How Easy is Code Equivalence over $GF(q)$?

Dimitris E. Simos
(joint work with Nicolas Sendrier)

Project-Team SECRET
INRIA Paris-Rocquencourt

October 10, 2012
Journées Codage et Cryptographie (C2)
Dinard, France

Outline of the Talk

1. Code Equivalence Problem
   - Motivation
   - Previous Work

Outline of the Talk

1. Code Equivalence Problem
   - Motivation
   - Previous Work
2. Support Splitting Algorithm
   - Mechanics
   - Generalization

Outline of the Talk

1. Code Equivalence Problem
   - Motivation
   - Previous Work
2. Support Splitting Algorithm
   - Mechanics
   - Generalization
3. Research Problems

Code Equivalence of Linear Codes

Equivalence of Linear Codes over $\mathbb{F}_q$

- Two linear codes $C, C' \subseteq \mathbb{F}_q^n$ are called semi-linear equivalent if there exist a permutation σ of $I_n = \{1, \dots, n\}$, an n-tuple $\lambda = (\lambda_i)_{i \in I_n}$ of $(\mathbb{F}_q^*)^n$ and a field automorphism $\alpha \in \mathrm{Aut}(\mathbb{F}_q)$:

$$(x_i)_{i \in I_n} \in C \iff (\alpha(\lambda_{\sigma^{-1}(i)} x_{\sigma^{-1}(i)}))_{i \in I_n} \in C'$$

- If q is prime, $Aut(\mathbb{F}_q)$ is trivial $\Longrightarrow$ C is linear equivalent to C'
- If $q = 2$, $\lambda_i = 1$, $i \in I_n \Longrightarrow$ C is permutation equivalent to C'
- **Notation:** $C \sim C'$

Code Equivalence Problem

- Input: Two $[n, k]$ linear codes C and C' over $\mathbb{F}_q$
- Decide: Are $C \sim C'$?
- Search: Given $C \sim C'$, find $\sigma \in \mathcal{S}_n, \lambda \in (\mathbb{F}_q^*)^n, \alpha \in \mathrm{Aut}(\mathbb{F}_q)$

Motivation for Code Equivalence

Relation to Error-Correcting Capability

Equivalent codes have the same error-correction properties (i.e. decoding)

Classification

Enumeration of equivalence classes of linear codes

Application in Code-based Cryptography

- The public key of the McEliece cryptosystem is a randomly permuted binary Goppa code [McEliece, 1978]
- McEliece-like cryptosystems over $\mathbb{F}_q$ have recently emerged
  - Wild Goppa codes [Bernstein, Lange and Peters, 2010]
- Identification schemes from error-correcting codes
  - Zero-knowledge protocols [Girault, 1990]

What is known for Code Equivalence?

Algorithms and Complexity

Complexity

PCE over $\mathbb{F}_2$ is difficult to decide in the worst case:

1. not NP-complete
2. at least as hard as GRAPH ISOMORPHISM [Petrank and Roth, 1997]
3. Recent result for $\mathbb{F}_q$: GI $\preceq$ PCE [Grochow, 2012]
4. Assuming an oracle for LCE or SLCE $\Longrightarrow$ PCE $\preceq$ LCE or SLCE
5. PCE over $\mathbb{F}_q$ resists quantum Fourier sampling; Reduction of PCE to the HIDDEN SUBGROUP PROBLEM [Dinh, Moore and Russell, 2011]

Recent Algorithms

- Adaptation of Hypergraph Isomorphism algorithms for PCE over $\mathbb{F}_q$ [Babai, Codenotti and Grochow, 2011]
- Computation of canonical forms of linear codes for LCE over $\mathbb{F}_q$, for q small [Feulner, 2009, 2011]
- Support splitting algorithm for PCE over $\mathbb{F}_q$ [Sendrier, 2000]
- No efficient algorithm for LCE or SLCE is known

Invariants and Signatures

for a given Linear Code

Invariants of a Code

- A mapping $\mathcal{V}$ is an invariant if $C \sim C' \Rightarrow \mathcal{V}(C) = \mathcal{V}(C')$
- Any invariant is a global property of a code

Weight Enumerators are Invariants

- $C \sim C' \Rightarrow \mathcal{W}_C(X) = \mathcal{W}_{C'}(X)$ or $\mathcal{W}_C(X) \neq \mathcal{W}_{C'}(X) \Rightarrow C \not\sim C'$
- $\mathcal{W}_C(X) = \sum_{i=0}^{n} A_i X^i$ and $A_i = \mid \{c \in C \mid w(c) = i\} \mid$

Signature of a Code

- A mapping S is a signature if $S(\sigma(C), \sigma(i)) = S(C, i)$
- Property of the code and one of its positions (local property)

Building a Signature from an Invariant

1. If $\mathcal{V}$ is an invariant, then $S_{\mathcal{V}} : (C, i) \mapsto \mathcal{V}(C_{\{i\}})$ is a signature
2. where $C_{\{i\}}$ is obtained by puncturing the code C on i
3. If $C' = \sigma(C) \Rightarrow \mathcal{V}(C_{\{i\}}) = \mathcal{V}(C'_{\{\sigma(i)\}})$, $\forall\ i \in I_n$, i.e. $\mathcal{V} = \mathcal{W}$

The Support Splitting Algorithm (I)

Design of the Algorithm

Discriminant Signatures

1. A signature S is discriminant for C if $\exists\, i \neq j, S(C,i) \neq S(C,j)$
2. S is fully discriminant for C if $\forall\, i \neq j, S(C,i) \neq S(C,j)$

The Procedure [Sendrier, 2000]

- From given signature S and code C, we wish to build a sequence $S_0 = S, S_1, \ldots, S_r$ of signatures of increasing "discriminancy" such that S_r is fully discriminant for C
- Achieved by succesive refinements of the signature S

Properties of $\mathcal{SSA}$

1. $\mathcal{SSA}(C)$ **returns** a labeled partition $\mathcal{P}(S, C)$ of I_n
2. Assuming the existence of a fully discriminant signature, $\mathcal{SSA}(C)$ recovers the desired permutation σ of $C' = \sigma(C)$

Fully Discriminant Signatures

Statement

If $C' = \sigma(C)$ and S is fully discriminant for C then $\forall\, i \in I_n \; \exists$ unique $j \in I_n$ such that $S(C,i) = S(C',j)$ and $\sigma(i) = j$

An Example of a Fully Discriminant Signature

$$C = \{1110, 0111, 1010\} \text{ and } C' = \{0011, 1011, 1101\}$$

$$\begin{cases} C_{\{1\}} = \{110, 111, 010\} & \rightarrow \quad \mathcal{W}_{C_{\{1\}}}(X) = X + X^2 + X^3 \\ C_{\{2\}} = \{110, 011\} & \rightarrow \quad \mathcal{W}_{C_{\{2\}}}(X) = 2X^2 \\ C_{\{3\}} = \{110, 011, 100\} & \rightarrow \quad \mathcal{W}_{C_{\{3\}}}(X) = X + 2X^2 \\ C_{\{4\}} = \{111, 011, 101\} & \rightarrow \quad \mathcal{W}_{C_{\{4\}}}(X) = 2X^2 + X^3 \end{cases}$$

$$\begin{cases} C'_{\{1\}} = \{011, 101\} & \rightarrow \quad \mathcal{W}_{C'_{\{1\}}}(X) = 2X^2 \\ C'_{\{2\}} = \{011, 111, 101\} & \rightarrow \quad \mathcal{W}_{C'_{\{2\}}}(X) = 2X^2 + X^3 \\ C'_{\{3\}} = \{001, 101, 111\} & \rightarrow \quad \mathcal{W}_{C'_{\{3\}}}(X) = X + X^2 + X^3 \\ C'_{\{4\}} = \{001, 101, 110\} & \rightarrow \quad \mathcal{W}_{C'_{\{4\}}}(X) = X + 2X^2 \end{cases}$$

$C' = \sigma(C)$ where $\sigma(1) = 3$, $\sigma(2) = 1$, $\sigma(3) = 4$ and $\sigma(4) = 2$

How to Refine a Signature

An Example of a Refined Signature

$$\begin{array}{rcl} C & = & \{01101, 01011, 01110, 10101, 11110\} \\ C' & = & \{10101, 00111, 10011, 11100, 11011\} \end{array}$$

$$\left\{ \begin{array}{lclcll} \mathcal{W}_{C_{\{1\}}}(X) & = & X^2 + 3X^3 & = & \mathcal{W}_{C'_{\{2\}}}(X) & \Rightarrow \sigma(1) = 2 \\ \mathcal{W}_{C_{\{4\}}}(X) & = & 2X^2 + 3X^3 & = & \mathcal{W}_{C'_{\{4\}}}(X) & \Rightarrow \sigma(4) = 4 \\ \mathcal{W}_{C_{\{5\}}}(X) & = & 3X^2 + X^3 + X^4 & = & \mathcal{W}_{C'_{\{3\}}}(X) & \Rightarrow \sigma(5) = 3 \\ \mathcal{W}_{C_{\{2\}}}(X) & = & 3X^2 + 2X^3 & = & \mathcal{W}_{C'_{\{1\}}}(X) & \\ \mathcal{W}_{C_{\{3\}}}(X) & = & 3X^2 + 2X^3 & = & \mathcal{W}_{C'_{\{5\}}}(X) & \end{array} \right.$$

Refinement: Positions $\{2, 3\}$ in C and $\{1, 5\}$ in C' cannot be discriminated, but

$$\left\{ \begin{array}{lclcll} \mathcal{W}_{C_{\{1,2\}}}(X) & = & 3X^2 & = & \mathcal{W}_{C'_{\{2,5\}}}(X) & \Rightarrow \sigma(\{1, 2\}) = \{2, 5\} \\ \mathcal{W}_{C_{\{1,3\}}}(X) & = & X + 2X^2 + X^3 & = & \mathcal{W}_{C'_{\{2,1\}}}(X) & \Rightarrow \sigma(\{1, 3\}) = \{2, 1\} \end{array} \right.$$

Thus $\sigma(1) = 2$, $\sigma(2) = 5$, $\sigma(3) = 1$, $\sigma(4) = 4$ and $\sigma(5) = 3$

Fundamental Properties of $\mathcal{SSA}$

1. If $C' = \sigma(C)$ then $\mathcal{P}'(S, C') = \sigma(\mathcal{P}(S, C))$
2. The **output** of $\mathcal{SSA}(C)$ where $C =< G >$ is independent of G

The Support Splitting Algorithm (II)

Practical Issues

A Good Signature

The mapping $(C, i) \mapsto \mathcal{W}_{\mathcal{H}(C_i)}(X)$ where $\mathcal{H}(C) = C \cap C^{\perp}$ is a signature which is, for random codes,

- easy to compute because of the small dimension [Sendrier, 1997]
- discriminant, i.e. $\mathcal{W}_{\mathcal{H}(C_i)}(X)$ and $\mathcal{W}_{\mathcal{H}(C_j)}(X)$ are “often” different

Algorithmic Cost

Let C be a binary code of length n, and let $h = \dim(\mathcal{H}(C))$:

- First step: $\mathcal{O}(n^3) + \mathcal{O}(n2^h)$
- Each refinement: $\mathcal{O}(hn^2) + \mathcal{O}(n2^h)$
- Number of refinements: $\approx \log n$

Total (heuristic) complexity: $\mathcal{O}(n^3 + 2^h n^2 \log n)$

- When $h \longrightarrow 0 \Longrightarrow \mathcal{SSA}$ runs in polynomial time

The Closure of a Linear Code (I)

Approach for the Generalization of $\mathcal{SSA}$

- Reduce LCE or SLCE to PCE
- Recall that $\mathcal{SSA}$ solves PCE in $\mathcal{O}(n^3)$ (for "several" instances)

Closure of a Code

Let p be a primitive element of $\mathbb{F}_q$. The closure $\overline{C}$ of a code $C \subseteq \mathbb{F}_q^n$ is a code of length $(q-1)n$ over the same field where:

$$(x_1, \ldots, x_n) \in C \Longrightarrow (px_1, \ldots, p^{q-1}x_1, \ldots, px_n, \ldots, p^{q-1}x_n) \in \overline{C}$$

Fundamental Properties of the Closure

- If $C \sim C'$ w.r.t. LCE $\Longrightarrow \overline{C} \sim \overline{C'}$ w.r.t. PCE
- $\exists$ a block-wise permutation σ of $\mathcal{M} \triangleleft \mathcal{S}_{(q-1)n}$ such that $\overline{C'} = \sigma(\overline{C})$
- If C is an $[n, k, d]$ code $\Longrightarrow \overline{C}$ is an $[(q-1)n, k, (q-1)d]$ code

The Closure of a Linear Code (II)

The Closure is a Weakly Self-Dual Code

$\forall\ \overline{x}, \overline{y} \in \overline{C}$ the Euclidean inner product is

$$\overline{x} \cdot \overline{y} = \underbrace{\left(\sum_{j=1}^{q-1} p^{2j} \right)}_{=0 \text{ over } \mathbb{F}_q,\ q \geq 5} \left(\sum_i x_i y_i \right) = 0$$

- Clearly $\dim(\mathcal{H}(\overline{C})) = \dim(\overline{C})$ and $\mathcal{SSA}$ runs in $\mathcal{O}(2^{\dim(\mathcal{H}(\overline{C}))})$
- The closure reduces LCE to the hard instances of $\mathcal{SSA}$ for PCE
- Exceptions are for $q = 3$ and $q = 4$ with the Hermitian inner product

Building Efficient Invariants from the Closure

- For any invariant $\mathcal{V}$ the mapping $C \longmapsto \mathcal{V}(\mathcal{H}(\overline{C}))$ is an invariant
- The dimension of the hull over $\mathbb{F}_q$ is on average a small constant

The Extension of the Dual Code

Extension of the Dual

Let β be a primitive element of $\mathbb{F}_q$ and $C^\perp$ the dual code of $C \subseteq \mathbb{F}_q^n$. Define $\widehat{C}_i = \{\beta^i x \mid \beta \in \mathbb{F}_q^*, x \in C^\perp\}$. The extension of the dual code is a code of length $(q-1)n$ and dimension $(q-1)(n-k)$ where $\dim(C) = k$ and is given by the direct sum

$$\widehat{C} = \bigoplus_{i=1}^{q-1} \widehat{C}_i = \widehat{C}_1 \bigoplus \dots \bigoplus \widehat{C}_{q-1}$$

Fundamental Properties of the Extension

- If $C^\perp \sim C'^\perp$ w.r.t. $\mathrm{LCE} \Longrightarrow \widehat{C} \sim \widehat{C'}$ w.r.t. PCE
- $\overline{\mathcal{H}(C)} = \overline{C} \cap \widehat{C}$
- If $\dim(\mathcal{H}(C)) = h \Longrightarrow \dim(\overline{C} \cap \widehat{C}) = h$

Towards a Generalization of $\mathcal{SSA}$

A Good Signature for $\mathbb{F}_3$ and $\mathbb{F}_4$

- $\overline{\mathcal{H}(C)} = \mathcal{H}(\overline{C}) = \overline{C} \cap \widehat{C}$ (valid only for these fields)
- $S(\overline{C}, i) = \mathcal{W}_{\mathcal{H}(\overline{C_i})}(X)$

An Efficient Algorithm for Solving LCE

- **Input**: C, C', S

1. Compute $\overline{C}, \overline{C'}$ and $\widehat{C}, \widehat{C'}$
2. $\mathcal{P}(S, \overline{C}) \longleftarrow \mathcal{SSA}(\overline{C})$ and $\mathcal{P}'(S, \overline{C'}) \longleftarrow \mathcal{SSA}(\overline{C'})$
3. If $\mathcal{P}'(S, \overline{C'}) = \sigma(\mathcal{P}(S, \overline{C}))$ return σ; else $C \nsim C'$ w.r.t. LCE
4. $\overline{C'} = \sigma(\overline{C})$ and a Gaussian elimination (GE) on the permuted generator matrices of the closures will reveal the scaling coefficients

- For SLCE we only have to consider an additional GE

Generalized Hulls of Linear Codes

What about $\mathbb{F}_q$, $q \geq 5$?

- If $C \sim C'$ w.r.t. LCE or SLCE $\Longrightarrow \mathcal{H}(C) \sim \mathcal{H}(C')$ w.r.t. LCE or SLCE is **not** true
- The hull is not an invariant for LCE or SLCE over $\mathbb{F}_q$, $q \geq 5$

The Generalized Hull

Let $C \subseteq \mathbb{F}_q^n$ and an n-tuple $a = (a_i)_{i \in I_n}$ of $(\mathbb{F}_q^*)^n$. Define the dual code $C_a^\perp = \{x \bullet c = 0 \mid x \in \mathbb{F}_q^n, c \in C\}$ w.r.t. to the inner product

$$x \bullet y = \sum_{i=1}^{n} a_i x_i y_i$$

- Hull w.r.t. a: $\mathcal{H}_a(C) = C \cap C_a^\perp$
- If we consider all $a \in (\mathbb{F}_q^*)^n$ we obtain $(q-1)^n$ different hulls
- The generalized hull is an invariant for LCE

Research Problems

Related to the Closure

- If $\overline{C'} = \sigma(\overline{C})$ for some σ of $\mathcal{M} \triangleleft \mathcal{S}_{(q-1)n}$ what is the structure of the subgroup $\mathcal{M}$?
- Other reductions of LCE or SLCE to PCE?

Conjecture

- LCE or SLCE seems to be hard over $\mathbb{F}_q$, $q \geq 5$
- Can we build zero-knowledge protocols based on the hardness of LCE or SLCE?

Related to the Generalized Hull

- Can we find a practical application of $\mathcal{H}_a(C)$?

Summary

Highlights

1. We defined the closure of a linear code and the extension of its dual
2. We presented a generalization of the support splitting algorithm for solving the LINEAR CODE EQUIVALENCE problem for $\mathbb{F}_3$ and $\mathbb{F}_4$
3. We conjectured that the (SEMI)-LINEAR CODE EQUIVALENCE problem over $\mathbb{F}_q$, $q \geq 5$ is hard on the average case

Summary

Highlights

1. We defined the closure of a linear code and the extension of its dual
2. We presented a generalization of the support splitting algorithm for solving the LINEAR CODE EQUIVALENCE problem for $\mathbb{F}_3$ and $\mathbb{F}_4$
3. We conjectured that the (SEMI)-LINEAR CODE EQUIVALENCE problem over $\mathbb{F}_q$, $q \geq 5$ is hard on the average case

Future Work

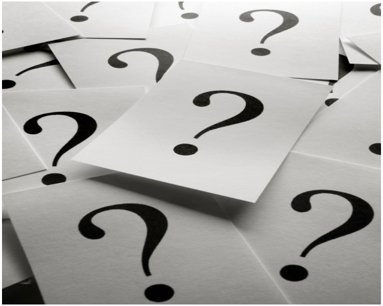
Solve (some) of the research problems..!

References

- László Babai, Paolo Codenotti, Joshua Grochow and Youming Qiao, "Code equivalence and group isomorphism," In Proc. 22nd Ann. Symp. on Discrete Algorithms (SODA 2011), pages 1395-1408. ACM-SIAM, 2011.
- D. J. Bernstein, T. Lange and C. Peters, "Wild McEliece," In SAC 2010, Lecture Notes in Computer Science, vol. 6544, pp. 143–158. Springer-Verlag, 2011.
- E. Petrank and R. M. Roth, "Is code equivalence easy to decide?," IEEE Trans. Inf. Theory, vol. 43, pp. 1602–1604, 1997.
- N. Sendrier, "On the dimension of the hull," SIAM J. Discete Math., vol. 10, pp. 282–293, 1997.
- N. Sendrier, "Finding the permutation between equivalent codes: the support splitting algorithm," IEEE Trans. Inf. Theory, vol. 46, pp. 1193–1203, 2000.

Questions - Comments

Thanks for your Attention!



Merci Beaucoup!