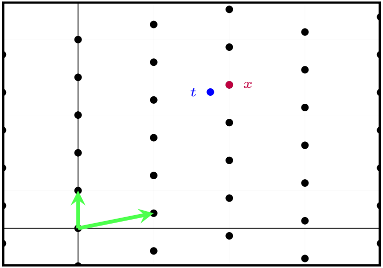
- Post-quantum cryptography.
- Lattice-based cryptography.
- Lattice isomorphism problem.
- Key-encapsulation mechanisms.

Post-quantum cryptography

- Classic cryptography is not secure against adversaries with access to quantum computers.
- Post-quantum cryptography results in cryptosystems that runs on classical computers and are secure against adversaries with access to quantum computers.
- We believe that lattice problems are hard to solve, even by quantum computers.

NIST's standardization process for post-quantum cryptography

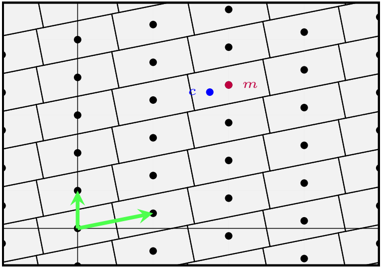
- Ever since the second round, all 12 lattice-based cryptosystems have LWE or NTRU as their security assumption.
- Both LWE and NTRU is related to finding close lattice vectors.

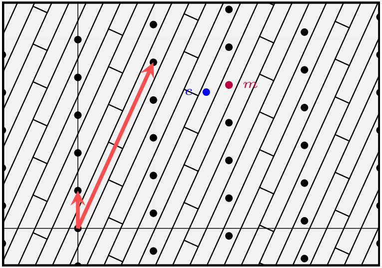


Definition (Bounded distance decoding (BDD))

Consider a basis $\boldsymbol{B}$, $\rho \in \mathbb{R}^+$, and an n-dimensional lattice $\mathcal{L}(\boldsymbol{B})$. Given a vector $\boldsymbol{t} \in \mathbb{R}^n$ such that $\boldsymbol{t} \notin \mathcal{L}(\boldsymbol{B})$, the bounded distance decoding consists of finding the unique vector $\boldsymbol{x} \in \mathcal{L}(\boldsymbol{B})$ such that $\|\boldsymbol{t} - \boldsymbol{x}\| \leq \rho$.

- The Goldreich-Goldwasser-Halevi is a nice introduction to the lattice concept of error-correction.





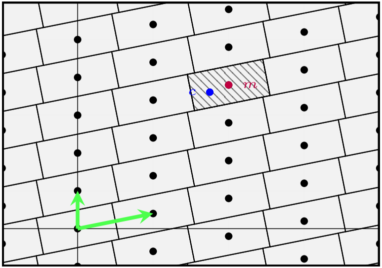
$\mathrm{sk} = \boldsymbol{B}$

$\mathrm{pk} = \boldsymbol{B}'$, s.t $\mathcal{L}(\boldsymbol{B}') = \mathcal{L}(\boldsymbol{B})$

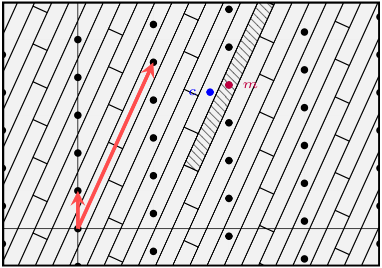
- A good basis contains smaller and more orthogonal vectors.
- A bad basis contains larger and less orthogonal vectors.
- The cipher text c is equal to the message m plus a random small error.
- Since the random error is small, decrypting the message is equal to solving BDD.

Motivation

Babai's nearest plane algorithm

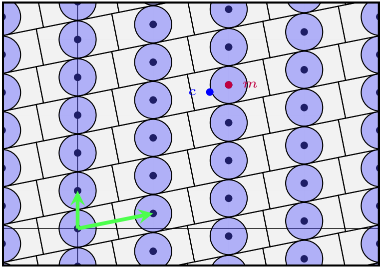


$\mathrm{sk} = \boldsymbol{B}$

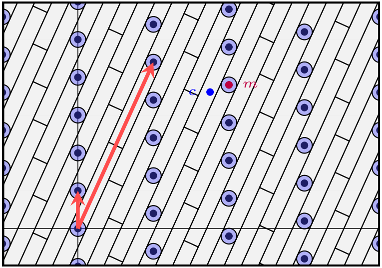


$\mathrm{pk} = \boldsymbol{B'}$, s.t $\mathcal{L}(\boldsymbol{B'}) = \mathcal{L}(\boldsymbol{B})$

- The Alg. could be seen as partitioning the space $\mathbb{R}^n$ with rectangles.
- Running the Alg. with a <u>bad</u> basis doesn't find the closest vector.
- BKZ is a basis reduction block-algorithm, having a blocksize β.
- A larger blocksize β results on reduced bases with <u>better quality</u>.



$\text{sk} = \boldsymbol{B}$

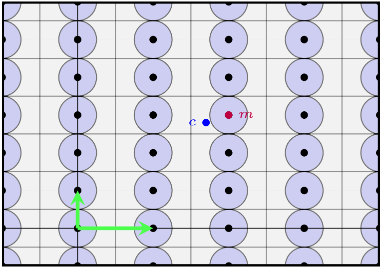


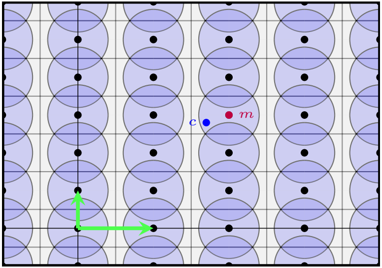
$\text{pk} = \boldsymbol{B'}$, s.t $\mathcal{L}(\boldsymbol{B'}) = \mathcal{L}(\boldsymbol{B})$

- We denote the length of the largest error sampled in the encryption as ρ.
- Let $\tilde{\boldsymbol{B}} = \text{GramSchmidt}(\boldsymbol{B})$ denote the orthogonalized basis.
- The error correcting length of Babai's algorithm is defined by $\min_{\tilde{\boldsymbol{b}} \in \tilde{\boldsymbol{B}}} \{\|\tilde{\boldsymbol{b}}\|/2\}$.

Motivation

When is decoding easy?





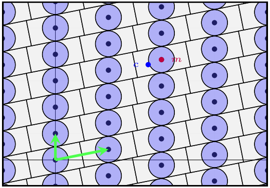
Definition (Decoding gap)

The decoding gap of a lattice $\mathcal{L}$ with decoding distance ρ is defined as $\mathrm{Gap}_\rho(\mathcal{L}) = \lambda_1/\rho$.

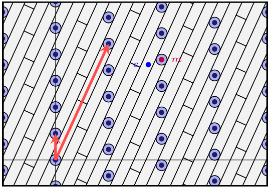
- The largest length where vectors can be uniquely decoded is $\lambda_1/2$.

Remark!

A larger decoding gap implies that the decoding problem can be solved by Babai's algorithm with a worst basis.



$\mathrm{sk} = \boldsymbol{B}$



$\mathrm{pk} = \boldsymbol{B}'$, s.t $\mathcal{L}(\boldsymbol{B}') = \mathcal{L}(\boldsymbol{B})$

Remark!

Consider that each one of these cryptosystems have a system parameter named decoding length ρ and that all sampled errors $\boldsymbol{e}$ have norm $\|\boldsymbol{e}\| \leq \rho$.

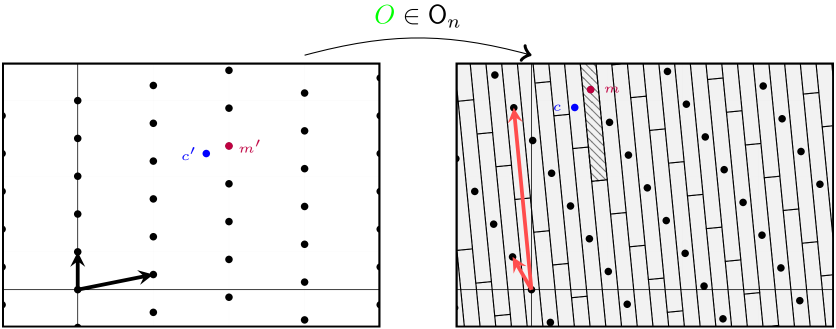
Heuristic

Most modern lattice-based cryptosystems are secure under the assumption that approximating solutions to BDD on an n-dimensional lattice $\mathcal{L}$ with decoding distance ρ and $\mathrm{Gap}_\rho(\mathcal{L}) > \omega(\sqrt{n})$ is hard; and are broken by block reduction algorithms with blocksize $\beta < n/2 + \mathrm{o}(n)$.

How to improve lattice-based cryptosystems?

R: Pick lattices with a small decoding gap . . . , and use lattice isomorphism.

Improving cryptosystems with isometries



$\mathrm{sk} = O \in \mathsf{O}_n$

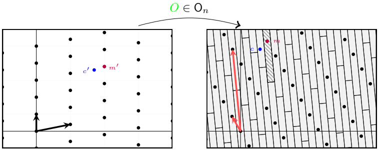
$\mathrm{pk} = \boldsymbol{B}'$, s.t $\mathcal{L}(\boldsymbol{B}') = O \cdot \mathcal{L}(\boldsymbol{B})$

Remark!

The adversary and the recipient no longer have access to the same lattice.

- Consider that we can efficiently decode erros in $\mathcal{L}(\boldsymbol{B})$.

Improving cryptosystems with isometries



sk $= O \in \mathsf{O}_n$

pk $= \boldsymbol{B}'$, s.t $\mathcal{L}(\boldsymbol{B}') = O \cdot \mathcal{L}(\boldsymbol{B})$

Definition (Heuristic)

Equivalent bases

Definition (Heuristic)

Lattice isomorphism

Definition (Heuristic)

Lattice isomorphism problem (LIP)

- Léo Ducas and van Woerden's frameworks:
  - Any lattice $\rightarrow$ identification scheme;
  - Decodable lattice $\rightarrow$ (~~encryption~~) key-encapsulation mechanism;
  - A gaussian samplable lattice $\rightarrow$ signature scheme.
- Open problems:
  - A concrete instance of a LIP-based key-encapsulation mechanism;
  - ~~A concrete instance of a LIP-based signature scheme~~;
  - ~~Investigate variations of LIP, such as Module-LIP~~;
  - ...

Overall objectives

- Review state-of-the art cryptoanalysis of LIP;
- present the current state of the art of modern cryptosystems having LIP as a foundation;
- propose a concrete instance of a LIP-based key-encapsulation mechanism.
- Investigate methods for improving our concrete KEM.

Key-encapsulation mechanism

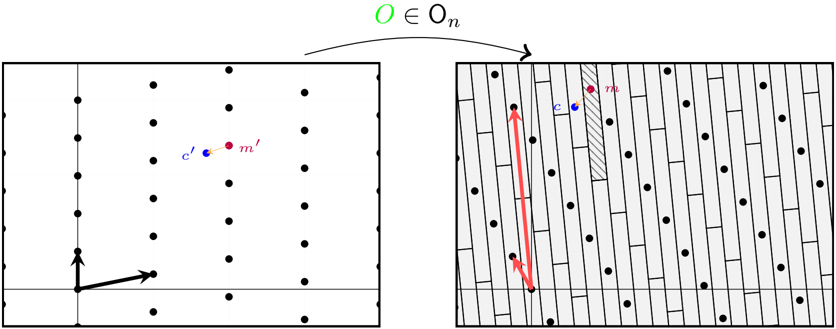
Léo Ducas and van Woerden's frameworks

Alice		Bob
$(pk, sk) := \textsc{KeyGen}()$		
	Send pk $\longrightarrow$	
		$(k, ch) := \textsc{Encapsulate}(pk)$
	$\longleftarrow$ Send ch	
$k := \textsc{Decapsulate}(sk, ch)$		

Definition (KEM)

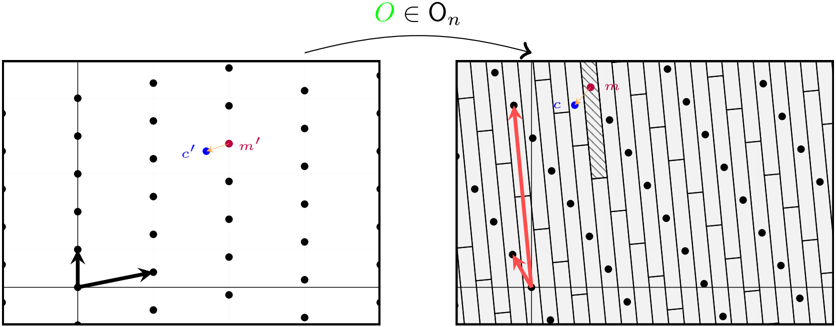
A KEM is defined as having three algorithms named:

- key generation (KeyGen),
- encapsulation (Encapsulate),
- decapsulation (Decapsulate).



Encapsulation

1. Choose an arbitrary m.
2. Sample a random error $\boldsymbol{e}$ such that $\|\boldsymbol{e}\| \leq \rho = \lambda_1/2$.
3. Compute the vector $\boldsymbol{c} = m + \boldsymbol{e}$.
4. Sample a random seed λ.
5. Sample a key $k = \mathcal{E}(\boldsymbol{e}, \lambda)$ using the randomness extractor $\mathcal{E}$.
6. Return $(k, (\boldsymbol{c}, \lambda))$ = shared secret $\times$ encapsulated key.



Decapsulation

1. Undo the rotation by computing $\boldsymbol{c}' = O^{-1} \cdot \boldsymbol{c}$.
2. Decode the error as $\boldsymbol{m}' = \mathrm{Decode}(\boldsymbol{c}')$.
3. Compute the erro vector $\boldsymbol{e}' = \boldsymbol{c}' - \boldsymbol{m}'$.
4. Rotate the error vector $\boldsymbol{e} = O \cdot \boldsymbol{e}'$.
5. Extract a random shared key $k = \mathcal{E}(\boldsymbol{e}, \lambda)$.
6. Return k.

Key-encapsulation mechanism

In practice, theory and practice are different

Gram matrix

The gram matrix o a lattice basis $\boldsymbol{B}$ is a positive-defined quadratic form $Q = \boldsymbol{B}^T \boldsymbol{B}$.

Gram matrix

The gram matrix o a lattice basis $\boldsymbol{B}$ is a positive-defined quadratic form $Q = \boldsymbol{B}^T\boldsymbol{B}$.

Gram matrix of a rotated basis

Let $O \in \mathrm{O}_n$ be an orthogonal transformation. Note that the Gram matrix of a basis $\boldsymbol{B}' = O \cdot \boldsymbol{B}$ is equal to $Q' = (\boldsymbol{B}')^T(\boldsymbol{B}') = (O \cdot \boldsymbol{B})^T(O \cdot \boldsymbol{B}) = \boldsymbol{B}^T O^T O \boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{B} = Q$.

Gram matrix

The gram matrix o a lattice basis $\boldsymbol{B}$ is a positive-defined quadratic form $Q = \boldsymbol{B}^T\boldsymbol{B}$.

Gram matrix of a rotated basis

Let $O \in \mathrm{O}_n$ be an orthogonal transformation. Note that the Gram matrix of a basis $\boldsymbol{B}' = O \cdot \boldsymbol{B}$ is equal to $Q' = (\boldsymbol{B}')^T(\boldsymbol{B}') = (O \cdot \boldsymbol{B})^T(O \cdot \boldsymbol{B}) = \boldsymbol{B}^T O^T O\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{B} = Q$.

Lattice isomorphism

Let $\boldsymbol{B}$ and $\boldsymbol{B}'$ be two bases having Gram matrices, respectivelly, Q and Q'. The lattice $\mathcal{L}(\boldsymbol{B}) \cong \mathcal{L}(\boldsymbol{B}')$ if there exist a uniform matrix $U \in \mathrm{U}_n$ such that $Q' = U^T Q U$.

Key-encapsulation mechanism

In practice, theory and practice are different

Gram matrix

The gram matrix o a lattice basis $\boldsymbol{B}$ is a positive-defined quadratic form $Q = \boldsymbol{B}^T\boldsymbol{B}$.

Gram matrix of a rotated basis

Let $O \in \mathrm{O}_n$ be an orthogonal transformation. Note that the Gram matrix of a basis $\boldsymbol{B}' = O \cdot \boldsymbol{B}$ is equal to $Q' = (\boldsymbol{B}')^T(\boldsymbol{B}') = (O \cdot \boldsymbol{B})^T(O \cdot \boldsymbol{B}) = \boldsymbol{B}^T O^T O\boldsymbol{B} = \boldsymbol{B}^T\boldsymbol{B} = Q$.

Lattice isomorphism

Let $\boldsymbol{B}$ and $\boldsymbol{B}'$ be two bases having Gram matrices, respectivelly, Q and Q'. The lattice $\mathcal{L}(\boldsymbol{B}) \cong \mathcal{L}(\boldsymbol{B}')$ if there exist a uniform matrix $U \in \mathrm{U}_n$ such that $Q' = U^T Q U$.

Remark!

- In practice we use quadratic forms.
- We can operate on the lattice vectors using quadratic forms and the integer coefficients. Consider $\boldsymbol{v} = \boldsymbol{B}\boldsymbol{x}$ and $\boldsymbol{u} = \boldsymbol{B}\boldsymbol{y}$, where $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{Z}^r$.
  1. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle_Q = \boldsymbol{x}^T Q \boldsymbol{y} = \boldsymbol{x}^T \boldsymbol{B}^T \boldsymbol{B}\boldsymbol{y} = (\boldsymbol{B}\boldsymbol{x})^T \boldsymbol{B}\boldsymbol{y} = \boldsymbol{v}^T\boldsymbol{u} = \langle \boldsymbol{v}, \boldsymbol{u} \rangle_2$.
  2. $\|\boldsymbol{x}\|_Q = \sqrt{\boldsymbol{x}^T Q \boldsymbol{x}} = \sqrt{\langle \boldsymbol{v}, \boldsymbol{v} \rangle} = \|\boldsymbol{v}\|_2$.

- It is common among lattice-based KEMs to use SHA3-256 or SHA-256 as a randomness extractor.
  - e.g. FrodoKEM, Crystal-Kyber.
  - First encode the vector in binary, and then use SHA3-256, ...

Example

Consider the basis $\boldsymbol{B} = [(11, 2), (-5, 1)]$ having Gram matrix Q such that $\mathcal{L}(\boldsymbol{B}) = \mathbb{Z}^2$. Let $\boldsymbol{x} = (5, 11)$, $\boldsymbol{y} = (10, 8)$ and $\boldsymbol{z} = (2, 40)$. If we encode $\boldsymbol{y}$ and $\boldsymbol{z}$ naively, the encoding is clearly not unique. Observe that

- $\boldsymbol{y} = (10, 8)$ would be encoded as $\overbrace{1010}^{10}\overbrace{1000}^{8}$,
- $\boldsymbol{z} = (2, 40)$ would be encoded as $\overbrace{\ 10\ }^{2}\overbrace{101000}^{40}$;

and $\|\boldsymbol{x}\|_Q = \|\boldsymbol{B}\boldsymbol{x}\|_2 = 1$ is very different from the norm $\|\boldsymbol{x}\|_2 \geq 12$.

- Build a randomness-extractor for the real coefficients (because we use quadratic forms).
  1. Adapts Ajtai's universal hash family to take as input small vectors considering the norm in respect to the quadratic forms.
  2. Build a randomness extractor from the unversal hash family using the Leftover hashing lemma.
- Handpick a decodable lattice having a small decoding gap.
- Build a lattice pair using the decodable lattice (IND-CPA).
- Then, ...

  choose parameters so that all theorems are respected!

[0]DOI: 10.1109/ACCESS.2024.3358670

Spoiler alert!

Tabela: Suggested parameter sets for our concrete KEM based on LIP.

Parameter set	$\mathcal{L}$	dim	g	$\sim s$	q	$\hat{q}$	$\hat{m}$	β
OurBW256g2	BW_{256}	512	2	51	3430	880729	12	$54880\sqrt{2}$
OurBW512g2	BW_{512}	1024	2	75	7479	1431655751	16	239328
OurBW256g4	BW_{256}	512	4	85	2858	880729	12	$91456\sqrt{2}$
OurBW512g4	BW_{512}	1024	4	125	6233	1431655751	16	398912
OurBW256g8	BW_{256}	512	8	152	2556	880729	12	$163584\sqrt{2}$
OurBW512g8	BW_{512}	1024	8	225	5610	1431655751	16	718080
OurBW256g12	BW_{256}	512	12	219	2455	880729	12	$235680\sqrt{2}$
OurBW512g12	BW_{512}	1024	12	325	5402	1431655751	16	1037184

Spoiler alert!

Tabela: Suggested parameter sets for our concrete KEM based on LIP.

Parameter set	$\mathcal{L}$	dim	g	$\sim s$	q	$\hat{q}$	$\hat{m}$	β
OurBW256g2	BW_{256}	512	2	51	3430	880729	12	$54880\sqrt{2}$
OurBW512g2	BW_{512}	1024	2	75	7479	1431655751	16	239328
OurBW256g4	BW_{256}	512	4	85	2858	880729	12	$91456\sqrt{2}$
OurBW512g4	BW_{512}	1024	4	125	6233	1431655751	16	398912
OurBW256g8	BW_{256}	512	8	152	2556	880729	12	$163584\sqrt{2}$
OurBW512g8	BW_{512}	1024	8	225	5610	1431655751	16	718080
OurBW256g12	BW_{256}	512	12	219	2455	880729	12	$235680\sqrt{2}$
OurBW512g12	BW_{512}	1024	12	325	5402	1431655751	16	1037184

Important question

Is this secure?

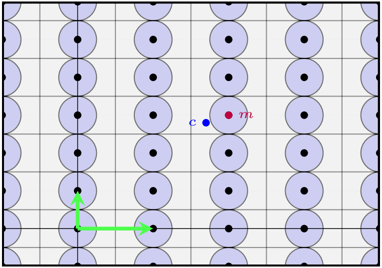
Definition (IND-CPA security)

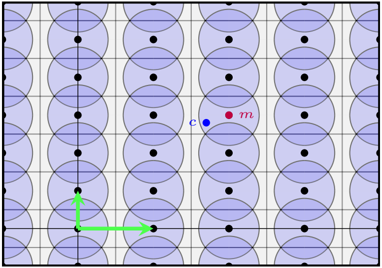
It is a security property common on encryption schemes, and key-encapsulation mechanisms. It is the concept that the advesary cannot distinguish a pair of ciphertexts given the message. On KEMS, the adversary is unable to distinguish a pair of shared secrets, given the encapsulated key.

IND-CPA Challenge

1. Let $(O, \boldsymbol{B}') := \text{KeyGen}(\boldsymbol{B})$.
2. Let $(k_0, (\boldsymbol{c}, \lambda)) := \text{Encaps}(\boldsymbol{B}')$.
3. Let $k_1 \xleftarrow{\$} \{0, 1\}^l$.
4. Sample $b \xleftarrow{\$} \{0, 1\}$.
5. Return to the adversary $(\boldsymbol{B}', (\boldsymbol{c}, \lambda), k_b)$.

- The adversary wins the challenge if he can guess the value of b.



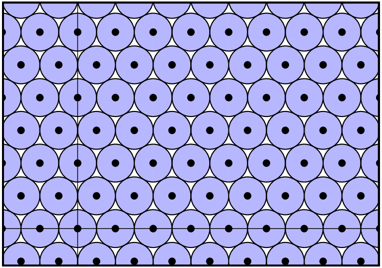


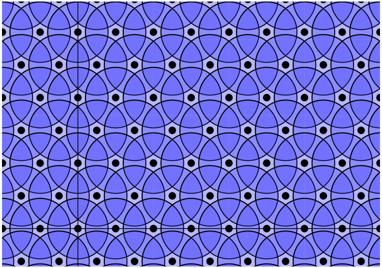
Overall idea

The idea is that the IND-CPA security challenge cannot be won when given a different lattice that is dense and has a much smaller decoding radius.

IND-CPA security

Why dense lattices?





Overall idea

The idea is that the IND-CPA security challenge cannot be won when given a different lattice that is dense and has a much smaller decoding radius.

IND-CPA Challenge

1. Let $(O, \boldsymbol{B}') := \mathrm{KeyGen}(\boldsymbol{B}_{\mathrm{dense}})$ (we replaced $\boldsymbol{B}$).
2. Let $(k_0, (\boldsymbol{c}, \lambda)) := \mathrm{Encaps}(\boldsymbol{B}')$.
3. Let $k_1 \xleftarrow{\$} \{0,1\}^l$.
4. Sample $b \xleftarrow{\$} \{0,1\}$.
5. Return to the adversary $(\boldsymbol{B}', (\boldsymbol{c}, \lambda), k_b)$.

- The adversary wins the challenge if he can guess the value of b.

Δ lattice isomorphism problem

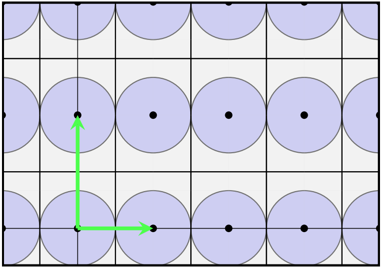
Let $\boldsymbol{B}_0$ and $\boldsymbol{B}_1$ be two basis. Consider a random secret $b \in \{0,1\}$ and a random public lattice $\mathcal{L} \cong \mathcal{L}(\boldsymbol{B}_b)$. The ΔLIP consists of finding b.

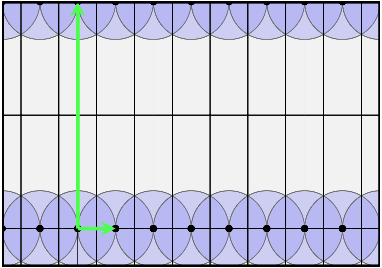
Remark!

Theorem foundation: The KEM is IND-CPA secure under the assumption that solving $\Delta\mathrm{LIP}(\boldsymbol{B}, \boldsymbol{B}_{\mathrm{dense}})$ is hard.

IND-CPA security

Minimal example





Example

Lattice pair example using $\mathcal{L}(\boldsymbol{B}_{dense}) = \mathbb{Z}^1$, and $g = 2$.

- $\mathcal{L}_S := g \cdot \mathcal{L}(\boldsymbol{B}_{dense}) \oplus (g+1) \cdot \mathcal{L}(\boldsymbol{B}_{dense}) = 2 \cdot \mathbb{Z}^1 \oplus 3 \cdot \mathbb{Z}^1$
- $\mathcal{L}_Q := \mathcal{L}(\boldsymbol{B}_{dense}) \oplus g(g+1) \cdot \mathcal{L}(\boldsymbol{B}_{dense}) = \mathbb{Z}^1 \oplus 6 \cdot \mathbb{Z}^1$

Note that

- $\lambda_1(\mathcal{L}_S) = g \cdot \lambda_1(\mathcal{L}(\boldsymbol{B}_{dense}))$,
- and $\lambda_1(\mathcal{L}_Q) = \lambda_1(\mathcal{L}(\boldsymbol{B}_{dense}))$.

Parameters!

Parameter set	$\mathcal{L}$	dim	g	$\sim s$	q	$\hat{q}$	$\hat{m}$	β
OurBW256g2	BW_{256}	512	2	51	3430	880729	12	$54880\sqrt{2}$
OurBW512g2	BW_{512}	1024	2	75	7479	1431655751	16	239328
OurBW256g4	BW_{256}	512	4	85	2858	880729	12	$91456\sqrt{2}$
OurBW512g4	BW_{512}	1024	4	125	6233	1431655751	16	398912
OurBW256g8	BW_{256}	512	8	152	2556	880729	12	$163584\sqrt{2}$
OurBW512g8	BW_{512}	1024	8	225	5610	1431655751	16	718080
OurBW256g12	BW_{256}	512	12	219	2455	880729	12	$235680\sqrt{2}$
OurBW512g12	BW_{512}	1024	12	325	5402	1431655751	16	1037184

Remark!

Important answer: The KEM is IND-CPA secure under the assumption that solving ΔLIP is hard.

Our work in perspective

Key sizes (IND-CPA)

IND-CPA KEM						
Parameter set	$\lvert pk \rvert$	$\lvert sk \rvert$	$\lvert ch \rvert$	$\lvert ss \rvert$	β	λ
OurBW256g2	439.25 KB	384 KB	781.88 B	240 b	352	2^{111}
OurBW512g2	1.91 MB	1.62 MB	1.65 KB	496 b	700	2^{213}
OurBW256g4	463.18 KB	384 KB	778.62 B	240 b	265	2^{85}
OurBW512g4	2.00 MB	1.62 MB	1.65 KB	496 b	557	2^{171}
OurBW256g8	489.93 KB	416 KB	776.25 B	240 b	201	2^{66}
OurBW512g8	2.11 MB	1.75 MB	1.64 KB	496 b	449	2^{139}
OurBW256g12	506.89 KB	448 KB	775.12 B	240 b	173	2^{57}
OurBW512g12	2.17 MB	1.88 MB	1.64 KB	496 b	398	2^{124}

Our work in perspective

Key sizes (IND-CCA2)

IND-CCA2 KEM

Parameter set	|pk|	|sk|	|ch|	|ss|	β	
OurBW256g2	439.25 KB	384.01 KB	797.88 B	111 b	352	2^{111}
OurBW512g2	1.91 MB	1.63 MB	1.68 KB	213 b	700	2^{213}
OurBW256g4	463.18 KB	384.01 KB	794.62 B	85 b	265	2^{85}
OurBW512g4	2.00 MB	1.63 MB	1.68 KB	171 b	557	2^{171}
OurBW256g8	489.93 KB	416.01 KB	792.25 B	66 b	201	2^{66}
OurBW512g8	2.11 MB	1.75 MB	1.67 KB	139 b	449	2^{139}
OurBW256g12	506.89 KB	448.01 KB	791.12 B	57 b	173	2^{57}
OurBW512g12	2.17 MB	1.88 MB	1.66 KB	124 b	398	2^{124}

Remark!

We converted the IND-CPA secure KEM to an IND-CCA2 secure KEM using well-known methods described in the literature, including the transformation of Fujisaki-Okamoto.

Our work in perspective

Modern lattice-based KEMs

Parameter set	$\lvert pk \rvert$	λ
lotus-params128	658.95 KB	2^{196}
lotus-params192	1.00 MB	2^{199}
Frodo-640	9.39 KB	2^{149}
Frodo-976	15.26 KB	2^{214}
NewHope512	928 B	2^{112}
Kyber512	800 B	2^{118}
Kyber768	1.15 KB	2^{182}
KEM_CATEGORY1_N536	1.07 MB	2^{133}
KEM_CATEGORY3_N816	1.64 MB	2^{193}
Titanium-CCA-Std128	14.37 KB	2^{146}
Titanium-CCA-Med160	16.06 KB	2^{192}

Our work in perspective

Conclusion

Remark!

It is easier to study specific optimizations once we have a first concrete instance.

Overall objectives

- propose a concrete instance of a LIP-based key-encapsulation mechanism ✓;

Our work in perspective
Conclusion

Remark!

It is easier to study specific optimizations once we have a first concrete instance.

Overall objectives

- Review state-of-the art cryptoanalysis of LIP $\cancel{\checkmark}$;
- present the current state of the art of modern cryptosystems having LIP as a foundation $\cancel{\checkmark}$;
- propose a concrete instance of a LIP-based key-encapsulation mechanism $\checkmark$;

Our work in perspective

Conclusion

Remark!

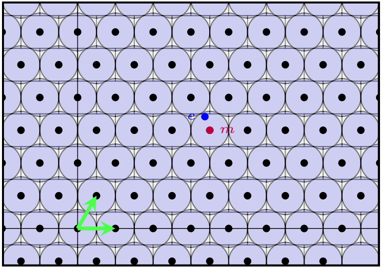
It is easier to study specific optimizations once we have a first concrete instance.

Overall objectives

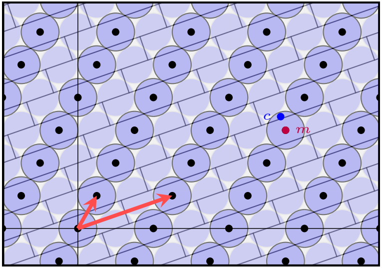
- Review state-of-the art cryptoanalysis of LIP $\bcancel{\checkmark}$;
- present the current state of the art of modern cryptosystems having LIP as a foundation $\bcancel{\checkmark}$;
- propose a concrete instance of a LIP-based key-encapsulation mechanism $\checkmark$;
- Investigate methods for improving our concrete KEM.

Open questions

Sublattice isomorphism



$\mathrm{sk} = \boldsymbol{B}$



$\mathrm{pk} = \boldsymbol{H}$, s.t $\mathcal{L}(\boldsymbol{H}) \subset \mathcal{L}(\boldsymbol{B})$

- In our example, the rank of $\mathcal{L}(\boldsymbol{H})$ and $\mathcal{L}(\boldsymbol{B})$ are the same.
- The same idea could be applied to the LIP Framework.
- However, the lattices are no longer isomorphic. For some $\mathcal{L}$,

$$\mathcal{L}(\boldsymbol{B}') \subset \mathcal{L} \cong \mathcal{L}(\boldsymbol{B}).$$

- Similar ideas exists in code-based cryptography, e.g subcode equivalence problem.