# Oblivious Ciphertext Compression via Linear Codes

Bruno Grenet

Joint work with P. Giorgi and M. Simkin

LABORATOIRE JEAN KUNTZMANN
MATHÉMATIQUES APPLIQUÉES - INFORMATIQUE

Université Grenoble Alpes

Séminaire CAS$^3$C$^3$

23 octobre 2025

# Representations of vectors with few non-zero entries

## Dense representation

$$\mathbf{v} = (0, 0, 1, 0, 0, 0, 0, 0, 0, 4, 6, 0, 0, 0, 0, 0, 7, 0, 0, 0, 8, 0, 0, 0, 0, 3, 1, 0, 0, 0, 0) \in \mathbb{F}_{37}^{31}$$

## Compact representations

$$\mathrm{sparse}(\mathbf{v}) = \left\{(2, 1), (9, 4), (10, 6), (16, 7), (20, 8), (25, 3), (26, 1)\right\}$$

$$(\mathbf{v}(2^i))_{0 \le i < 14} = (30, 2, 5, 23, 15, 35, 23, 16, 2, 2, 28, 30, 2, 16)$$

$$\dots$$

## Hamming weight

- $\mathrm{hw}(\mathbf{v}) = \#\{i : v_i \neq 0\}$ $\qquad$ $\mathrm{hw}(\mathbf{v}) = 7$

# Oblivious compression of ciphertexts [Choi *et al.*'21,Liu-Tromer'22,Fleischhacker-Larsen-Simkin'23]

**Server** — pk of $\boxed{\cdot}$

$$\boxed{\mathbf{v}} = (\boxed{v_0}, \ldots, \boxed{v_{n-1}})$$

*oblivious compression*

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

(*some* compact representation)

A bound $t \geq \mathrm{hw}(\mathbf{v})$ is given as a public parameter

**Client** — sk of $\boxed{\cdot}$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

*decryption*

$$\mathbf{c} = (c_0, \ldots, c_{\delta n-1})$$

*decompression*

$$\mathrm{sparse}(\mathbf{v}) = \{(i, v_i) : v_i \neq 0\}$$

# Oblivious compression of ciphertexts [Choi *et al.*'21,Liu-Tromer'22,Fleischhacker-Larsen-Simkin'23]

**Server** pk of $\boxed{\cdot}$

$$\boxed{\mathbf{v}} = (\boxed{v_0}, \ldots, \boxed{v_{n-1}})$$

*oblivious compression*

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

(*some* compact representation)

A bound $t \geq \mathrm{hw}(\mathbf{v})$ is given as a public parameter

**Client** sk of $\boxed{\cdot}$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

*decryption*

$$\mathbf{c} = (c_0, \ldots, c_{\delta n-1})$$

*decompression*

$$\mathrm{sparse}(\mathbf{v}) = \{(i, v_i) : v_i \neq 0\}$$

## Goals

- $\delta < 1$ as small as possible *depends on $t$*
- Efficiency for oblivious compression and for decompression
- *With known support*: Client knows $\{i : v_i \neq 0\}$ [Bienstock *et al*'24]

# Oblivious decompression of ciphertexts

[Angel *et al.*'18,Bienstock *et al.*'24]

**Server**

pk of $\boxed{\cdot}$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

*oblivious decompression*

$$\downarrow$$

$$\boxed{\mathbf{w}} = (\boxed{w_0}, \ldots, \boxed{w_{n-1}})$$

where $w_i = v_i$ for all $v_i \neq 0$

**Client**

**pk** of $\boxed{\cdot}$

$$\mathbf{v} = (v_0, \ldots, v_{n-1})$$

*compression + encryption*

$$\downarrow$$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

(*some* compact representation)

# Oblivious decompression of ciphertexts

[Angel *et al.*'18,Bienstock *et al.*'24]

**Server** pk of $\boxed{\cdot}$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}}) \longleftarrow$$

*oblivious decompression*

$$\downarrow$$

$$\boxed{\mathbf{w}} = (\boxed{w_0}, \ldots, \boxed{w_{n-1}})$$

where $w_i = v_i$ for all $v_i \neq 0$

**Client** **pk** of $\boxed{\cdot}$

$$\mathbf{v} = (v_0, \ldots, v_{n-1})$$

*compression + encryption*

$$\downarrow$$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{\delta n-1}})$$

(*some* compact representation)

## Goals

- $\delta$ as small as possible
- Efficiency for compression and oblivious decompression
- Remark: no condition on $w_i$ if $v_i = 0$

# Applications: Secure Search [Choi *et al.*'18] / Obl. Message Retrieval [Liu-Tromer'22]

**Server** pk of $\boxed{\cdot}$

database
$$\mathbf{d} = (d_0, \ldots, d_{n-1})$$

$$\boxed{q}$$

*evaluate* $\boxed{q}$ *on* $\mathbf{d}$

$$\boxed{\mathbf{v}} = (\boxed{v_0}, \ldots, \boxed{v_{n-1}})$$

$$v_i = \begin{cases} d_i & \text{if } d_i \text{ satisfies } q \\ 0 & \text{otherwise} \end{cases}$$

**Client** sk of $\boxed{\cdot}$

query $q$

*encryption*

$$\boxed{q}$$

$$\boxed{\mathbf{v}}$$

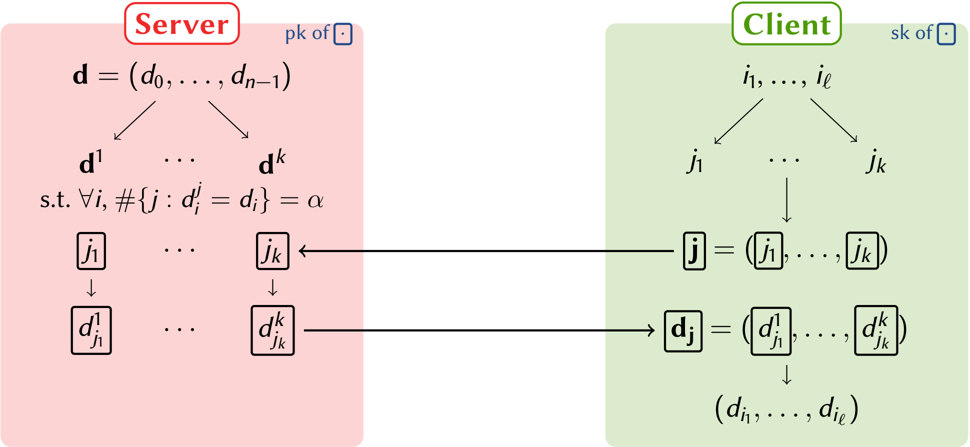
*decryption*

sparse$(\mathbf{v})$

Using oblivious compression:

- Reduced (last) communication better than full database!
- Reduced work for the client

# Application: Batch-PIR from any PIR [Angel *et al.*'18, Bienstock *et al.*'24]




## Reduce communications

- Using oblivious decompression: compress $\boxed{\mathbf{j}}$
- Using oblivious compression with known support: compress $\boxed{\mathbf{d_j}}$

# Linear codes

## Definitions

- *Linear* $[n, k, d]$ *code* $\mathcal{C}$ *over* $\mathbb{F}_q$: linear subspace of dimension $k$ of $\mathbb{F}_q^n$
  - $d \leq n - k + 1$: *distance* of $\mathcal{C} = \min_{\mathbf{c} \in \mathcal{C} \setminus \{\mathbf{0}\}} \mathrm{hw}(\mathbf{c})$
- *Generator and parity-check matrices:*
  - $G \in \mathbb{F}_q^{k \times n}$ such that $\mathcal{C} = \{\mathbf{m} \cdot G : \mathbf{m} \in \mathbb{F}_q^k\}$ $\qquad$ $\mathcal{C} = \mathrm{rowsp}(G)$
  - $H \in \mathbb{F}_q^{(n-k) \times n}$ such that $\mathcal{C} = \{\mathbf{c} \in \mathbb{F}_q^n : H \cdot \mathbf{c} = \mathbf{0}\}$ $\qquad$ $\mathcal{C} = \ker(H)$

## Dual code

Dual code $\mathcal{C}^\perp$ of $\mathcal{C}$: linear $[n, n-k, d^\perp]$ code defined either by

- $\mathcal{C}^\perp = \mathrm{rowsp}(H)$ $\qquad$ $H$ is a generator matrix
- $\mathcal{C}^\perp = \ker(G)$ $\qquad$ $G$ is a parity-check matrix

# Classical use of linear codes

**Sender**

$$\mathbf{m} = (m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k$$

*encoding*

$$\mathbf{c} = (c_0, \ldots, c_{n-1}) \in \mathcal{C} \xrightarrow{\text{noisy transmission}} \mathbf{r} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$$

$+\mathbf{e} \in \mathbb{F}_q^n$
$\mathrm{hw}(\mathbf{e})$ *small*

**Receiver**

$$\mathbf{r} = \mathbf{c} + \mathbf{e} \in \mathbb{F}_q^n$$

*decoding*

$$\begin{aligned} \mathbf{c} &= (c_0, \ldots, c_{n-1}) \in \mathcal{C} \\ \mathbf{e} &= (e_0, \ldots, e_{n-1}) \in \mathbb{F}_q^n \\ \mathbf{m} &= (m_0, \ldots, m_{k-1}) \in \mathbb{F}_q^k \end{aligned}$$

- Decoding from errors: $\mathbf{r} \mapsto$ either $\mathbf{c}$, $\mathbf{m}$ or $\mathbf{e}$ $\qquad \mathrm{hw}(\mathbf{e}) < \frac{d}{2}$
- Decoding from erasures: $\mathbf{r}$ *and* $\{i : e_i \neq 0\} \mapsto \mathbf{c}$, $\mathbf{m}$ or $\mathbf{e}$ $\qquad \mathrm{hw}(\mathbf{e}) < d$

# Syndrome and decoding

## Syndrome

Let $\mathbf{r} = \mathbf{c} + \mathbf{e}$ with $\mathbf{c} \in \mathcal{C}$ and $\mathrm{hw}(\mathbf{e}) < \frac{d}{2}$.

- The syndrome of $\mathbf{r}$ is $\mathbf{s} = H \cdot \mathbf{r}$ where $H$ is a parity-check matrix of $\mathcal{C}$
- The syndrome does not depend on $\mathbf{c}$: $\mathbf{s} = H \cdot \mathbf{e}$

## Syndrome Decoding Problem

Parameter: a parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$ of a linear $[n, k, d]$ code

Input: a *syndrome* $\mathbf{s} = H \cdot \mathbf{e}$ where $\mathrm{hw}(\mathbf{e}) < \frac{d}{2}$

Output: the *error vector* $\mathbf{e} \in \mathbb{F}_q^n$

## Remarks

- Variant: Erasure-SDP $\to \{i : e_i \neq 0\}$ also given as input
- Input and output sizes: $O(n-k)$ using sparse($\mathbf{e}$)

# Oblivious compression based on linear codes

**Server** pk of $\boxed{\cdot}$

$$\boxed{\mathbf{v}} = (\boxed{v_0}, \dots, \boxed{v_{n-1}})$$

$$\downarrow$$

$$\boxed{\mathbf{c}} = H \cdot \boxed{\mathbf{v}} \longrightarrow$$

*additively homomorphic matrix-vector product*

$\mathcal{C}$: linear $[n, k, d]$ code with parity-check matrix $H$

**Client** sk of $\boxed{\cdot}$

$$\boxed{\mathbf{c}} = (\boxed{c_0}, \dots, \boxed{c_{n-k-1}})$$

$$\Big\downarrow \textit{decryption}$$

$$\mathbf{c} = (c_0, \dots, c_{n-k-1})$$

$$\Big\downarrow \textit{syndrome decoding problem}$$

$$\text{sparse}(\mathbf{v}) = \{(i, v_i) : v_i \neq 0\}$$

## Coding-theoretic interpretation

- $\mathbf{v}$ = *error vector*
- $\mathbf{c}$ = *syndrome*
- If the client knows $\{i : v_i \neq 0\}$, SDP $\rightsquigarrow$ Erasure-SDP

# Analysis of the oblivious compression scheme

## Requirements

- Additively homomorphic encryption scheme $\boxed{\cdot}$, with message space $\mathbb{F}_q$
- Linear $[n, k, d]$ code $\mathcal{C}$ over $\mathbb{F}_q$, with
  - $d > 2t$ — $\mathrm{hw}(\mathbf{v}) < \frac{d}{2}$
  - $d > t$ in the *know support* variant — $\mathrm{hw}(\mathbf{v}) < d$
- Efficiency:
  - Fast syndrome computation
  - Fast (Erasure) Syndrome Decoding Problem

## Theorem

If $\mathcal{C}$ is a linear $[n, k, d]$ code over $\mathbb{F}_q$ with $d > 2t$ (resp. $d > t$), the construction is an oblivious compression scheme (resp. with known support) with compression rate $\frac{n-k}{n}$.

## Remarks

- Since $d \leq n - k + 1$, compression rate is $\geq 2t/n$ (resp. $\geq t/n$)
- If $d = n - k + 1$, optimal compression rate $2t/n$ (resp. $t/n$) — MDS code

# Oblivious decompression based on linear codes

**Server** pk of $\boxed{\cdot}$

$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{n-k-1}})$

$\downarrow$

$\boxed{\mathbf{w}} = \boxed{\mathbf{c}} \cdot H$

*additively homomorphic matrix-vector product*

$\mathcal{C}$: linear $[n, k, d]$ code with parity-check matrix $H$

**Client** pk of $\boxed{\cdot}$

$\text{sparse}(\mathbf{v}) = \{(i, v_i) : i \in I\}$

$\downarrow$

Solve $\mathbf{v}_I = \mathbf{c} \cdot H_{*,I}$

$\downarrow$

$\boxed{\mathbf{c}} = (\boxed{c_0}, \ldots, \boxed{c_{n-k-1}})$

## Coding-theoretic interpretation

$H =$ generator matrix of the dual code $\mathcal{C}^{\perp}$

- $\mathbf{v} =$ vector with erasures
- $\mathbf{c} =$ message
- $\mathbf{w} =$ codeword

# Analysis of the oblivious decompression scheme

## Requirements

- Additively homomorphic encryption scheme $\boxed{\cdot}$, with message space $\mathbb{F}_q$
- Linear $[n, k, d]$ code $\mathcal{C}$ over $\mathbb{F}_q$, with $d > t$
- Efficiency: for the dual code $\mathcal{C}^\perp$,
  - Fast encoding — message $\to$ codeword
  - Fast decoding from erasures

## Theorem

If $\mathcal{C}$ is a linear $[n, k, d]$ code over $\mathbb{F}_q$ with $d > t$, the construction is an oblivious decompression scheme with compression rate $\frac{n-k}{n}$.

## Remarks

- Since $d \leq n - k + 1$, compression rate is $\geq t/n$
- If $d = n - k + 1$, optimal compression rate $t/n$ — MDS code

# Generalized Reed-Solomon Codes

## Definition: $[n, k, d]$ generalized Reed-Solomon code

$$\mathcal{C} = \left\{ (\lambda_0 \cdot f(\alpha_0), \ldots, \lambda_{n-1} \cdot f(\alpha_{n-1})) : f \in \mathbb{F}_q[x]_{<k} \right\}$$

- *code locators*: $\alpha_0, \ldots, \alpha_{n-1} \in \mathbb{F}_q$ $\quad \alpha_i \neq \alpha_j$ for $i \neq j$
- *column multipliers*:: $\lambda_0, \ldots, \lambda_{n-1} \in \mathbb{F}_q^\times$

(classical RS codes: $\lambda_0 = \cdots = \lambda_{n-1} = 1$ and $\alpha_i = \gamma^i$ for some $\gamma$ of order $n$)

## Properties

- GRS codes are MDS: $d = n - k + 1$
- The dual of a GRS code is a GRS code:

$$G = \begin{bmatrix} \lambda_0 & \ldots & \lambda_{n-1} \\ \lambda_0\alpha_0 & \ldots & \lambda_{n-1}\alpha_{n-1} \\ \vdots & & \vdots \\ \lambda_0\alpha_0^{k-1} & \ldots & \lambda_{n-1}\alpha_{n-1}^{k-1} \end{bmatrix} \qquad H = \begin{bmatrix} \mu_0 & \ldots & \mu_{n-1} \\ \mu_0\alpha_0 & \ldots & \mu_{n-1}\alpha_{n-1} \\ \vdots & & \vdots \\ \mu_0\alpha_0^{n-k-1} & \ldots & \mu_{n-1}\alpha_{n-1}^{n-k-1} \end{bmatrix}$$

# GRS codes: encoding, syndrome, erasure decoding

## Expression in terms of evaluation / interpolation

- In the dual code $\mathcal{C}^\perp$ with generator matrix $H \in \mathbb{F}_q^{(n-k)\times n}$:
  - Encoding: compute $\mathbf{m} \cdot H = (\mu_0 m(\alpha_0), \ldots, \mu_{n-1} m(\alpha_{n-1}))$ — multipoint evaluation
  - Erasure decoding: solve $\mathbf{r}_I = \mathbf{m} \cdot H_{*,I}$ — interpolation
- In the code $\mathcal{C}$ with parity-check matrix $H \in \mathbb{F}_q^{(n-k)\times n}$:
  - Syndrome computation: compute $H \cdot \mathbf{r}$ — *transposed* multipoint evaluation
  - Erasure-SDP: solve $\mathbf{s} = H_{*,I} \cdot \mathbf{e}_I$ — *transposed* interpolation

## Complexities

- Encoding and syndrome: $O(\mathsf{M}(n)\log(n-k))$ operations in $\mathbb{F}_q$
- Erasure decoding and Erasure-SDP: $O(\mathsf{M}(n-k)\log(n-k))$ operations in $\mathbb{F}_q$

## Remarks

- If $\alpha_i = \gamma^i$ for $0 \le i < n$, encoding and syndrome in $O(\mathsf{M}(n))$ operations
- In the general case, $\mathsf{M}(n) = O(n \log n \log\log n)$

# Syndrome Decoding Problem for GRS Codes

## The problem

Input: $\mathbf{s} = \begin{bmatrix} \mu_0 & \dots & \mu_{n-1} \\ \mu_0\alpha_0 & \dots & \mu_{n-1}\alpha_{n-1} \\ \vdots & & \vdots \\ \mu_0\alpha_0^{n-k-1} & \dots & \mu_{n-1}\alpha_{n-1}^{n-k-1} \end{bmatrix} \cdot \mathbf{e}$ where $\mathrm{hw}(\mathbf{e}) = t < \frac{d}{2}$

Output: $\mathrm{sparse}(\mathbf{e})$

## Basic idea

- $s_j = \sum_{i:e_i \neq 0} e_i \mu_i \cdot \alpha_i^j$ for $0 \leq j < n - k$
- $(s_j)_j$ is linearly recurrent with minimal polynomial $\prod_{i:e_i \neq 0}(x - \alpha_i)$

## Algorithm

1. Compute the minimal polynomial of $\mathbf{s}$ — Berlekamp-Massey alg.: $O(\mathsf{M}(t)\log(t))$
2. Compute its roots $\alpha_{i_0}, \dots, \alpha_{i_{t-1}}$ — Berlekamp-Rabin alg.: $O(\mathsf{M}(t)\log(t)\log(q))$
3. Compute the set $I = \{i_0, \dots, i_{t-1}\}$ — Requires $\alpha_i \mapsto i$
4. Solve the Erasure-SDP $\mathbf{s} = H_{*,I} \cdot \mathbf{e}_I$ — $O(\mathsf{M}(t)\log(t))$

# GRS codes for oblivious (de)compression

## Oblivious compression scheme

- Using an $[n, n-2t, 2t+1]$ GRS code with $\alpha_i = i \in \mathbb{F}_q$, we obtain
  - an optimal compression rate $2t/n$
  - oblivious compression in $O(\mathsf{M}(n)\log(t))$ add. homomorphic operations
  - decompression in $O(\mathsf{M}(t)\log(t)\log q)$ operations in $\mathbb{F}_q$

## Oblivious compression scheme with known support

- Using an $[n, n-t, t+1]$ GRS code with $\alpha_i = \gamma^i \in \mathbb{F}_q$, we obtain
  - an optimal compression rate $t/n$
  - oblivious compression in $O(\mathsf{M}(n))$ add. homomorphic operations
  - decompression in $O(\mathsf{M}(t)\log(t))$ operations

## Oblivious decompression scheme

- Using an $[n, n-t, t+1]$ GRS code with $\alpha_i = \gamma^i \in \mathbb{F}_q$, we obtain
  - an optimal compression rate $t/n$
  - compression in $O(\mathsf{M}(t)\log(t))$ operations
  - oblivious decompression in $O(\mathsf{M}(n))$ add. homomorphic operations

# Complexities for special finite fields

### Binary field $\mathbb{F}_{2^\ell}$

Use of the LCH basis of $\mathbb{F}_2[x]_{<2^\ell}$: alternative to monomial basis with fast algorithms

- Server: $O(n \log t)$ add. homomorphic operations
- Client: $O(t \log^2 t)$ / $O(t \log^2 t\ell)$ operations in $\mathbb{F}_{2^\ell}$

### Prime finite field $\mathbb{F}_q$ with $q \equiv 1 \bmod 2^\ell$

$\mathbb{F}_q$ has a $2^\ell$th primitive root of unity: *FFT-friendly finite field*

- Server: $O(n \log t)$ / $O(n \log^2 t)$ add. homomorphic operations
- Client: $O(t \log^2 t)$ / $O(t \log^2 t \log q)$ operations in $\mathbb{F}_q$

# Comparisons of oblivious compression schemes

| compressed size | obl. comp. | decomp. | perfect | $\mathbb{F}_q$ | |
|---|---|---|---|---|---|
| $O(t(\log^2 t + \log \kappa))$ | $O(nt)$ | $O(t^3)$ | no | $q > 2^\kappa$ | [Liu-Tromer'22] |
| $O(\kappa t / \log t)$ | $O(n\kappa / \log t)$ | $O(t\kappa / \log t)$ | no | $q > 2^\kappa$ | [FLS'23] |
| $2t$ | $O(n \log n)$ | $O(t\sqrt{n})$ | yes | $q > n$ | [FLS'23] |
| $O(t + \kappa \log \kappa)$ | $O(n\kappa)$ | $O(t\kappa)$ | no | $q > 2^\kappa$ | [FLOS'24] |
| $2t$ | $O(n \log^2 t)$ | $O(t \log^2 t \log q)$ | yes | $q > n$ | This work |
| $2t$ | $O(n \log t)$ | $O(t \log^2 t\ell)$ | yes | $q = 2^\ell$ | This work |

- $\kappa =$ failure probability, $\kappa > \log n$

# Conclusion

## A new framework

- Oblivious compression $\leftrightarrow$ Syndrome Decoding Problem
- Oblivious compression with known support $\leftrightarrow$ Erasure Syndrome Decoding Problem
- Oblivious decompression $\leftrightarrow$ Erasure Decoding + Encoding

## Results

- Optimal compression rates with MDS codes
- Deterministic and perfectly correct schemes
- Very good asymptotic complexities in particular over $\mathbb{F}_{2^\ell}$

## Open questions

- Behavior in practice, in particular within complete protocols
- Use of other codes?

# Conclusion

## A new framework

- Oblivious compression $\leftrightarrow$ Syndrome Decoding Problem
- Oblivious compression with known support $\leftrightarrow$ Erasure Syndrome Decoding Problem
- Oblivious decompression $\leftrightarrow$ Erasure Decoding + Encoding

## Results

- Optimal compression rates with MDS codes
- Deterministic and perfectly correct schemes
- Very good asymptotic complexities in particular over $\mathbb{F}_{2^\ell}$

## Open questions

- Behavior in practice, in particular within complete protocols
- Use of other codes?

Thank you!