Where did my RAM go? Using algebraic cryptanalysis in practice (modelling exercises)

Lars Ran

Monika Trimoska

Summer school on RWC and privacy
July 1, Dubrovnik, Croatia

TU/e

Algebraic cryptanalysis

A type of cryptanalytic methods where the problem of finding the secret key (or any attack goal) is reduced to the problem of finding a solution to a nonlinear multivariate polynomial system of equations.

Exercise 1: Trivium

Trivium

Initialisation:

$(s_1, \ldots, s_{93}) \leftarrow (K_1, \ldots, K_{80}, 0, \ldots, 0)$

$(s_{94}, \ldots, s_{177}) \leftarrow (IV_1, \ldots, IV_{80}, 0, \ldots, 0)$

$(s_{178}, \ldots, s_{288}) \leftarrow (0, \ldots, 0, 1, 1, 1)$

Algorithm 8.1 Trivium's iterative function for keystream generation.

Input: The number of bits to be generated, denoted Z.
Output: Keystream vector z.

for $i = 1$ to Z **do**
  $t_1 \leftarrow s_{66} + s_{93}$
  $t_2 \leftarrow s_{162} + s_{177}$
  $t_3 \leftarrow s_{243} + s_{288}$
  $z_i \leftarrow t_1 + t_2 + t_3$
  $t_1 \leftarrow t_1 + s_{91} \cdot s_{92} + s_{171}$
  $t_2 \leftarrow t_2 + s_{175} \cdot s_{176} + s_{264}$
  $t_3 \leftarrow t_3 + s_{286} \cdot s_{287} + s_{69}$
  $(s_1, s_2 \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$
  $(s_{94}, s_{95} \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$
  $(s_{178}, s_{179} \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$
end for

Iterate for 1155 rounds without producing any output

Trivium

Keystream generation:

Algorithm 8.1 Trivium's iterative function for keystream generation.

Input: The number of bits to be generated, denoted Z.

Output: Keystream vector z.

for $i = 1$ to Z **do**

- $t_1 \leftarrow s_{66} + s_{93}$
- $t_2 \leftarrow s_{162} + s_{177}$
- $t_3 \leftarrow s_{243} + s_{288}$
- $z_i \leftarrow t_1 + t_2 + t_3$
- $t_1 \leftarrow t_1 + s_{91} \cdot s_{92} + s_{171}$
- $t_2 \leftarrow t_2 + s_{175} \cdot s_{176} + s_{264}$
- $t_3 \leftarrow t_3 + s_{286} \cdot s_{287} + s_{69}$
- $(s_1, s_2 \ldots, s_{93}) \leftarrow (t_3, s_1, \ldots, s_{92})$
- $(s_{94}, s_{95} \ldots, s_{177}) \leftarrow (t_1, s_{94}, \ldots, s_{176})$
- $(s_{178}, s_{179} \ldots, s_{288}) \leftarrow (t_2, s_{178}, \ldots, s_{287})$

end for

Exercise 2: MQ-Sign

The trapdoor construction

The trapdoor construction

- Central map:
  $f : (x_1, \ldots, x_n) \in \mathbb{F}_q^n \to \left(f^{(1)}(x_1, \ldots, x_n), \ldots, f^{(m)}(x_1, \ldots, x_n)\right) \in \mathbb{F}_q^m$

- Two bijective linear (or affine) transformations:
  $\mathbf{S} \in \mathrm{GL}_n(\mathbb{F}_q)$ and $\mathbf{T} \in \mathrm{GL}_m(\mathbb{F}_q)$

- Public map:
  $p = \mathbf{T} \circ f \circ \mathbf{S}$

The trapdoor construction

- Central map:
  $f : (x_1, \ldots, x_n) \in \mathbb{F}_q^n \to \left(f^{(1)}(x_1, \ldots, x_n), \ldots, f^{(m)}(x_1, \ldots, x_n)\right) \in \mathbb{F}_q^m$

- Two bijective linear (or affine) transformations:
  $\mathbf{S} \in \mathrm{GL}_n(\mathbb{F}_q)$ and $\mathbf{T} \in \mathrm{GL}_m(\mathbb{F}_q)$

- Public map:
  $p = \mathbf{T} \circ f \circ \mathbf{S}$

Main idea:

- The central map has a structure such that it is easy to find preimages: it is easy (polynomial time) to compute $f^{-1}(\mathbf{x})$ for a target vector $\mathbf{x}$.

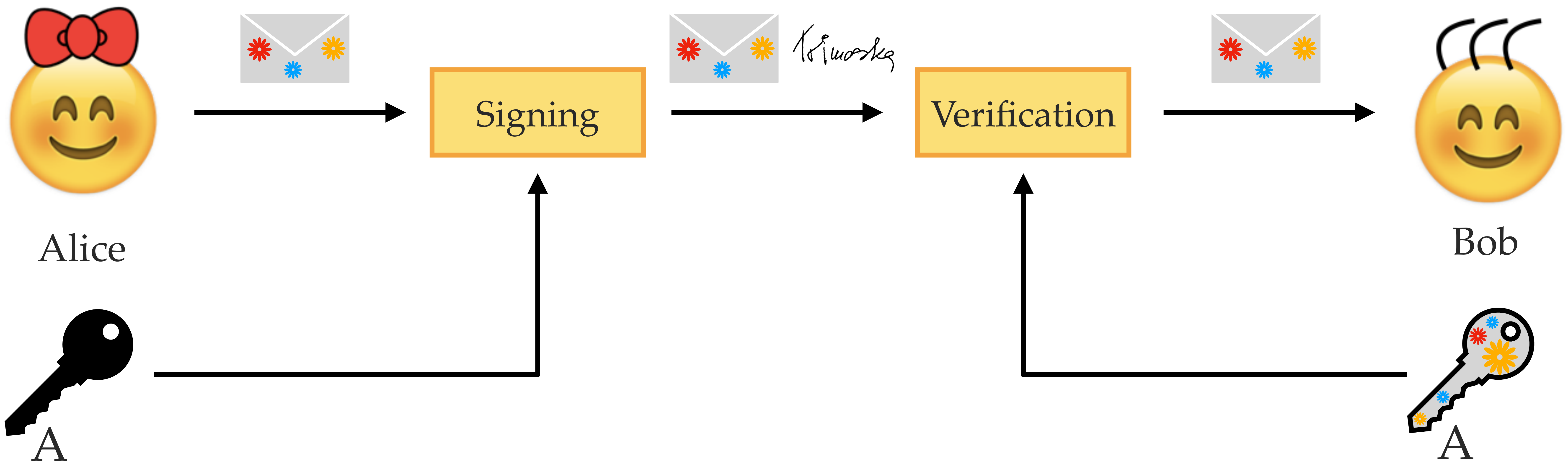
- The linear transformations hide the structure of the central map.

The trapdoor construction

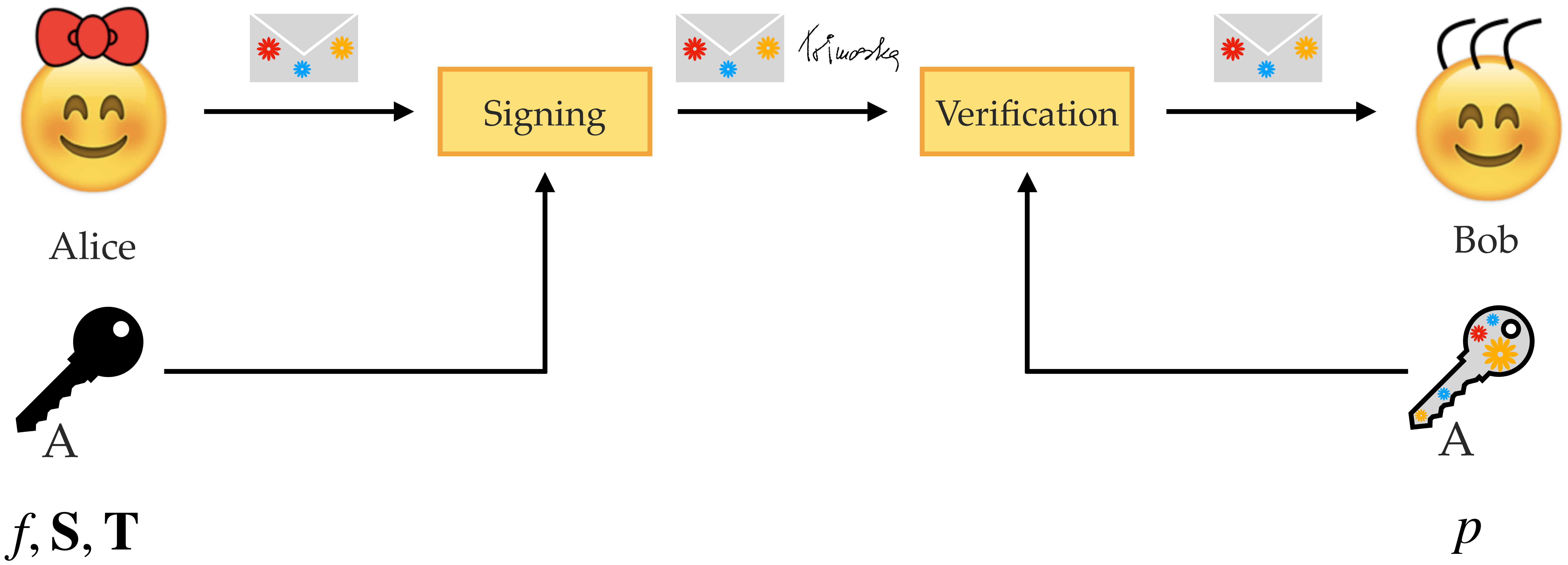
$$
\begin{tikzcd}
\mathbf{w} \in \mathbb{F}_q^m \arrow[r, "\mathbf{T}^{-1}"] & \mathbf{x} \in \mathbb{F}_q^m \arrow[r, "f^{-1}"] & \mathbf{y} \in \mathbb{F}_q^n \arrow[r, "\mathbf{S}^{-1}"] & \mathbf{z} \in \mathbb{F}_q^n \arrow[lll, bend left, "p"]
\end{tikzcd}
$$

General workflow

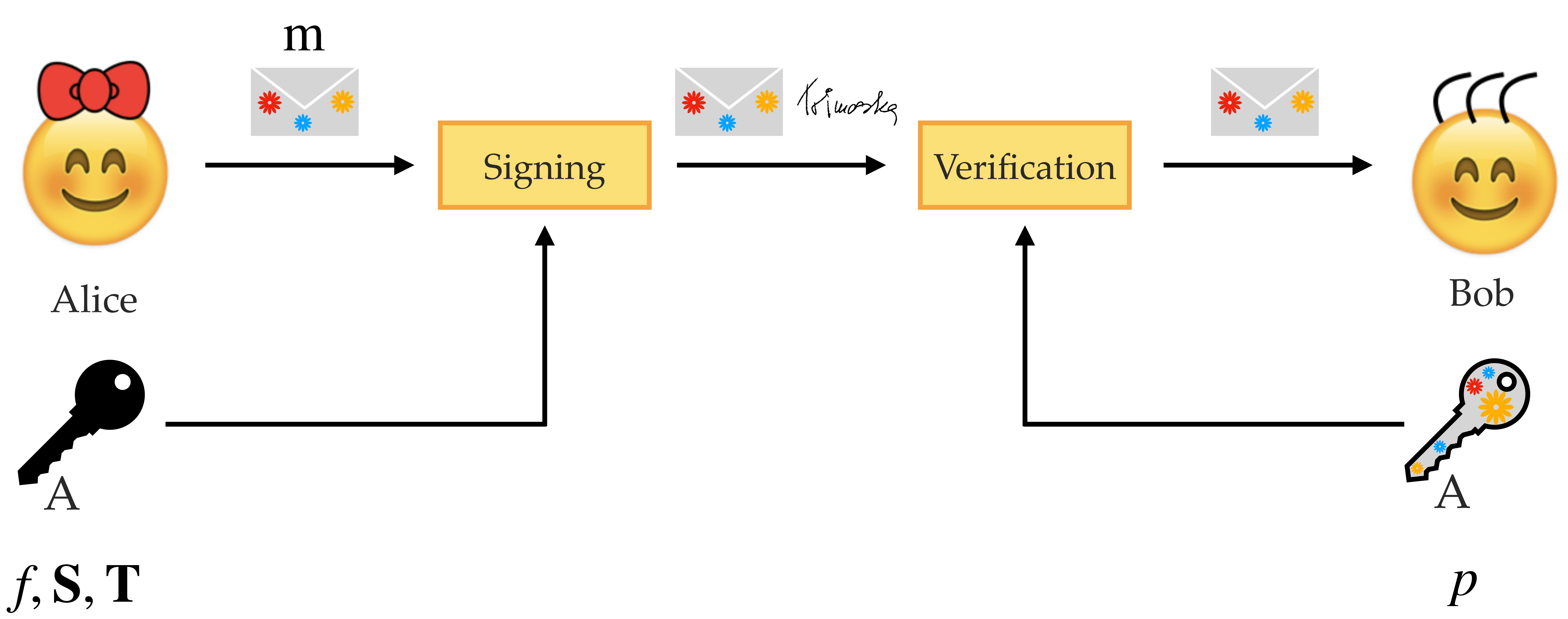
The trapdoor construction



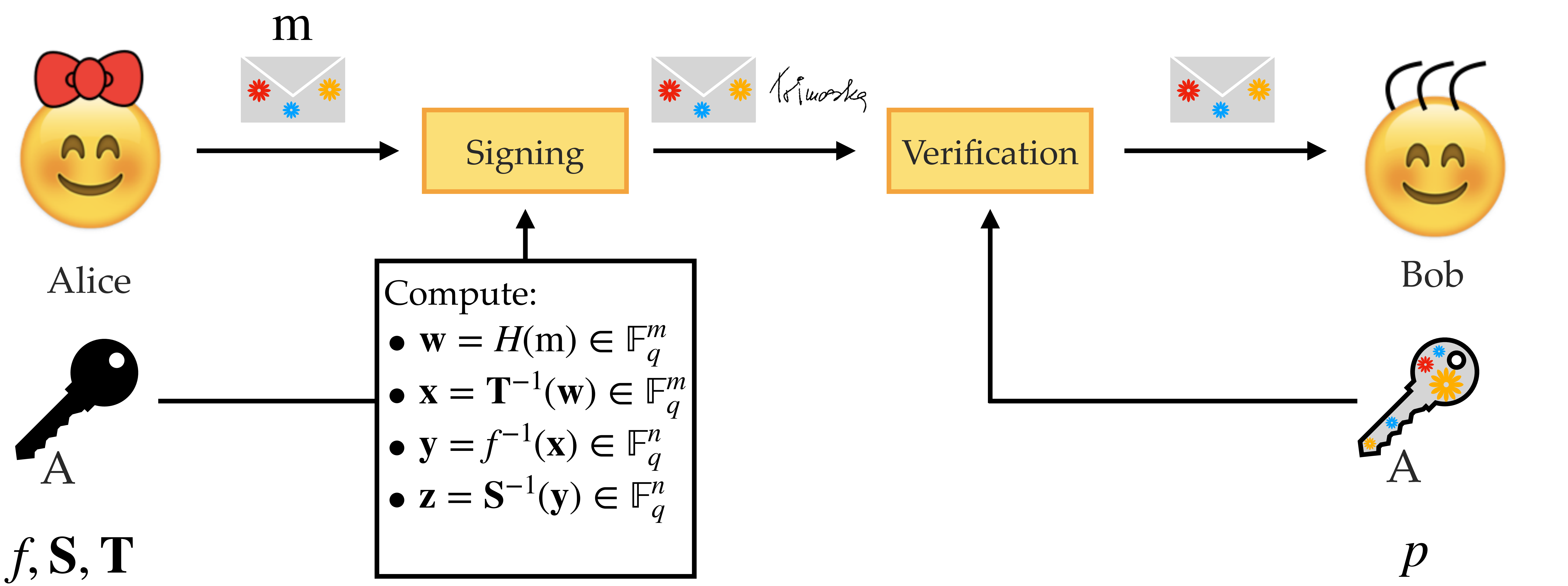
The trapdoor construction



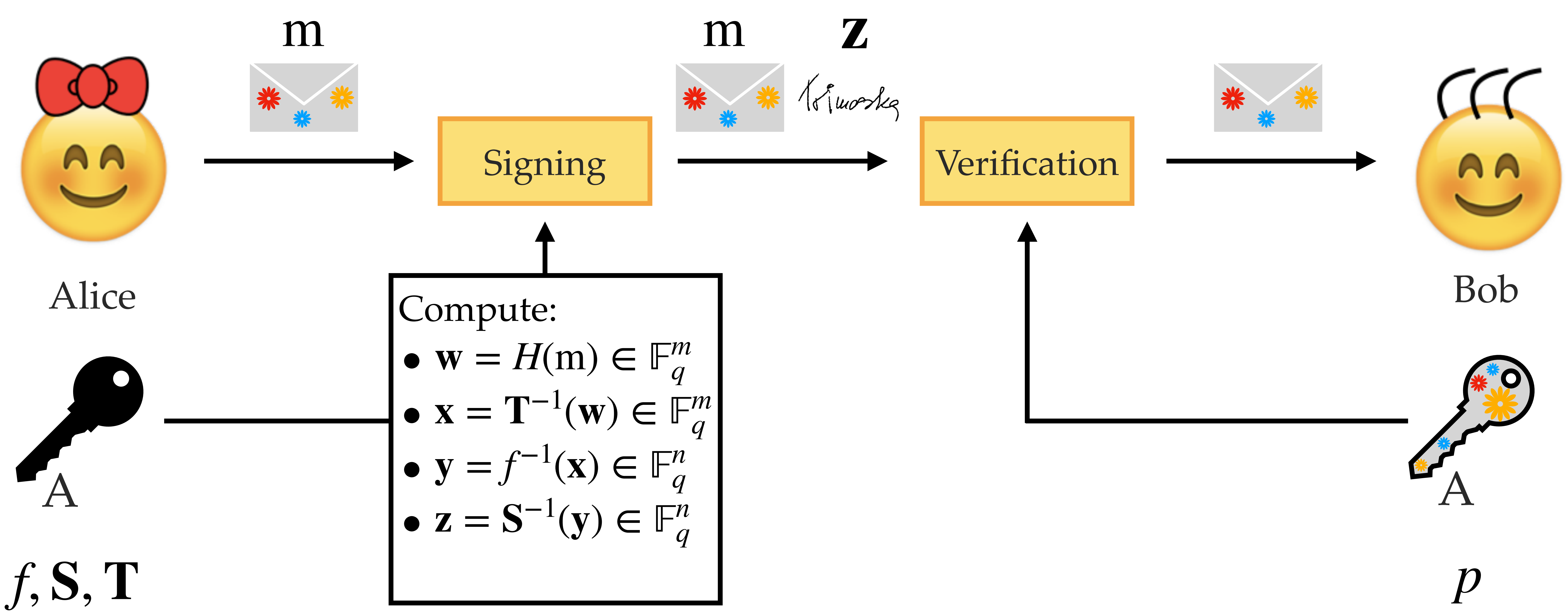
The trapdoor construction



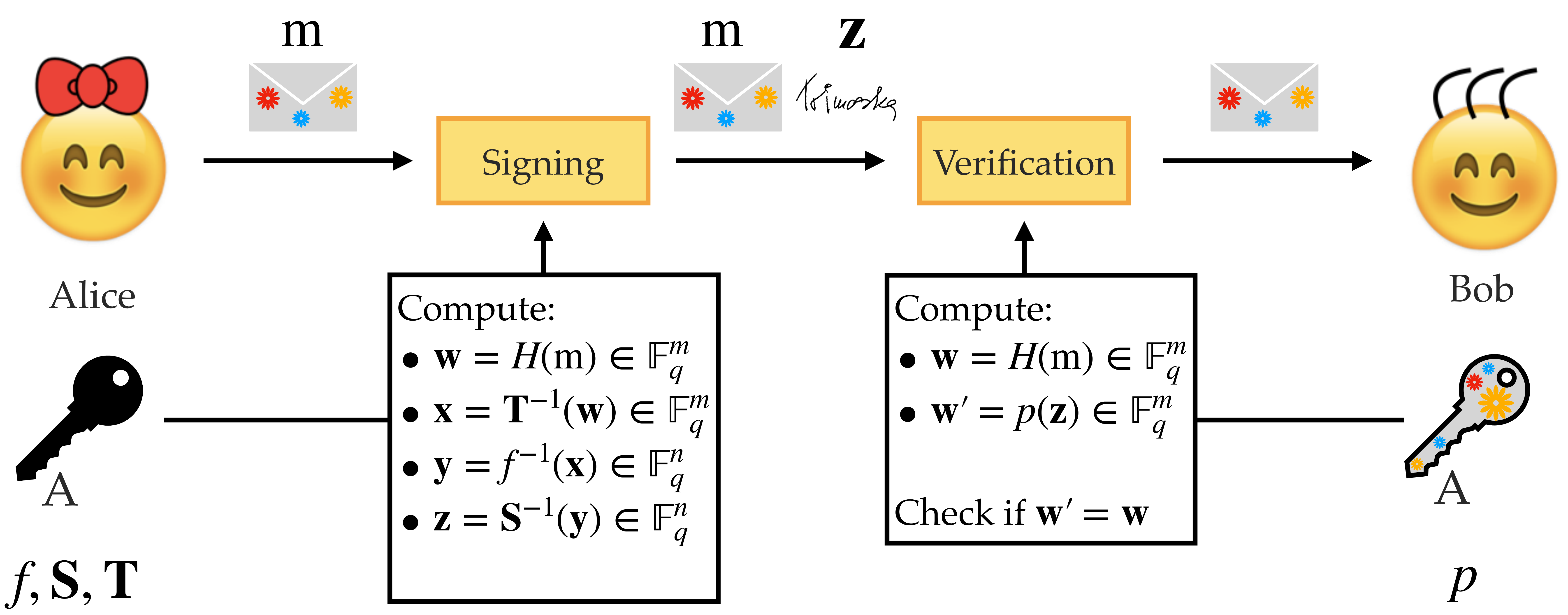
The trapdoor construction



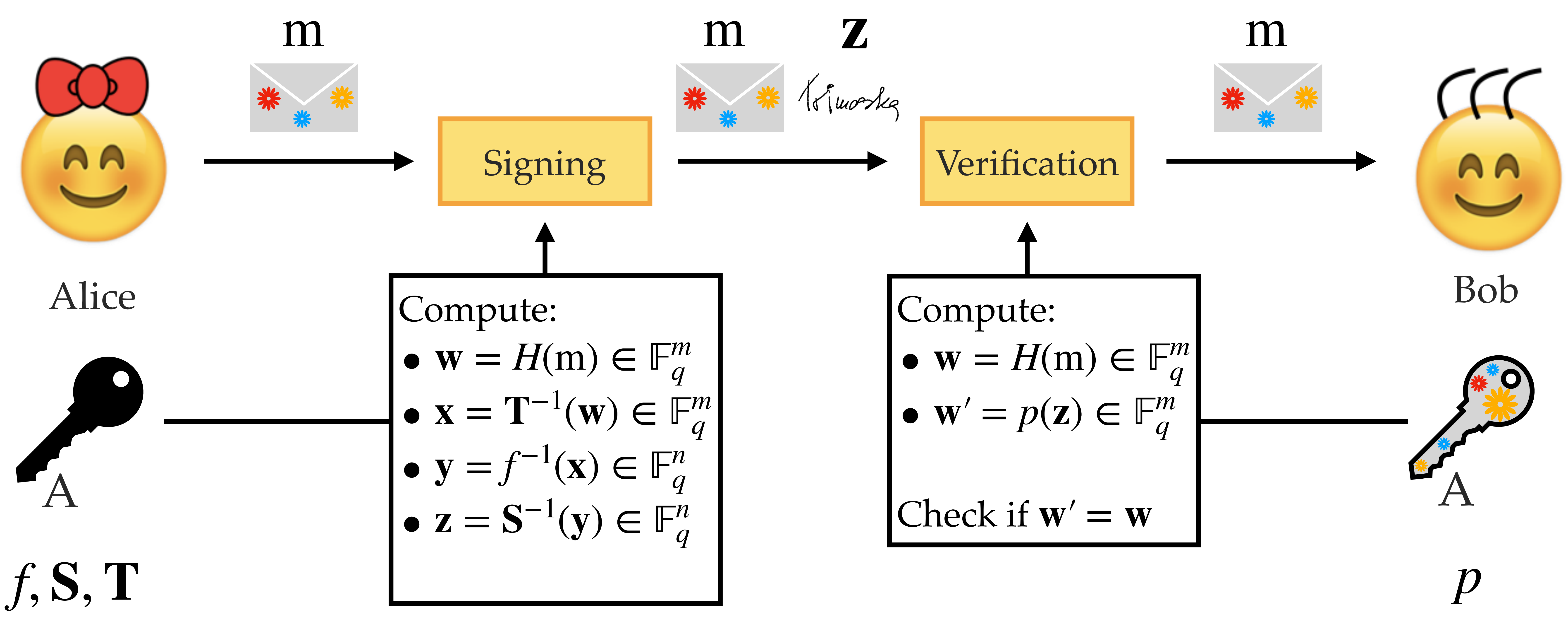
The trapdoor construction



The trapdoor construction



The trapdoor construction



The UOV central map

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, '99]

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i \in V, j \in V} \gamma_{ij}^{(k)} x_i x_j + \sum_{i \in V, j \in O} \gamma_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \alpha^{(k)}$$

Index set of vinegar variables: $V = \{1, \ldots, v\}$

Index set of oil variables: $O = \{v+1, \ldots, n\}$

The UOV central map

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, '99]

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i \in V, j \in V} \gamma_{ij}^{(k)} x_i x_j + \sum_{i \in V, j \in O} \gamma_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \alpha^{(k)}$$

Index set of vinegar variables: $V = \{1, \ldots, v\}$

Index set of oil variables: $O = \{v + 1, \ldots, n\}$

The central map is constructed in such a way that enumerating all of the vinegar variables leaves us with a linear system in the oil variables (oil does not mix with oil).

The UOV central map

Unbalanced Oil and Vinegar [Kipnis, Patarin, Goubin, '99]

$$f^{(k)}(x_1, \ldots, x_n) = \sum_{i \in V, j \in V} \gamma_{ij}^{(k)} x_i x_j + \sum_{i \in V, j \in O} \gamma_{ij}^{(k)} x_i x_j + \sum_{i=1}^{n} \beta_i^{(k)} x_i + \alpha^{(k)}$$

Index set of vinegar variables: $V = \{1, \ldots, v\}$

Index set of oil variables: $O = \{v+1, \ldots, n\}$

- The central map is constructed in such a way that enumerating all of the vinegar variables leaves us with a linear system in the oil variables (oil does not mix with oil).
- Everything is as described in the previous slides, except that we do not have a linear transformation on the output: $\mathbf{T} = \mathbf{I}$.

Matrix representation of quadratic forms

Quadratic form: $f(\mathbf{x}) = \sum \gamma_{ij} x_i x_j$

$$\begin{array}{ccc} \mathbf{x}^\top & \mathbf{F} & \mathbf{x} \\ \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix} & \begin{bmatrix} \gamma_{1,1} & \frac{\gamma_{1,2}}{2} & \frac{\gamma_{1,3}}{2} & \frac{\gamma_{1,4}}{2} \\ \frac{\gamma_{2,1}}{2} & \gamma_{2,2} & \frac{\gamma_{2,3}}{2} & \frac{\gamma_{2,4}}{2} \\ \frac{\gamma_{3,1}}{2} & \frac{\gamma_{3,2}}{2} & \gamma_{3,3} & \frac{\gamma_{3,4}}{2} \\ \frac{\gamma_{4,1}}{2} & \frac{\gamma_{4,2}}{2} & \frac{\gamma_{4,3}}{2} & \gamma_{4,4} \end{bmatrix} & \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} \end{array}$$

so with $\mathbf{x} = (x_1, \ldots, x_n)$, we get $\mathbf{x}^\top \mathbf{F} \mathbf{x}$.

Key generation

In matrix representation

$$\mathbf{P}^{(k)} = \mathbf{S}^{\top}\mathbf{F}^{(k)}\mathbf{S}, \text{ for all } k \in \{1, \ldots, m\}.$$

Key generation

In matrix representation

$$\mathbf{P}^{(k)} = \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}, \text{ for all } k \in \{1, \ldots, m\}.$$

Why ?

Key generation

In matrix representation

$$\mathbf{P}^{(k)} = \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}, \text{ for all } k \in \{1, \ldots, m\}.$$

Why ?

By definition, $p = f \circ \mathbf{S}$.

Key generation

In matrix representation

$\mathbf{P}^{(k)} = \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}$, for all $k \in \{1, \ldots, m\}$.

Why ?

By definition, $p = f \circ \mathbf{S}$.

In matrix representation, we need:

$$\mathbf{x}^\top \mathbf{P}^{(k)} \mathbf{x} = (\mathbf{S}\mathbf{x})^\top \mathbf{F}^{(k)} (\mathbf{S}\mathbf{x})$$

Key generation

In matrix representation

$\mathbf{P}^{(k)} = \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}$, for all $k \in \{1, \ldots, m\}$.

Why ?

By definition, $p = f \circ \mathbf{S}$.

In matrix representation, we need:

$$\mathbf{x}^\top \mathbf{P}^{(k)} \mathbf{x} = (\mathbf{S}\mathbf{x})^\top \mathbf{F}^{(k)} (\mathbf{S}\mathbf{x})$$

$$\mathbf{x}^\top \mathbf{P}^{(k)} \mathbf{x} = \mathbf{x}^\top \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}\mathbf{x}$$

Key generation

In matrix representation

$$\mathbf{P}^{(k)} = \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S}, \text{ for all } k \in \{1, \ldots, m\}.$$

Why ?

By definition, $p = f \circ \mathbf{S}$.

In matrix representation, we need:

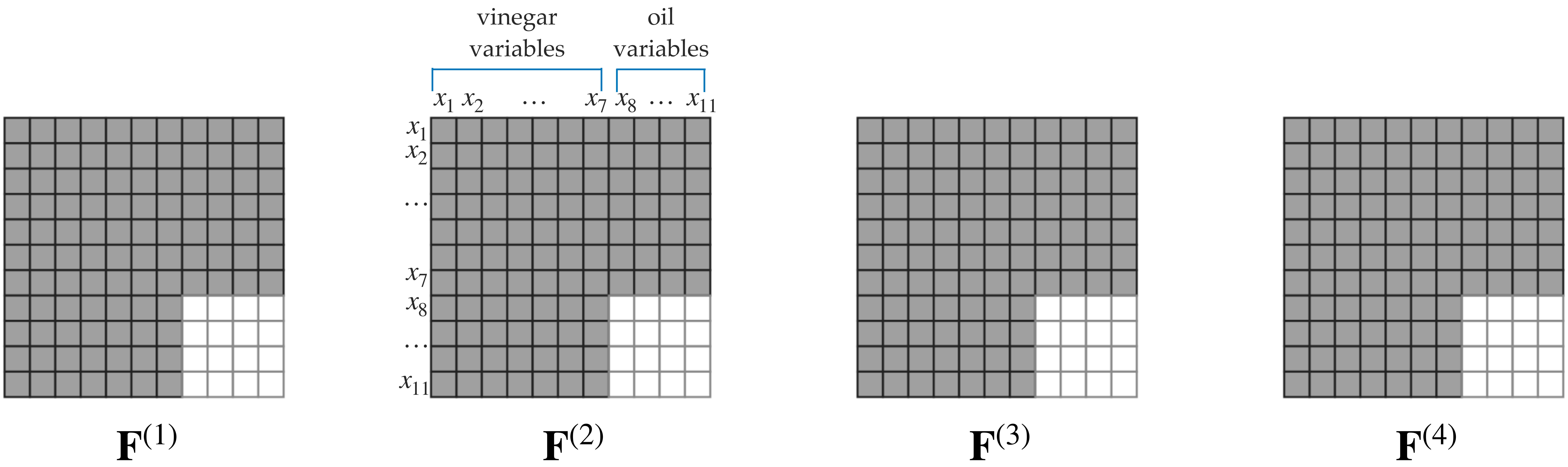
$$\mathbf{x}^\top \mathbf{P}^{(k)} \mathbf{x} = (\mathbf{S}\mathbf{x})^\top \mathbf{F}^{(k)} (\mathbf{S}\mathbf{x})$$

$$\mathbf{x}^\top \mathbf{P}^{(k)} \mathbf{x} = \mathbf{x}^\top \mathbf{S}^\top \mathbf{F}^{(k)} \mathbf{S} \mathbf{x}$$

Spoilers ahead !

The UOV central map

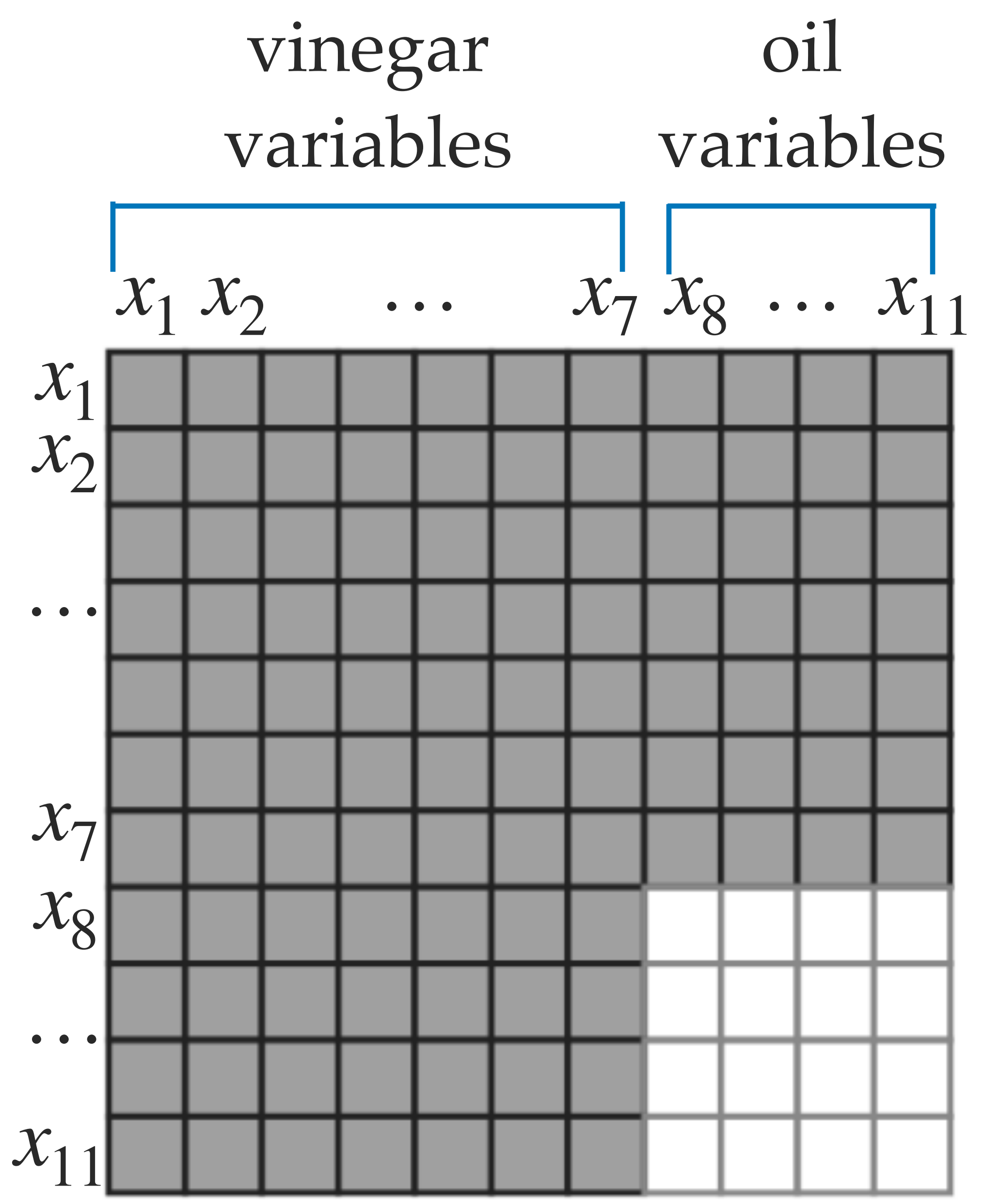
Toy example: $v = 7, m = 4$



*Grayed areas represent the entries that are possibly nonzero; blank areas denote the zero entries;

MQ-Sign

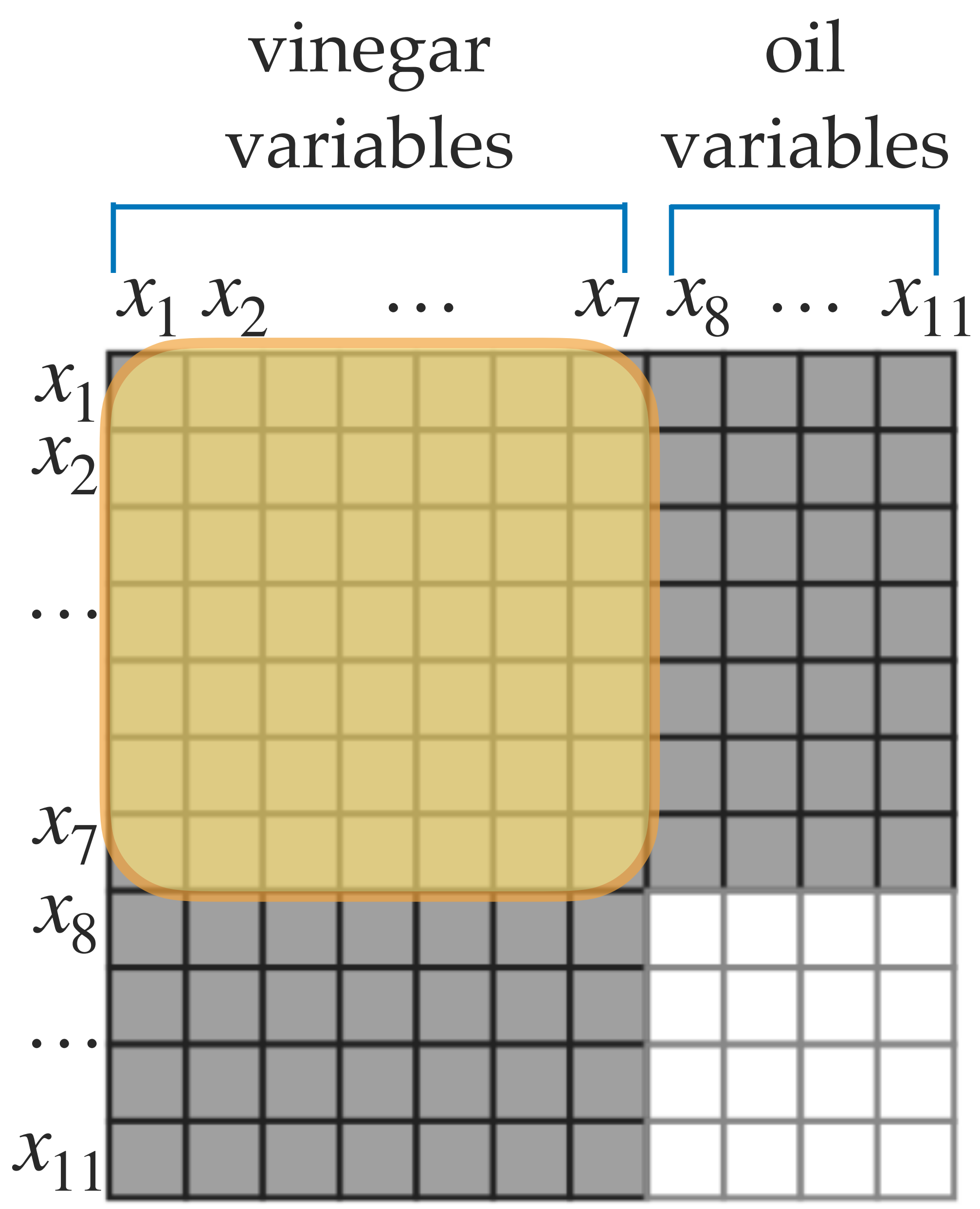
Variants with additional structure to the vinegar-vinegar or/and the vinegar-oil part, with the goal to reduce the size of the secret key.



$\mathbf{F}^{(i)}$

MQ-Sign

Variants with additional structure to the vinegar-vinegar or/and the vinegar-oil part, with the goal to reduce the size of the secret key.

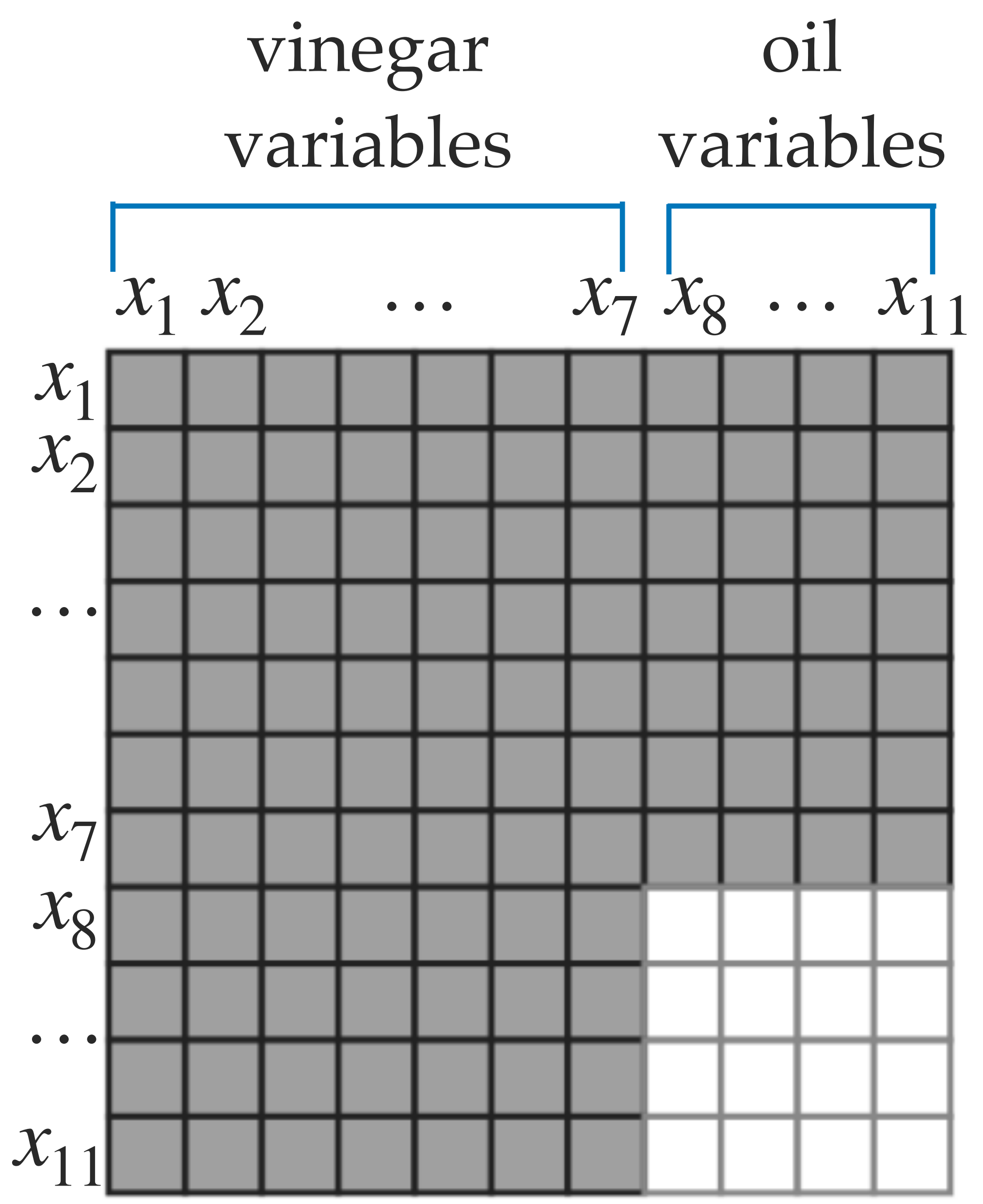


$\mathbf{F}^{(i)}$

The vinegar-vinegar part

MQ-Sign

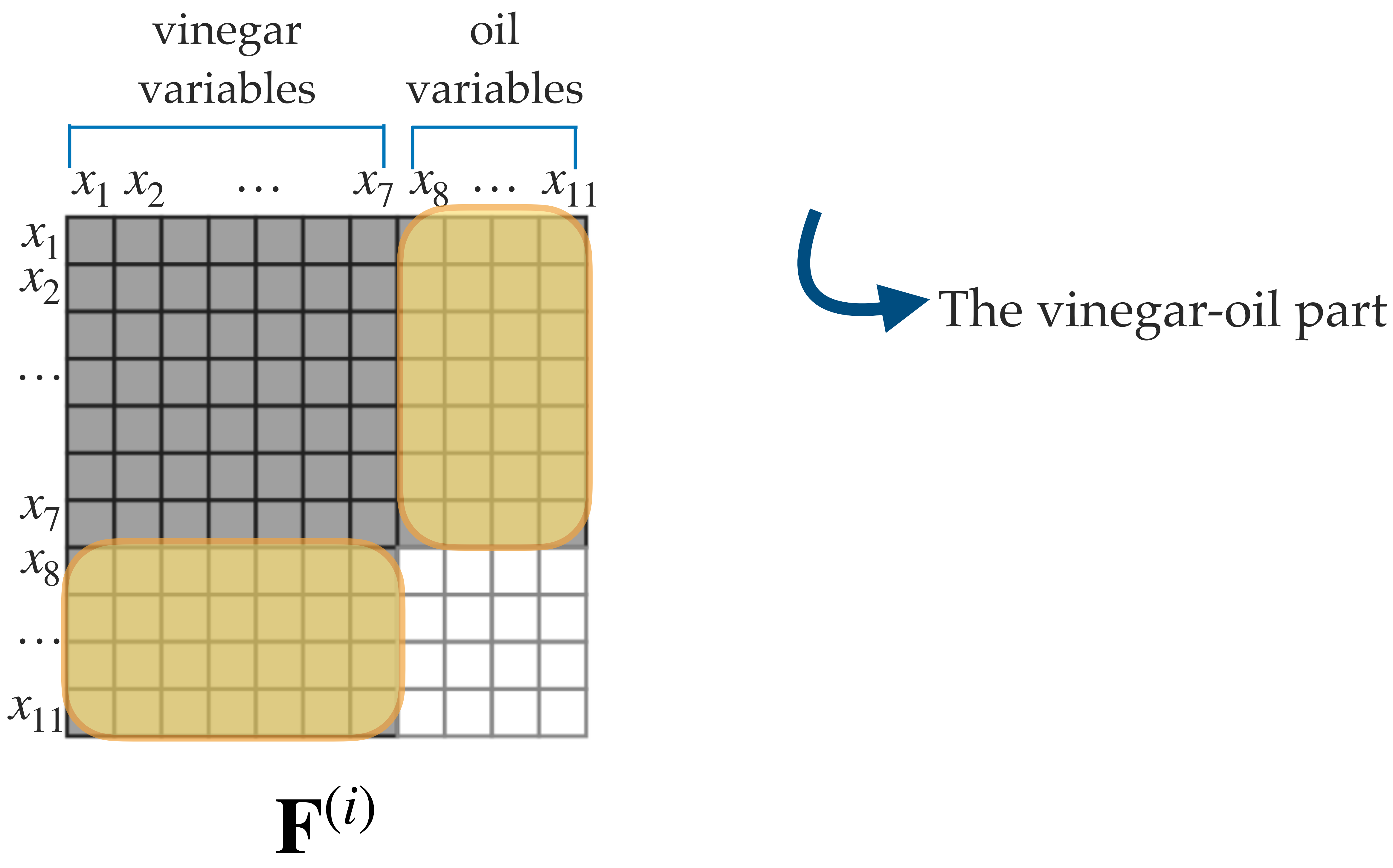
Variants with additional structure to the vinegar-vinegar or/and the vinegar-oil part, with the goal to reduce the size of the secret key.



$\mathbf{F}^{(i)}$

MQ-Sign

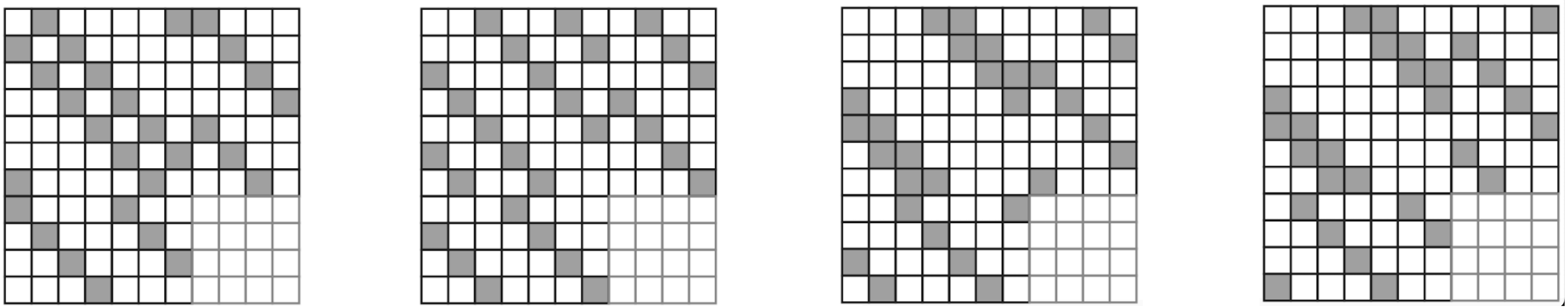
Variants with additional structure to the vinegar-vinegar or/and the vinegar-oil part, with the goal to reduce the size of the secret key.



MQ-Sign (round 1)

Variants with additional structure to the vinegar-vinegar or/and the vinegar-oil part, with the goal to reduce the size of the secret key.

Toy example: $v = 7, m = 4$



$\mathbf{F}^{(1)}$ $\mathbf{F}^{(2)}$ $\mathbf{F}^{(3)}$ $\mathbf{F}^{(4)}$

Equivalent secret keys

For any instance of a UOV secret key $(f', \mathbf{S}')$, there exists an equivalent secret key $(f, \mathbf{S})$ with

$$\mathbf{S} = \begin{pmatrix} \mathbf{I}_{v\times v} & \mathbf{S}_1 \\ \mathbf{0}_{m\times v} & \mathbf{I}_{m\times m} \end{pmatrix}.$$

- A key of this *equivalent keys* form is used for efficiency (fewer entries in $\mathbf{S}$).

Equivalent secret keys optimisation

Key generation $\mathbf{P} = \mathbf{S}^\top \mathbf{F} \mathbf{S}$

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & 0 \\ \mathbf{S}_1^\top & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{F}_1^{(k)} & \mathbf{F}_2^{(k)} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{S}_1 \\ 0 & \mathbf{I} \end{pmatrix}$$

Equivalent secret keys optimisation

Key generation $\mathbf{P} = \mathbf{S}^\top \mathbf{F} \mathbf{S}$

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{I} & 0 \\ \mathbf{S}_1^\top & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{F}_1^{(k)} & \mathbf{F}_2^{(k)} \\ 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{I} & \mathbf{S}_1 \\ 0 & \mathbf{I} \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ 0 & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}$$

Recovering the central map

From the equivalence:

Recovering the central map

From the equivalence:

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ 0 & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}$$

Recovering the central map

From the equivalence:

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ 0 & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}$$

Recovering the central map

From the equivalence:

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ 0 & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}$$

Recovering the central map

From the equivalence:

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ 0 & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}$$

We obtain constraints:

$$\mathbf{P}_1^{(k)} = \mathbf{F}_1^{(k)}$$

$$\mathbf{P}_2^{(k)} = (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)}$$

Recovering the central map

From the equivalence:

$$\begin{pmatrix} \mathbf{P}_1^{(k)} & \mathbf{P}_2^{(k)} \\ 0 & \mathbf{P}_4^{(k)} \end{pmatrix} = \begin{pmatrix} \mathbf{F}_1^{(k)} & (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)} \\ 0 & \mathsf{Upper}(\mathbf{S}_1^\top \mathbf{F}_1^{(k)} \mathbf{S}_1 + \mathbf{S}_1^\top \mathbf{F}_2^{(k)}) \end{pmatrix}$$

We obtain constraints:

$$\mathbf{P}_1^{(k)} = \mathbf{F}_1^{(k)}$$
$$\mathbf{P}_2^{(k)} = (\mathbf{F}_1^{(k)} + \mathbf{F}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)}$$

$$\longrightarrow \qquad \mathbf{P}_2^{(k)} = (\mathbf{P}_1^{(k)} + \mathbf{P}_1^{(k)\top})\mathbf{S}_1 + \mathbf{F}_2^{(k)}$$

Recovering the central map

We obtain equations:

*$\tilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1(k) + \mathbf{P}_1(k)^\top$

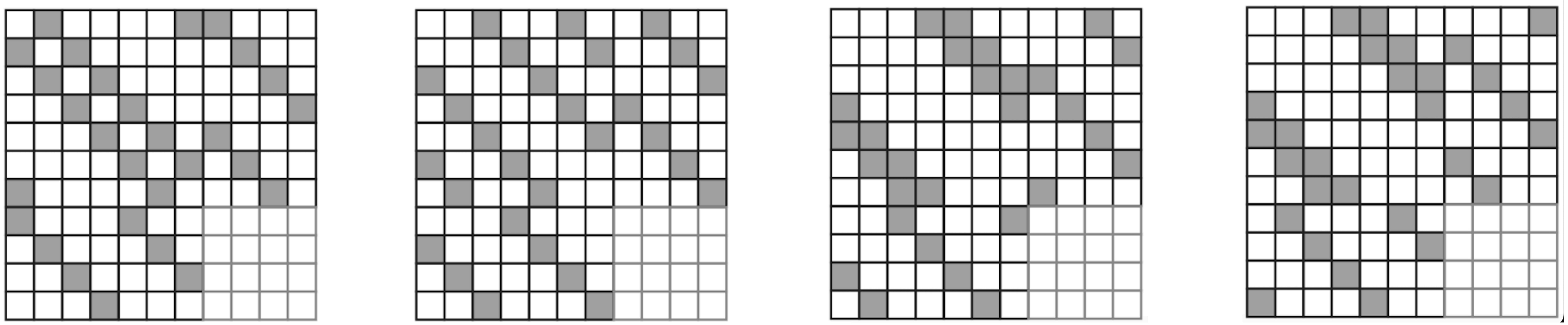
$$\sum_{1\le p\le v} \tilde{\mathbf{P}}^{(k)}_{1[ip]}\mathbf{S}_{1[pj]} - \mathbf{P}_{2[ij]} = 0, \quad \forall (i,j,k) \text{ s.t. } \mathbf{F}^{(k)}_{2[ij]} = 0$$

Recovering the central map

We obtain equations:

$^*\tilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1(k) + \mathbf{P}_1(k)^\top$

Recall the structure of the central map



Recovering the central map

We obtain equations:

$^*\tilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1(k) + \mathbf{P}_1(k)^\top$

$$\sum_{1 \le p \le v} \tilde{\mathbf{P}}_{1[ip]}^{(k)} \mathbf{S}_{1[pj]} - \mathbf{P}_{2[ij]} = 0, \quad \forall (i, j, k) \text{ s.t. } \mathbf{F}_{2[ij]}^{(k)} = 0$$

Recovering the central map

We obtain equations:

$*\tilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1(k) + \mathbf{P}_1(k)^\top$

$$\sum_{1 \le p \le v} \tilde{\mathbf{P}}^{(k)}_{1[ip]} \mathbf{S}_{1[pj]} - \mathbf{P}_{2[ij]} = 0, \quad \forall (i, j, k) \text{ s.t. } \mathbf{F}^{(k)}_{2[ij]} = 0$$

for $mv(m-1)$ entries

Recovering the central map

We obtain equations:

*$\tilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1(k) + \mathbf{P}_1(k)^\top$

$$\sum_{1 \le p \le v} \tilde{\mathbf{P}}_{1[ip]}^{(k)} \mathbf{S}_{1[pj]} - \mathbf{P}_{2[ij]} = 0, \quad \forall (i, j, k) \text{ s.t. } \mathbf{F}_{2[ij]}^{(k)} = 0$$

for $mv(m-1)$ entries

huge probability to obtain vm linearly independent equations

Recovering the central map

We obtain equations:

*$\tilde{\mathbf{P}}_1^{(k)} = \mathbf{P}_1(k) + \mathbf{P}_1(k)^\top$

$$\sum_{1 \le p \le v} \tilde{\mathbf{P}}^{(k)}_{1[ip]} \mathbf{S}_{1[pj]} - \mathbf{P}_{2[ij]} = 0, \quad \forall (i, j, k) \text{ s.t. } \mathbf{F}^{(k)}_{2[ij]} = 0$$

for $mv(m-1)$ entries

huge probability to obtain vm linearly independent equations

Complexity of solving the system column-by-column:

$$\mathcal{O}(mv^\omega)$$

Exercise 3:
Matrix Code Equivalence

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 $\mathbb{F}_q^{m\times n}$.

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 $\mathbb{F}_q^{m\times n}$.

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

Matrix code

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

Basis of a matrix code

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

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.

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

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.

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.

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?

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 $\mathbf{M} \in \mathbb{F}_q^{n \times r}$

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 $\mathbf{M} \in \mathbb{F}_q^{n \times r}$ ✗

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 $\mathbf{B} \in \mathrm{GL}_n$

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 $\mathbf{B} \in \mathrm{GL}_n$ ✓

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 $\mathbf{A} \in \mathrm{GL}_m$

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 $\mathbf{A} \in \mathrm{GL}_m$ ✓

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 $\mathbf{ACB} \in \mathscr{D}$.

Matrix code equivalence

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

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

Matrix code equivalence

The MCE problem in matrix form

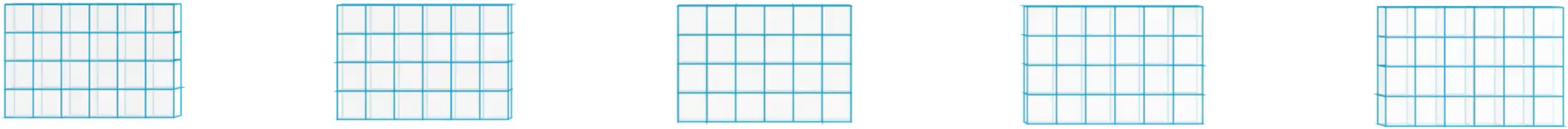
Let $(\mathbf{C}^{(1)}, \ldots, \mathbf{C}^{(k)})$ be a basis of code $\mathscr{C}$ and let $(\mathbf{D}^{(1)}, \ldots, \mathbf{D}^{(k)})$ be a basis of code $\mathscr{D}$. Find $\mathbf{A} \in \mathrm{GL}_m(\mathbb{F}_q)$, $\mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ and $\mathbf{T} \in \mathrm{GL}_k(\mathbb{F}_q)$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

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

change of basis

From matrix codes to 3-tensors

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

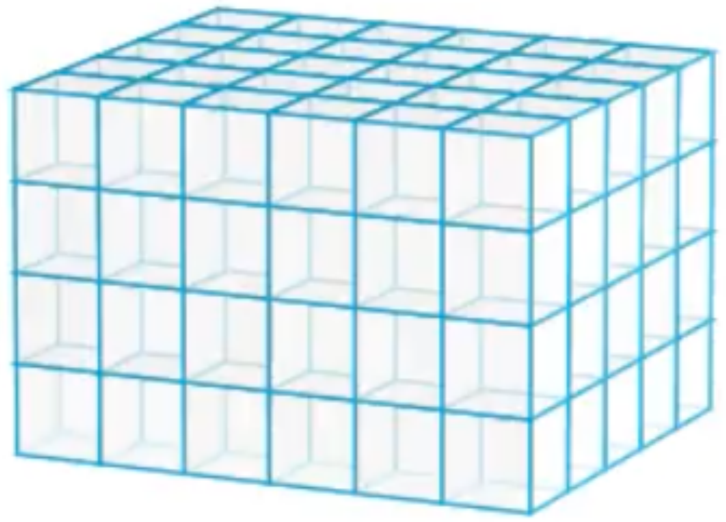


$\mathbf{C}_1$ $\mathbf{C}_2$ $\mathbf{C}_3$ $\mathbf{C}_4$ $\mathbf{C}_5$

From matrix codes to 3-tensors

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

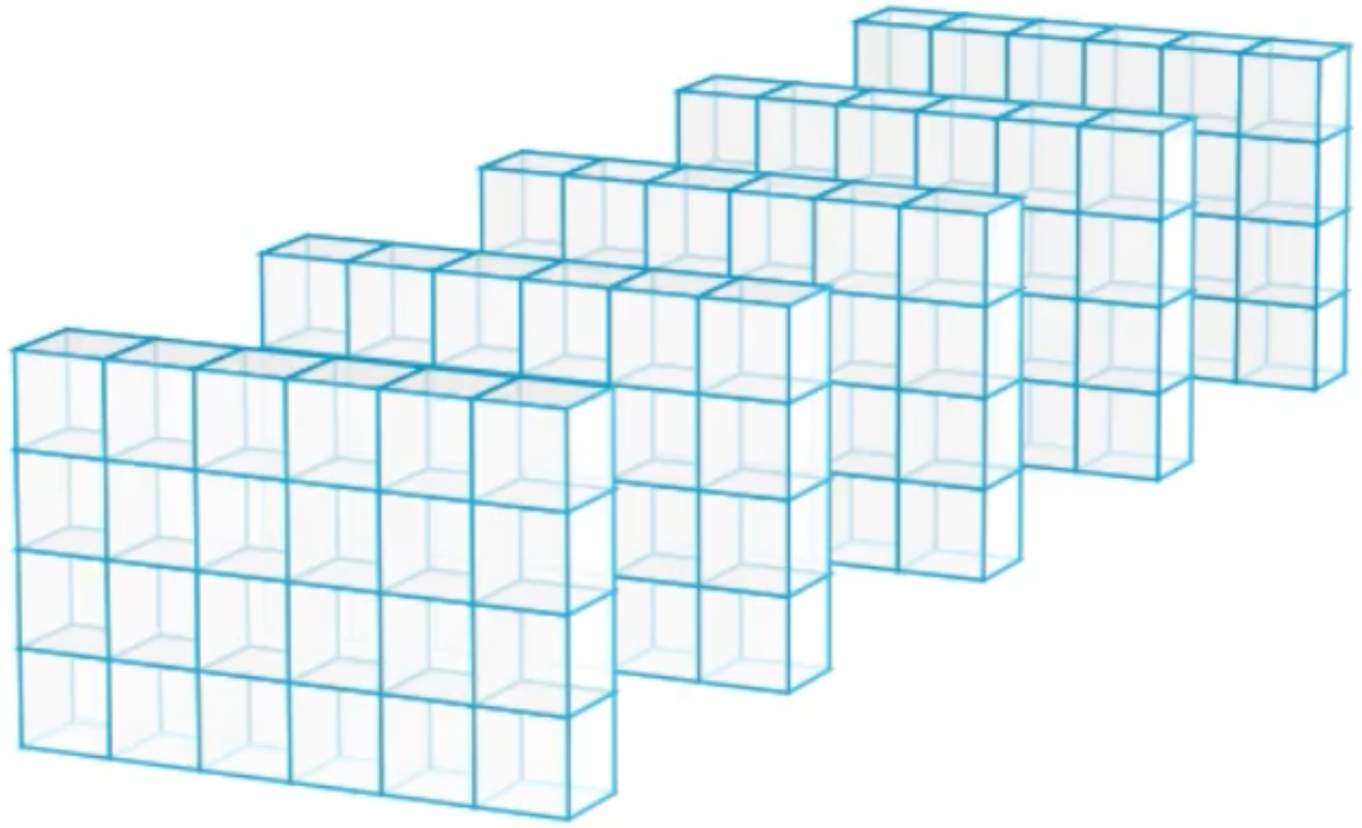
$$\mathcal{C} \subseteq \mathbb{F}_q^{m \times n \times k}$$



From matrix codes to 3-tensors

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

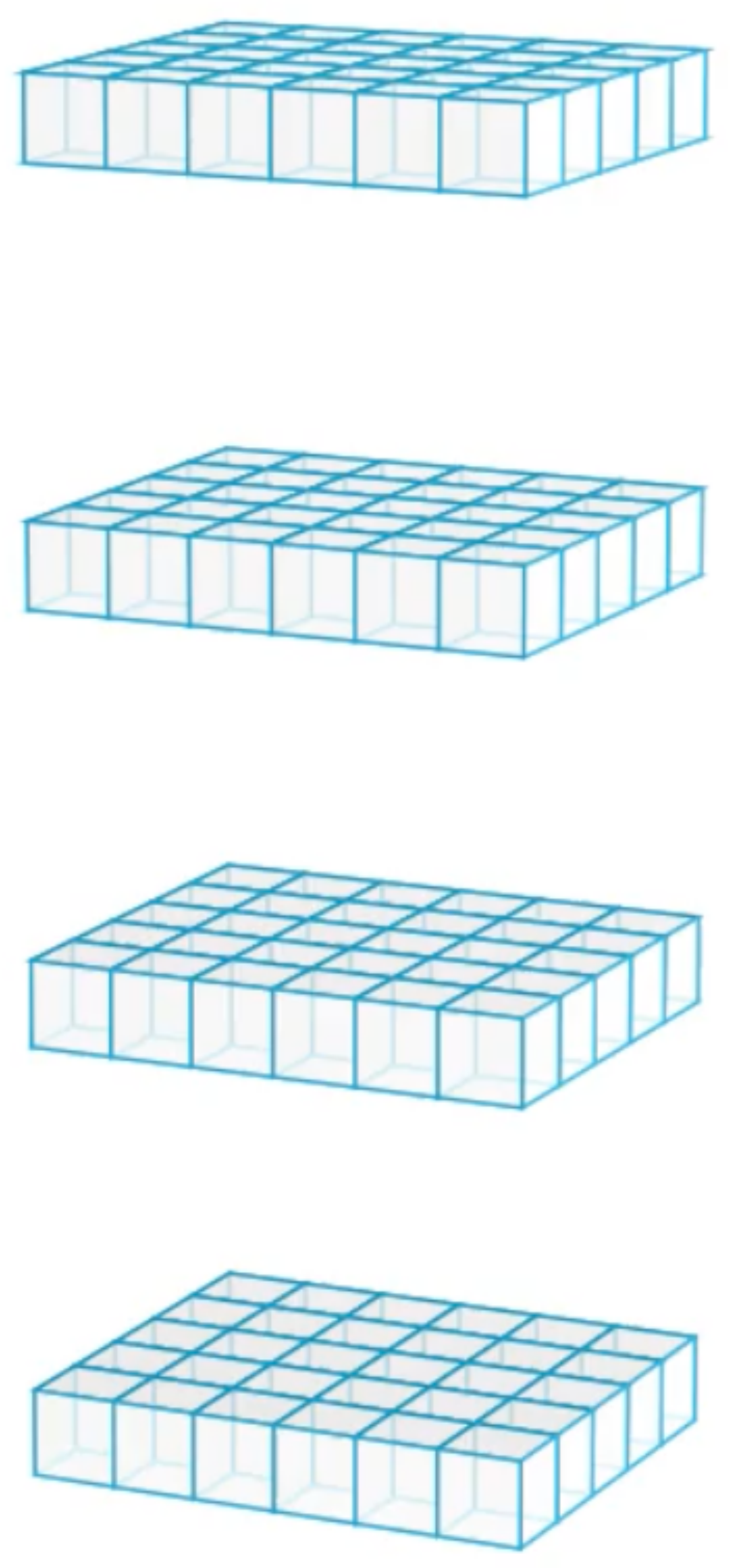
- a k-dimensional code in $\mathbb{F}_q^{m\times n}$
- an m-dimensional code in $\mathbb{F}_q^{n\times k}$
- an n-dimensional code in $\mathbb{F}_q^{m\times k}$



From matrix codes to 3-tensors

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

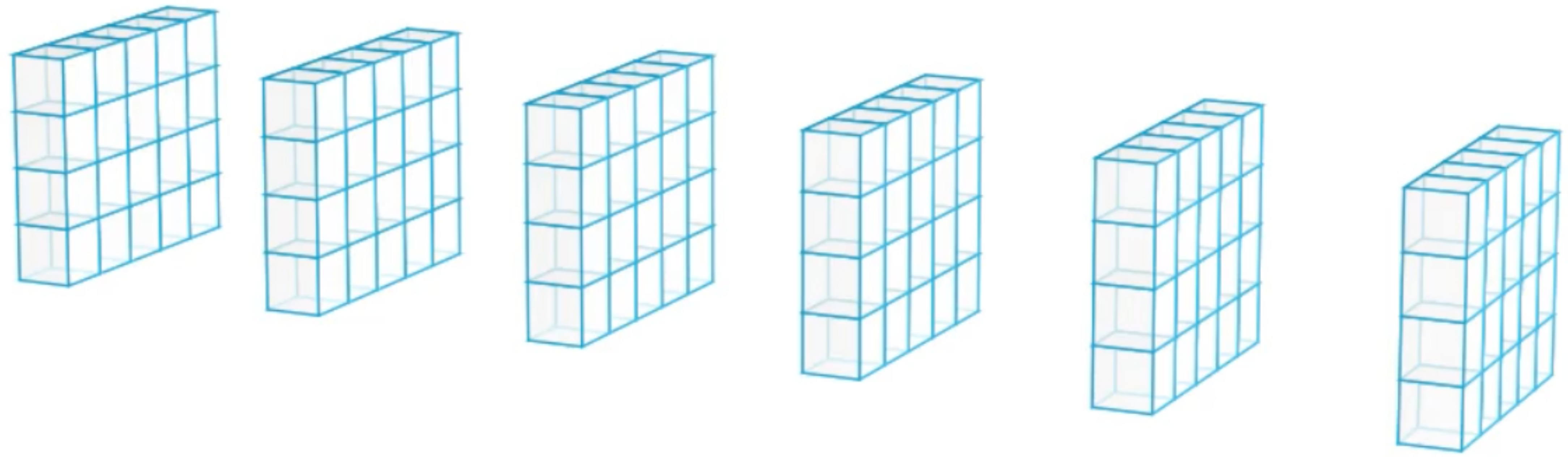
- a k-dimensional code in $\mathbb{F}_q^{m\times n}$
- an m-dimensional code in $\mathbb{F}_q^{n\times k}$
- an n-dimensional code in $\mathbb{F}_q^{m\times k}$



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 $\mathbb{F}_q^{m\times k}$



Spoilers ahead !

Direct 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

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

Direct 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

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

Alternatively, this gives a better modelling:

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

Exercise 4: Matrix Code Equivalence with hints

Collision

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

$$\mathbf{D} = \mathbf{ACB}$$

Collision

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

$$\mathbf{D} = \mathbf{A}\mathbf{C}\mathbf{B}$$

Recall how we can represent codewords with their coordinate vectors

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

$$(q = 13, \quad m = 4, \quad n = 6, \quad k = 5)$$

Spoilers ahead !

Collision

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

$$\mathbf{D} = \mathbf{A}\mathbf{C}\mathbf{B}$$

We can then infer linear constraints from

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