EuroTechPostdoc2 Programme
Excellence in Science and Technology

FuLeeca

Violetta Weger

2nd Oxford Post-Quantum Cryptography Summit 2023

September 5, 2023

The Rise and Fall of FuLeeca

Violetta Weger

2nd Oxford Post-Quantum Cryptography Summit 2023

September 5, 2023

Outline

FuLeeca: Hash & Sign scheme based on:

- Lee metric
- Quasi-cyclic codes
- Sign matching

Outline

FuLeeca: Hash & Sign scheme based on:

- Lee metric
- Quasi-cyclic codes
- Sign matching

→ vulnerable to lattice-based attacks

Outline

FuLeeca: Hash & Sign scheme based on:

- Lee metric
- Quasi-cyclic codes
- Sign matching

→ vulnerable to lattice-based attacks

1. Basics: Lee metric, sign matching
2. FuLeeca: Scheme description
3. Rise: Performance
4. Fall: Attack, repairs?

Outline & Disclaimer

FuLeeca: Hash & Sign scheme based on:

- Lee metric
- Quasi-cyclic codes
- Sign matching

→ vulnerable to lattice-based attacks

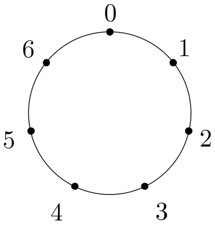
1. Basics: Lee metric, sign matching
2. FuLeeca: Scheme description
3. Rise: Performance
4. Fall: Attack, repairs?

⚠ Not a lattice-based expert ⚠

Basics

Lee Metric

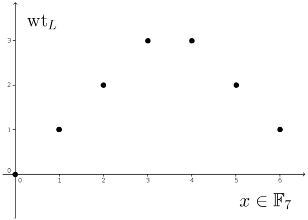
- $x \in \mathbb{Z}/m\mathbb{Z} = \{0, \ldots, m-1\}$ $\rightarrow$ $\text{wt}_L(x) = \min\{x, |m-x|\}$



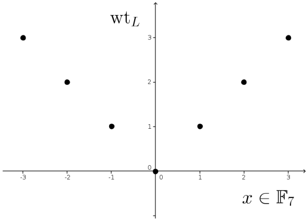
Basics

Lee Metric

- $x \in \{-\lfloor \frac{m}{2} \rfloor, \dots, \lfloor \frac{m}{2} \rfloor\}$ $\rightarrow$ $\mathrm{wt}_L(x) = |x|$



$\longrightarrow$



Basics

Ambient Space: prime field $\mathbb{F}_p$ with p odd

Lee Metric

- $x \in \{-\frac{p-1}{2}, \dots, \frac{p-1}{2}\}$ → $\mathrm{wt}_L(x) = |x|$
- $x \in \mathbb{F}_p^n$ → $\mathrm{wt}_L(x) = \sum_{i=1}^n \mathrm{wt}_L(x_i)$
- $x, y \in \mathbb{F}_p^n$ → $d_L(x, y) = \mathrm{wt}_L(x - y)$
- $\mathcal{C} \subseteq \mathbb{F}_p^n$ linear code → $d_L(\mathcal{C}) = \min\{\mathrm{wt}_L(x) \mid x \in \mathcal{C}, x \neq 0\}$

⚠ → Maximal Lee weight $M = \frac{p-1}{2}$ → $d_H(\mathcal{C}) \leq d_L(\mathcal{C})$

Basics

A random code can correct more Lee-metric errors than Hamming-metric errors

$\rightarrow$ Generic decoders have a larger cost

Hash & Sign schemes suffer from large public key sizes

Basics

A random code can correct more Lee-metric errors than Hamming-metric errors

→ Generic decoders have a larger cost

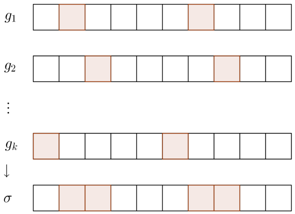
Hash & Sign schemes suffer from large public key sizes

→ reduce key sizes:

→ low density generators

→ quasi-cyclic codes

→ statistical attacks



M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, D. Schipani. "Using LDGM codes and sparse syndromes to achieve digital signatures.", PQCrypto, 2013.

Basics

A random code can correct more Lee-metric errors than Hamming-metric errors

→ Generic decoders have a larger cost

Hash & Sign schemes suffer from large public key sizes

→ reduce key sizes:

→ low Lee density generators

→ quasi-cyclic codes

→ low Lee weight but large Hamming weight

g_1	1	1	2	1	3	0
g_2	0	1	1	2	1	3
$\vdots$						
g_k	1	2	1	3	0	1
$\downarrow$						
σ	2	4	4	6	4	4

M. Baldi, M. Bianchi, F. Chiaraluce, J. Rosenthal, D. Schipani. "Using LDGM codes and sparse syndromes to achieve digital signatures.", PQCrypto, 2013.

Basics

- Lee GV: $R \geq 1 - \lim_{n\to\infty} \frac{1}{n} \log_p |\{x \mid \mathrm{wt}_L(x) = \delta n M\}|$, rel. min. Lee distance δ

$\rightarrow$ Random codes attain the Lee-metric GV bound w.h.p.

E. Byrne, A.-L. Horlemann, K. Khathuria, **V.W.** "Density of free modules over finite chain rings."Linear Algebra and its Applications, 2022

Basics

- Lee GV: $R \geq 1 - \lim_{n\to\infty} \frac{1}{n} \log_p |\{x \mid \mathrm{wt}_L(x) = \delta n M\}|$, rel. min. Lee distance δ

$\rightarrow$ Random codes attain the Lee-metric GV bound w.h.p.

E. Byrne, A.-L. Horlemann, K. Khathuria, **V.W.** "Density of free modules over finite chain rings."Linear Algebra and its Applications, 2022

- Lee SDP: given H, s, t, is there e with $\mathrm{wt}_L(e) \leq t$ and $eH^\top = s$?

$\rightarrow$ Lee SDP is NP-hard

V.W., K. Khathuria, A.-L. Horlemann, M. Battaglioni,P. Santini, E. Persichetti. "On the hardness of the Lee syndrome decoding problem."AMC, 2022

Basics

- Lee GV: $R \geq 1 - \lim_{n \to \infty} \frac{1}{n} \log_p |\{x \mid \mathrm{wt}_L(x) = \delta n M\}|$, rel. min. Lee distance δ

$\rightarrow$ Random codes attain the Lee-metric GV bound w.h.p.

E. Byrne, A.-L. Horlemann, K. Khathuria, **V.W.** "Density of free modules over finite chain rings."Linear Algebra and its Applications, 2022

- Lee SDP: given H, s, t, is there e with $\mathrm{wt}_L(e) \leq t$ and $eH^\top = s$?

$\rightarrow$ Lee SDP is NP-hard

V.W., K. Khathuria, A.-L. Horlemann, M. Battaglioni,P. Santini, E. Persichetti. "On the hardness of the Lee syndrome decoding problem."AMC, 2022

- Typical set for vectors of fixed Lee weight w: entry $x_i = \alpha \in \mathbb{F}_p$ with prob. $p_w(\alpha)$

$\rightarrow$ $T(w, n) = \{x \mid x_i = \alpha \text{ for } p_w(\alpha) n \text{ many } i\}$

J. Bariffi, H. Bartz, G. Liva, J. Rosenthal. "On the Properties of Error Patterns in the Constant Lee Weight Channel."IZS, 2021

Basics

Sign Matching

- $x \in \{-\frac{p-1}{2}, \dots, \frac{p-1}{2}\}$ → $\text{sgn}(x) = -1$, if $x < 0, \text{sgn}(x) = 1$, if $x > 0$, $\text{sgn}(0) = 0$

Example: $\mathbb{F}_7 : \text{sgn}(0, 1, 5, 3) = (0, 1, -1, 1)$

- $x, y \in \mathbb{F}_p^n$ → $\text{mt}(x, y) = |\{i \mid \text{sgn}(x_i) = \text{sgn}(y_i) \neq 0\}|$

→ How likely that a random vector matches signs with fixed one?

- $x \in \mathbb{F}_p^n$ fix, $y \in \{\pm 1\}^n$ rand. → $\mathbb{P}(\text{mt}(x, y) = \mu) = B(\mu, \text{wt}_H(x), \frac{1}{2})$ (binom. distr.)

Logarithmic Matching Probability

$\text{LMP}(x, y) = -\log_2(B(\mu, \text{wt}_H(x), \frac{1}{2}))$ → cost to find y with $\text{LMP} = \lambda$ is 2^λ

FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

Connection to message not $\mathsf{Hash}(m) = eH^\top$
but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\operatorname{sgn}(xG)$

FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

Connection to message not $\mathsf{Hash}(m) = eH^\top$
but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\operatorname{sgn}(xG)$

KEY GENERATION

FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

Connection to message not $\mathsf{Hash}(m) = eH^\top$
but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\mathrm{sgn}(xG)$

KEY GENERATION

- $G = \big(A \mid B\big) = \big(\mathrm{circ}(a) \mid \mathrm{circ}(b)\big)$ quasi-cylcic code
- $\tilde{G} = \big(\mathrm{Id}_{n/2} \mid A^{-1}B\big) = \big(\mathrm{Id}_{n/2} \mid T\big) \rightarrow$ public key: T

$\rightarrow$ How to sample secret generators a, b?

FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

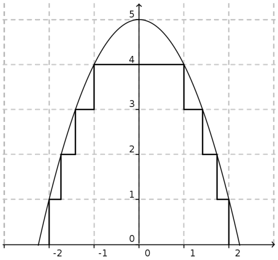
Connection to message not $\mathsf{Hash}(m) = eH^\top$ but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\mathrm{sgn}(xG)$

KEY GENERATION

→ How to sample secret generators a, b?

→ $d = w_{\text{key}}$: min. Lee distance from GV
→ $a, b \in T(d/2, n/2)$

- hidden detail: fancy rounding function f to get close to weight $d/2$
- hidden detail: random sign swapping of a to get invertible $\mathrm{circ}(a)$



FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

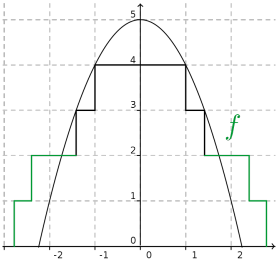
Connection to message not $\mathsf{Hash}(m) = eH^\top$ but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\operatorname{sgn}(xG)$

KEY GENERATION

→ How to sample secret generators a, b?

→ $d = w_{\text{key}}$: min. Lee distance from GV
→ $a, b \in T(d/2, n/2)$

- hidden detail: fancy rounding function f to get close to weight $d/2$
- hidden detail: random sign swapping of a to get invertible $\operatorname{circ}(a)$



FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

Connection to message not $\mathsf{Hash}(m) = eH^\top$
but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\operatorname{sgn}(xG)$

SIGNATURE GENERATION

VERIFICATION

FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

Connection to message not $\mathsf{Hash}(m) = eH^\top$
but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\operatorname{sgn}(xG)$

SIGNATURE GENERATION

VERIFICATION

- Iterative algorithm: go through rows of G
- add / subtract rows until $xG = v$ is s.t.
  1. $\mathrm{wt}_L(v) \in [w_{\mathrm{sig}} - 2w_{\mathrm{key}}, w_{\mathrm{sig}}]$
  2. $\mathrm{LMP}(v, \mathsf{Hash}(m)) \geq \lambda + 64$
- $\tilde{G} = \left(\mathrm{Id}_{n/2} \mid T\right) \to v = (y, yT)$

→ Signature y

FuLeeca

Similarity

Hide code $\langle G \rangle$ and publish $\tilde{G}$

Difference

Connection to message not $\mathsf{Hash}(m) = eH^\top$
but $\mathsf{Hash}(m) \in \{\pm 1\}^n$ is close to $\mathrm{sgn}(xG)$

SIGNATURE GENERATION

- Iterative algorithm: go through rows of G
- add / subtract rows until $xG = v$ is s.t.
  1. $\mathrm{wt}_L(v) \in [w_{\mathrm{sig}} - 2w_{\mathrm{key}}, w_{\mathrm{sig}}]$
  2. $\mathrm{LMP}(v, \mathsf{Hash}(m)) \geq \lambda + 64$
- $\tilde{G} = \left(\mathrm{Id}_{n/2} \mid T\right) \to v = (y, yT)$

$\to$ Signature y

VERIFICATION

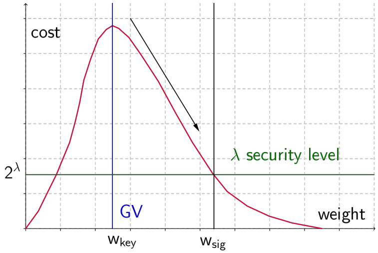
- Given T, message m and signature y
- recover $v = (y, yT)$ and check
  1. $\mathrm{wt}_L(v) \in [w_{\mathrm{sig}} - 2w_{\mathrm{key}}, w_{\mathrm{sig}}]$
  2. $\mathrm{LMP}(v, \mathsf{Hash}(m)) \geq \lambda + 64$

$\to$ Accept/ Reject

Rise

Parameter Choices

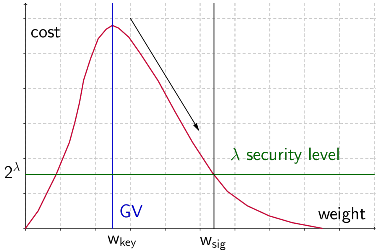
- $p = 65'521$
- $w_{\text{key}}/(nM) = 0.001437$ on GV
- $w_{\text{sig}}/(nM) = 0.03$ s.t. generic decoders cost 2^λ



Rise

Parameter Choices

- $p = 65'521$
- $w_{\text{key}}/(nM) = 0.001437$ on GV
- $w_{\text{sig}}/(nM) = 0.03$ s.t. generic decoders cost 2^λ



Sizes in bytes, times in MCycles

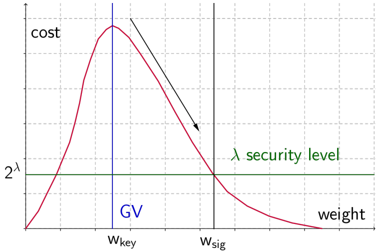
Level	| pk |	| sign |	t_{sign}	t_{verify}
I	1'318	1'100	1'803	1.4
III	1'982	1'620	2'139	2.5
V	2'638	2'130	11'805	3.8

Rise

Parameter Choices

- $p = 65'521$
- $w_{\text{key}}/(nM) = 0.001437$ on GV
- $w_{\text{sig}}/(nM) = 0.03$ s.t. generic decoders cost 2^λ

Sizes in bytes, times in MCycles



Level	| pk |	| sign |	t_{sign}	t_{verify}
I	1'318	1'100	1'803	1.4
III	1'982	1'620	2'139	2.5
V	2'638	2'130	11'805	3.8

→ Total size: 2.4 KB

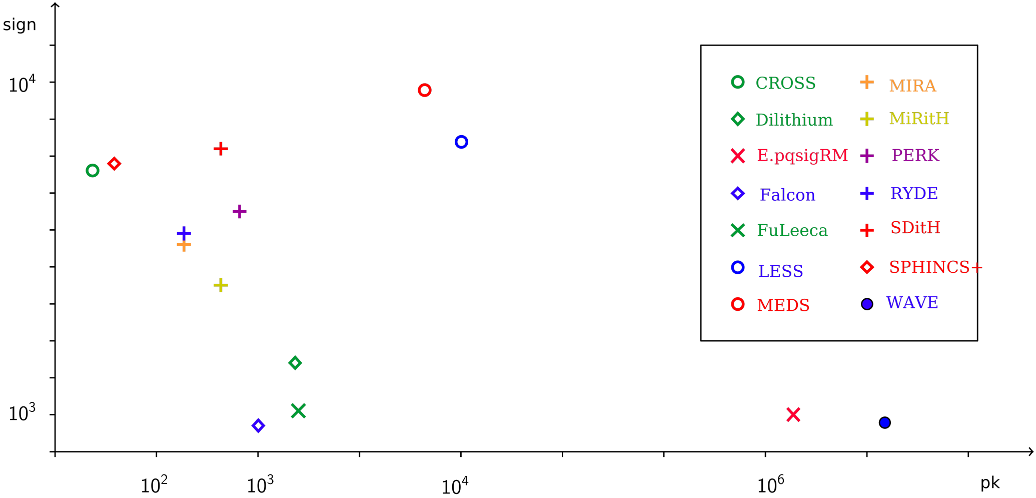
→ Falcon: 1.5 KB

→ Dilithium: 3.7 KB

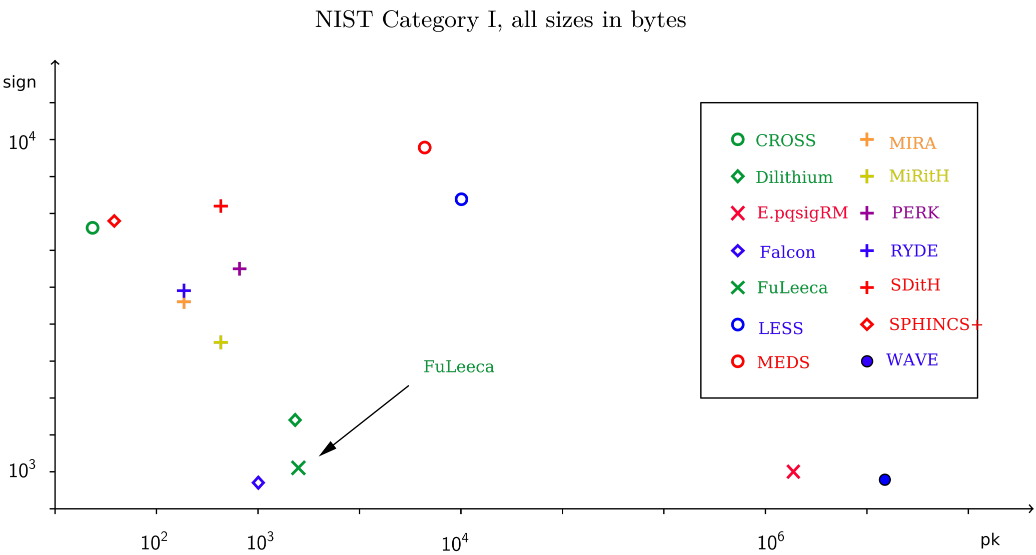
→ SPHINCS$^+$: 7.7 KB

Rise

NIST Category I, all sizes in bytes

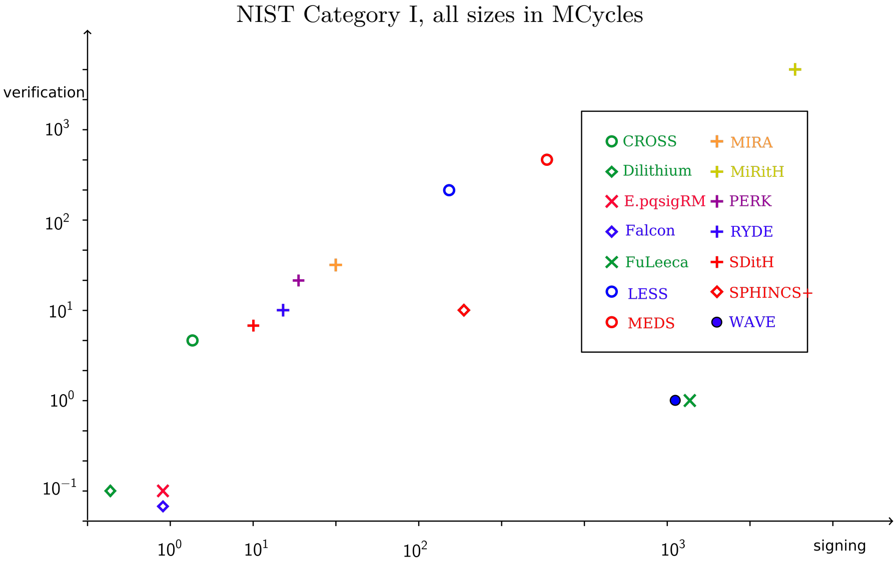


Rise

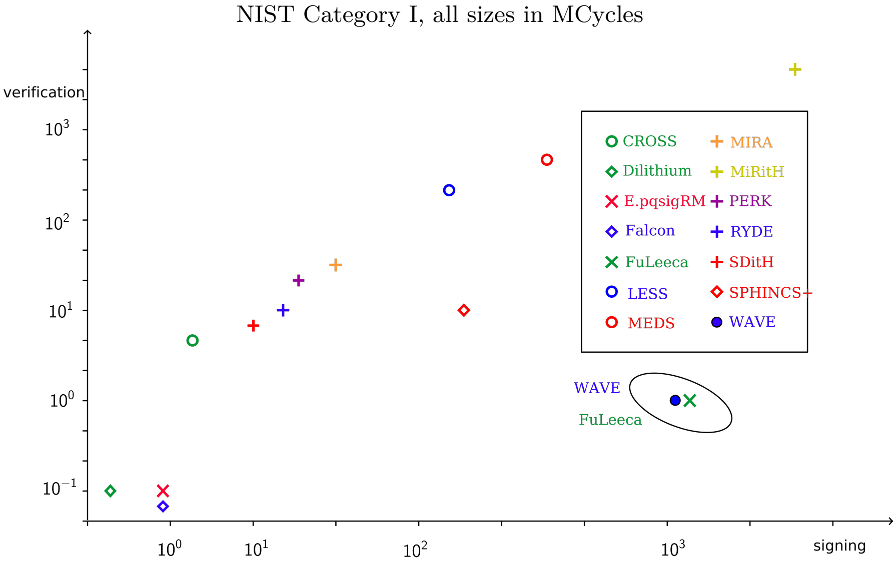


NIST Category I, all sizes in bytes
sign
10^4
10^3
10^2
10^3
10^4
10^6
pk
FuLeeca
CROSS
Dilithium
E.pqsigRM
Falcon
FuLeeca
LESS
MEDS
MIRA
MiRitH
PERK
RYDE
SDitH
SPHINCS+
WAVE

Rise


NIST Category I, all sizes in MCycles
verification
signing
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
CROSS
Dilithium
E.pqsigRM
Falcon
FuLeeca
LESS
MEDS
MIRA
MiRitH
PERK
RYDE
SDitH
SPHINCS+
WAVE

Rise



NIST Category I, all sizes in MCycles
verification
10^3
10^2
10^1
10^0
10^{-1}
10^0
10^1
10^2
10^3
signing
CROSS
MIRA
Dilithium
MiRitH
E.pqsigRM
PERK
Falcon
RYDE
FuLeeca
SDitH
LESS
SPHINCS+
MEDS
WAVE
WAVE
FuLeeca

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \;\rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad\quad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

$\rightarrow$ can use Lee to solve Euclidean

$\rightarrow$ use Euclidean to solve Lee: not known/hard

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \ \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\qquad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

$\rightarrow$ can use Lee to solve Euclidean

$\rightarrow$ use Euclidean to solve Lee: our instances ✓

Fall

Not a lattice-based expert

Euclidean Metric

$x \in \mathbb{Z}^n \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad\left(\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|\right)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

1. No modular reduction: $v = xG \mod p$

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

1. No modular reduction: $v = xG$ ~~mod p~~

→ directly use integer lattice $L(G)$

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

1. No modular reduction: $v = xG$ ~~mod p~~

→ directly use integer lattice $L(G)$

Countermeasure: larger values in x

→ large w_{sig}: forgery attacks

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\qquad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

1. No modular reduction: $v = xG \, \cancel{\bmod p}$

→ directly use integer lattice $L(G)$

Countermeasure: larger values in x

→ large w_{sig}: forgery attacks

2. Quasi-cyclic: $G = \begin{pmatrix} A \mid B \end{pmatrix}$, $v = (xA, xB)$

→ enough to work with $L(A)$

→ $L(A)$ circulant lattice → subexponential

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \ \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

1. No modular reduction: $v = xG$

→ directly use integer lattice $L(G)$

Countermeasure: larger values in x

→ large w_{sig}: forgery attacks

2. Quasi-cyclic: $G = \left(A \mid B\right)$, $v = (xA, xB)$

→ enough to work with $L(A)$

→ $L(A)$ circulant lattice → subexponential

Countermeasure: no quasi-cyclic code

→ public keys > 3.5 MB

Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \ \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

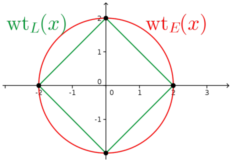
L_2-Norm can be reduced to any L_p-Norm (also L_1)

O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

3. Short Euclidean = Small Lee weight



Fall

⚠ Not a lattice-based expert ⚠

Euclidean Metric

$x \in \mathbb{Z}^n \ \rightarrow \mathrm{wt}_E(x) = \sqrt{\sum_{i=1}^n |x_i|^2}$ $\quad (\mathrm{wt}_L(x) = \sum_{i=1}^n |x_i|)$

L_2-Norm can be reduced to any L_p-Norm (also L_1)

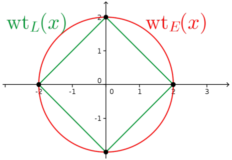
O. Regev, R. Rosen. "Lattice problems and norm embeddings.", ACM symposium on Theory of Computing, 2006.

→ can use Lee to solve Euclidean

→ use Euclidean to solve Lee: our instances ✓

3. Short Euclidean = Small Lee weight

→ How to avoid this situation?



Hörmann & van Woerden: experimentally ✓

Questions?

Many thanks from the FuLeeca-Team

- Stefan Ritterhoff
- Sebastian Bitzer
- Patrick Karl
- Georg Maringer
- Thomas Schamberger
- Jonas Schupp
- Georg Sigl
- Antonia Wachter-Zeh
- Violetta Weger

Slides

Website

Code-Based Submissions

All sizes in bytes, times in MCycles.

Scheme	Based on	Technique	| Pk |	| Sig |	Sign	Verify
CROSS	Restricted SDP	ZK	32	7'625	11	7.4
Enh. pqsigRM	Reed-Muller	Hash & Sign	2'000'000	1'032	1.3	0.2
FuLeeca	Lee SDP	Hash & Sign	1'318	1'100	1'846	1.3
LESS	Code equiv.	ZK	13'700	8'400	206	213
MEDS	Matrix rank equiv.	ZK	9'923	9'896	518	515
MIRA	Matrix rank SDP	MPC	84	5'640	46'8	43'9
MiRitH	Matrix rank SDP	MPC	129	4'536	6'108	6'195
PERK	Permuted Kernel	MPC	150	6'560	39	27
RYDE	Rank SDP	MPC	86	5'956	23.4	20.1
SDitH	SDP	MPC	120	8'241	13.4	12.5
WAVE	Large wt $(U, U+V)$	Hash & Sign	3'677'390	822	1'160	1.23

Not all schemes have optimized implementations → Numbers may change