Generalization of the Ball-Collision Algorithm

Violetta Weger

joint work with Carmelo Interlando, Karan Khathuria, Nicole Rohrer and Joachim Rosenthal

University of Zurich

Munich

18 July 2019

Outline

Proposing a code-based cryptosystem

- Structural Attacks
- Nonstructural Attacks

Have to consider Information Set Decoding (ISD)

Motivation

Proposing a code-based cryptosystem

- Structural Attacks
- Nonstructural Attacks

Have to consider Information Set Decoding (ISD)

1978 Berlekamp, McEliece and van Tilborg: Decoding a random linear code is NP-complete

Problem (Syndrome decoding problem)

Given a parity check matrix H of a (binary) code of length n and dimension k and a syndrome s:

$$s = Hx^\intercal \in \mathbb{F}_2^{n-k}$$

and the error correction capacity t, we want to find $e \in \mathbb{F}_2^n$ of weight t such that

$$s = He^\intercal.$$

ISD algorithms and syndome decoding problem

- Syndrome decoding problem is equivalent to the decoding problem and

Problem (Decoding problem)

Given a generator matrix G of a (binary) code of length n and dimension k and a corrupted codeword c:

$$c = mG + e \in \mathbb{F}_2^n$$

and the error correction capacity t, we want to find $e \in \mathbb{F}_2^n$ of weight t.

- equivalent to finding a minimum weight codeword, since in $\mathcal{C} + \{0, c\}$ the error vector e is now the minimum weight codeword.

Information set

Notation

Let $c \in \mathbb{F}_q^n$ and $A \in \mathbb{F}_q^{k \times n}$, let $S \subset \{1, \ldots, n\}$, then we denote by c_S the restriction of c to the entries indexed by S and by A_S the columns of A indexed by S. For a code $\mathcal{C} \subset \mathbb{F}_q^n$, we denote by

$$\mathcal{C}_S = \{c_S \mid c \in \mathcal{C}\}.$$

Definition (Information set)

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a code of dimension k. If $I \subset \{1, \ldots, n\}$ of size k is such that

$$| \mathcal{C} | = | \mathcal{C}_I |,$$

then we call I an information set of $\mathcal{C}$.

Information set

Notation

Let $c \in \mathbb{F}_q^n$ and $A \in \mathbb{F}_q^{k \times n}$, let $S \subset \{1, \ldots, n\}$, then we denote by c_S the restriction of c to the entries indexed by S and by A_S the columns of A indexed by S. For a code $\mathcal{C} \subset \mathbb{F}_q^n$, we denote by

$$\mathcal{C}_S = \{c_S \mid c \in \mathcal{C}\}.$$

Definition (Information set)

Let $\mathcal{C} \subset \mathbb{F}_q^n$ be a code of dimension k. If $I \subset \{1, \ldots, n\}$ of size k is such that

$$| \mathcal{C} | = | \mathcal{C}_I |,$$

then we call I an information set of $\mathcal{C}$.

Information set

Definition (Information set)

Let G be the $k \times n$ generator matrix of $\mathcal{C}$. If $I \subset \{1, \ldots, n\}$ of size k is such that G_I is invertible, then I is an information set of $\mathcal{C}$.

Definition (Information set)

Let H be the $n-k \times n$ parity check matrix of $\mathcal{C}$. If $I \subset \{1, \ldots, n\}$ of size k is such that H_{I^c} is invertible, then I is an information set of $\mathcal{C}$.

Information set

Definition (Information set)

Let G be the $k \times n$ generator matrix of $\mathcal{C}$. If $I \subset \{1, \ldots, n\}$ of size k is such that G_I is invertible, then I is an information set of $\mathcal{C}$.

Definition (Information set)

Let H be the $n-k \times n$ parity check matrix of $\mathcal{C}$. If $I \subset \{1, \ldots, n\}$ of size k is such that H_{I^c} is invertible, then I is an information set of $\mathcal{C}$.

Prange's algorithm

1962 Prange proposes the first ISD algorithm.
Assumption: All t errors occur outside of the information set.

Input: $H \in \mathbb{F}_2^{n-k \times n}, s \in \mathbb{F}_2^{n-k}, t \in \mathbb{N}$
Output: $e \in \mathbb{F}_2^n, wt(e) = t$ and $He^\intercal = s$.

1. Choose an information set $I \subset \{1, \ldots, n\}$ of size k.
2. Find an invertible matrix $U \in \mathbb{F}_2^{n-k \times n-k}$ such that $(UH)_I = A$ and $(UH)_{I^c} = \mathrm{Id}_{n-k}$.
3. If $wt(Us) = t$, then $e_I = 0$ and $e_{I^c} = Us$.
4. Else start over.

Prange's algorithm

1 Choose an information set $I \subset \{1, \ldots, n\}$ of size k.

Let us assume for simplicity that $I = \{1, \ldots, k\}$.

Prange's algorithm

1. Choose an information set $I \subset \{1, \dots, n\}$ of size k.
2. Find an invertible matrix $U \in \mathbb{F}_2^{n-k \times n-k}$ such that $(UH)_I = A$ and $(UH)_{I^c} = \mathrm{Id}_{n-k}$.

Let us assume for simplicity that $I = \{1, \dots, k\}$.

$$UH = \begin{pmatrix} A & \mathrm{Id}_{n-k} \end{pmatrix},$$

hence

$$UHe^\intercal = \begin{pmatrix} A & \mathrm{Id}_{n-k} \end{pmatrix} \begin{pmatrix} 0 \\ e_{I^c} \end{pmatrix} = Us.$$

Prange's algorithm

1. Choose an information set $I \subset \{1, \dots, n\}$ of size k.
2. Find an invertible matrix $U \in \mathbb{F}_2^{n-k \times n-k}$ such that $(UH)_I = A$ and $(UH)_{I^c} = \mathrm{Id}_{n-k}$.
3. If $wt(Us) = t$, then $e_I = 0$ and $e_{I^c} = Us$.

Let us assume for simplicity that $I = \{1, \dots, k\}$.

$$UH = \begin{pmatrix} A & \mathrm{Id}_{n-k} \end{pmatrix},$$

hence

$$UHe^\intercal = \begin{pmatrix} A & \mathrm{Id}_{n-k} \end{pmatrix} \begin{pmatrix} 0 \\ e_{I^c} \end{pmatrix} = Us.$$

From which we get the condition $e_{I^c} = Us$.

Prange's algorithm

The cost of an ISD algorithm is given by the product of

- the cost of one iteration,
- inverted success probability = average number of iterations needed.

The success probability is given by the weight distribution of the error vector.

Example (Success probability of Prange's algorithm)

$$\binom{n-k}{t}\binom{n}{t}^{-1}.$$

Remark

Brute force $\neq$ ISD.

Prange's algorithm

The cost of an ISD algorithm is given by the product of

- the cost of one iteration,
- inverted success probability = average number of iterations needed.

The success probability is given by the weight distribution of the error vector.

Example (Success probability of Prange's algorithm)

$$\binom{n-k}{t}\binom{n}{t}^{-1}.$$

Remark

Brute force $\neq$ ISD.

Prange's algorithm

The cost of an ISD algorithm is given by the product of

- the cost of one iteration,
- inverted success probability = average number of iterations needed.

The success probability is given by the weight distribution of the error vector.

Example (Success probability of Prange's algorithm)

$$\binom{n-k}{t}\binom{n}{t}^{-1}.$$

Remark

Brute force $\neq$ ISD.

Prange's algorithm

The cost of an ISD algorithm is given by the product of

- the cost of one iteration,
- inverted success probability = average number of iterations needed.

The success probability is given by the weight distribution of the error vector.

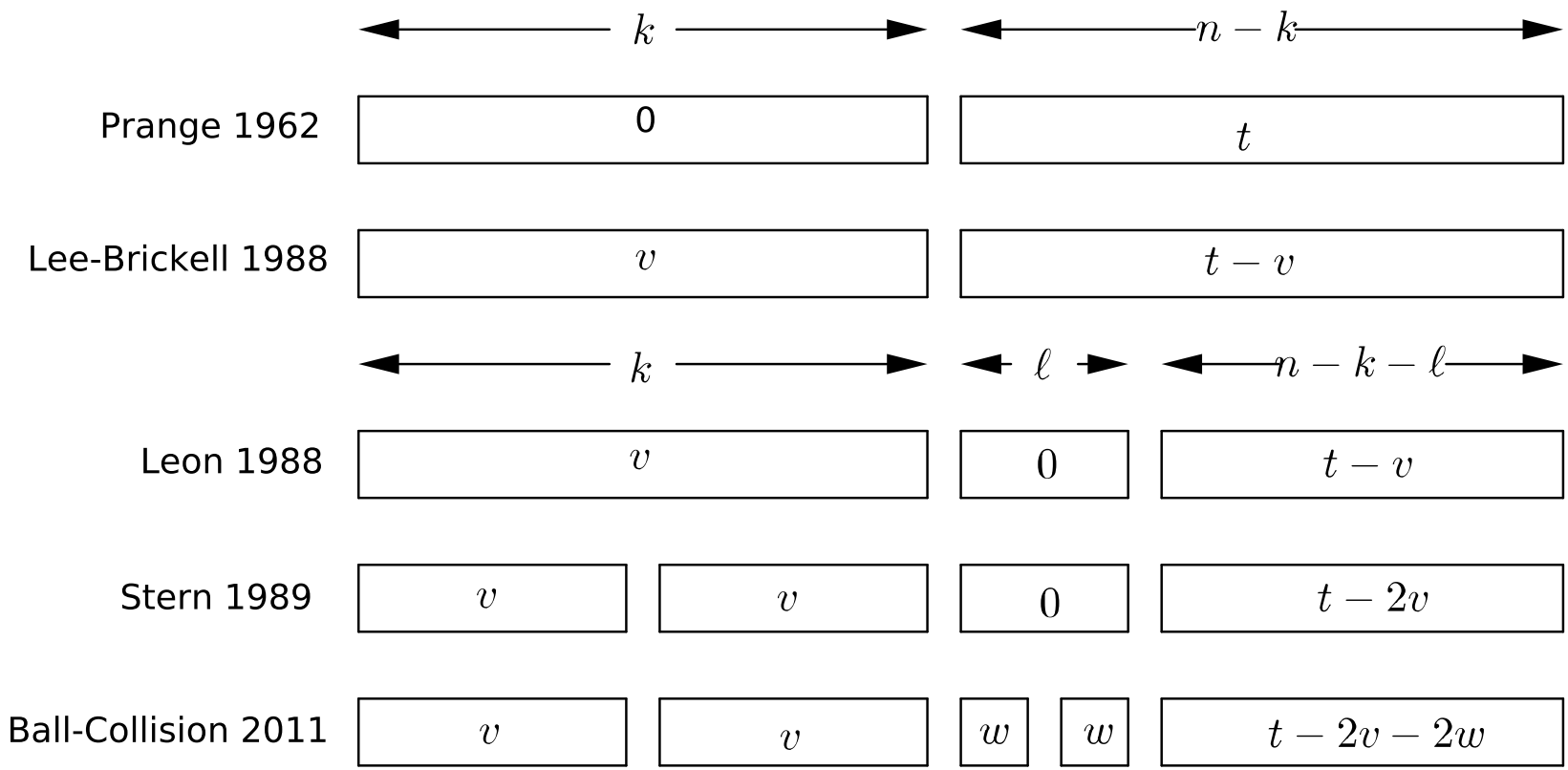
Example (Success probability of Prange's algorithm)

$$\binom{n-k}{t}\binom{n}{t}^{-1}.$$

Remark

Brute force $\neq$ *ISD.*

Improvements Overview



Ball-collision Algorithm

Algorithm 1 Ball-collision over $\mathbb{F}_q$

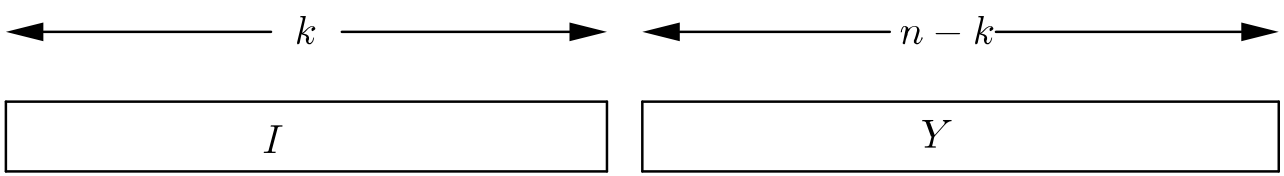
Input: The $(n-k) \times n$ parity check matrix H, the syndrome $s \in \mathbb{F}_q^{n-k}$ and the positive integers p_1, p_2, q_1, q_2, k_1, k_2, ℓ_1, $\ell_2 \in \mathbb{Z}$, such that $k = k_1 + k_2$, $p_i \leq k_i$, $q_i \leq \ell_i$ and $t - p_1 - p_2 - q_1 - q_2 \leq n - k - \ell_1 - \ell_2$.
Output: $e \in \mathbb{F}_q^n$ with $He^\top = s$ and $w(e) = t$.

1: Choose an information set $I \subseteq \{1, ..., n\}$ of H of size k.
2: Partition I into two disjoint subsets X_1 and X_2 of size k_1 and $k_2 = k - k_1$ respectively.
3: Partition $Y = \{1, ..., n\} \backslash I$ into disjoint subsets Y_1 of size ℓ_1, Y_2 of size ℓ_2 and Y_3 of size $\ell_3 = n - k - \ell_1 - \ell_2$.
4: Find an invertible matrix $U \in \mathbb{F}_q^{(n-k)\times(n-k)}$, such that $(UH)_Y = \mathrm{Id}_{n-k}$ and $(UH)_I = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$, where $A_1 \in \mathbb{F}_q^{(\ell_1+\ell_2)\times k}$ and $A_2 \in \mathbb{F}_q^{\ell_3 \times k}$.
5: Compute $Us = \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}$, where $s_1 \in \mathbb{F}_q^{\ell_1+\ell_2}$ and $s_2 \in \mathbb{F}_q^{\ell_3}$.
6: Compute $S = \{(A_1(\pi_I(x_1)) + \pi_{Y_1 \cup Y_2}(y_1), x_1, y_1) \mid x_1 \in \mathbb{F}_q^n(X_1), wt(x_1) = v_1, y_1 \in \mathbb{F}_q^n(Y_1), wt(y_1) = w_1\}$,
7: Compute $T = \{(-A_1(\pi_I(x_2)) + s_1 - \pi_{Y_1 \cup Y_2}(y_2), x_2, y_2) \mid x_2 \in \mathbb{F}_q^n(X_2), wt(x_2) = v_2, y_2 \in \mathbb{F}_q^n(Y_2), wt(y_2) = w_2\}$.
8: **for** $(v, x_1, y_1) \in S$ **do**
9: **for** $(v, x_2, y_2) \in T$ **do**
10: **if** $w(-A_2(\pi_I(x_1 + x_2)) + s_2) = t - p_1 - p_2 - q_1 - q_2$ **then**
Output: $e = x_1 + x_2 + y_1 + y_2 + \sigma_{Y_3}(-A_2(\pi_I(x_1 + x_2)) + s_2)$
11: **else** go to Step 1 and choose new information set I.

Ball-collision Algorithm

1 Choose an information set I.

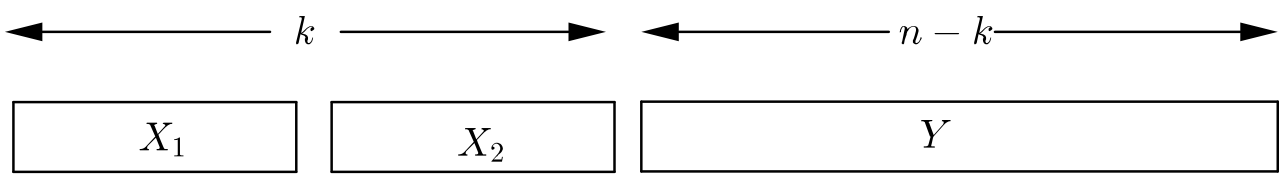
Let us assume for simplicity that $I = \{1, \ldots, k\}$.



Ball-collision Algorithm

1. Choose an information set I.
2. Partition I into X_1 and X_2.

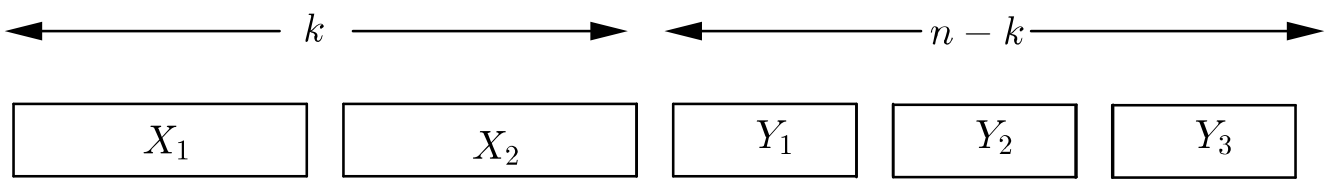
Let us assume for simplicity that $I = \{1, \ldots, k\}$.



Ball-collision Algorithm

1. Choose an information set I.
2. Partition I into X_1 and X_2.
3. Partition Y into Y_1, Y_2, Y_3.

Let us assume for simplicity that $I = \{1, \ldots, k\}$.



Ball-collision Algorithm

1. Choose an information set I.
2. Partition I into X_1 and X_2.
3. Partition Y into Y_1, Y_2, Y_3.
4. Bring H in systematic form.

$$UHe^\intercal = \begin{pmatrix} A_1 & \mathrm{Id}_{\ell_1+\ell_2} & 0 \\ A_2 & 0 & \mathrm{Id}_{\ell_3} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = Us.$$

We get the conditions

$$A_1 e_1 + e_2 = s_1,$$
$$A_2 e_1 + e_3 = s_2.$$

Ball-collision Algorithm

1. Choose an information set I.
2. Partition I into X_1 and X_2.
3. Partition Y into Y_1, Y_2, Y_3.
4. Bring H in systematic form.

$$UHe^\intercal = \begin{pmatrix} A_1 & \mathrm{Id}_{\ell_1+\ell_2} & 0 \\ A_2 & 0 & \mathrm{Id}_{\ell_3} \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix} = \begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = Us.$$

We get the conditions

$$A_1 e_1 + e_2 = s_1,$$
$$A_2 e_1 + e_3 = s_2.$$

Ball-collision Algorithm

Conditions:

$$A_1 e_1 + e_2 = s_1,$$
$$A_2 e_1 + e_3 = s_2.$$

Assumptions:

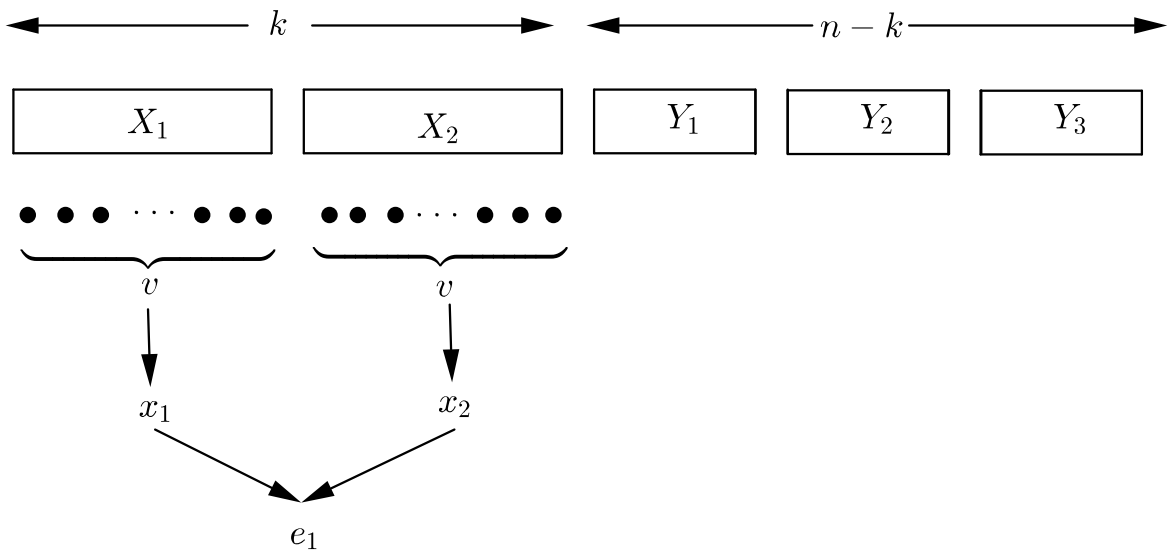
a e_1 has support in $I = X_1 \cup X_2$ and weight $2v$

b e_2 has support in $Y_1 \cup Y_2$ and weight $2w$

c e_3 has support in Y_3 and weight $t - 2v - 2w$

Ball-collision Algorithm

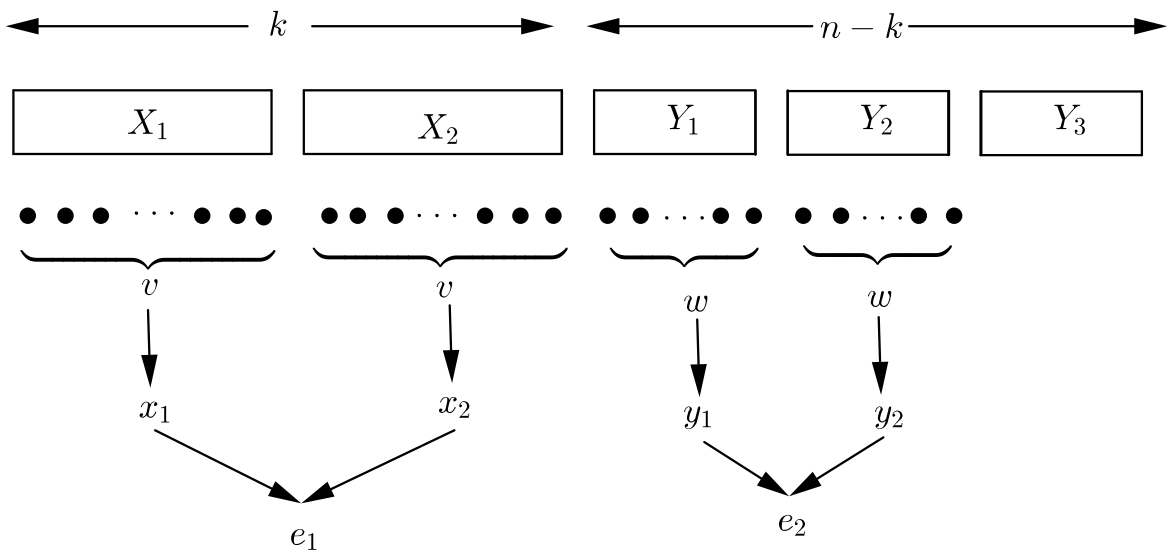
a e_1 has support in $I = X_1 \cup X_2$ and weight $2v$



Ball-collision Algorithm

a e_1 has support in $I = X_1 \cup X_2$ and weight $2v$

b e_2 has support in $Y_1 \cup Y_2$ and weight $2w$



Ball-collision Algorithm

$$A_1 e_1 + e_2 = s_1, \tag{1}$$
$$A_2 e_1 + e_3 = s_2. \tag{2}$$

Condition (1): Go through all choices of e_1 and e_2 and check with collision if (1) is satisfied.
Condition (2): Define $e_3 = s_2 - A_2 e_1$ and check if e_3 has weight $t - 2v - 2w$.

Hidden details

- Collision check

For $e_1 = x_1 + x_2$ and $e_2 = y_1 + y_2$ we want to check that $A_1(x_1 + x_2) + (y_1 + y_2) = s_1$.

1. For all x_1 having support in X_1 and weight v, for all y_1 having support in Y_1 and weight w: compute $A_1 x_1 + y_1$.
2. For all x_2 having support in X_2 and weight v, for all y_2 having support in Y_2 and weight w: compute $s_1 - y_2 - A_1 x_2$.

If $A_1 x_1 + y_1 = s_1 - y_2 - A_1 x_2$ it follows $A_1(x_1 + x_2) + (y_1 + y_2) = s_1$.

Hidden details

- Collision check Cost

Given $A \in \mathbb{F}_q^{k \times n}$. Instead of computing $Ax^\intercal$ for all x of a fixed weight w, one can use intermediate sums:

1. Compute $Ax_1^\intercal$ for all x_1 of weight 1, i.e. compute scalar multiples of the columns of A
   Cost: $n(q-1)\log_2(q)^2$ bit operations.
2. Compute $Ax_2^\intercal$ for all x_2 having weight 2, i.e. choose two columns and add their *already computed* scalar multiples
   Cost: $\binom{n}{2}(q-1)^2 k \log_2(q)$ bit operations.

$\vdots$

Total cost: $\sum_{i=2}^{w} \binom{n}{i}(q-1)^i k \log_2(q) + n(q-1)\log_2(q)^2$ bit operations.

Hidden details

- Average number of collisions: assuming uniform distribution the average number of collision between a set S and T in $\mathbb{F}_q^\ell$ is given by

$$\frac{|\ S\ ||\ T\ |}{q^\ell}$$

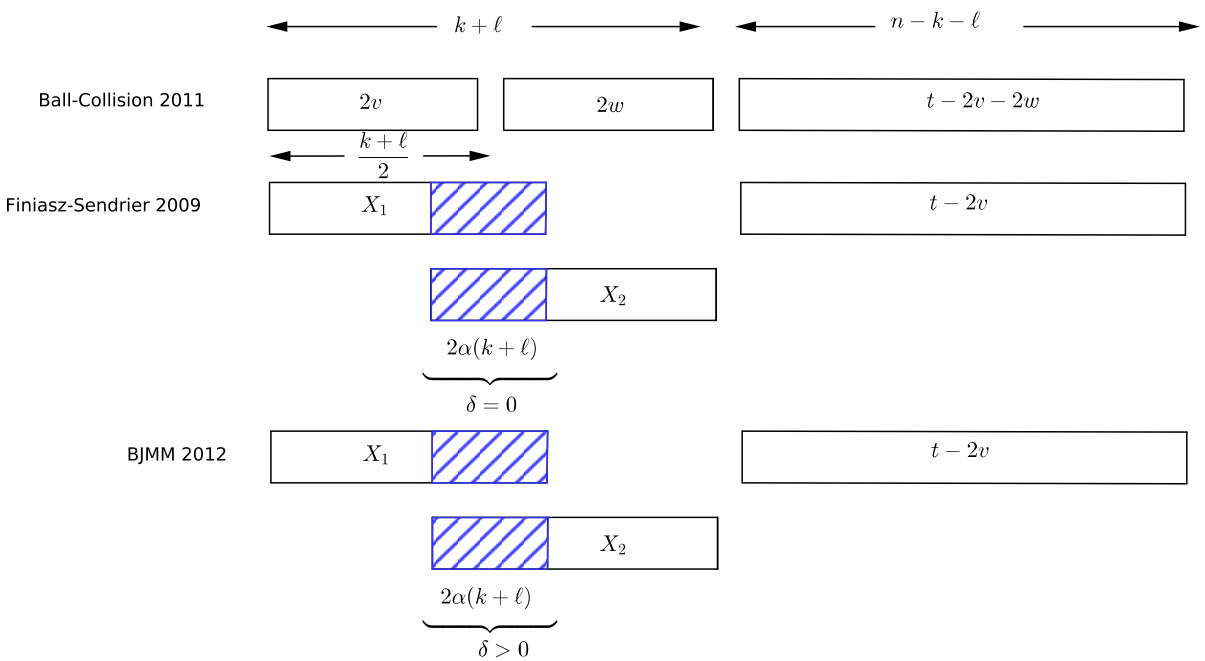
- Early abort: we want to check wheter $x + y$ has weight t, we compute and simultaneously check the weight, as soon as the weight $t + 1$ is reached we abort. An entry in $\mathbb{F}_q$ has on average the weight $\frac{q-1}{q}$, hence *on average* after computing $\frac{q}{q-1}(t + 1)$ entries of the solution we can abort.

New directions

Idea of overlapping sets:

2009 Finiasz and Sendrier: X_1 and X_2 can overlap

2012 Becker, Joux, May and Meurer: can add redundant errors in the overlap



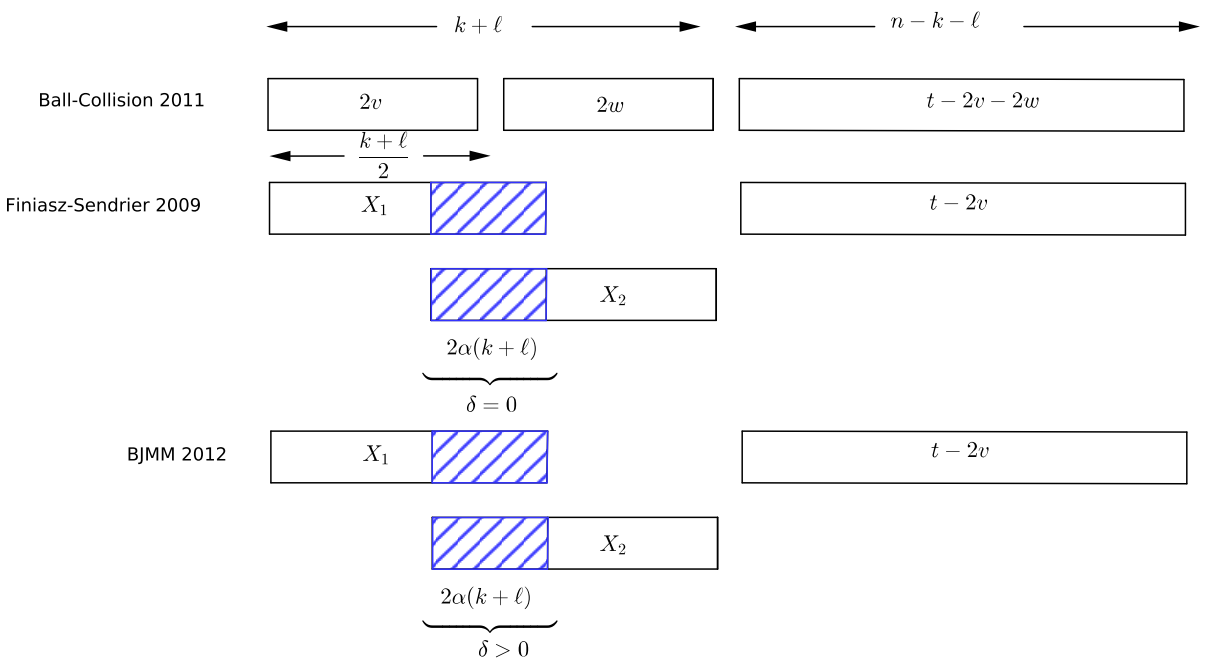
New directions

New parameters:

- α overlap-ratio
- δ amount of redundant errors

2009 Finiasz and Sendrier: $\alpha = 1/2, \delta = 0$

2012 BJMM: $\alpha = 1/2, \delta > 0$



Comparison of Complexities

Let $F(q, R)$ be the exponent of the optimized asymptotic complexity. The asymptotic complexity of half-distance decoding at rate R over $\mathbb{F}_q$ is then given by $q^{F(q,R)n+o(n)}$.

q	$q-$Stern	$q-$Stern-MO	$q-$Ball-collision	$q-$BJMM-MO
2	0.05563	0.05498	0.055573	0.04730
3	0.05217	0.05242	0.052145	0.04427
4	0.04987	0.05032	0.049846	0.04294
5	0.04815	0.04864	0.048140	0.03955
7	0.04571	0.04614	0.045697	0.03706
8	0.04478	0.04519	0.044770	0.03593
11	0.04266	0.04299	0.042656	0.03335

Comparison of Complexities

If prefered in base 2: The asymptotic complexity of half-distance decoding at rate R over $\mathbb{F}_q$ is then given by $2^{F(q,R)\log_2(q)n+o(n)}$.

q	$q-$Stern	$q-$Stern-MO	$q-$Ball-collision	$q-$BJMM-MO
2	0.05563	0.05498	0.055573	0.04730
3	0.08269	0.08308	0.082648	0.07017
4	0.09974	0.10064	0.099692	0.08588
5	0.11180	0.11294	0.111778	0.09183
7	0.12832	0.12953	0.128288	0.10404
8	0.13434	0.13557	0.13431	0.10779
11	0.14758	0.14872	0.147566	0.11537

Open questions

- Partitioning more
- Mixture of both directions
- Cyclic structure

- Rank metric ISD
- Other metrics

Open questions

- Partitioning more
- Cyclic structure
  Tillich and Canto-Torres: *Speeding up decoding a code with a non-trivial automorphism group up to an exponential factor*
- Rank metric ISD
- Other metrics
  BONUS ROUND

Bonus Round: Lee metric

Joint work with Annalena Horlemann-Trautmann

Joint work with Marco Baldi, Franco Chiaraluce, Paolo Santini, Massimo Battaglioni

Definitions

We will consider the integer-residue ring $\mathbb{Z}/m\mathbb{Z} := \mathbb{Z}_m$, and for $x \in \mathbb{Z}_m$ we will always consider its representative in $\{0, \ldots, m-1\}$.

Definition (Lee weight)

For $x \in \mathbb{Z}_m$, the Lee weight is defined as

$$wt_L(x) = \min\{x, m-x\}.$$

For $v \in \mathbb{Z}_m^n$, the Lee weight is defined as

$$wt_L(v) = \sum_{i=1}^{n} wt_L(v_i).$$

Definition (Lee distance)

For $x, y \in \mathbb{Z}_m$, the Lee distance is defined as

$$d_L(x, y) = wt_L(x - y) = \min\{\mid x - y \mid, m - \mid x - y \mid\}.$$

For $v, w \in \mathbb{Z}_m^n$, the Lee distance is defined as

$$d_L(v, w) = \sum_{i=1}^{n} d_L(v_i, w_i).$$

Example

Example (Lee weight in $\mathbb{Z}_7$)

x	0	1	2	3	4	5	6
$wt_L(x)$	0	1	2	3	3	2	1

Quaternary codes

The case $m = 4$ is the most studied one in *ring linear* coding theory.

Definition (Quaternary Codes)

We say that $\mathcal{C}$ is a quaternary code of length n, if $\mathcal{C}$ is an additive subgroup of $\mathbb{Z}_4^n$.

Reason: We have a connection between ring linear coding theory and classical coding theory over finite fields:

Definition (Gray Isometry)

$$\begin{aligned} \phi : (\mathbb{Z}_4, wt_L) &\rightarrow (\mathbb{F}_2^2, wt_H) \\ 0 &\mapsto (0,0), \\ 1 &\mapsto (0,1), \\ 2 &\mapsto (1,1), \\ 3 &\mapsto (1,0). \end{aligned}$$

The Gray isometry can be extended componentwise to

$$\overline{\phi} : (\mathbb{Z}_4^n, wt_L) \rightarrow (\mathbb{F}_2^{2n}, wt_H).$$

Main Differences

Main changes from $\mathbb{F}_q$ to $\mathbb{Z}_4$

- Systematic form

$\Rightarrow$ Have to change the algorithm

- Amount of vectors having fixed Lee weight

$\Rightarrow$ New concept of intermediate sums and success probability

- Systematic form

Definition (Parity Check matrix)

Let $\mathcal{C}$ be a quaternary code of length n and type $|\mathcal{C}| = 4^{k_1}2^{k_2}$. $\mathcal{C}$ has a $(n-k_1)\times n$ parity check matrix

$$H = \begin{pmatrix} D & E & Id_{n-k_1-k_2} \\ 2F & 2Id_{k_2} & 0 \end{pmatrix},$$

where $D \in \mathbb{Z}_4^{(n-k_1-k_2)\times k_1}$, $E \in \mathbb{Z}_2^{(n-k_1-k_2)\times k_2}$, $F \in \mathbb{Z}_2^{k_2\times k_1}$.

- Systematic form

Definition (Generator matrix)

Let $\mathcal{C}$ be a quaternary code of length n and type $|\mathcal{C}| = 4^{k_1}2^{k_2}$. $\mathcal{C}$ has a $(k_1 + k_2) \times n$ generator matrix

$$G = \begin{pmatrix} Id_{k_1} & A & B \\ 0 & 2Id_{k_2} & 2C \end{pmatrix},$$

where $A \in \mathbb{Z}_2^{k_1 \times k_2}$, $B \in \mathbb{Z}_4^{k_1 \times (n-k_1-k_2)}$, $C \in \mathbb{Z}_2^{k_2 \times (n-k_1-k_2)}$.

Main Differences

- Amount of vectors having fixed Lee weight

The amount of all vectors in $\mathbb{Z}_4^n$ having Lee weight w is

$$c(n,w) = \sum_{i=0}^{\lfloor w/2 \rfloor} \binom{n}{i} \binom{n-i}{w-2i} 2^{w-2i}.$$

$$c(n,w) = \binom{2n}{w}.$$

Hence many concepts can be easily translated using the Gray isometry, getting the parameter $2n$ instead of n.

- Amount of vectors having fixed Lee weight

The amount of all vectors in $\mathbb{Z}_4^n$ having Lee weight w is

$$c(n,w) = \sum_{i=0}^{\lfloor w/2 \rfloor} \binom{n}{i} \binom{n-i}{w-2i} 2^{w-2i}.$$

$$c(n,w) = \binom{2n}{w}.$$

Hence many concepts can be easily translated using the Gray isometry, getting the parameter $2n$ instead of n.

GV bound

Again using the Gray isometry, we get a Gilbert-Varshamov bound for quaternary codes:

Proposition (Quaternary GV bound)

Let $\mathcal{C}$ be a quaternary code of length n and minimum Lee distance d, then

$$|\mathcal{C}| \geq \frac{4^n}{\sum_{j=0}^{d-1} \binom{2n}{j}}.$$

Theoretical Key Sizes

Assuming quaternary GV bound and half minimum distance error correction:
Theoretical parameters for the McEliece cryptosystem using quaternary codes:

system	n	k	k_1	k_2	t	Key size	security
quaternary	425		55	370	42	20355	128
binary	425	240			42	44400	62
binary	850	2240			42	146400	37
Goppa	2960	2288				1537536	128

Difficulties and open questions

- Find a suitable quaternary code
  For example Kerdock codes are having length $n = 2^m$ has a generator matrix of size $(2 + m) \times 2^m$. Increasing m to $m + 1$ gives exponential growth in the key size and adds 2 bits to the security level of m.
- Generalizing ISD algorithms to $\mathbb{Z}_{p^m}$
  It is even difficult to compute an exact formula for the amount of vectors in $\mathbb{Z}_{p^m}^n$ having Lee weight w.
- NP completeness:

  1978 Berlekamp, McEliece, van Tilborg: Syndrome Decoding Problem in the binary is NP-complete

  1994 Barg: Syndrome Decoding Problem over $\mathbb{F}_q$ is NP-complete

  2016 Gaborit, Zemor: *Probabilistic* Proof of NP-completeness of the rank metric Syndrome Decoding Problem
- Other metrics?

Thank you!