IRIF
INSTITUT
DE RECHERCHE
EN INFORMATIQUE
FONDAMENTALE

Université Paris Cité

Inria

GENERALIZED SYNDROME DECODING PROBLEM AND ITS APPLICATION TO POST-QUANTUM CRYPTOGRAPHY

PhD thesis in theoretical computer science

28 June 2023, Paris

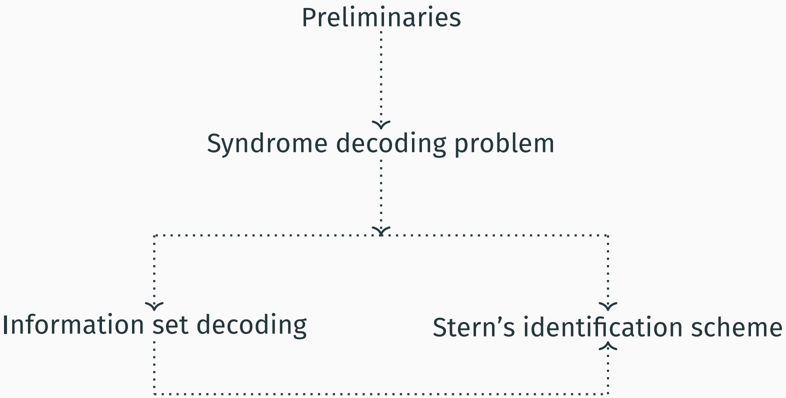
Candidate: Simona Etinski

Advisors: André Chailloux,
Frédéric Magniez

Reviewers: Daniel Augot,
Elena Kirshanova

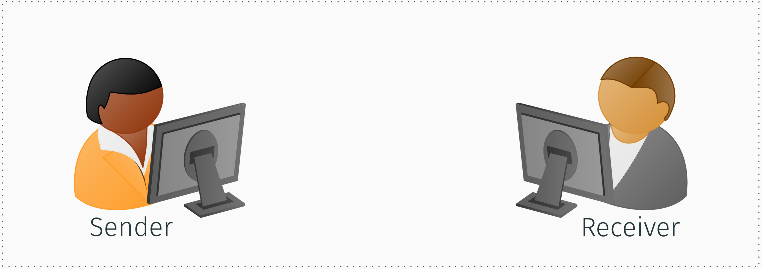
Examiners: Sophie Laplante,
Nicolas Resch,
Nicolas Sendrier

OUTLINE


Preliminaries
Syndrome decoding problem
Information set decoding
Stern's identification scheme

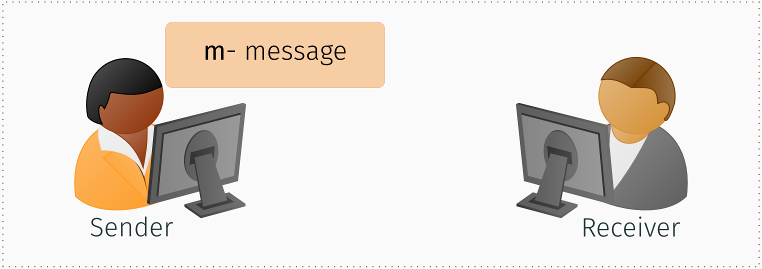
PRELIMINARIES

Coding theory



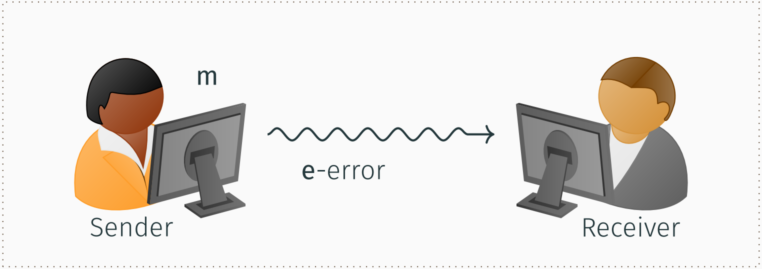
Basic setting

CODING THEORY



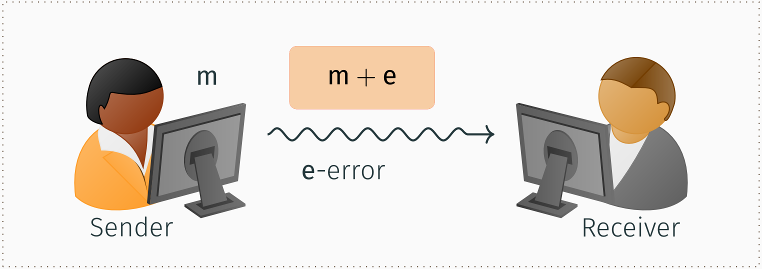
Basic setting

Coding theory



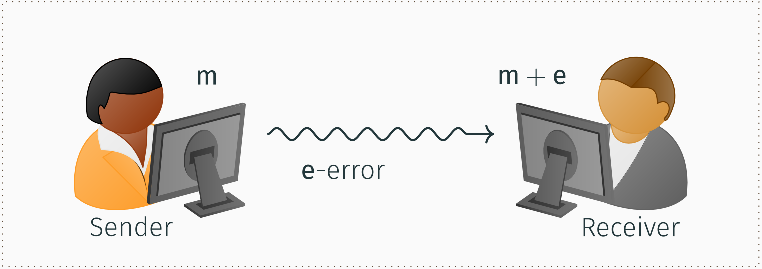
Basic setting

Coding theory



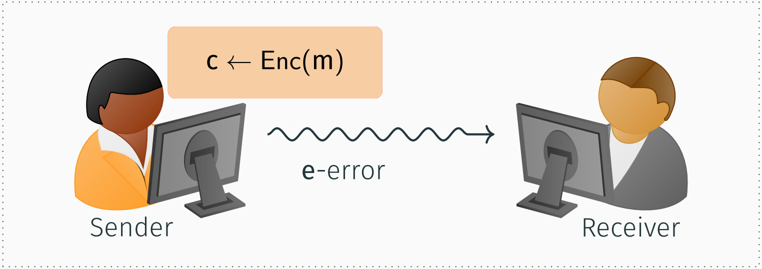
Basic setting

Coding theory



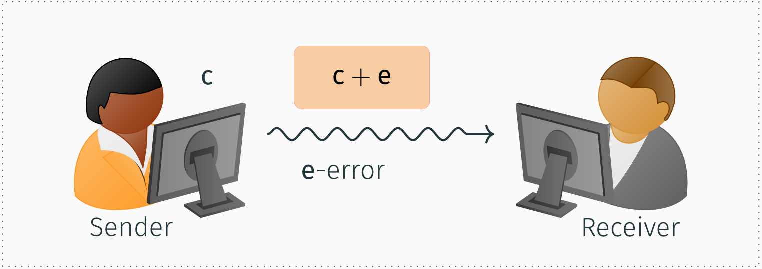
Basic setting

CODING THEORY



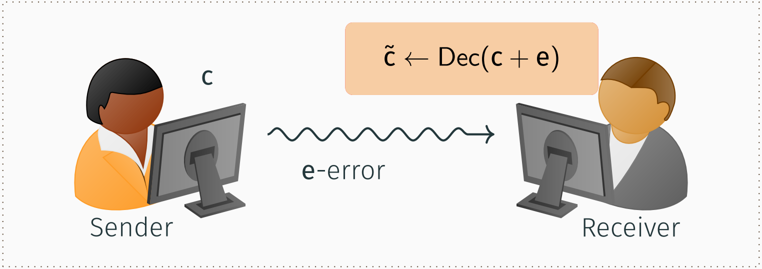
Basic setting

CODING THEORY



Basic setting

CODING THEORY



Basic setting

Message encoding

- message $\mathbf{m}$ of length k, with symbols from alphabet of size q

$$\rightarrow \boxed{\mathbf{m} \in \mathbb{F}_q^k}$$

Message encoding

- message $\mathbf{m}$ of length k, with symbols from alphabet of size q

$$\rightarrow \mathbf{m} \in \mathbb{F}_q^k$$

- codeword $\mathbf{c}$ of length n, with symbols from alphabet of size q

$$\rightarrow \mathbf{c} \in \mathbb{F}_q^n$$

Message encoding

- message $\mathbf{m}$ of length k, with symbols from alphabet of size q

$$\rightarrow \mathbf{m} \in \mathbb{F}_q^k$$

- codeword $\mathbf{c}$ of length n, with symbols from alphabet of size q

$$\rightarrow \mathbf{c} \in \mathbb{F}_q^n$$

- encoding algorithm Enc that maps message into codeword

$$\rightarrow \mathsf{Enc} : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$$

Message encoding

Encoding is commonly defined via a **generator matrix**, $\mathsf{G} \in \mathbb{F}_q^{k \times n}$:

$$\forall \mathsf{m} \in \mathbb{F}_q^k, \quad \mathsf{Enc}(\mathsf{m}) := \mathsf{m}^\mathsf{T}\mathsf{G}.$$

Message encoding

Encoding is commonly defined via a **generator matrix**, $\mathsf{G} \in \mathbb{F}_q^{k \times n}$:

$$\forall \mathbf{m} \in \mathbb{F}_q^k, \quad \mathsf{Enc}(\mathbf{m}) := \mathbf{m}^\top \mathsf{G}.$$

A **code**, $\mathcal{C}$, is then defined as:

$$\mathcal{C} := \{\mathbf{c} \in \mathbb{F}_q^n \mid (\exists \mathbf{m} \in \mathbb{F}_q^k)\ \mathbf{c} = \mathsf{Enc}(\mathbf{m})\}.$$

Message encoding

Equivalently, linear code can be defined via a **parity check matrix**, $\mathsf{H} \in \mathbb{F}_q^{(n-k)\times n}$, which is a matrix of maximal rank that satisfies:

$$\mathsf{H}\mathsf{G}^\top = 0.$$

Message encoding

Equivalently, linear code can be defined via a **parity check matrix**, $\mathsf{H} \in \mathbb{F}_q^{(n-k)\times n}$, which is a matrix of maximal rank that satisfies:

$$\mathsf{H}\mathsf{G}^\top = 0.$$

A **code**, $\mathcal{C}$, is then defined as:

$$\mathcal{C} := \{\mathsf{c} \in \mathbb{F}_q^n \mid \mathsf{H}\mathsf{c} = 0\}.$$

Message decoding

- error $\mathbf{e}$ of length n, with symbols from alphabet of size q

$$\rightarrow \boxed{\mathbf{e} \in \mathbb{F}_q^n}$$

Message decoding

- error $\mathbf{e}$ of length n, with symbols from alphabet of size q

$$\rightarrow \boxed{\mathbf{e} \in \mathbb{F}_q^n}$$

- noisy codeword $\tilde{\mathbf{c}} := \mathbf{c} + \mathbf{e}$ of length n, with symbols from alphabet of size q

$$\rightarrow \boxed{\tilde{\mathbf{c}} \in \mathbb{F}_q^n}$$

Message decoding

- error $\mathbf{e}$ of length n, with symbols from alphabet of size q

$$\rightarrow \boxed{\mathbf{e} \in \mathbb{F}_q^n}$$

- noisy codeword $\tilde{\mathbf{c}} := \mathbf{c} + \mathbf{e}$ of length n, with symbols from alphabet of size q

$$\rightarrow \boxed{\tilde{\mathbf{c}} \in \mathbb{F}_q^n}$$

- decoding algorithm Dec that maps noisy codeword, $\tilde{\mathbf{c}}$, into codeword $\mathbf{c} \in \mathcal{C}$

$$\rightarrow \boxed{\mathsf{Dec} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^n}$$

Hamming weight

Hamming distance, $\mathrm{dist}_H(\cdot)$

$$\forall \mathbf{c} = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n, \quad \forall \tilde{\mathbf{c}} = (\tilde{c}_0, \ldots, \tilde{c}_{n-1}) \in \mathbb{F}_q^n,$$
$$\mathrm{dist}_H(\mathbf{c}, \tilde{\mathbf{c}}) = |\{i \in [n] : c_i \neq \tilde{c}_i\}|$$

Hamming weight

Hamming distance, $\mathrm{dist}_H(\cdot)$

$$\forall \mathbf{c} = (c_0, \ldots, c_{n-1}) \in \mathbb{F}_q^n, \quad \forall \tilde{\mathbf{c}} = (\tilde{c}_0, \ldots, \tilde{c}_{n-1}) \in \mathbb{F}_q^n,$$
$$\mathrm{dist}_H(\mathbf{c}, \tilde{\mathbf{c}}) = |\{i \in [n] : c_i \neq \tilde{c}_i\}|$$

Hamming weight, $\mathrm{wt}_H(\cdot)$

$$\forall \mathbf{e} = (e_1, \ldots, e_n) \in \mathbb{F}_q^n, \quad \mathrm{wt}_H(\mathbf{e}) := \mathrm{dist}_H(\mathbf{e}, 0).$$

Message decoding

Decoding methods:

Message decoding

Decoding methods:

- **minimum distance decoding** - given the noisy codeword, $\tilde{\mathbf{c}}$, find the codeword, $\mathbf{c}$, at smallest **Hamming distance**;

Message decoding

Decoding methods:

- **minimum distance decoding** - given the noisy codeword, $\tilde{\mathbf{c}}$, find the codeword, $\mathbf{c}$, at smallest **Hamming distance**;
- **syndrome decoding**: calculate the syndrome, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, defined as:

$$\mathbf{s} := \mathbf{H}\tilde{\mathbf{c}} = \mathbf{H}(\mathbf{c} + \mathbf{e}) = \mathbf{H}\mathbf{e},$$

  find the error, $\mathbf{e}$, of the smallest **Hamming weight** that corresponds to $\mathbf{s}$.

SYNDROME DECODING PROBLEM (SDP)

Computational problem derived from the **syndrome decoding method**.

Syndrome Decoding Problem, SDP

Input – A parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a weight $w \in \mathbb{N}$.

Goal – Find an error $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{H}\mathbf{e}$ and $\mathrm{wt}(\mathbf{e}) = w$.

Syndrome Decoding Problem

An **NP-complete** problem.[1]

[1] Elwyn R. Berlekamp, Robert J. McEliece, and Henk C. A. van Tilborg. "On the inherent intractability of certain coding problems (Corresp.)". In: (1978), pp. 384–386. DOI: 10.1109/TIT.1978.1055873.

Syndrome Decoding Problem

An **NP-complete** problem.

For conveniently chosen parameters, the problem is exponentially hard for the best known **classical** and **quantum** algorithms.

Syndrome Decoding Problem

An **NP-complete** problem.

For conveniently chosen parameters, the problem is exponentially hard for the best known **classical** and **quantum** algorithms.

$\Rightarrow$ It is believed to be **post-quantum**.

Syndrome Decoding Problem

An **NP-complete** problem.

For conveniently chosen parameters, the problem is exponentially hard for the best known **classical** and **quantum** algorithms.

Used as basis of different cryptographic protocols.[1],[2]

[1] R. J. McEliece. "A Public-Key Cryptosystem Based On Algebraic Coding Theory". In: Deep Space Network Progress Report 44 (Jan. 1978), pp. 114–116.

[2] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Generalized Syndrome Decoding Problem, GSDP

Input – A parity check matrix $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, a syndrome $\mathbf{s} \in \mathbb{F}_q^{n-k}$, and a weight $w \in \mathbb{N}$.

Goal – Find an error $\mathbf{e} \in \mathbb{F}_q^n$ such that $\mathbf{s} = \mathbf{H}\mathbf{e}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

Elementwise weight functions, $\mathrm{wt}_M : \mathbb{F}_q^n \to \mathbb{N}$

$$\forall \mathbf{e} = (e_0, \ldots, e_{n-1}) \in \mathbb{F}_q^n, \quad \mathrm{wt}_M(\mathbf{e}) = \sum_i \mathrm{dist}(e_i, 0),$$

where $\mathrm{dist} : \mathbb{F}_q \times \mathbb{F}_q \to \mathbb{N}$ is a distance function (metric).

Examples of elementwise weight functions

Hamming distance, $\mathrm{dist}_H(\cdot,\cdot)$

$$\forall a, b \in \mathbb{F}_q, \quad \mathrm{dist}_H(a,b) = \begin{cases} 0, & a = b \\ 1, & \text{otherwise} \end{cases} .$$

Examples of elementwise weight functions

Hamming distance, $\mathrm{dist}_H(\cdot,\cdot)$

$$\forall a, b \in \mathbb{F}_q, \quad \mathrm{dist}_H(a,b) = \begin{cases} 0, & a = b \\ 1, & \text{otherwise} \end{cases}.$$

Hamming weight, $\mathrm{wt}_H(\cdot)$

$$\forall \mathbf{e} = (e_1, \dots, e_n) \in \mathbb{F}_q^n, \quad \mathrm{wt}_H(\mathbf{e}) = |\{i \in [n] : e_i \neq 0\}|.$$

Examples of elementwise weight functions

Lee distance, $\mathrm{dist}_L(\cdot,\cdot)$

$$\forall a, b \in \mathbb{F}_q, \quad \mathrm{dist}_L(a, b) = \min(|a - b|, q - |a - b|).$$

Examples of elementwise weight functions

Lee distance, $\mathrm{dist}_L(\cdot,\cdot)$

$$\forall a, b \in \mathbb{F}_q, \quad \mathrm{dist}_L(a,b) = \min(|a-b|, q-|a-b|).$$

Lee weight, $\mathrm{wt}_L(\cdot)$

$$\forall \mathbf{e} = (e_1, \ldots, e_n) \in \mathbb{F}_q^n, \quad \mathrm{wt}_L(\mathbf{e}) = \sum_i \mathrm{wt}_l(e_i).$$

Our goals

Estimate the **asymptotic complexity** of the generalized syndrome decoding problem.

Our goals

Estimate the **asymptotic complexity** of the generalized syndrome decoding problem.

Apply the generalized syndrome decoding problem to a **concrete cryptographic setting**.

INFORMATION SET DECODING (ISD)

Information Set Decoding

The best generic algorithms for solving the syndrome decoding problem.

Information Set Decoding

The best generic algorithms for solving the syndrome decoding problem.

Exploit the **linear structure** of the linear codes.

Algorithm Information set decoding

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $w, d, l \in \mathbb{N}$.

Output: $\mathbf{e} \in \mathbb{F}_q^{n}$ s.t. $\mathbf{He} = \mathbf{s}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

Algorithm Information set decoding

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $w, d, l \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ s.t. $\mathbf{He} = \mathbf{s}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

while e is not found **do**

Algorithm Information set decoding

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $w, d, l \in \mathbb{N}$.

Output: $\mathbf{e} \in \mathbb{F}_q^n$ s.t. $\mathbf{He} = \mathbf{s}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

while e is not found **do**

permutation step: permutes columns of $\mathbf{H}$

Algorithm Information set decoding

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $w, d, l \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ s.t. $\mathbf{He} = \mathbf{s}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

while $\mathbf{e}$ is not found **do**

permutation step: permutes columns of $\mathbf{H}$

partial Gaussian elimination step: given permuted $\mathbf{H}$ and $\mathbf{s}$, as well as d and l, creates a GSDP subinstance

Algorithm Information set decoding

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $w, d, l \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ s.t. $\mathbf{He} = \mathbf{s}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

while $\mathbf{e}$ is not found **do**

- **permutation step:** permutes columns of $\mathbf{H}$
- **partial Gaussian elimination step:** given permuted $\mathbf{H}$ and $\mathbf{s}$, as well as d and l, creates a GSDP subinstance
- **multi-solution GSDP step:** returns a list $\mathcal{L}$ of solution to the GSDP subinstance

Algorithm Information set decoding

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\mathbf{s} \in \mathbb{F}_q^{n-k}$, $w, d, l \in \mathbb{N}$.

Output: $\mathbf{e} \in \mathbb{F}_q^{n}$ s.t. $\mathbf{He} = \mathbf{s}$ and $wt_M(\mathbf{e}) = w$.

while e is not found **do**

- **permutation step:** permutes columns of **H**
- **partial Gaussian elimination step:** given permuted **H** and **s**, as well as d and l, creates a GSDP subinstance
- **multi-solution GSDP step:** returns a list $\mathcal{L}$ of solution to the GSDP subinstance
- **test step:** checks if any solution from the list $\mathcal{L}$ yields a solution to the original problem

end

return e

partial Gaussian elimination step: given permuted $\mathbf{H}$ and $\mathbf{s}$, as well as d and l, creates a GSDP subinstance

$$\mathbf{U}\mathbf{H}_\pi = \begin{pmatrix} \mathbf{I} & \mathbf{H}_1 \\ \mathbf{0} & \mathbf{H}_2 \end{pmatrix}, \quad \mathbf{U}\mathbf{s} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} \quad \Rightarrow \quad \begin{cases} \mathbf{e}_1 + \mathbf{H}_1\mathbf{e}_2 = \mathbf{s}_1 \\ \boxed{\mathbf{H}_2\mathbf{e}_2 = \mathbf{s}_2} \end{cases}.$$

where $\mathbf{e}_{\pi^{-1}} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}$ is a permuted solution to the problem

partial Gaussian elimination step: given permuted $\mathbf{H}$ and $\mathbf{s}$, as well as d and l, creates a GSDP subinstance

$$\mathbf{U}\mathbf{H}_\pi = \begin{pmatrix} \mathbf{I} & \mathbf{H}_1 \\ 0 & \mathbf{H}_2 \end{pmatrix}, \quad \mathbf{U}\mathbf{s} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} \quad \Rightarrow \quad \begin{cases} \mathbf{e}_1 + \mathbf{H}_1\mathbf{e}_2 = \mathbf{s}_1 \\ \mathbf{H}_2\mathbf{e}_2 = \mathbf{s}_2 \end{cases}.$$

where $\mathbf{e}_{\pi^{-1}} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}$ is a permuted solution to the problem

multi-solution GSDP step: return $\mathcal{L}$ as a list of solutions $\mathbf{e}_2$ to the GSDP-subinstance given on $(\mathbf{H}_2, \mathbf{s}_2, d)$

partial Gaussian elimination step: given permuted $\mathbf{H}$ and $\mathbf{s}$, as well as d and l, creates a GSDP subinstance

$$\mathbf{U}\mathbf{H}_\pi = \begin{pmatrix} \mathbf{I} & \mathbf{H}_1 \\ \mathbf{0} & \mathbf{H}_2 \end{pmatrix}, \quad \mathbf{U}\mathbf{s} = \begin{pmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \end{pmatrix} \quad \Rightarrow \quad \begin{cases} \mathbf{e}_1 + \mathbf{H}_1\mathbf{e}_2 = \mathbf{s}_1 \\ \mathbf{H}_2\mathbf{e}_2 = \mathbf{s}_2 \end{cases}.$$

where $\mathbf{e}_{\pi^{-1}} = \begin{pmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \end{pmatrix}$ is a permuted solution to the problem

multi-solution GSDP step: return $\mathcal{L}$ as a list of solutions $\mathbf{e}_2$ to the GSDP-subinstance given on $(\mathbf{H}_2, \mathbf{s}_2, d)$

test step: for each $\mathbf{e}_2 \in \mathcal{L}$, calculate $\mathbf{e}_1 \leftarrow \mathbf{s}_1 - \mathbf{H}_1\mathbf{e}_2$ and verify if $\mathrm{wt}_M(\mathbf{e}_1) = w - d$

Different (classical) ISD variants

ISD algorithms differ primarily in the last two steps of the algorithm, namely, **Multi-solution SDP step** and **Test step**.

Different (classical) ISD variants

ISD algorithms differ primarily in the last two steps of the algorithm, namely, **Multi-solution SDP step** and **Test step**.

Prange's algorithm[a] takes $\mathbf{e} \leftarrow \begin{pmatrix} s_1 \\ 0 \end{pmatrix}$ and verify if $\mathrm{wt}_M(\mathbf{e}) = w$.

[a] E. Prange. "The use of information sets in decoding cyclic codes". In: IRE Transactions on Information Theory (1962), pp. 5–9. DOI: 10.1109/TIT.1962.1057777.

Different (classical) ISD variants

ISD algorithms differ primarily in the last two steps of the algorithm, namely, **Multi-solution SDP step** and **Test step**.

Lee-Brickel's algorithm[a], for each $\mathbf{e}_2$ of weight d, calculates

$$\mathbf{e}_1 \leftarrow \mathbf{s}_1 - \mathbf{H}_1\mathbf{e}_2$$

and verify if $\mathrm{wt}_M(\mathbf{e}_1) = w - d$.

[a] Pil Joong Lee and Ernest F. Brickell. "An Observation on the Security of McEliece's Public-Key Cryptosystem". In: 1988.

Different (classical) ISD variants

ISD algorithms differ primarily in the last two steps of the algorithm, namely, **Multi-solution SDP step** and **Test step**.

Stern's/Dumer's algorithm[a], merges two lists of elements of weight $d/2$ to obtain a list, $\mathcal{L}$, of elements of weight d.

For each $\mathbf{e}_2$ in $\mathcal{L}$, the algorithm calculates

$$\mathbf{e}_1 \leftarrow \mathbf{s}_1 - \mathbf{H}_1\mathbf{e}_2$$

and verify if $\mathrm{wt}_M(\mathbf{e}_1) = w - d$.

[a] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Different (classical) ISD variants

ISD algorithms differ primarily in the last two steps of the algorithm, namely, **Multi-solution SDP step** and **Test step**.

Wagner's algorithm[a], for a chosen a, merges 2^a lists of elements of weight $d/2^a$ to obtain a list, $\mathcal{L}$, of elements of weight d.

For each $\mathbf{e}_2$ in $\mathcal{L}$, the algorithm calculates

$$\mathbf{e}_1 \leftarrow \mathbf{s}_1 - \mathbf{H}_1\mathbf{e}_2$$

and verify if $\mathrm{wt}_M(\mathbf{e}_1) = w - d$.

[a] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Our contributions: Part 1

Generalized ISD framework solving the generalized syndrome decoding problem.

Our contributions: Part 1

Generalized ISD framework solving the generalized syndrome decoding problem.

Derivation of a hybrid quantum-classical ISD algorithm.

Our contributions: Part 1

Generalized ISD framework solving the generalized syndrome decoding problem.

Derivation of a hybrid quantum-classical ISD algorithm.

Numerical results on the asymptotic analysis of the running time of ISD when solving GSDP over q-ary Hamming and Lee weight.

Classical ISD algorithms

The choice of l, d, and a give us the following algorithms:

Classical ISD algorithms

The choice of $l, \ d$, and a give us the following algorithms:

- $l = 0, \ d = 0, \ a = 1 \Rightarrow$ Prange's algorithm[3];

[3] E. Prange. "The use of information sets in decoding cyclic codes". In: IRE Transactions on Information Theory (1962), pp. 5–9. DOI: 10.1109/TIT.1962.1057777.

Classical ISD algorithms

The choice of $l,\ d$, and a give us the following algorithms:

- $l = 0,\ d = 0,\ a = 1 \Rightarrow$ Prange's algorithm;
- $l = 0,\ d \geq 0,\ a = 1 \Rightarrow$ Lee-Brickel's algorithm[3];

[3] Pil Joong Lee and Ernest F. Brickell. "An Observation on the Security of McEliece's Public-Key Cryptosystem". In: 1988.

Classical ISD algorithms

The choice of l, d, and a give us the following algorithms:

- $l = 0,\ d = 0,\ a = 1 \Rightarrow$ Prange's algorithm;
- $l = 0,\ d \geq 0,\ a = 1 \Rightarrow$ Lee-Brickel's algorithm ;
- $l \geq 0,\ d \geq 0,\ a = 1 \Rightarrow$ Stern's/Dumer's algorithm[3];

[3] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Classical ISD algorithms

The choice of l, d, and a give us the following algorithms:

- $l = 0,\ d = 0,\ a = 1 \Rightarrow$ Prange's algorithm;
- $l = 0,\ d \geq 0,\ a = 1 \Rightarrow$ Lee-Brickel's algorithm ;
- $l \geq 0,\ d \geq 0,\ a = 1 \Rightarrow$ Stern's/Dumer's algorithm ;
- $l \geq 0,\ d \geq 0,\ a \geq 1 \Rightarrow$ Wagner's algorithm[3].

[3] David A. Wagner. “A Generalized Birthday Problem”. In: ed. by Moti Yung. 2002, pp. 288–303. DOI: 10.1007/3-540-45708-9_19.

Input : $\mathbf{H} \in \mathbb{F}_q^{(n-k)\times n}$, $\quad \mathbf{s} \in \mathbb{F}_q^{n-k}$, $\quad w, d, l \in \mathbb{N}$.
Output: $\mathbf{e} \in \mathbb{F}_q^n$ s.t. $\mathbf{He} = \mathbf{s}$ and $\mathrm{wt}_M(\mathbf{e}) = w$.

while e is not found **do**

- **permutation step:** permutes columns of **H** — poly(n)
- **partial Gaussian elimination step:** given permuted **H** and **s**, as well as d and l, creates a GSDP subinstance — poly(n)
- **multi-solution GSDP step:** returns a list $\mathcal{L}$ of solution to the GSDP subinstance — T_{SUB}
- **test step:** checks if any solution from the list $\mathcal{L}$ yields a solution to the original problem — $|\mathcal{L}|\,\mathrm{poly}(n)$

end
return e

Running time of classical ISD algorithms

$$T_C(n, l, d, a) = \frac{\text{poly}(n) + T_{SUB}(n, l, d, a) + |\mathcal{L}|\,\text{poly}(n)}{p(n, l, d, a)},$$

where $p(\cdot, \cdot, \cdot, \cdot)$ is the probability of success in the test step.

Probability of success

$$p(n, l, d, a) = \min\left(1, \frac{\mathrm{surf}_M(q, n-k-l, w-d)}{\max\left(q^{n-k}, \mathrm{surf}_M(q, n, w)\right) q^{-l}} |\mathcal{L}|\right).$$

where

- $\mathrm{surf}_M(q, n, w)$ is the surface area of a sphere of radius w in $\mathbb{F}_q^n$,
- $\mathrm{surf}_M(q, n-k-l, w-d)$ is the surface area of a sphere of radius $w-d$ in $\mathbb{F}_q^{n-k-l}$.

Probability of success

$$p(n,l,d,a) = \min\left(1, \frac{\mathrm{surf}_M(q, n-k-l, w-d)}{\max\left(q^{n-k}, \mathrm{surf}_M(q,n,w)\right) q^{-l}} |\mathcal{L}|\right).$$

where

- $\mathrm{surf}_M(q,n,w)$ is the surface area of a sphere of radius w in $\mathbb{F}_q^n$,
- $\mathrm{surf}_M(q,n-k-l,w-d)$ is the surface area of a sphere of radius $w-d$ in $\mathbb{F}_q^{n-k-l}$.

Major obstacle: calculating the surface area of a sphere in a vector space endowed with arbitrary elementwise weight function.

Quantum Wagner's algorithm

A hybrid classical-quantum algorithm was obtained as a combination of:

- classical Wagner's algorithm,
- Grover's search[4],
- amplitude amplification[5].

[4] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: 1996, pp. 212–219. DOI: 10.1145/237814.237866.

[5] Gilles Brassard, Peter Høyer, et al. Quantum amplitude amplification and estimation. 2002.

Quantum Wagner's algorithm

Definition: Grover's algorithm[4]

Let $f : \{0,1\}^n \to \{0,1\}$ has an efficient classical description.

Grover's algorithm can find i such $f(i) = 1$ in time $O(poly(n)2^{n/2})$ if such an i exists and output 'no solution' otherwise.

[4] Lov K. Grover. "A Fast Quantum Mechanical Algorithm for Database Search". In: 1996, pp. 212–219. DOI: 10.1145/237814.237866.

Quantum Wagner's algorithm

Definition: Amplitude amplification[4]

Let $f : \{0,1\}^n \rightarrow \{0,1\}$ has an efficient classical description.

Consider an algorithm $\mathcal{A}$ that outputs i such that $f(i) = 1$ with probability p, and $f(i) = 0$ with probability $1 - p$.

Using amplitude amplification, one can find i such that $f(i) = 1$ by making $O(\frac{1}{\sqrt{p}})$ calls to $\mathcal{A}$.

[4] Gilles Brassard and Peter Hoyer. "An Exact Quantum Polynomial-Time Algorithm for Simon's Problem". In: 1997, pp. 12–23. DOI: 10.1109/ISTCS.1997.595153.

Quantum Wagner's algorithm

The difference appears only in the **multi-solution GSDP step** and **test step**:

Quantum Wagner's algorithm

The difference appears only in the **multi-solution GSDP step** and **test step**:

- in the **multi-solution GSDP step**, the algorithm returns a **description of a function** $f : [|\mathcal{L}|] \to \mathbb{F}_q^n$

Quantum Wagner’s algorithm

The difference appears only in the **multi-solution GSDP step** and **test step**:

- in the **multi-solution GSDP step**, the algorithm returns a **description of a function** $f : [|\mathcal{L}|] \to \mathbb{F}_q^n$
- in the **test step** the algorithm checks if any output of $f(\cdot)$ yields a solution to the original problem using **Grover’s search**

Quantum Wagner's algorithm

Input : $H \in \mathbb{F}_q^{(n-k)\times n}$, $s \in \mathbb{F}_q^{n-k}$, $w, d, l, a \in \mathbb{N}$.
Output : $e \in \mathbb{F}_q^n$ s.t. $He = s$ and $wt_M(e) = w$.

while e is not found **do**

- **permutation and partial Gaussian elimination step:** permute columns of H and create a GSDP subinstance — $\text{poly}(n)$
- **multi-solution GSDP step:** returns a **description of** $f : [|\mathcal{L}|] \to \mathbb{F}_q^n$ that outputs solutions to the GSDP subinstance — T_{SUB}
- **test step:** using **Grover's search**, checks if any output of $f(\cdot)$ yields a solution to the original problem — $\sqrt{|\mathcal{L}|}\,\text{poly}(n)$

end
return e

Quantum Wagner's algorithm

Running time

$$T_Q(n, l, d, a) = \frac{\mathrm{poly}(n) + T_{SUB}(n, l, d, a) + \sqrt{|\mathcal{L}|}\,\mathrm{poly}(n)}{\sqrt{p}(n, l, d, a)},$$

where p is the probability of success in the test step.

Numerical results

The asymptotic running time is evaluated when parameters l, d, and a are optimized to yield the shortest running time.

Numerical results

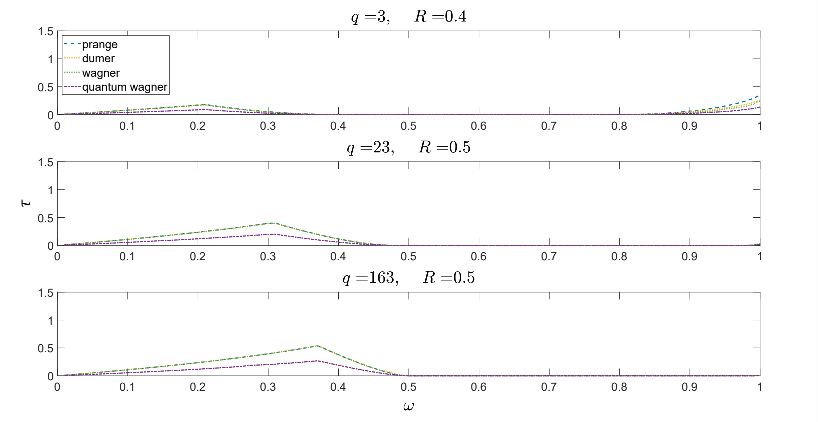
The asymptotic running time is evaluated when parameters l, d, and a are optimized to yield the shortest running time.

Exponent of the asymptotic running time, τ

$$\tau(q, R, \omega) = \lim_{n\to\infty} \frac{1}{n} \log_2 T(n),$$

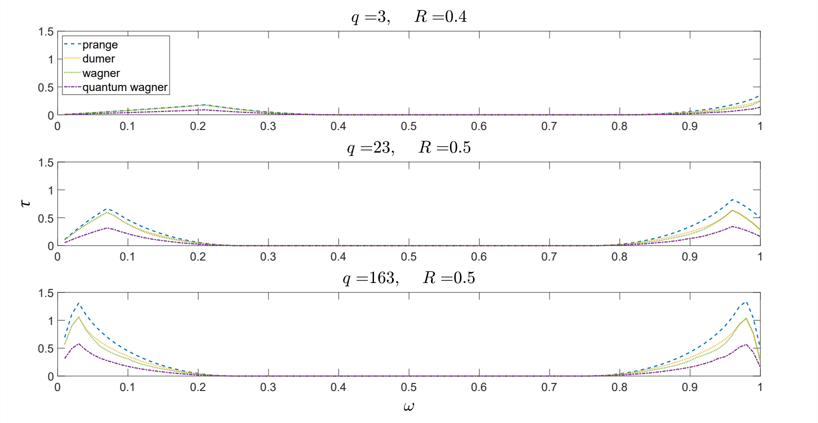
where $R := \frac{k}{n}$ and $\omega := \frac{w}{n}$.

Numerical results



Hamming weight setting: $\tau(q, R, \omega) = \lim_{n\to\infty} \frac{1}{n} \log_2 T$, $R := \frac{k}{n}$, and $\omega : \frac{w}{n}$

Numerical results



Lee weight setting: $\tau(q, R, \omega) = \lim_{n\to\infty} \frac{1}{n} \log_2 T$, $R := \frac{k}{n}$, and $\omega : \frac{w}{n}$

Summary of the first part

The asymptotic complexity of the **hardest instances** of GSDP problem is in the **Lee weight** setting is at least as long as in the Hamming weight case.

Summary of the first part

The asymptotic complexity of the **hardest instances** of GSDP problem is in the **Lee weight** setting is at least as long as in the Hamming weight case.

For the **quantum setting**, our algorithms have almost a **quadratic improvement** over the classical setting.

Summary of the first part

The asymptotic complexity of the **hardest instances** of GSDP problem is in the **Lee weight** setting is at least as long as in the Hamming weight case.

For the **quantum setting**, our algorithms have almost a **quadratic improvement** over the classical setting.

The **GSDP problem** remains **exponentially har**d for conveniently chosen parameters both in the classical and quantum setting.

STERN'S IDENTIFICATION PROTOCOL

Stern's identification protocol[4]

Belongs to the class of so-called **sigma** or **three-round** protocols.

[4] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Stern's identification protocol[4]

Belongs to the class of so-called **sigma** or **three-round** protocols.

The security of the original protocol relies on the hardness of **binary SDP** over the **Hamming weight**.

[4] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Stern's identification protocol[4]

Belongs to the class of so-called **sigma** or **three-round** protocols.

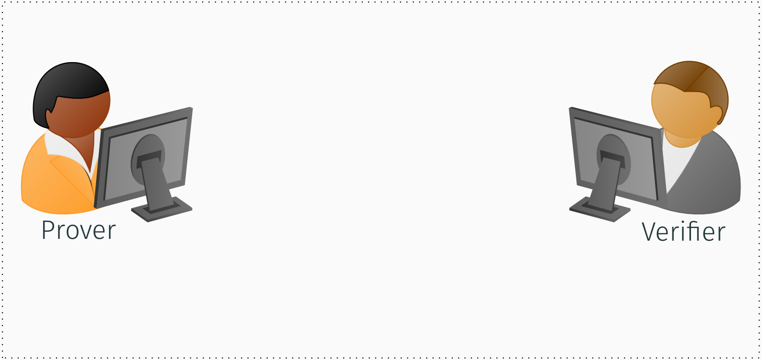
The security of the original protocol relies on the hardness of **binary SDP** over the **Hamming weight**.

The protocol is unbroken for almost 30 years now, but suffers from rather **high communication costs**.

[4] Jacques Stern. "A New Identification Scheme Based on Syndrome Decoding". In: 1993, pp. 13–21. DOI: 10.1007/3-540-48329-2_2.

Sigma (3-round) protocol

A two-party, public-key protocol.



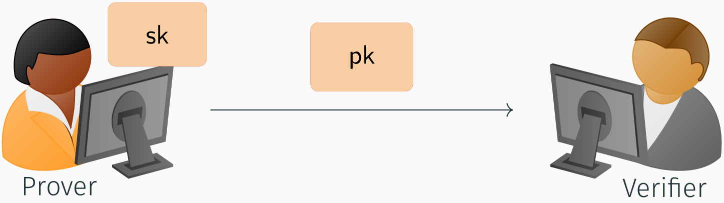
Sigma (3-round) protocol

Prover

$\mathsf{sk}, \mathsf{pk} \leftarrow \mathsf{Keygen}(1^\lambda)$

Key generation

Sigma (3-round) protocol



$\mathsf{sk}, \mathsf{pk} \leftarrow \mathsf{Keygen}(1^\lambda)$

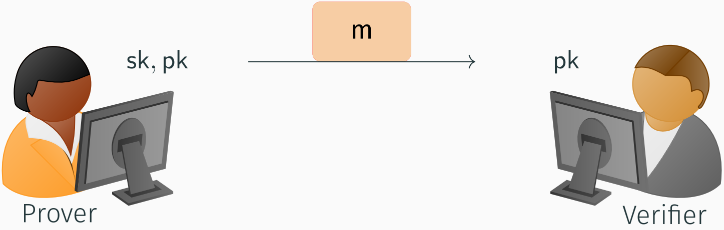
Key generation

SIGMA (3-ROUND) PROTOCOL

$(\mathbf{m}, \mathsf{init}) \leftarrow P_1(\mathsf{sk})$

Interaction

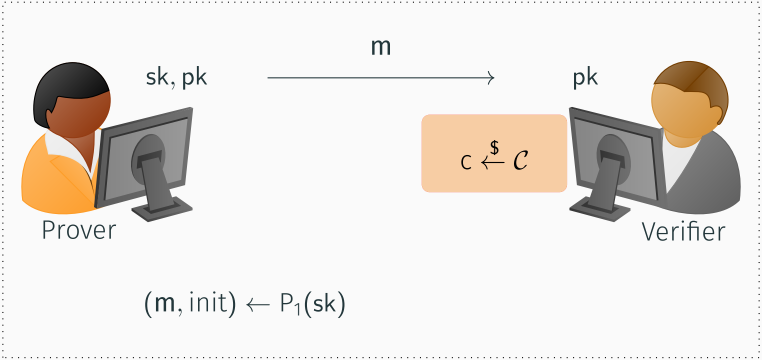
Sigma (3-round) protocol



$(\mathbf{m}, \text{init}) \leftarrow P_1(\mathbf{sk})$

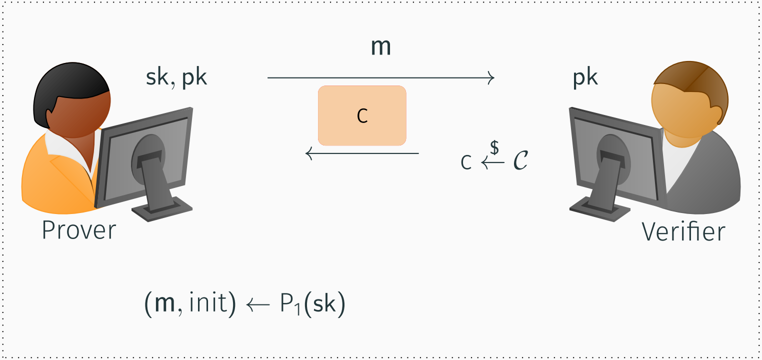
Interaction

Sigma (3-round) protocol



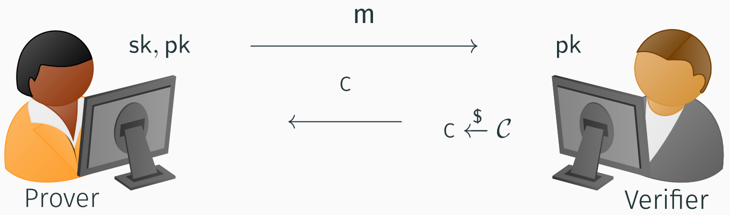
Interaction

Sigma (3-round) protocol



Interaction

Sigma (3-round) protocol

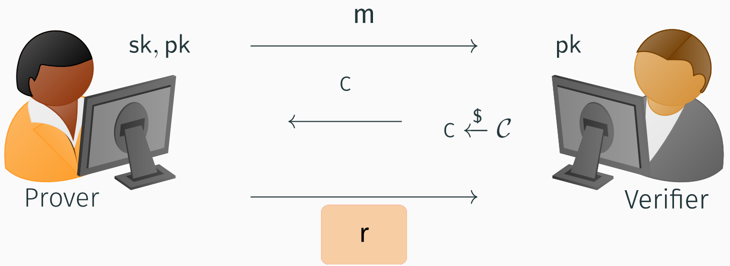


$(\mathbf{m}, \text{init}) \leftarrow P_1(\text{sk})$

$\mathbf{r} \leftarrow P_2(\text{sk}, \text{init}, c)$

Interaction

Sigma (3-round) protocol

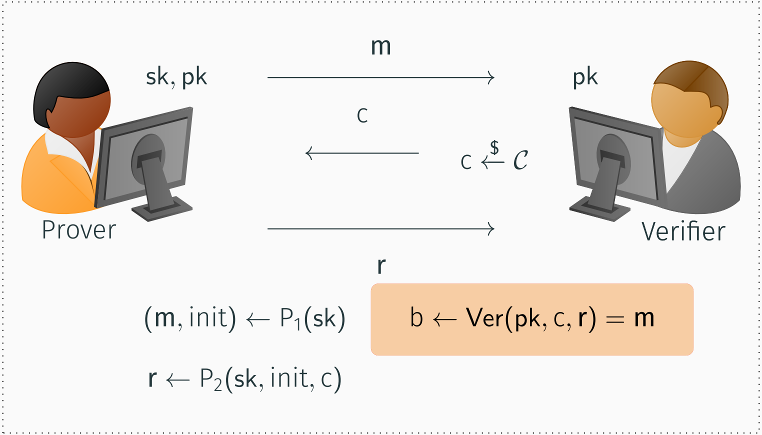


$(\mathbf{m}, \text{init}) \leftarrow P_1(\text{sk})$

$\mathbf{r} \leftarrow P_2(\text{sk}, \text{init}, c)$

Interaction

SIGMA (3-ROUND) PROTOCOL



Verifying

Sigma (3-round) protocol

Basic properties:

Sigma (3-round) protocol

Basic properties:

- **completness**: honest prover needs to be able to convince verifier it knows sk;

Sigma (3-round) protocol

Basic properties:

- **completness**: honest prover needs to be able to convince verifier it knows sk;
- **soundness**: dishonest prover is not able to convince verifier it knows sk with probability 1;

Sigma (3-round) protocol

Basic properties:

- **completness**: honest prover needs to be able to convince verifier it knows sk;
- **soundness**: dishonest prover is not able to convince verifier it knows sk with probability 1;
- **zero-knolwedge**: communication reveals only if prover knows sk and nothing else.

Stern's identification protocol

$$H \xleftarrow{\$} \mathbb{F}_q^{(n-k)\times n}, \quad e \xleftarrow{\$} \mathbb{F}_q^n, \quad s = He$$

Key generation

Stern's identification protocol

$$H \xleftarrow{\$} \mathbb{F}_q^{(n-k)\times n}, \quad e \xleftarrow{\$} \mathbb{F}_q^n, \quad s \leftarrow He$$

$$pk \leftarrow (H, s), \quad sk \leftarrow e$$

Key generation

Stern's identification protocol

$$\pi \xleftarrow{\$} \mathrm{Perm}[n],\ \mathbf{y} \xleftarrow{\$} \mathbb{F}_q^n,\ \mathbf{t} \leftarrow \mathbf{H}\mathbf{y},$$

Interaction

Stern's identification protocol

sk

Prover

pk

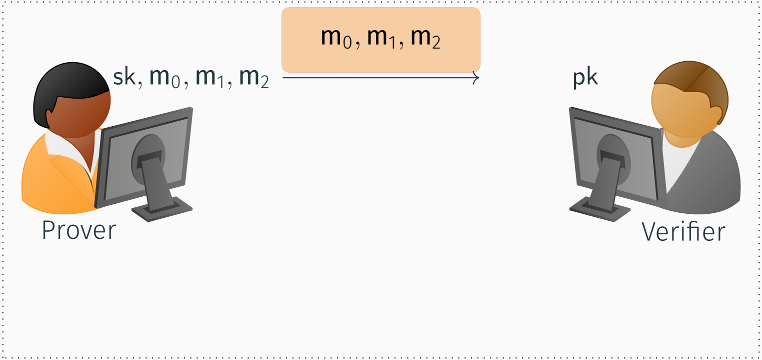
Verifier

$$\pi \xleftarrow{\$} \mathrm{Perm}[n],\ \mathbf{y} \xleftarrow{\$} \mathbb{F}_q^n,\ \mathbf{t} \leftarrow \mathbf{H}\mathbf{y},$$

$$m_0 \leftarrow \mathcal{H}(\pi, \mathbf{t}),\ m_1 \leftarrow \mathcal{H}(\pi(\mathbf{y})),\ m_2 \leftarrow \mathcal{H}(\pi(\mathbf{y} + \mathbf{e}))$$

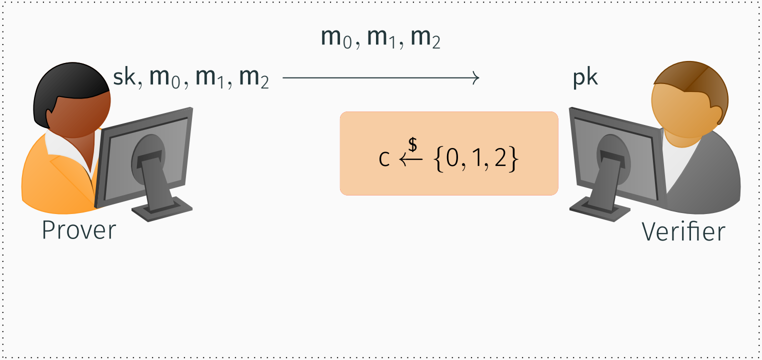
Interaction

Stern's identification protocol



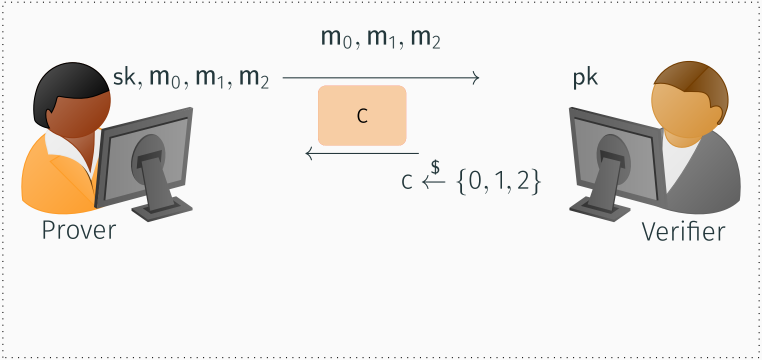
Interaction

Stern's identification protocol



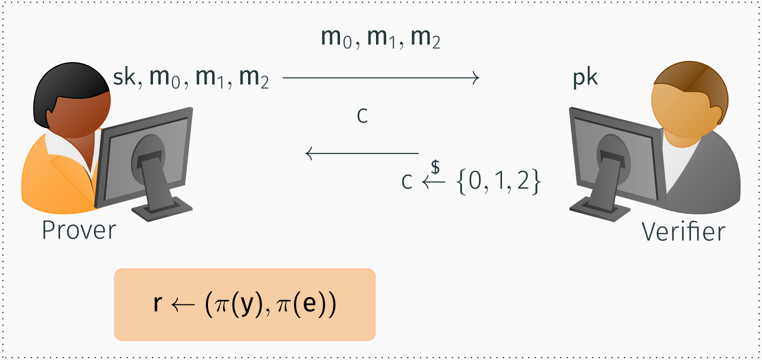
Interaction

Stern's identification protocol



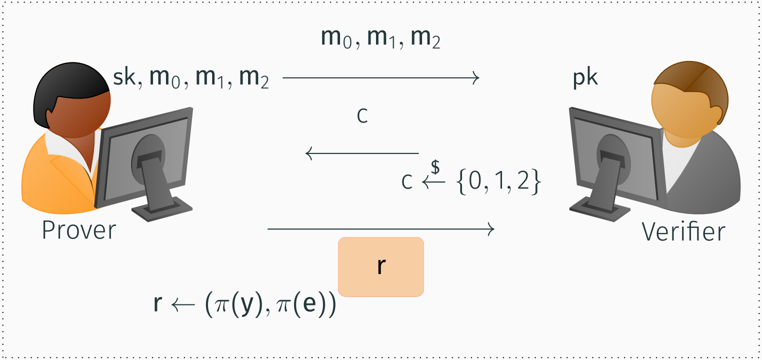
Interaction

Stern's identification protocol



Interaction: case $c = 0$

Stern's identification protocol



Interaction

Stern's identification protocol

m_0, m_1, m_2

Prover: sk, m_0, m_1, m_2 $\longrightarrow$ Verifier: pk

c $\longleftarrow$ $c \xleftarrow{\$} \{0,1,2\}$

r $\longrightarrow$

Prover

Verifier

$r \leftarrow (\pi(y), \pi(e))$

$b \leftarrow \quad m_1 \stackrel{?}{=} H(\pi(y)) \wedge$

$m_2 \stackrel{?}{=} H(\pi(y) + \pi(e)) \;\wedge\; wt_M(\pi(e)) \stackrel{?}{=} w$

Verifying: case c = 0

Stern's identification protocol

Basic properties:

- the scheme is **complete**;
- it is **sound**, with **soundness error** of 2/3;
- it is proven to be **honest verifier zero-knowledge** in the random oracle model.

Stern's identification protocol

Basic properties:

- the scheme is **complete**;
- it is **sound**, with **soundness error** of 2/3;
- it is proven to be **honest verifier zero-knowledge**.

Soundness error can be reduced arbitrarily close to zero by repeating the protocol r times.

Stern's identification protocol

Major drawback: high communication costs (order of 100 kB).

Stern's identification protocol

Major drawback: high communication costs (order of 100 kB).

$\rightarrow$ Reduction of communication cost can be achieved using **pseudo random generators** and **deterministic commitments**.

Our contributions[5]

A **honest verifier zero-knowledge** variant of Stern's identification scheme adapted to the **generalized syndrome decoding problem**.

[5]André Chailloux and Simona Etinski. On the (In)security of optimized Stern-like signature schemes. Cryptology ePrint Archive, Paper 2021/552. 2022.

Our contributions[5]

A **honest verifier zero-knowledge** variant of Stern's identification scheme adapted to the **generalized syndrome decoding problem**.

A proof that using **deterministic commitments** in combination **pseudo random generated** random vectors is secure.

[5] André Chailloux and Simona Etinski. On the (In)security of optimized Stern-like signature schemes. Cryptology ePrint Archive, Paper 2021/552. 2022.

Numerical results

Obtained for concrete parameters of GSDP that guarantee that the analyzed algorithms run in $2^{128} \Rightarrow 128$ bits of security.

Numerical results

Obtained for concrete parameters of GSDP that guarantee that the analyzed algorithms run in $2^{128} \Rightarrow 128$ bits of security.

The optimized scheme is constructed using **deterministic commitments** in combination with **pseudo-random generators**.

Numerical results

q	Non-optimized scheme		Optimized scheme	
	wt_H	wt_L	wt_H	wt_L
2	253.05	253.05	26.21	26.21
3	116.54	116.54	21.81	21.81
5	138.54	95.48	27.62	21.41
7	126.47	90.94	28.29	22.71
13	113.23	79.27	29.38	23.29

Table: Communication cost of non-optimized and optimized schemes

Summary of the second part

Communication cost can be significantly reduced by using **deterministic commitments** in combination with the **pseudo-random generation**.

Summary of the second part

Communication cost can be significantly reduced by using **deterministic commitments** in combination with the **pseudo-random generation**.

Without loss in security, **additional reduction** can be obtained by replacing the original SDP with it's **generalized version over Lee weight**.

Future directions

Generalize the asymptotic analysis to the ISD algorithms based on **representation techniques**, **nearest neighbour search**, and **statistical decoding**.

Future directions

Generalize the asymptotic analysis to the ISD algorithms based on **representation techniques**, **nearest neighbour search**, and **statistical decoding**.

Apply more advanced communication reduction techniques such as **shared permutations**, **"MPC in the head"**, use **quasi-cyclic matrices**.

Merci pour votre attention !

Thank you for your attention!

Hvala vam na pažnji!