Code-Based Cryptography in the Lee Metric

Anna-Lena Horlemann

Institute of Computer Science, University of St.Gallen, Switzerland

Munich Workshop in Coding Theory and Cryptography (MWCC),
TU München, Germany, 2024

Recap – generalized McEliece framework

- The receiver chooses
  - a code C with generator matrix G and an efficient decoding algorithm,
  - a disguising function ϕ that is a (near-)isometry.
- **Private key:** ϕ and the decoding algorithm for C
- **Public key:** $\phi(G)$ together with the error correction capability of the code generated by $\phi(G)$, say $\hat{t}$
- **Encryption:** Choose a random error vector e of weight at most $\hat{t}$ and encrypt the message m as

$$c = m\,\phi(G) + e.$$

- **Decryption:** Compute $\phi^{-1}(c)$ and decode this in the secret code C to recover m.

Recap – generalized McEliece framework

- The receiver chooses
  - a code C with generator matrix G and an efficient decoding algorithm,
  - a disguising function ϕ that is a (near-)isometry.
- **Private key:** ϕ and the decoding algorithm for C
- **Public key:** $\phi(G)$ together with the error correction capability of the code generated by $\phi(G)$, say $\hat{t}$
- **Encryption:** Choose a random error vector e of weight at most $\hat{t}$ and encrypt the message m as

$$c = m\,\phi(G) + e.$$

- **Decryption:** Compute $\phi^{-1}(c)$ and decode this in the secret code C to recover m.

What if we use the Lee metric instead of the Hamming metric?

Recap – generalized McEliece framework

- The receiver chooses
  - a code C with generator matrix G and an efficient decoding algorithm,
  - a disguising function ϕ that is a (near-)isometry.
- **Private key:** ϕ and the decoding algorithm for C
- **Public key:** $\phi(G)$ together with the error correction capability of the code generated by $\phi(G)$, say $\hat{t}$
- **Encryption:** Choose a random error vector e of weight at most $\hat{t}$ and encrypt the message m as

$$c = m\,\phi(G) + e.$$

- **Decryption:** Compute $\phi^{-1}(c)$ and decode this in the secret code C to recover m.

What if we use the Lee metric instead of the Hamming metric?

Lee metric and linear isometries

Definition (Lee distance and weight)

For $u, v \in \mathbb{Z}_m^n$:

$$wt_L((v_1, \ldots, v_n)) := \sum_{i=1}^{n} \min\{|v_i|, |m - v_i|\}$$

$$d_L((u_1, \ldots, u_n), (v_1, \ldots, v_n)) := wt_L((v_1 - u_1, \ldots, v_n - u_n))$$

Example in $\mathbb{Z}_5$:

$$wt_L(0) = 0 \quad wt_L(1) = 1 \quad wt_L(2) = 2 \quad wt_L(3) = 2 \quad wt_L(4) = 1$$

Lee metric and linear isometries

Definition (Lee distance and weight)

For $u, v \in \mathbb{Z}_m^n$:

$$wt_L((v_1, \ldots, v_n)) := \sum_{i=1}^{n} \min\{|v_i|, |m - v_i|\}$$

$$d_L((u_1, \ldots, u_n), (v_1, \ldots, v_n)) := wt_L((v_1 - u_1, \ldots, v_n - u_n))$$

Example in $\mathbb{Z}_5$:

$$wt_L(0) = 0 \quad wt_L(1) = 1 \quad wt_L(2) = 2 \quad wt_L(3) = 2 \quad wt_L(4) = 1$$

Theorem

The linear isometries for the metric space $(\mathbb{Z}_m^n, d_L)$ are represented by the monomial matrices with non-zero entries in $\{1, -1\}$.

Lee metric and linear isometries

Definition (Lee distance and weight)

For $u, v \in \mathbb{Z}_m^n$:

$$wt_L((v_1, \ldots, v_n)) := \sum_{i=1}^{n} \min\{|v_i|, |m - v_i|\}$$

$$d_L((u_1, \ldots, u_n), (v_1, \ldots, v_n)) := wt_L((v_1 - u_1, \ldots, v_n - u_n))$$

Example in $\mathbb{Z}_5$:

$$wt_L(0) = 0 \quad wt_L(1) = 1 \quad wt_L(2) = 2 \quad wt_L(3) = 2 \quad wt_L(4) = 1$$

Theorem

The linear isometries for the metric space $(\mathbb{Z}_m^n, d_L)$ are represented by the monomial matrices with non-zero entries in $\{1, -1\}$.

Near-isometries: Extend to monomial matrices with other non-zero entries, or extend to row/column weight-2 matrices.
(Not trivial to estimate the distance bounds.)

Decodable codes

Idea of Goppa codes works also in the Lee metric!

Theorem (Roth-Siegel '94, Byrne '02)

Let $\alpha_i \in \mathbb{Z}_{p^s}$ be pairwise distinct, $\gamma_i \in \mathbb{Z}_{p^s}$ be invertible, and C be a code over a subring $\mathbb{Z}_{p^{s'}}$ with parity check matrix

$$\begin{pmatrix} \gamma_1 & \gamma_2 & \dots & \gamma_n \\ \gamma_1 a_1 & \gamma_2 a_2 & \dots & \gamma_n a_n \\ \vdots & & & \vdots \\ \gamma_1 a_1^{r-1} & \gamma_2 a_2^{r-1} & \dots & \gamma_n a_n^{r-1} \end{pmatrix}$$

(called an alternant code). Then there exists a decoding algorithm in quadratic time that can uniquely correct any r Lee errors.

A Lee McEliece scheme

- The receiver chooses an alternant code $C \subseteq \mathbb{Z}_p^n$ of dimension k with generator matrix G and minimum Lee distance $2t+1$. Moreover, they choose a random Lee isometry M and some $A \in \mathrm{GL}_k(p)$.
- **Private key:** G and (A, M) (decoder for C given by G)
- **Public key:** $G' = AGM$
- **Encryption:** Choose a random error vector e of Lee weight at most t and encrypt the message m as

$$c = m\,G' + e.$$

- **Decryption:** Compute $c' = cM^{-1}$ and decode this in C to recover mA. Recover m by multiplying with A^{-1}.

Open questions about security

- (Distinguisher) attacks on alternant code over finite (extension) fields are well studied under the Hamming metric isometries.

  $\rightarrow$ But what happens over $\mathbb{Z}_{p^s}$ under Lee-(near-)isometries?

Open questions about security

- (Distinguisher) attacks on alternant code over finite (extension) fields are well studied under the Hamming metric isometries.

  $\rightarrow$ But what happens over $\mathbb{Z}_{p^s}$ under Lee-(near-)isometries?
- What is the complexity of generic Lee decoding?

  $\rightarrow$ Certainly lower bounded by Hamming generic decoding.

Generic decoding (message recovery attacks)

Question: What is the right approach in the Lee metric?

- **ISD approach:** ISD (and follow-up approaches) find vectors with a lot of zeros. This is a good indicator for a low Lee weight, as well (but not exactly the same). We have a few Hamming-ISD adaptations to the Lee metric.

Generic decoding (message recovery attacks)

Question: What is the right approach in the Lee metric?

- **ISD approach:** ISD (and follow-up approaches) find vectors with a lot of zeros. This is a good indicator for a low Lee weight, as well (but not exactly the same). We have a few Hamming-ISD adaptations to the Lee metric.
- **Lattice approach:** The Lee metric corresponds to the ℓ_1-metric over $\mathbb{Z}_m$. For large m Lee metric decoding in $\mathbb{Z}_m^n$ corresponds to LWE in the corresponding lattice in the ℓ_1-norm. In some cases this can be approximated with ℓ_2-norm.

Generic decoding (message recovery attacks)

Question: What is the right approach in the Lee metric?

- **ISD approach:** ISD (and follow-up approaches) find vectors with a lot of zeros. This is a good indicator for a low Lee weight, as well (but not exactly the same). We have a few Hamming-ISD adaptations to the Lee metric.
- **Lattice approach:** The Lee metric corresponds to the ℓ_1-metric over $\mathbb{Z}_m$. For large m Lee metric decoding in $\mathbb{Z}_m^n$ corresponds to LWE in the corresponding lattice in the ℓ_1-norm. In some cases this can be approximated with ℓ_2-norm.

$$\text{Hamming metric} \xleftarrow{\text{small } m} \text{Lee metric on } \mathbb{Z}_m^n \xrightarrow{\text{large } m} \ell_1\text{-lattice}$$

(A few) generic decoding references

- **ISD approach:**
  - Information set decoding in the Lee metric with applications to cryptography (AL Horlemann-Trautmann, V Weger – 2019/21)
  - On the hardness of the Lee syndrome decoding problem (Weger, Khathuria, Horlemann, Battaglioni, Santini, Persichetti – 2020/24)
  - Information set decoding for Lee-metric codes using restricted balls (Bariffi, Khathuria, Weger – 2022)
- **Lattice approach:**
  - FuLeakage: Breaking FuLeeca by Learning Attacks (Hörmann, van Woerden – 2024)

— What about other systems? —

Digital signatures

We can do the same as in Hamming or rank metric...

- Hash and sign[1]
  - given a message m, find a random m' such that $Hash(m, H_{pub}, m')$ is a decodable syndrome (error vector is main part of signature)
  - Problem: finding this m' is inefficient

[1] See also FuLeeca `https://eprint.iacr.org/2023/377`.

Digital signatures

We can do the same as in Hamming or rank metric...

- Hash and sign[1]
  - given a message m, find a random m' such that $Hash(m, H_{pub}, m')$ is a decodable syndrome (error vector is main part of signature)
  - Problem: finding this m' is inefficient
- Fiat-Shamir transform of zero-knowledge identification (ZKID) schemes
  - public information: random parity check matrix
  - secret: decodable error vector
  - interactive (5-pass) scheme, where one reveals either the weight of the error vector, or that it is a solution to the syndrome equation

[1] See also FuLeeca `https://eprint.iacr.org/2023/377`.

Digital signatures

We can do the same as in Hamming or rank metric...

- Hash and sign[1]
  - given a message m, find a random m' such that $Hash(m, H_{pub}, m')$ is a decodable syndrome (error vector is main part of signature)
  - Problem: finding this m' is inefficient
- Fiat-Shamir transform of zero-knowledge identification (ZKID) schemes
  - public information: random parity check matrix
  - secret: decodable error vector
  - interactive (5-pass) scheme, where one reveals either the weight of the error vector, or that it is a solution to the syndrome equation

  Problem: Because of the nature of the isometries we reveal a lot of information about the whole error vector.

  $\Longrightarrow$ restricted error model (and hence no more Lee metric) (Baldi-...-H-...-Santini-Weger '21)

[1] See also FuLeeca `https://eprint.iacr.org/2023/377`.

— More interesting properties —

Minimum distance of random codes

In the Lee metric there is no one analog of the Singleton bound, and all known bounds are not attainable for most parameters.

Theorem (Byrne-H-Khathuria-Weger '22)

For $n \to \infty$ a random linear code in $\mathbb{F}_q^n$ gets arbitrarily close to the Gilbert-Varshamov bound by probability 1.

$\implies$ Use these results as an estimate for the minimum distance of the random linear code in an ZKID scheme.

How to sample error vectors?

Theorem (Bariffi-Bartz-Liva-Rosenthal '22)

Assume that $x \in \mathbb{Z}_q^n$ has been drawn uniformly at random among all vectors of Lee weight $t(n) = \delta n$, for some constant δ. Let X denote the random variable defining the realizations of an entry of x. Then for $n \to \infty$, the probability of X taking the value i is given by

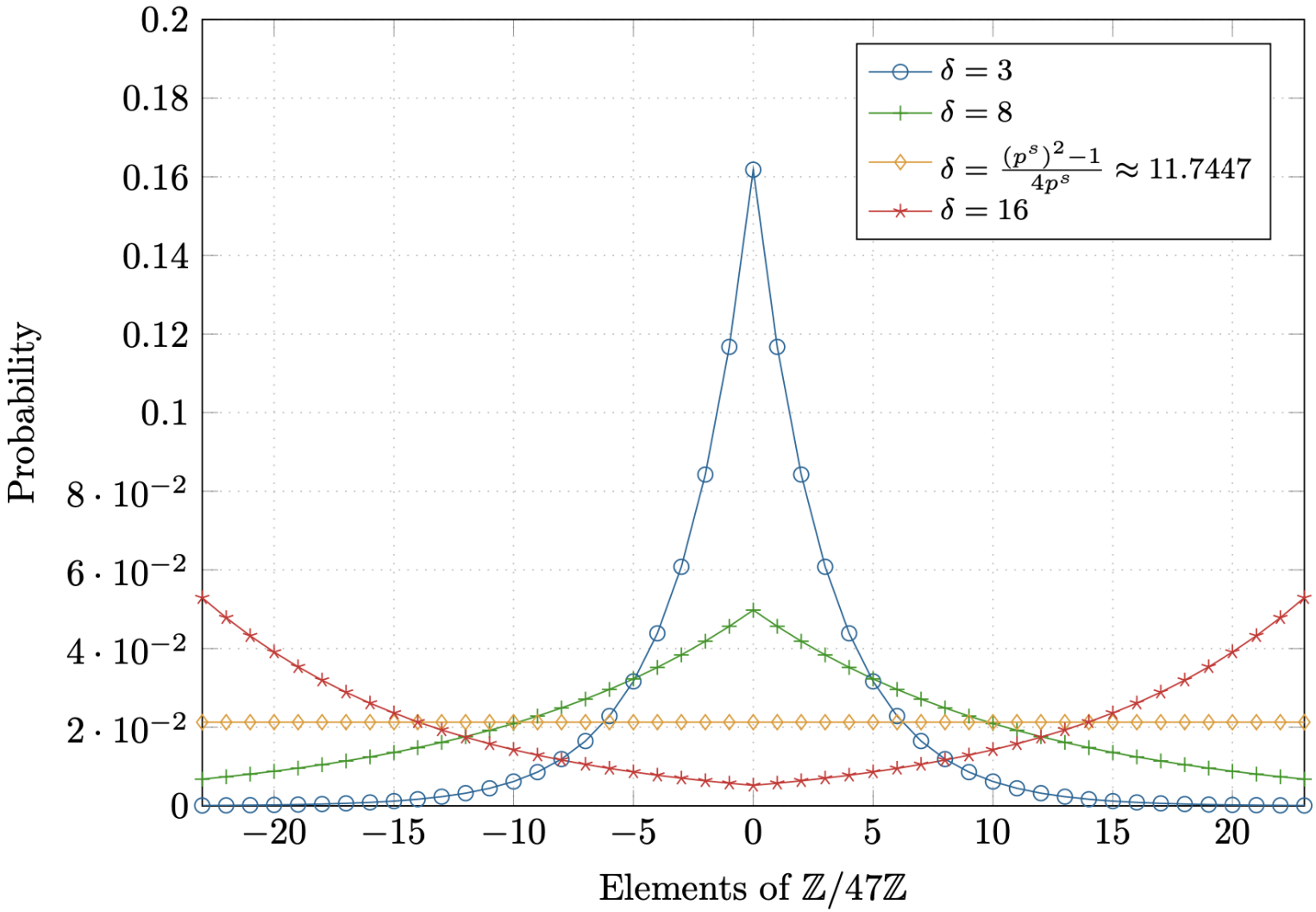
$$P(X = i) = \frac{1}{Z(\beta)} e^{-\beta w t_L(i)}$$

where β is the unique real solution to the weight constraint

$$\delta = \frac{\lfloor \frac{q}{2} \rfloor e^{(\lfloor \frac{q}{2} \rfloor + 2)\beta} - (\lfloor \frac{q}{2} \rfloor + 1) e^{(\lfloor \frac{q}{2} \rfloor + 1)\beta} + e^{\beta}}{(e^{\beta(\lfloor \frac{q}{2} \rfloor + 1)} - 1)(e^{\beta} - 1)}$$

and $Z(\beta)$ denotes the normalization constant.

Marginal constant Lee channel distribution



Thanks to Jessica Bariffi for this graphic.

NP-Completeness of the Lee-SDP

Definition (Syndrome Decoding Problem)

For a ring R, given some $t \in \mathbb{N}$, a parity check matrix $H \in R^{r \times n}$ and a syndrome $s \in R^r$, find a vector $e \in R^n$ of weight at most t that fulfills

$$eH^\top = s.$$

The Lee weight is an additive weight, and we can hence use:

Theorem (Weger-Khathuria-H-Battaglioni-Santini-Persichetti '22)

Let R be a ring with unity and $wt : R \to \mathbb{R}_{\geq 0}$ be a function that satisfies the following properties:

1. $wt(0) = 0$,
2. $wt(1) = 1$ *and* $wt(x) \geq 1$ *for all* $x \neq 0$.

If wt is extended additively to R^n (i.e., by adding the weights of the coordinates) then the SDP with respect to wt is NP-complete.

— Let's wrap up —

Summary and current work

- Lee metric promises better parameters than Hamming, since generic decoding is most likely more difficult.
- However, lots of open questions/problems about possible attacks.
- Generic decoding:
  - When is the ISD approach better, when the lattice approach? Should we do something completely different?
  - If lattice: when do ℓ_2 approximations work for ℓ_1-norm?

	Hamming	rank	Lee
NP-hard	YES	probabilistic	YES
good random codes	YES	YES	YES
eff. decodable codes	YES	YES	YES
generic decoding	x	$> x$	$> x$ lattice tech. ?!

Summary and current work

- Lee metric promises better parameters than Hamming, since generic decoding is most likely more difficult.
- However, lots of open questions/problems about possible attacks.
- Generic decoding:
  - When is the ISD approach better, when the lattice approach? Should we do something completely different?
  - If lattice: when do ℓ_2 approximations work for ℓ_1-norm?

	Hamming	rank	Lee
NP-hard	YES	probabilistic	YES
good random codes	YES	YES	YES
eff. decodable codes	YES	YES	YES
generic decoding	x	$> x$	$> x$ lattice tech. ?!

Thank you for your attention!

Questions? – Comments?