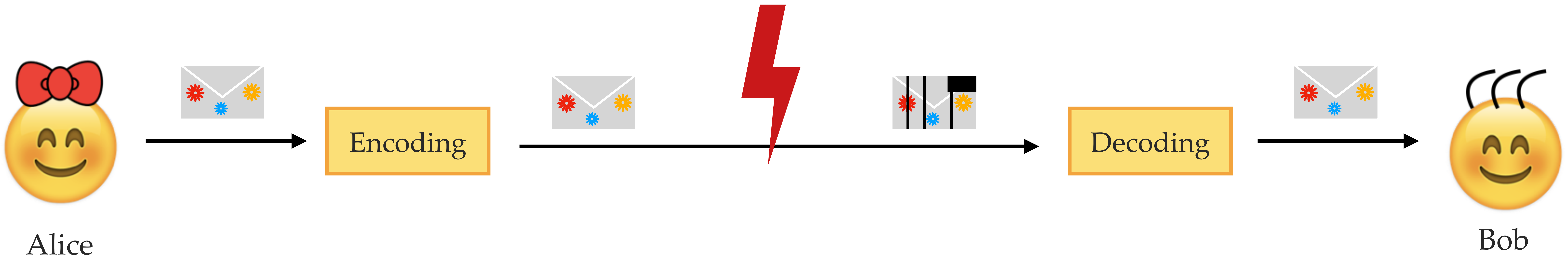
Code-based cryptography I

Monika Trimoska

Selected Areas in Cryptology - Part 1
Spring, 2024

TU/e

Error-correcting codes



- Primary use case: communication over a noisy channel.
- Main idea: introduce some redundancy in order to be able to correct the errors.
- Some structured error-correcting codes have efficient decoding algorithms.
- Decoding is, in general, a hard problem - so it is hard for *random* codes.

Hard problems (often) find their use in cryptography.

Linear codes

Linear code

An $[n, k]$ linear code $\mathscr{C}$ over $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$.

- The parameter n is called the length of the code.
- The parameter k is called the dimension of the code.
- The elements in the code are called codewords.

Hamming metric

For $\mathbf{x} \in \mathbb{F}_q^n$, the Hamming weight of $\mathbf{x}$ is the number of nonzero elements, aka.

$$\mathrm{wt}(\mathbf{x}) = |\{i \in \{1, \ldots, n\} \mid x_i \neq 0\}|.$$

Generator matrix

The matrix $\mathbf{G} \in \mathbb{F}_q^{k \times n}$ is called a generator matrix of $\mathscr{C}$, if

$$\mathscr{C} = \{\mathbf{xG} \mid \mathbf{x} \in \mathbb{F}_q^k\}.$$

Linear codes

Linear code

An $[n, k]$ linear code $\mathscr{C}$ over $\mathbb{F}_q$ is a k-dimensional subspace of $\mathbb{F}_q^n$.

Example. $q = 2, n = 5, k = 3$

$$\mathbf{G} = \begin{pmatrix} 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \end{pmatrix}$$

Codewords: $\lambda_1(10101) + \lambda_2(11000) + \lambda_3(11110)$

Example. $\mathbf{c}_1 = (111)\mathbf{G} = (10011),$
$\mathbf{c}_2 = (100)\mathbf{G} = (10101)$

Linear code equivalence

Isometry

An isometry (for our purposes) between two codes $\mathscr{C}$ and $\mathscr{D}$ is a linear map $\mu : \mathscr{C} \to \mathscr{D}$ that preserves the metric.

In our case: an isometry preserves the Hamming weight of codewords.

Which linear transformations preserve the Hamming weight?

Multiply a codeword by $\mathbf{A} \in \mathrm{GL}_n$?

Example. $q = 7, n = 5, k = 3,\ \mathbf{G} = \begin{pmatrix} 2 & 0 & 4 & 0 & 1 \\ 6 & 1 & 0 & 0 & 0 \\ 3 & 4 & 1 & 3 & 0 \end{pmatrix}$

Let $\mathbf{c} = (100)\mathbf{G} = (20401)$ $\longrightarrow$ $\mathbf{cA} = (20401)\mathbf{A} = (12533)$ $\longrightarrow$ $\mathrm{wt}(\mathbf{c}) \neq \mathrm{wt}(\mathbf{cA})$ ✗

Let $\mathbf{A} = \begin{pmatrix} 0 & 2 & 1 & 5 & 6 \\ 5 & 6 & 4 & 1 & 4 \\ 5 & 0 & 3 & 2 & 1 \\ 6 & 6 & 4 & 6 & 4 \\ 2 & 5 & 6 & 6 & 1 \end{pmatrix}$

Linear code equivalence

Isometry

An isometry (for our purposes) between two codes $\mathscr{C}$ and $\mathscr{D}$ is a linear map $\mu : \mathscr{C} \to \mathscr{D}$ that preserves the metric.

In our case: an isometry preserves the Hamming weight of codewords.

Which linear transformations preserve the Hamming weight?

→ Multiply a codeword by a permutation matrix **P** ?

Example. $q = 7, n = 5, k = 3,\ \mathbf{G} = \begin{pmatrix} 2 & 0 & 4 & 0 & 1 \\ 6 & 1 & 0 & 0 & 0 \\ 3 & 4 & 1 & 3 & 0 \end{pmatrix}$

Let $\mathbf{c} = (100)\mathbf{G} = (20401)$ → $\mathbf{cP} = (20401)\mathbf{P} = (20014)$ → $\mathrm{wt}(\mathbf{c}) = \mathrm{wt}(\mathbf{cP})$ ✓

Let $\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \end{pmatrix}$

Linear code equivalence

Isometry

An isometry (for our purposes) between two codes $\mathscr{C}$ and $\mathscr{D}$ is a linear map $\mu : \mathscr{C} \to \mathscr{D}$ that preserves the metric.

In our case: an isometry preserves the Hamming weight of codewords.

Which linear transformations preserve the Hamming weight?

→ Multiply a codeword by a monomial matrix $\mathbf{Q}$?

Example. $q = 7, n = 5, k = 3, \; \mathbf{G} = \begin{pmatrix} 2 & 0 & 4 & 0 & 1 \\ 6 & 1 & 0 & 0 & 0 \\ 3 & 4 & 1 & 3 & 0 \end{pmatrix}$

Let $\mathbf{c} = (100)\mathbf{G} = (20401)$ → $\mathbf{cQ} = (20401)\mathbf{Q} = (20023)$ → $\mathrm{wt}(\mathbf{c}) = \mathrm{wt}(\mathbf{cQ})$ ✓

Let $\mathbf{Q} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 6 \\ 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \end{pmatrix}$

Linear code equivalence

Isometry

An isometry (for our purposes) between two codes $\mathscr{C}$ and $\mathscr{D}$ is a linear map $\mu : \mathscr{C} \to \mathscr{D}$ that preserves the metric.

In our case: an isometry preserves the Hamming weight of codewords.

Which linear transformations preserve the Hamming weight?

We can also multiply $\mathbf{G}$ on the left by $\mathbf{T} \in \mathrm{GL}_k$.

This is just a change of basis (because we defined the code $\mathscr{C}$ as the row span of $\mathbf{G}$). ✓

Linear code equivalence

Isometry

An isometry (for our purposes) between two codes $\mathscr{C}$ and $\mathscr{D}$ is a linear map $\mu : \mathscr{C} \to \mathscr{D}$ that preserves the metric.

In our case: an isometry preserves the Hamming weight of codewords.

Equivalent codes

Two codes $\mathscr{C}$ and $\mathscr{D}$ are equivalent if there is an isometry between them.

Linear code equivalence

The Linear Code Equivalence (LCE) problem

Input: Two generator matrices $\mathbf{G}_1, \mathbf{G}_2 \in \mathbb{F}_q^{k\times n}$ for two linear codes $\mathscr{C}$ and $\mathscr{D}$.
Question: Find - if any - $\mathbf{Q}$, a monomial matrix, and $\mathbf{T} \in \mathrm{GL}_k(\mathbb{F}_q)$ such that $\mathbf{G}_2 = \mathbf{T}\mathbf{G}_1\mathbf{Q}$.

Matrix (rank-metric) codes

Matrix code

A matrix code $\mathscr{C}$ over $\mathbb{F}_q$ is a k-dimensional $\mathbb{F}_q$-linear subspace of $\mathbb{F}_q^{m\times n}$.

Rank metric

For $\mathbf{C} \in \mathbb{F}_q^{m\times n}$, the rank weight of $\mathbf{C}$ is given by the rank of $\mathbf{C}$, aka.

$$\mathrm{wt}(\mathbf{C}) = \mathrm{rk}(\mathbf{C}).$$

Basis of a matrix code

The basis of a matrix code $\mathscr{C}$ is given by the k-tuple $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$.

Matrix (rank-metric) codes

Example. $q = 13, \quad m = 4, \quad n = 6, \quad k = 5$

$$\mathbf{C} = \lambda_1 \cdot \begin{pmatrix} 2 & 8 & 10 & 4 & 5 & 7 \\ 1 & 11 & 7 & 9 & 6 & 12 \\ 3 & 0 & 13 & 5 & 4 & 8 \\ 9 & 6 & 3 & 2 & 10 & 11 \end{pmatrix} + \lambda_2 \cdot \begin{pmatrix} 12 & 0 & 4 & 11 & 9 & 3 \\ 5 & 6 & 8 & 13 & 2 & 1 \\ 10 & 7 & 3 & 9 & 4 & 6 \\ 2 & 5 & 11 & 8 & 1 & 10 \end{pmatrix} + \lambda_3 \cdot \begin{pmatrix} 5 & 2 & 9 & 11 & 4 & 8 \\ 3 & 7 & 1 & 10 & 12 & 0 \\ 6 & 9 & 2 & 13 & 11 & 8 \\ 1 & 5 & 6 & 3 & 10 & 7 \end{pmatrix} + \lambda_4 \cdot \begin{pmatrix} 9 & 4 & 6 & 1 & 13 & 2 \\ 8 & 0 & 5 & 12 & 6 & 11 \\ 3 & 7 & 10 & 9 & 4 & 5 \\ 2 & 8 & 11 & 3 & 7 & 1 \end{pmatrix} + \lambda_5 \cdot \begin{pmatrix} 7 & 10 & 4 & 6 & 8 & 3 \\ 1 & 5 & 2 & 11 & 9 & 0 \\ 13 & 7 & 6 & 4 & 12 & 2 \\ 8 & 3 & 1 & 9 & 5 & 10 \end{pmatrix} \qquad \lambda_i \in \mathbb{F}_q$$

Matrix code equivalence

Isometry

An isometry (for our purposes) between two codes $\mathscr{C}$ and $\mathscr{D}$ is a linear map $\mu : \mathscr{C} \to \mathscr{D}$ that preserves the metric.

In this case: an isometry preserves the rank weight of codewords.

Which linear transformations preserve the rank?

- Multiply a codeword on the right by any $\mathbf{M} \in \mathbb{F}_q^{n \times r}$ ✗
- Multiply a codeword on the right by $\mathbf{B} \in \mathrm{GL}_n$ ✓
- Multiply a codeword on the left by $\mathbf{A} \in \mathrm{GL}_m$ ✓
- Take the transposition of a codeword (only when $m = n$, does not make the equivalence problem harder) ✓

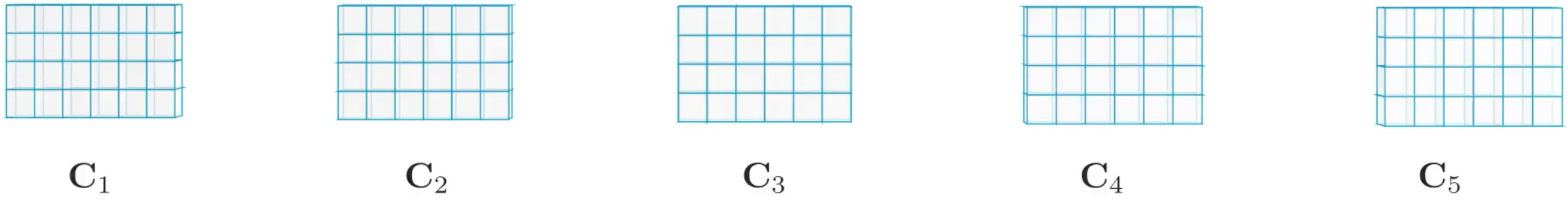
Matrix code equivalence

The Matrix Code Equivalence (MCE) problem

Input: Two k-dimensional matrix codes $\mathscr{C}, \mathscr{D} \subset \mathbb{F}_q^{m \times n}$ for two matrix codes $\mathscr{C}$ and $\mathscr{D}$.
Question: Find - if any - a map $(\mathbf{A}, \mathbf{B})$, where $\mathbf{A} \in \mathrm{GL}_m(\mathbb{F}_q)$ and $\mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ such that for all $\mathbf{C} \in \mathscr{C}$, it holds that $\mathbf{ACB} \in \mathscr{D}$.

From matrix codes to 3-tensors

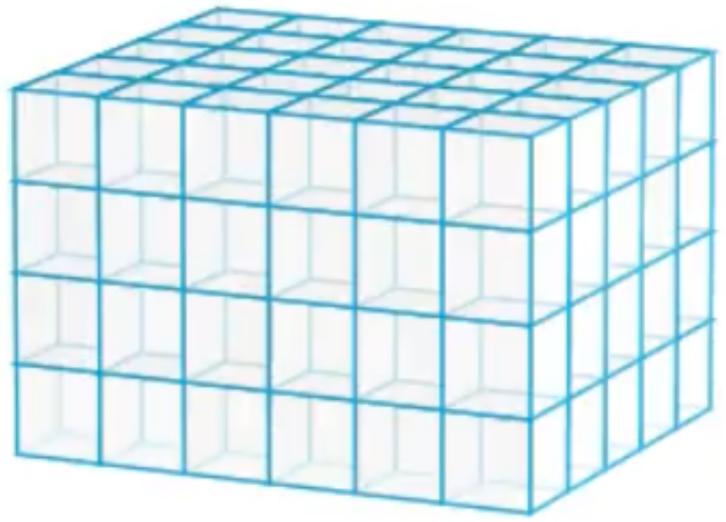
We can think of a matrix code as a 3-tensor over $\mathbb{F}_q$.



From matrix codes to 3-tensors

We can think of a matrix code as a 3-tensor over $\mathbb{F}_q$.

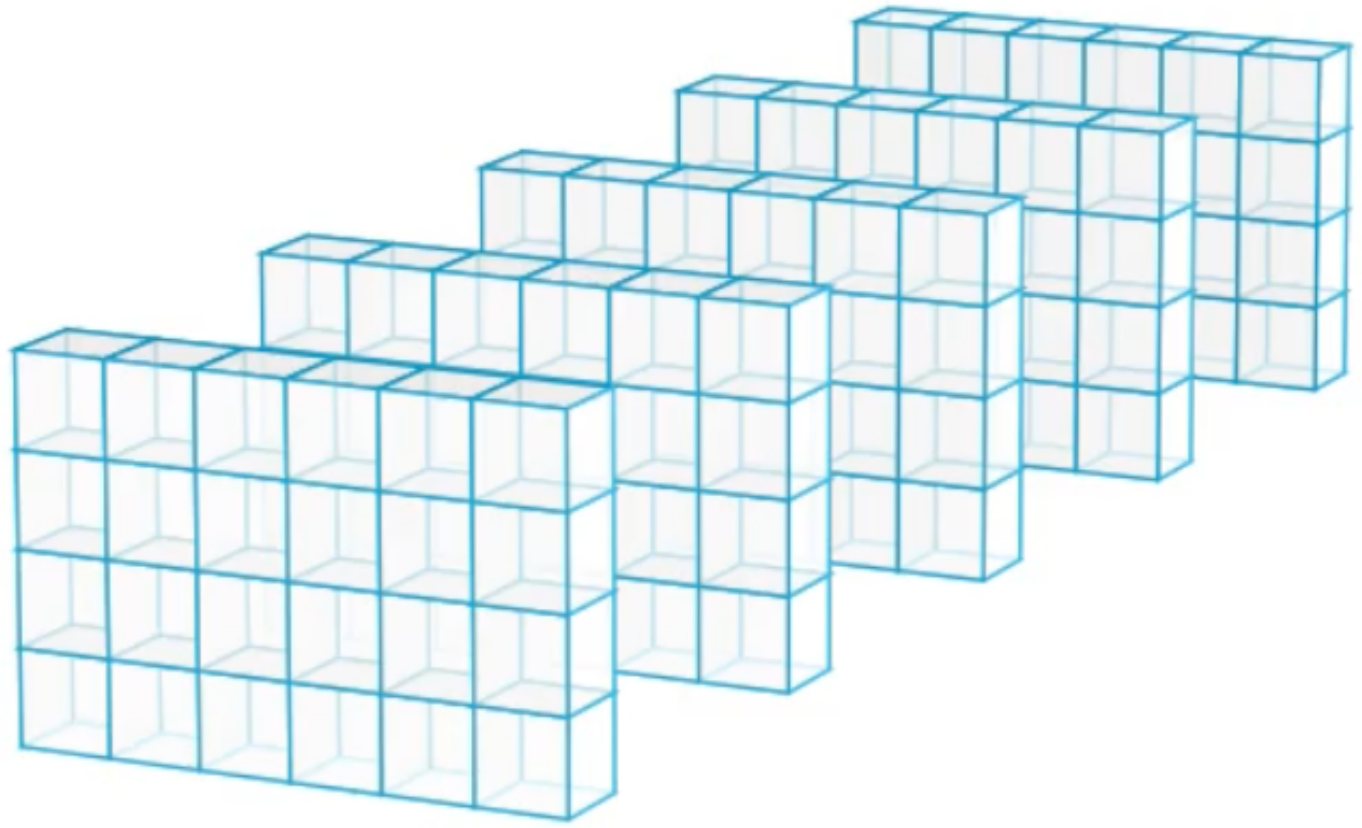
$$\mathcal{C} \subseteq \mathbb{F}_q^{m\times n\times k}$$



From matrix codes to 3-tensors

Viewed as a 3-tensor, we can see $\mathscr{C}$ from three directions

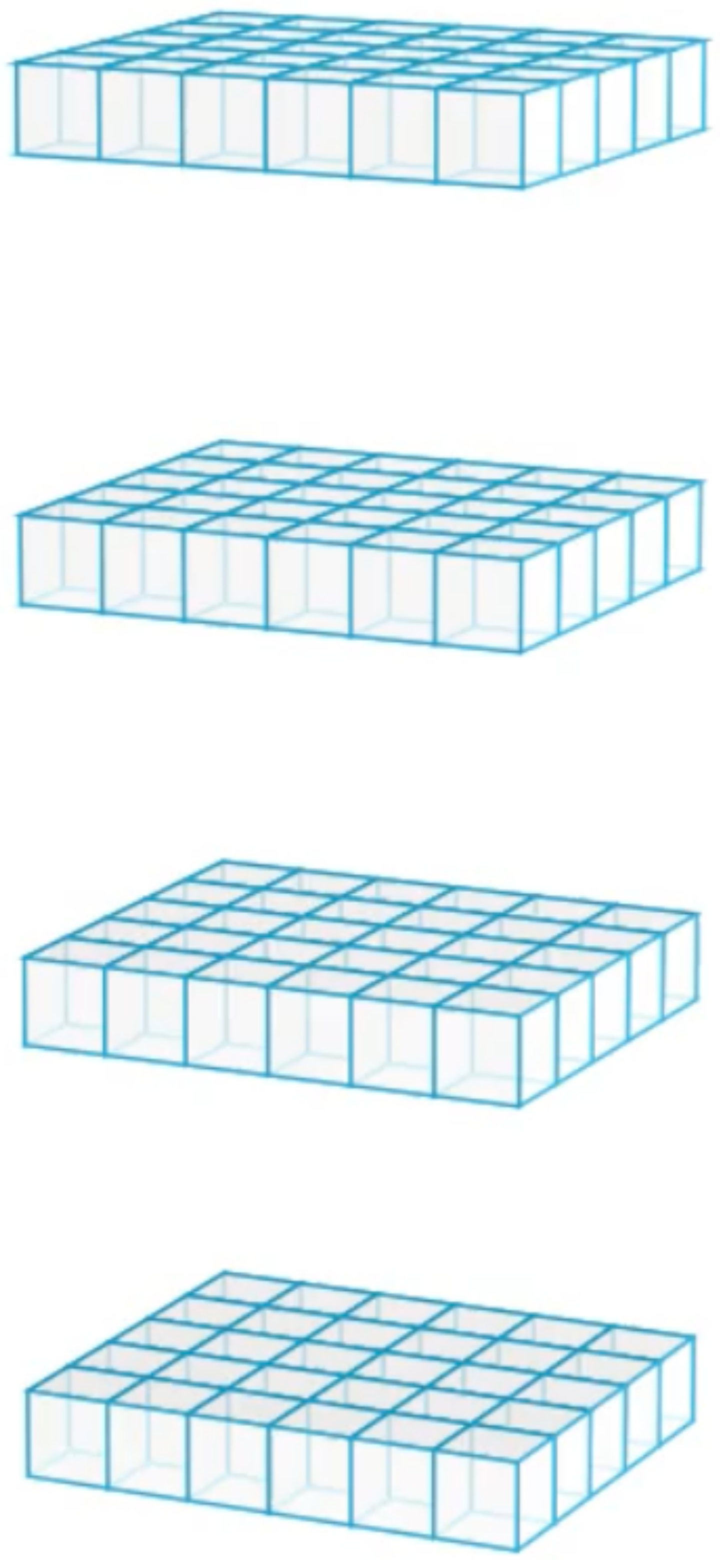
- a k-dimensional code in $\mathbb{F}_q^{m\times n}$
- an m-dimensional code in $\mathbb{F}_q^{n\times k}$
- an n-dimensional code in $\mathbb{F}_q^{m\times k}$



From matrix codes to 3-tensors

Viewed as a 3-tensor, we can see $\mathscr{C}$ from three directions

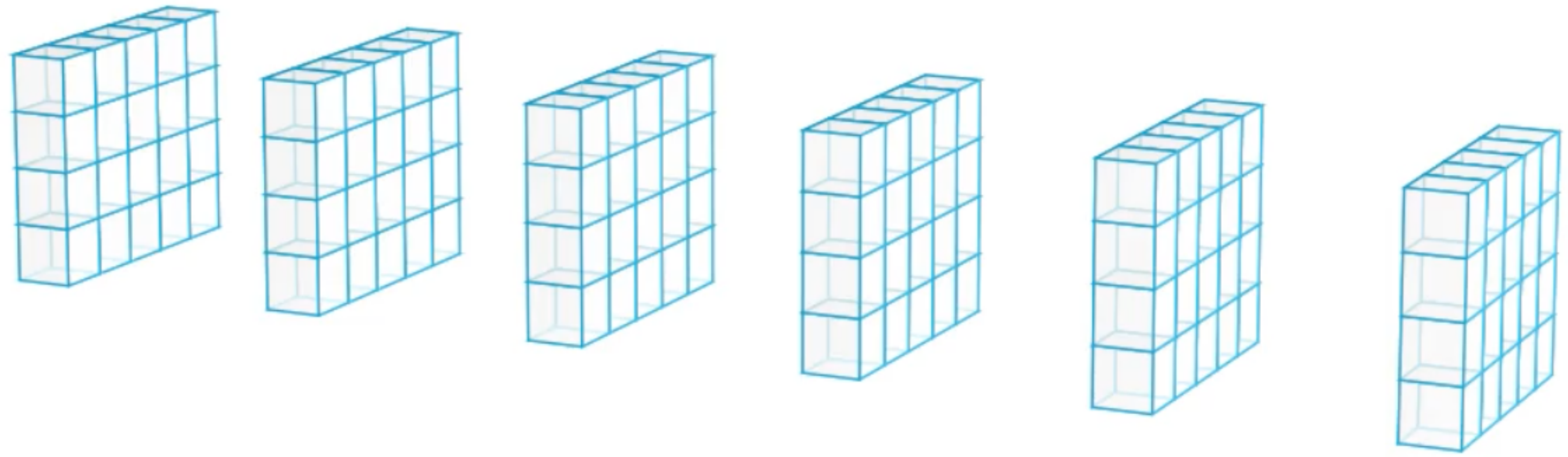
- a k-dimensional code in $\mathbb{F}_q^{m\times n}$
- an m-dimensional code in $\mathbb{F}_q^{n\times k}$
- an n-dimensional code in $\mathbb{F}_q^{m\times k}$



From matrix codes to 3-tensors

Viewed as a 3-tensor, we can see $\mathscr{C}$ from three directions

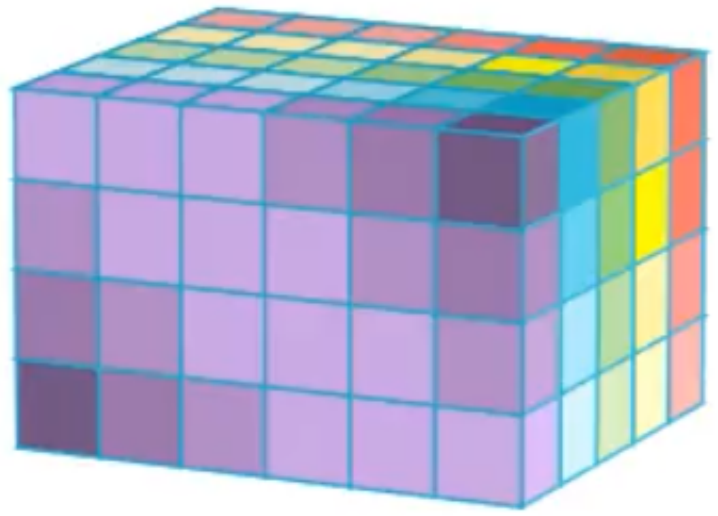
- a k-dimensional code in $\mathbb{F}_q^{m\times n}$
- an m-dimensional code in $\mathbb{F}_q^{n\times k}$
- an n-dimensional code in $\mathbb{F}_q^{m\times k}$



Tensor isomorphism

The equivalence then becomes tensor isomorphism.

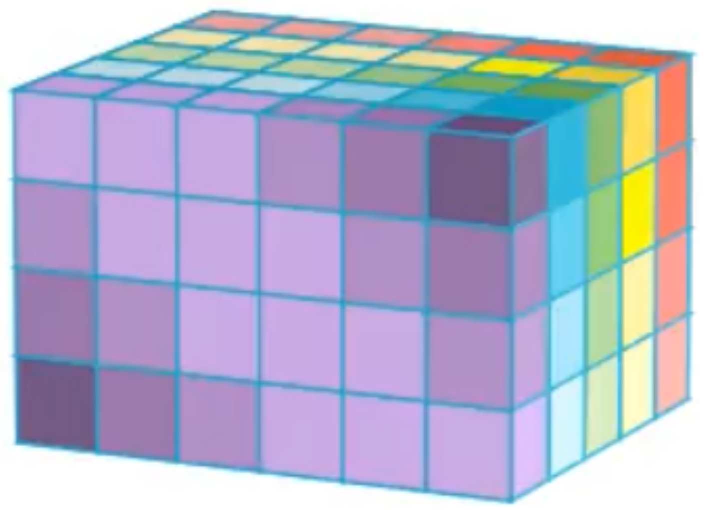
$\mathcal{C} \subseteq \mathbb{F}_q^{m \times n \times k}$



Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$\mathbf{T} \in \mathrm{GL}_k(q)$



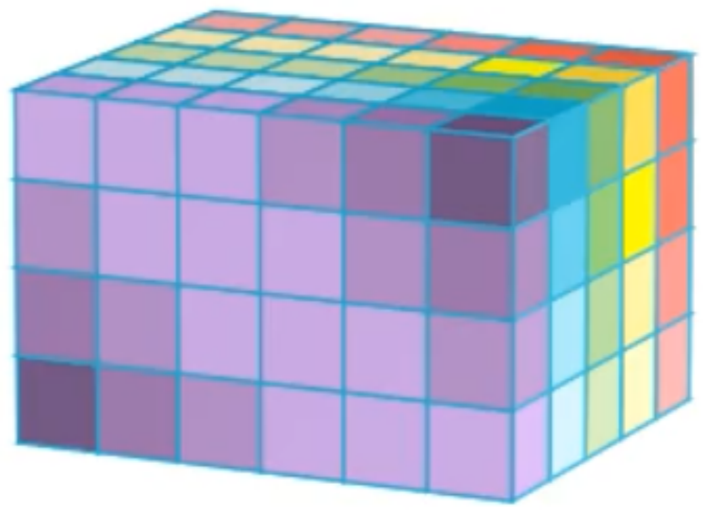
$\mathbf{A} \in \mathrm{GL}_m(q)$

$\mathbf{B} \in \mathrm{GL}_n(q)$

Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$\mathbf{T} \in \mathrm{GL}_k(q)$



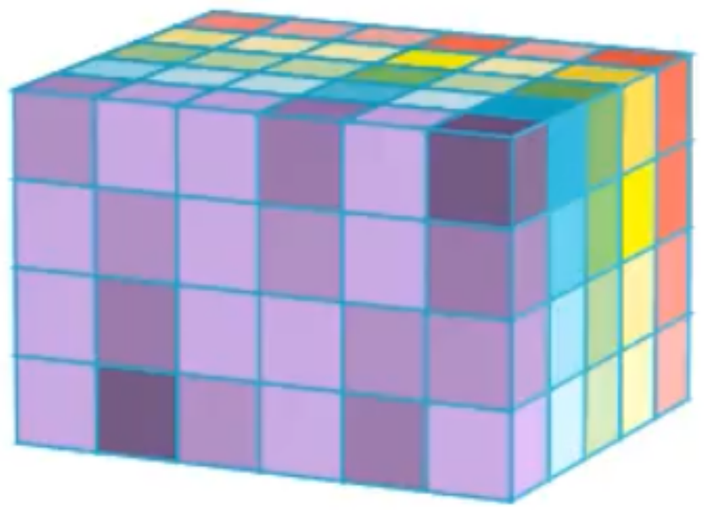
$\mathbf{A} \in \mathrm{GL}_m(q)$

$\mathbf{B} \in \mathrm{GL}_n(q)$

Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$\mathbf{T} \in \mathrm{GL}_k(q)$



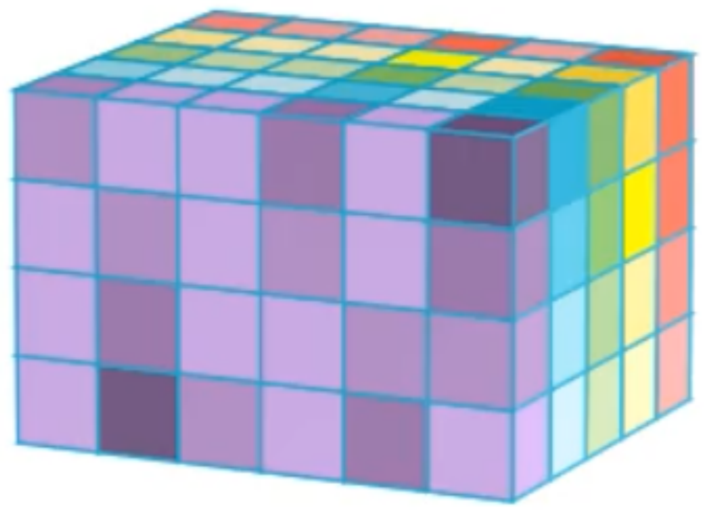
$\mathbf{A} \in \mathrm{GL}_m(q)$

$\mathbf{B} \in \mathrm{GL}_n(q)$

Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$\mathbf{T} \in \mathrm{GL}_k(q)$



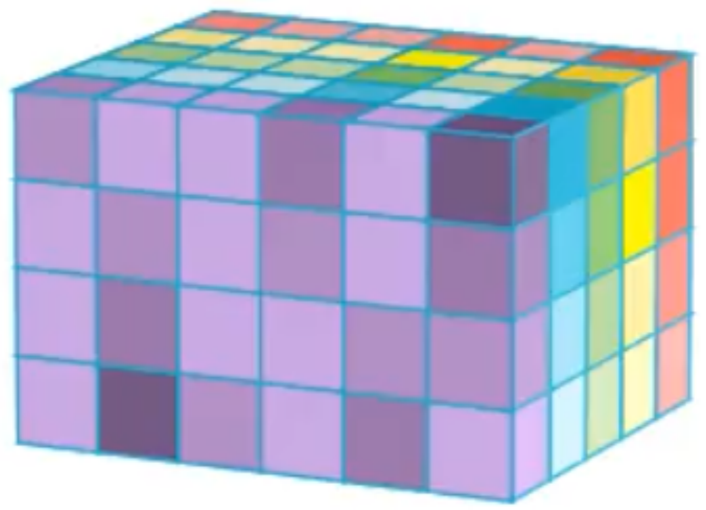
$\mathbf{A} \in \mathrm{GL}_m(q)$

$\mathbf{B} \in \mathrm{GL}_n(q)$

Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$\mathbf{T} \in \mathrm{GL}_k(q)$



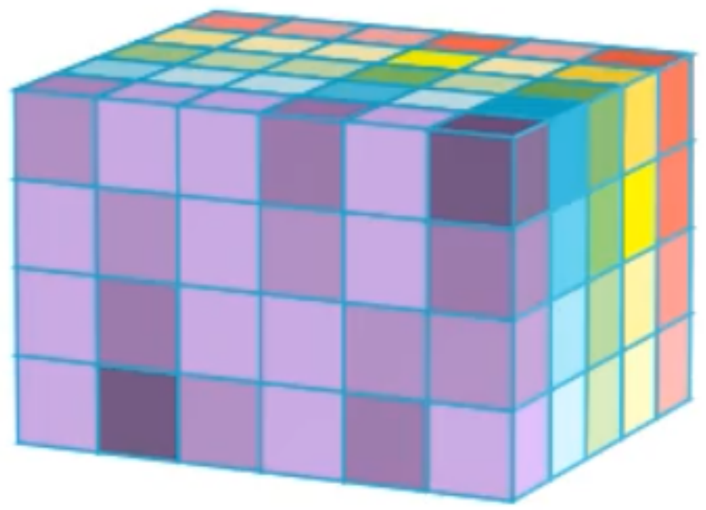
$\mathbf{A} \in \mathrm{GL}_m(q)$

$\mathbf{B} \in \mathrm{GL}_n(q)$

Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$\mathbf{T} \in \mathrm{GL}_k(q)$



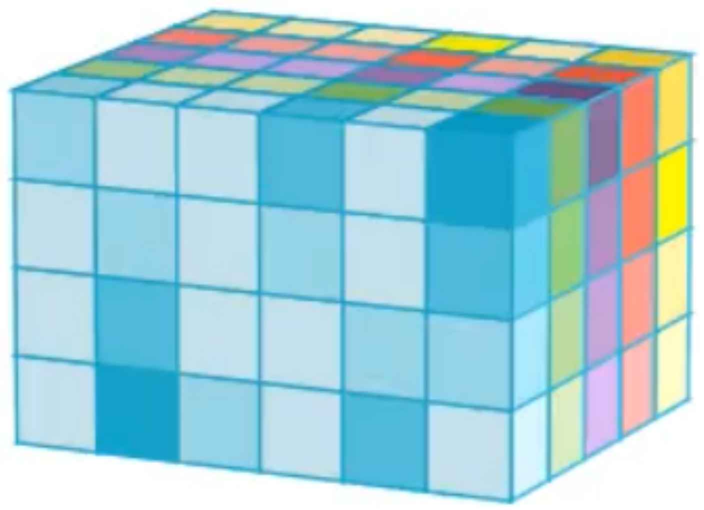
$\mathbf{A} \in \mathrm{GL}_m(q)$

$\mathbf{B} \in \mathrm{GL}_n(q)$

Tensor isomorphism

The equivalence then becomes tensor isomorphism.

$$\mathcal{D} \subseteq \mathbb{F}_q^{m \times n \times k}$$



Cryptanalysis

(The MCE case)

Algebraic attack

Algebraic attack

The MCE problem in matrix form

Let $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$ be a basis of code $\mathscr{C}$ and let $(\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(k)})$ be a basis of code $\mathscr{D}$. Find $\mathbf{A} \in \mathrm{GL}_m(\mathbb{F}_q)$, $\mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ and $\mathbf{T} \in \mathrm{GL}_k(\mathbb{F}_q)$ such that

$$\mathbf{D}^{(i)} = \sum_{1 \le j \le k} t_{j,i} \mathbf{A}\mathbf{C}^{(j)}\mathbf{B}, \quad \forall 1 \le i \le k$$

Alternatively, this gives a better modelisation:

$$\sum_{1 \le j \le k} t_{j,i} \mathbf{D}^{(j)} = \mathbf{A}\mathbf{C}^{(i)}\mathbf{B}, \quad \forall 1 \le i \le k$$

Combinatorial attack

Collision

We have a collision when we know a codeword $\mathbf{C}$ in $\mathscr{C}$ that maps to a codeword $\mathbf{D}$ in $\mathscr{D}$.

We can then infer linear constraints from

$$\mathbf{A}^{-1}\mathbf{D} = \mathbf{C}\mathbf{B}$$

If we add these linear constraints to the system obtained from the algebraic attack, we can efficiently solve the system of equations and recover the isometry (the resolution being efficient - close to polynomial - is an empirical result, not yet proven).

Collision

With two collisions, we get the following system

$$\mathbf{A}^{-1}\mathbf{D}_1 = \mathbf{C}_1\mathbf{B}$$
$$\mathbf{A}^{-1}\mathbf{D}_2 = \mathbf{C}_2\mathbf{B}$$

- Results in a linear system with the same number of variables and equations.
  - If $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2$ are all full rank, we should have a unique solution.
  - We can easily recover $\mathbf{A}$ from $\mathbf{A}^{-1}$.

The birthday paradox

What is the probability that, in a set of N randomly chosen people, at least two will share a birthday?

How big should N be to get a probability of 50% ?

$N = 23$

The birthday problem in collision search algorithms

We draw, randomly, elements from a set of size N.
How many times do we expect to draw an element before we get the same element twice?

The birthday paradox

The birthday problem in collision search algorithms

We draw, randomly, elements from a set of size N.
How many times do we expect to draw an element before we get the same element twice?

- Probability that there is no collision when the first element is drawn: 1
- Probability that there is no collision when the second element is drawn: $1 - \frac{1}{N}$
- Probability that there is no collision when the third element is drawn: $1 - \frac{2}{N}$
- …

The birthday paradox

The birthday problem in collision search algorithms

We draw, randomly, elements from a set of size N.
How many times do we expect to draw an element before we get the same element twice?

$$P(X > T) = (1 - \frac{1}{N})(1 - \frac{2}{N})\cdots(1 - \frac{T-1}{N})$$

Using a first-order Taylor approximation $e^x \approx 1 + x$, this simplifies to

$$\begin{aligned} P(X > T) &\approx e^{-\frac{1}{N}} \cdot e^{-\frac{2}{N}} \cdot \cdots \cdot e^{-\frac{T-1}{N}} \approx \\ &\approx e^{-(1+2+\cdots+(T-1))/N} \approx \\ &\approx e^{-\frac{T(T-1)}{2N}} \end{aligned}$$

For $P(X > T) \approx 63\,\%$, we get $T \approx 1.41\sqrt{N}$.

General collision attack

Algorithm 1 General Birthday-based Equivalence Finder

function SampleSet$(S, \mathbb{P}, \ell)$
    $L \leftarrow \emptyset$
    repeat
        $a \xleftarrow{\$} S$
        if $\mathbb{P}(a)$ **then** $L \leftarrow L \cup \{a\}$
        end if
    until $\mathsf{L} = \ell$
    return L
end function

function CollisionFind(S_1, S_2)
    $L_1 \leftarrow$ SampleSet$(S_1, \mathbb{P}, \ell)$
    $L_2 \leftarrow$ SampleSet$(S_2, \mathbb{P}, \ell)$
    for all $(a, b) \in L_1 \times L_2$ **do**
        $\phi \leftarrow$FindFunction(a, b)
        if $\phi \neq \perp$ **then**
            return solution ϕ
        end if
    end for
    return $\perp$
end function

Collision attack : complexity

Depends on the choice of the predicate $\mathbb{P}$. The choice is made such that we obtain the optimal balance between the two parts of the algorithm, aka. they take approximately the same time (whenever possible).

We will get an intuition for the complexity with an exercise in the assignment.

Digital signatures from equivalence problems

Zero-knowledge proof of knowledge

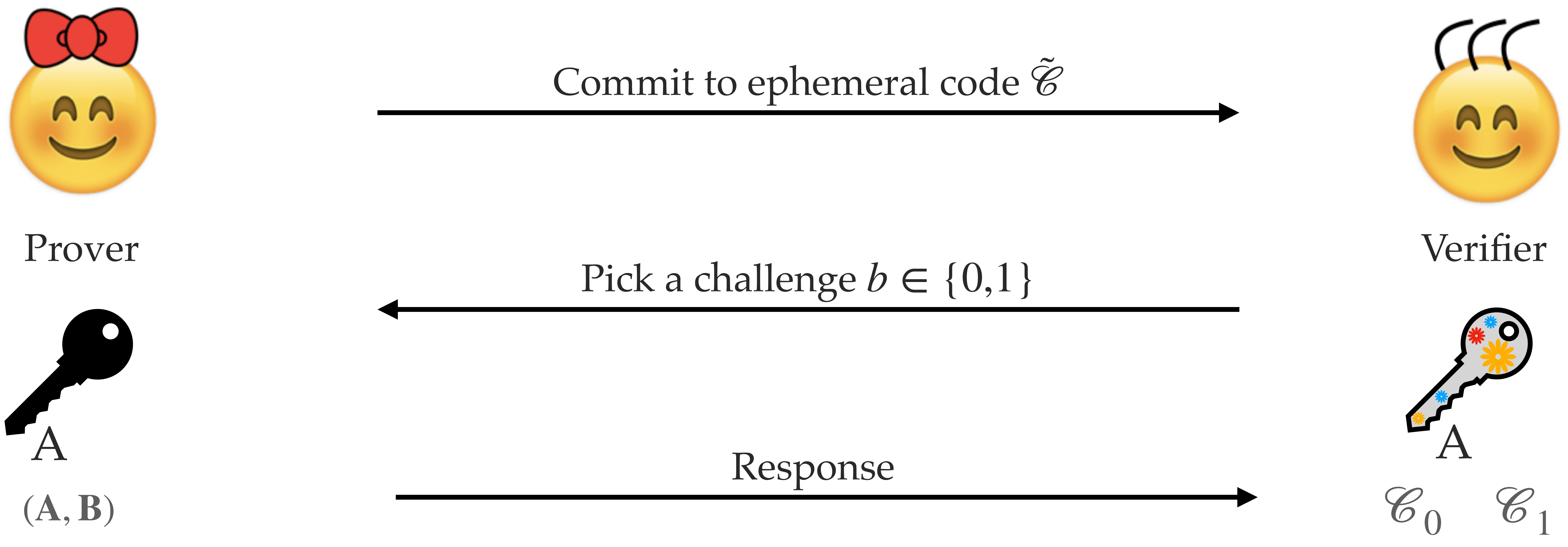
The prover needs to prove to the verifier that they know a secret, without revealing the secret or anything about the secret.

ZK identification scheme

$$\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}} \\
\mathscr{C}_1 \arrow[ur, "(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})"] &
\end{tikzcd}$$

ZK identification scheme

$$\begin{tikzcd} \mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}} \\ \mathscr{C}_1 \arrow[ur, "(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})"] & \end{tikzcd}$$



ZK identification scheme

$$
\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}} \\
\mathscr{C}_1 \arrow[ur, "(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})"] &
\end{tikzcd}
$$

Prover

A

$(\mathbf{A}, \mathbf{B})$

Commit to ephemeral code $\tilde{\mathscr{C}}$ →

← Pick a challenge $b \in \{0,1\}$

$b = 0$

Response →

$(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})$

Verifier

A

$\mathscr{C}_0$ $\mathscr{C}_1$

ZK identification scheme

$$\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}} \\
\mathscr{C}_1 \arrow[ur, "(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})"] &
\end{tikzcd}$$

Prover

A

$(\mathbf{A}, \mathbf{B})$

Commit to ephemeral code $\tilde{\mathscr{C}}$ →

← Pick a challenge $b \in \{0,1\}$

$b = 1$

Response →

$(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})$

Verifier

A

$\mathscr{C}_0$ $\mathscr{C}_1$

ZK identification scheme

→ To get a security level of 2^{λ}

$$\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}, \tilde{\mathbf{B}})"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}} \\
\mathscr{C}_1 \arrow[ur, "(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})"] &
\end{tikzcd}$$

→ Repeat λ times

$$\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}_1, \tilde{\mathbf{B}}_1)"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}}_1 \\
\mathscr{C}_1 &
\end{tikzcd}
\quad
\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}_2, \tilde{\mathbf{B}}_2)"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}}_2 \\
\mathscr{C}_1 &
\end{tikzcd}
\quad
\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}_3, \tilde{\mathbf{B}}_3)"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}}_3 \\
\mathscr{C}_1 &
\end{tikzcd}
\quad \ddots \quad
\begin{tikzcd}
\mathscr{C}_0 \arrow[r, "(\tilde{\mathbf{A}}_\lambda, \tilde{\mathbf{B}}_\lambda)"] \arrow[d, "(\mathbf{A}, \mathbf{B})"'] & \tilde{\mathscr{C}}_\lambda \\
\mathscr{C}_1 \arrow[ur, "(\tilde{\mathbf{A}} \cdot \mathbf{A}^{-1}, \mathbf{B}^{-1} \cdot \tilde{\mathbf{B}})"] &
\end{tikzcd}$$

Properties : completeness

If the statement is true, an honest prover is always able to convince an honest verifier.

Properties : soundness

A dishonest prover cannot convince an honest verifier other than with a small probability.

2-Special soundness

Having obtained two valid transcripts with the same commitment and a different challenge, we can extract a solution for the underlying problem.

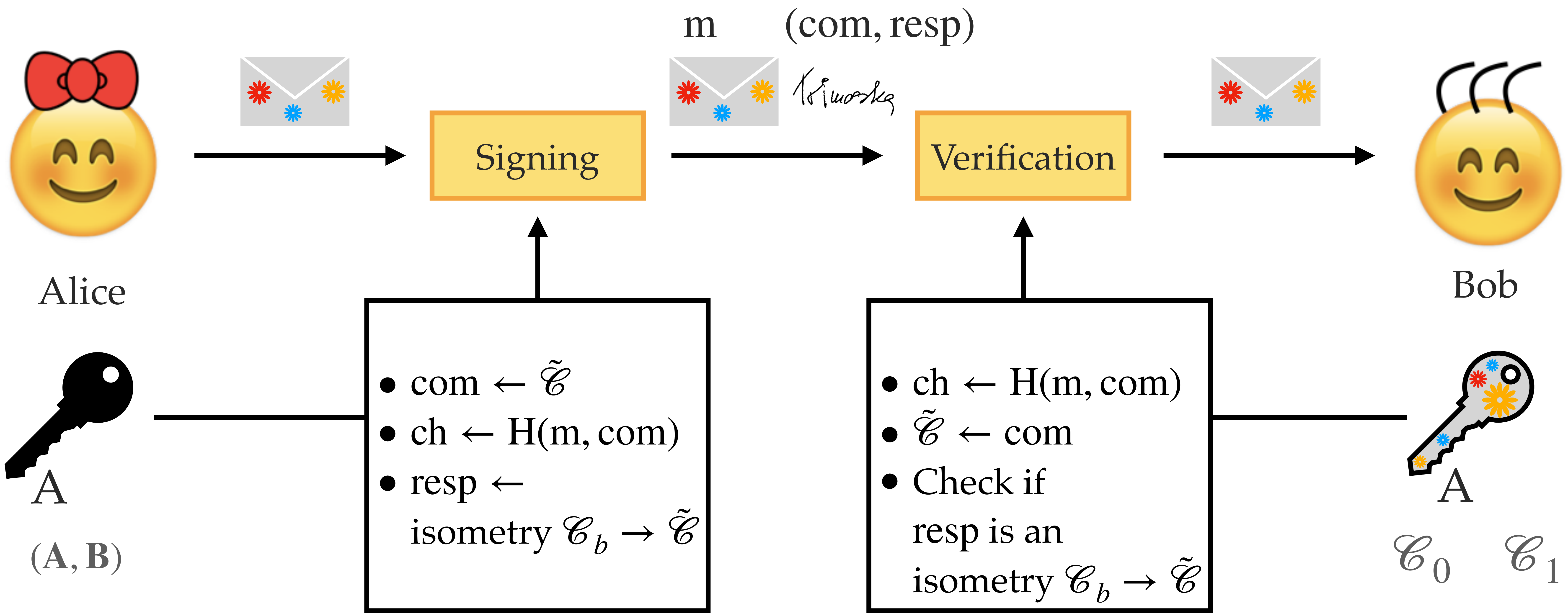
Properties : zero-knowledge

Anyone observing the transcript (including the verifier) learns nothing other than the fact that the statement is true.

The Fiat-Shamir transform

The goal is to transform an interactive identification scheme into a digital signature scheme.

Instead of the prover choosing a challenge, the challenge is determined by the hash of the message and commitments.



Soundness amplification

Multiple public keys

$$
\begin{tikzcd}
 & & \mathscr{C}_0 \arrow[dll] \arrow[d] \arrow[drr] & & \\
\mathscr{C}_1 & \cdots & \mathscr{C}_i & \cdots & \mathscr{C}_s
\end{tikzcd}
$$

- Provide s public keys
- Challenge is $b \in \{0, \ldots, s\}$
- Response is an isometry $\mathscr{C}_b \to \tilde{\mathscr{C}}$

Signatures from equivalence problems

Equivalence-based digital signature schemes in the NIST competition (and elsewhere):

LESS Linear code equivalence

MEDS Matrix code equivalence

ALTEQ Alternating trilinear form equivalence

Patarin's signature scheme: Isomorphism of polynomials (seen in previous lecture)

SeaSign, SQISign: Isogeny between elliptic curves

…