Multivariate cryptography

Monika Trimoska

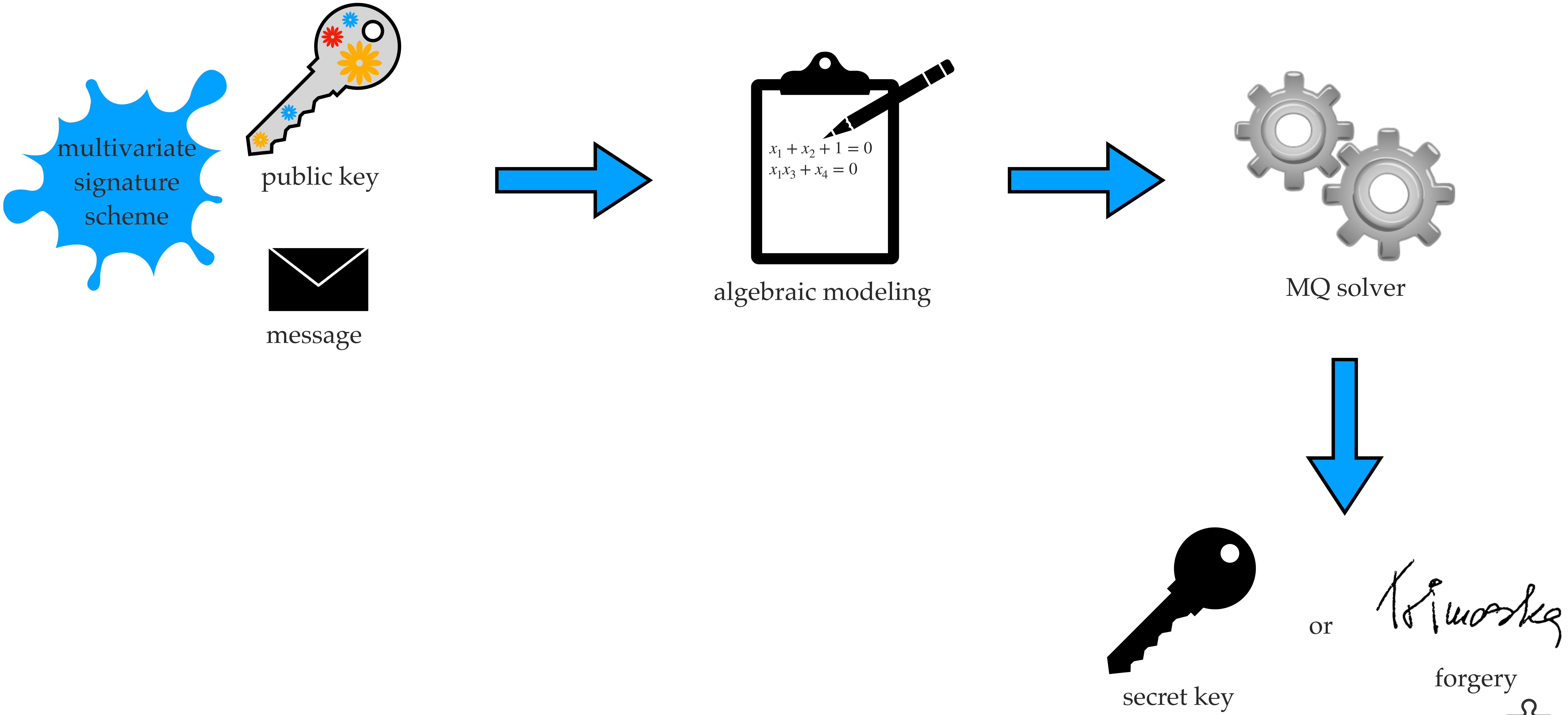
PQC Spring School
Porto, March 15 2024

TU/e

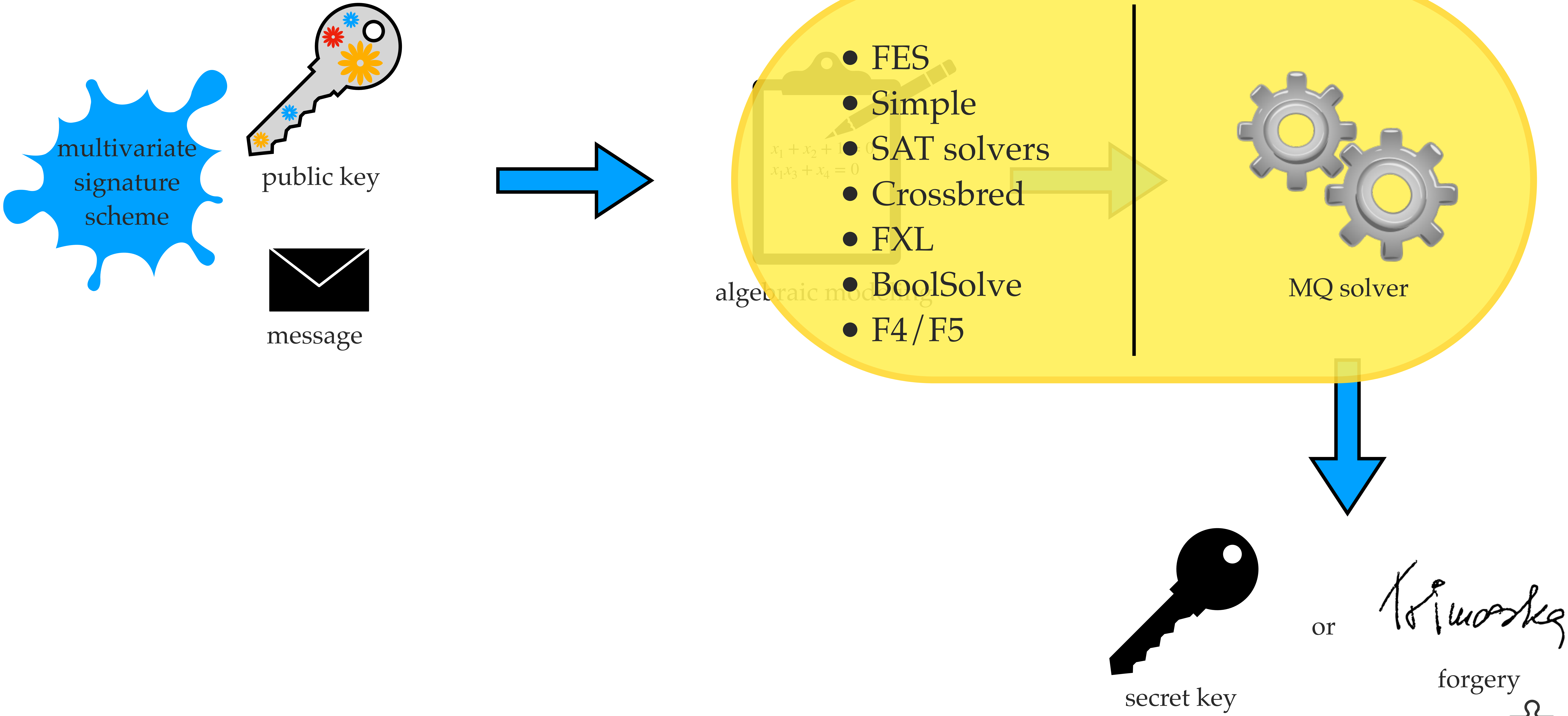
Algebraic cryptanalysis

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.

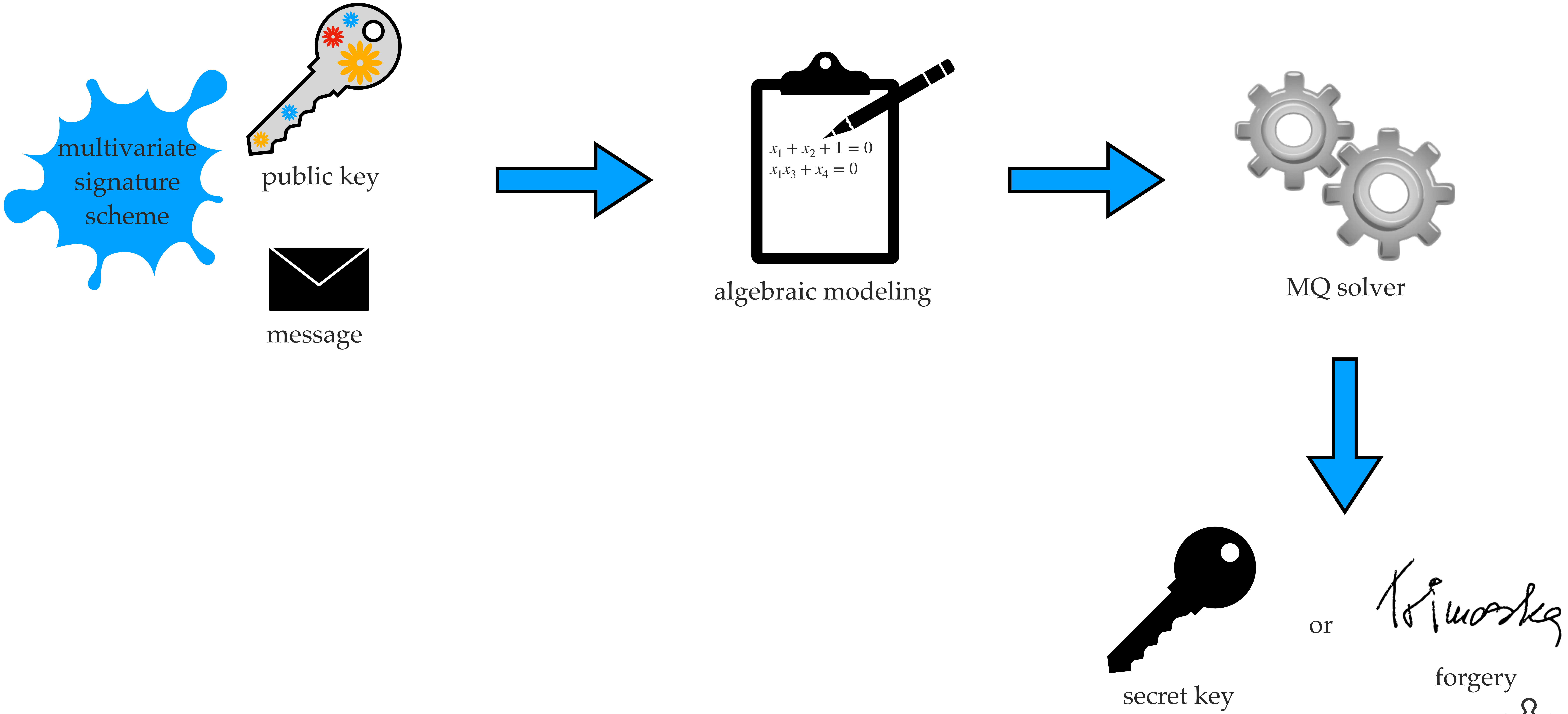
Algebraic cryptanalysis



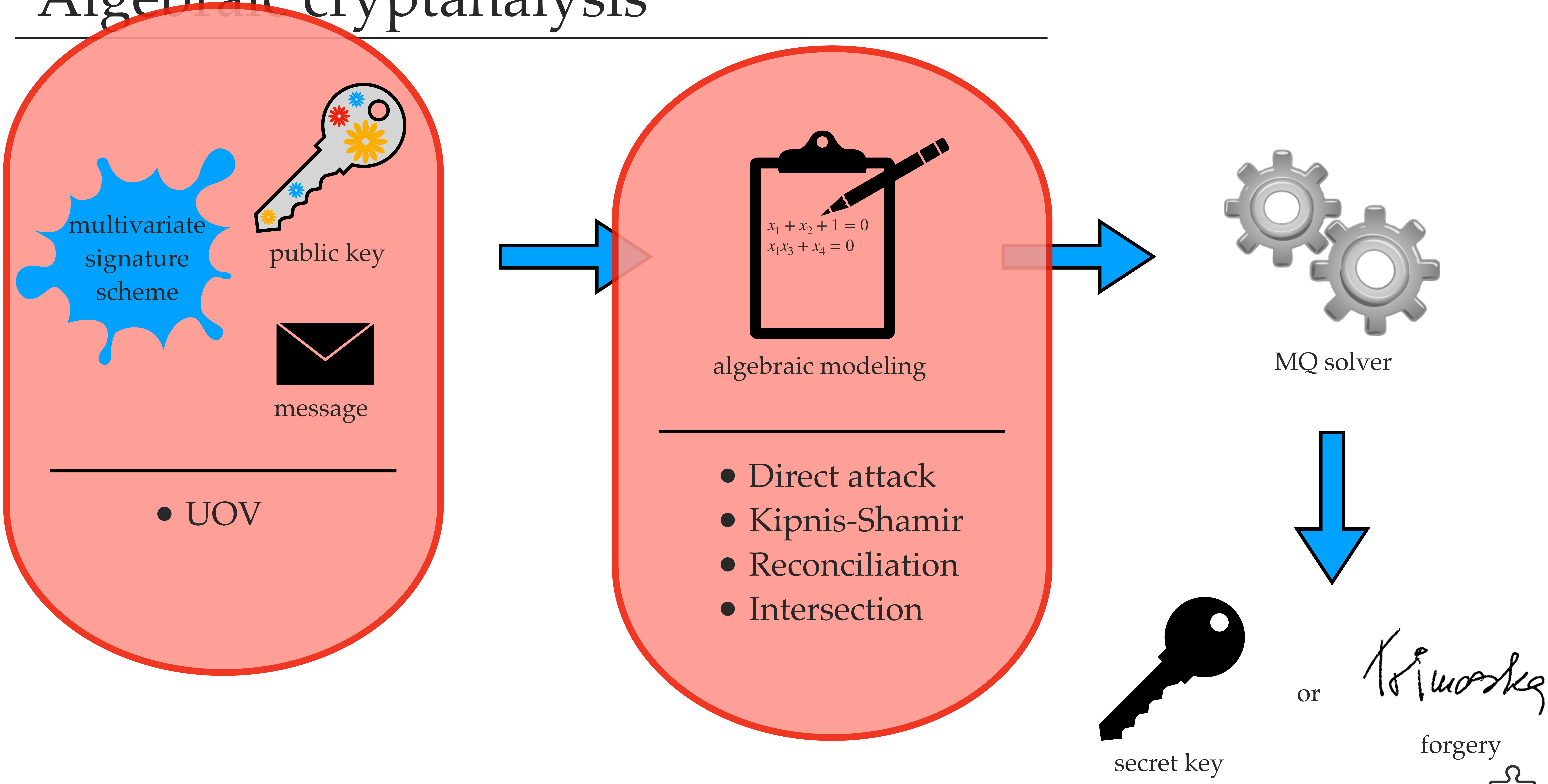
Algebraic cryptanalysis



Algebraic cryptanalysis



Algebraic cryptanalysis



The MQ problem

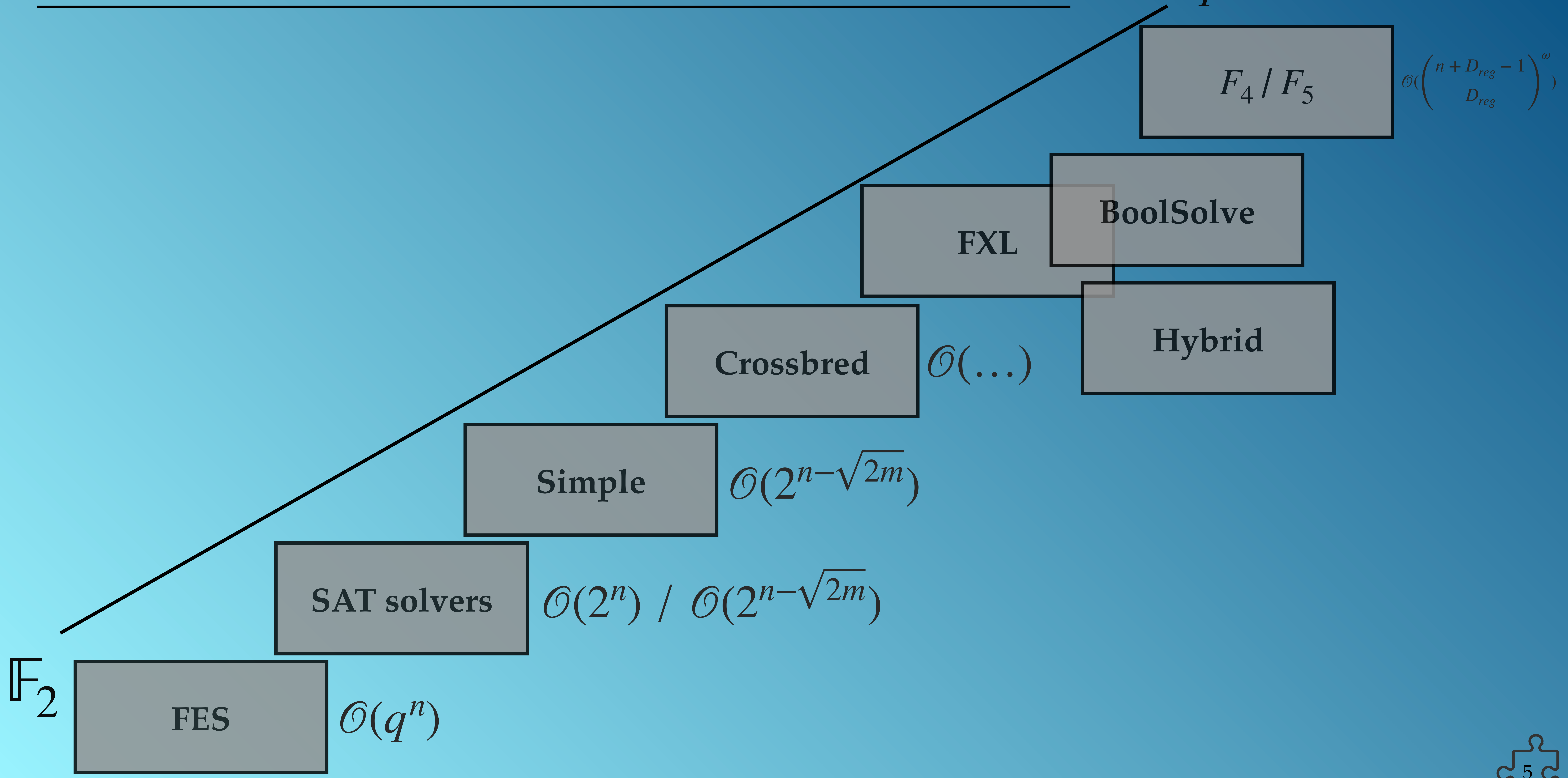
The MQ problem

Given m multivariate quadratic polynomials $f_1, \ldots, f_m$ of n variables over a finite field $\mathbb{F}_q$, find a tuple $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{F}_q^n$, such that $f_1(\mathbf{x}) = \ldots = f_m(\mathbf{x}) = 0$.

Example.

$$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$$
$$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$$
$$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$$
$$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$$
$$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$$
$$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$$

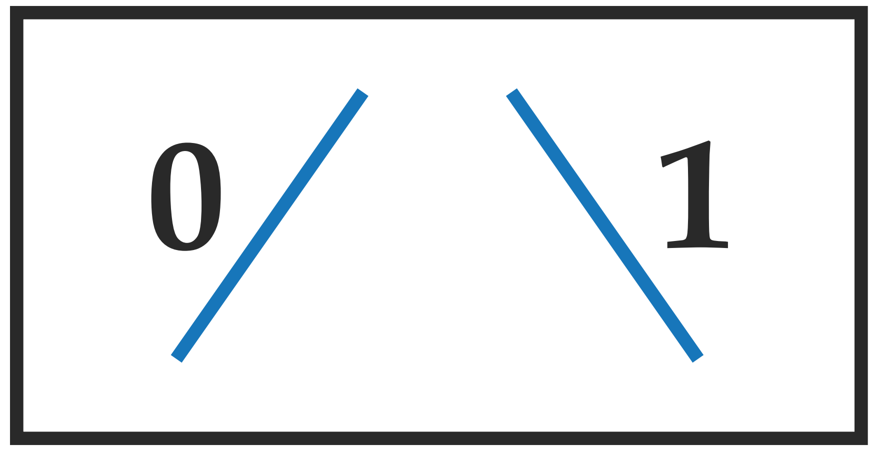
Overview of solvers

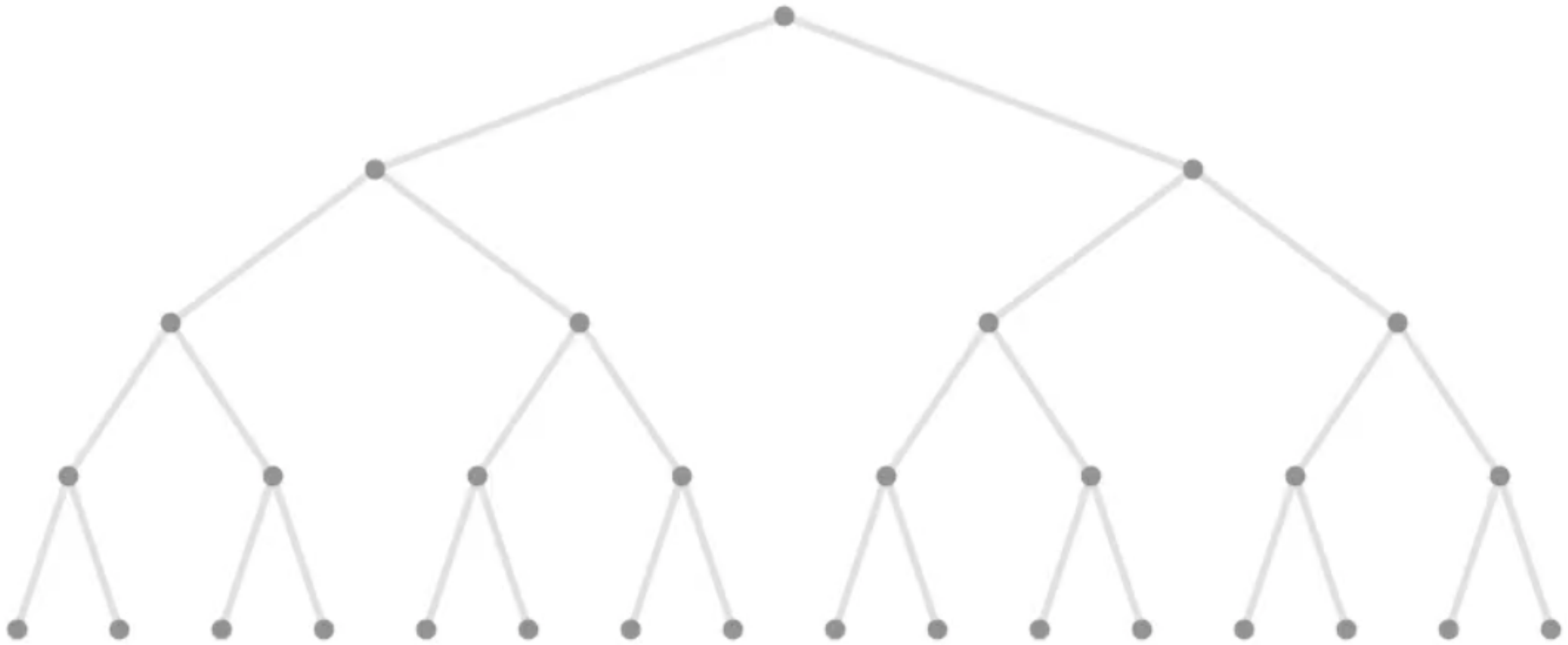


(Fast) Exhaustive Search

[Bouillaguet, Chen, Cheng, Chou, Niederhagen, Shamir, Yang, 2010]

Exhaustive Search





Binary search tree

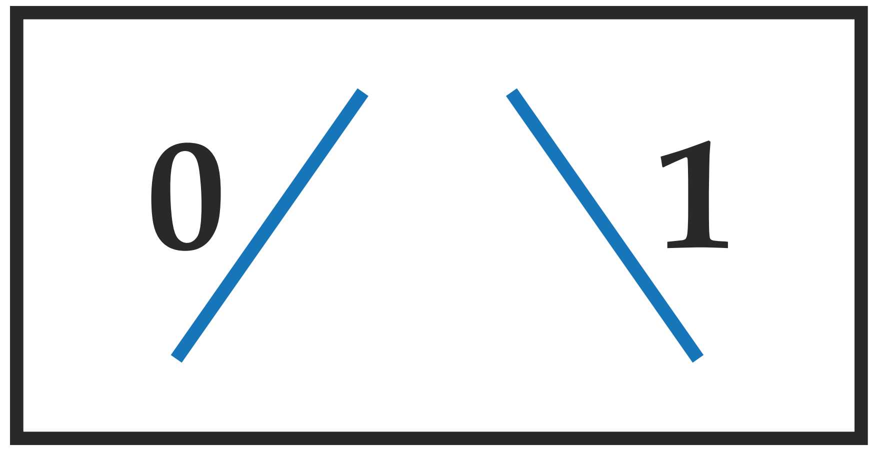
$$x_1 \cdot x_2 + x_1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$

$$x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_2 + 1 = 0$$

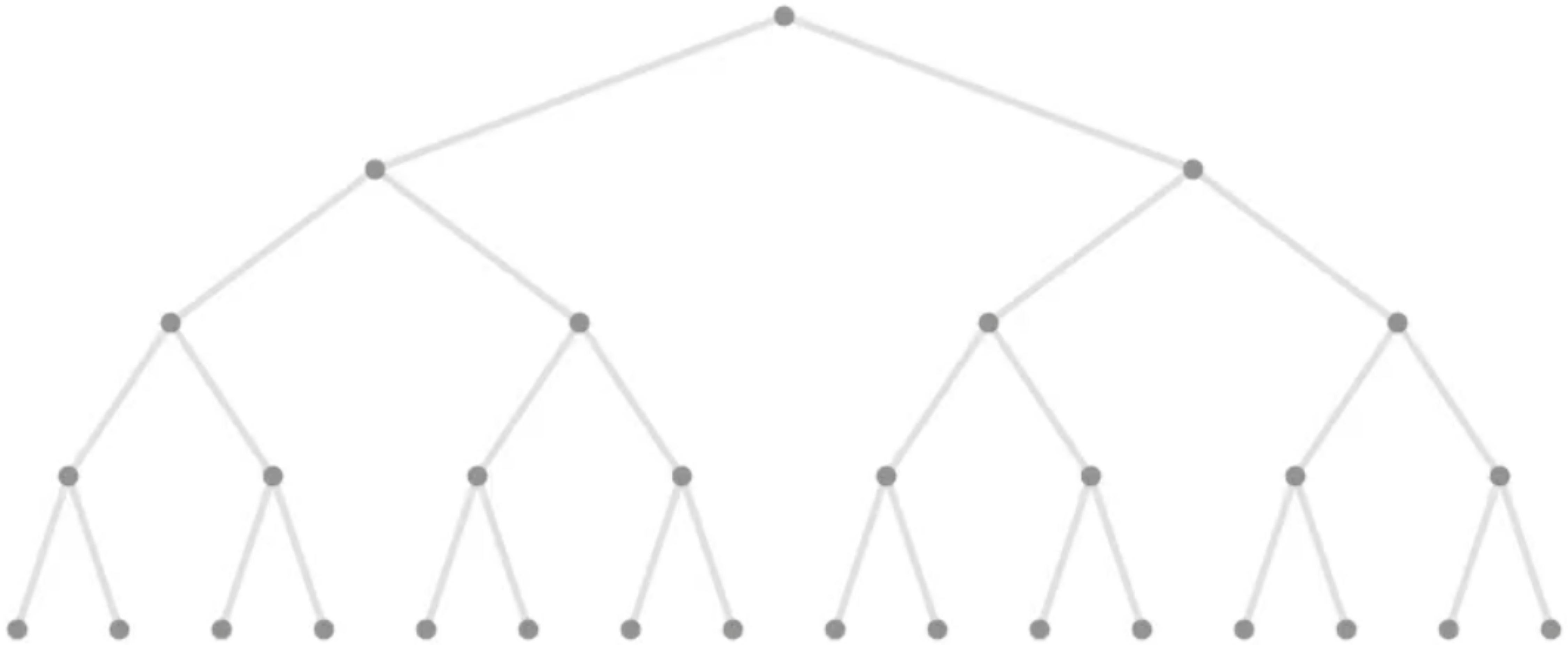
$$x_1 \cdot x_2 + x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_4 = 0$$

$$x_1 \cdot x_4 + x_2 \cdot x_3 + x_2 + x_3 + x_4 = 0$$

Exhaustive Search



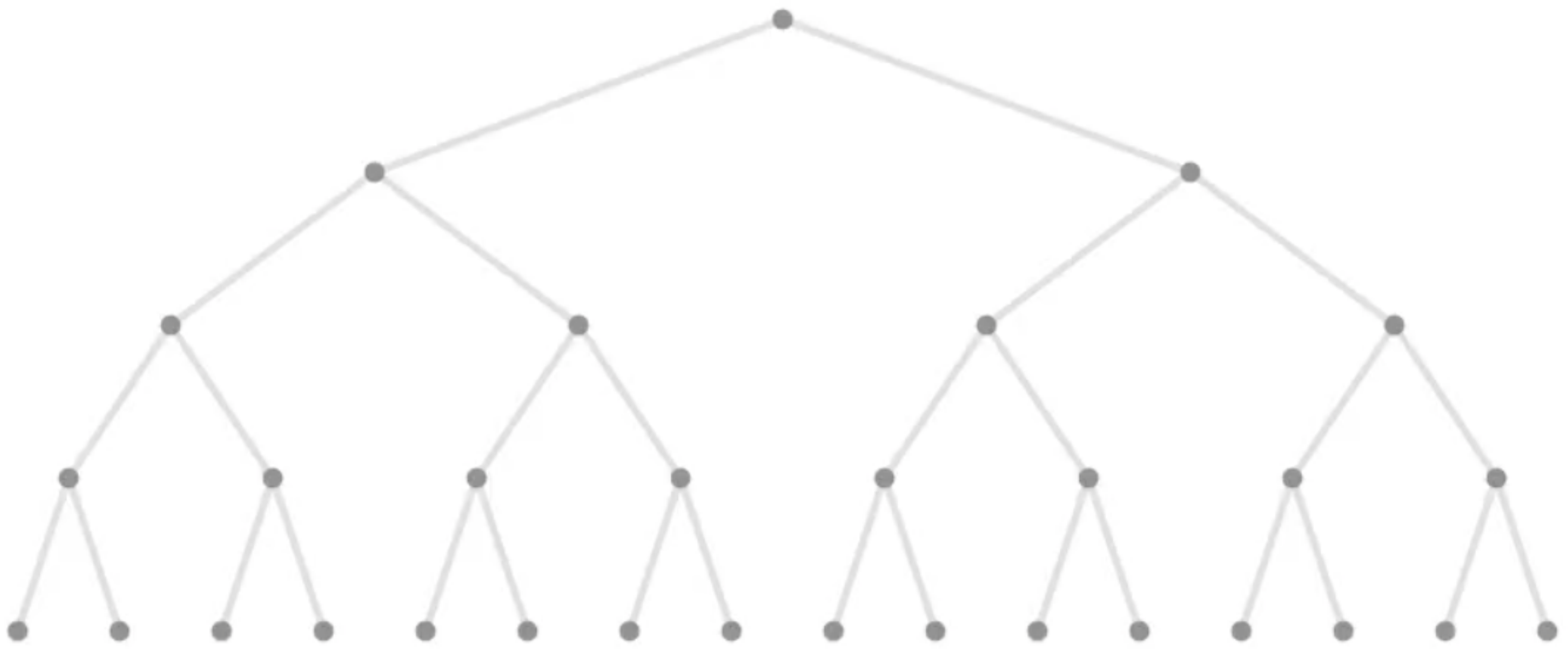
Worst-case complexity: $\mathcal{O}(2^n)$



Binary search tree

$$x_1 \cdot x_2 + x_1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$
$$x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_2 + 1 = 0$$
$$x_1 \cdot x_2 + x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_4 = 0$$
$$x_1 \cdot x_4 + x_2 \cdot x_3 + x_2 + x_3 + x_4 = 0$$

Exhaustive Search



Binary search tree

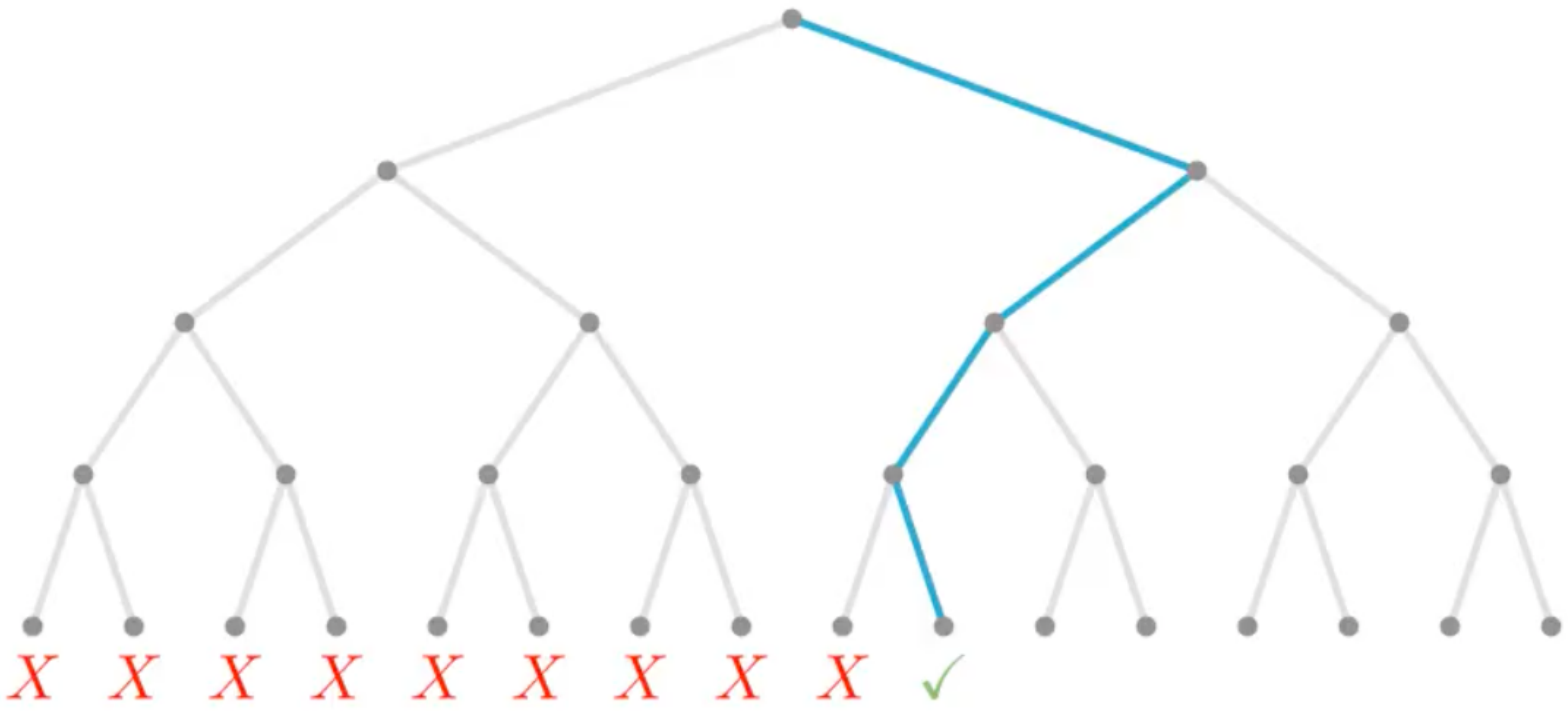
$$x_1 \cdot x_2 + x_1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$

$$x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_2 + 1 = 0$$

$$x_1 \cdot x_2 + x_2 \cdot x_3 + x_2 \cdot x_4 + x_1 + x_4 = 0$$

$$x_1 \cdot x_4 + x_2 \cdot x_3 + x_2 + x_3 + x_4 = 0$$

Exhaustive Search



Binary search tree

$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

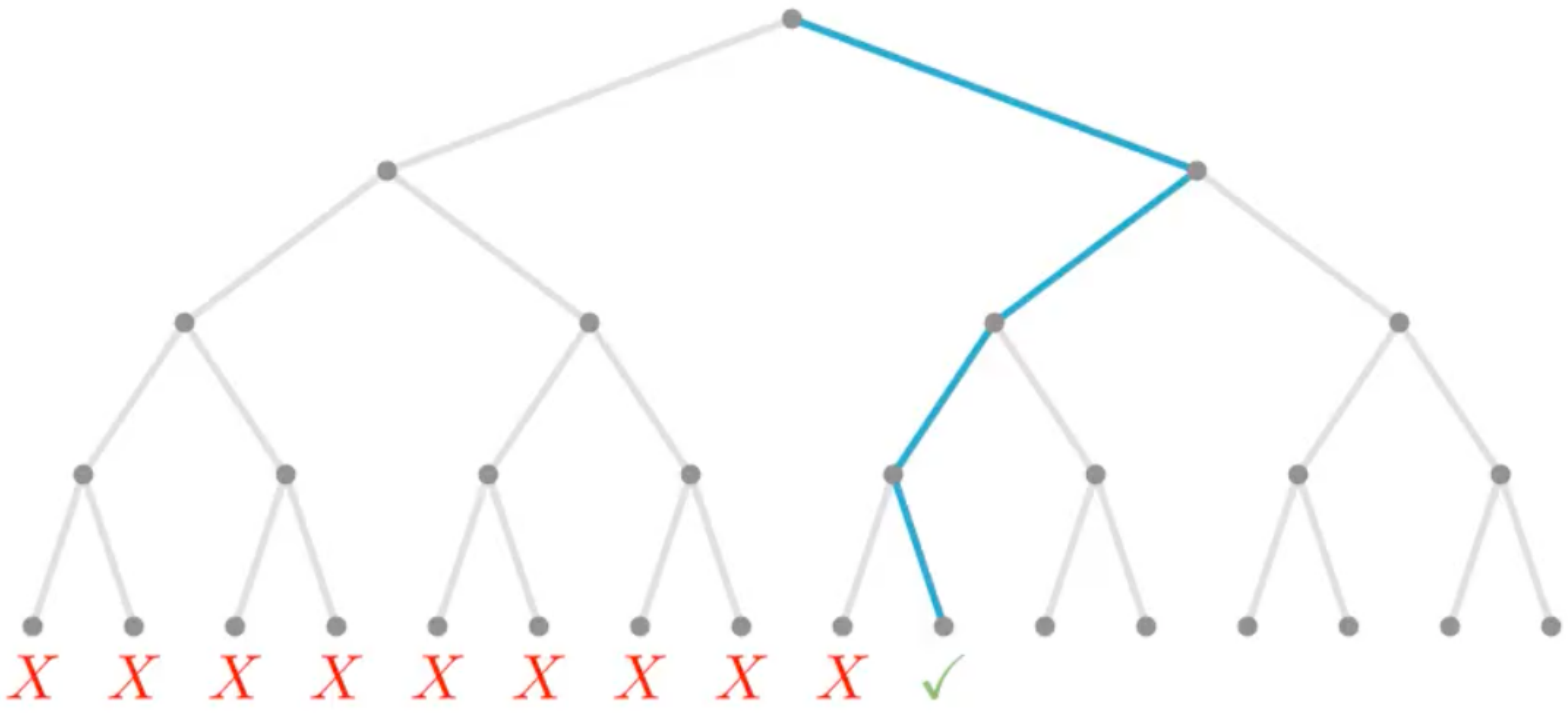
$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

Exhaustive Search

Worst-case complexity: $\mathcal{O}(2^n)$



Binary search tree

$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

Fast Exhaustive Search

* The libFES solver

Gray code

- An ordering of the binary system where two successive values differ in only one bit.

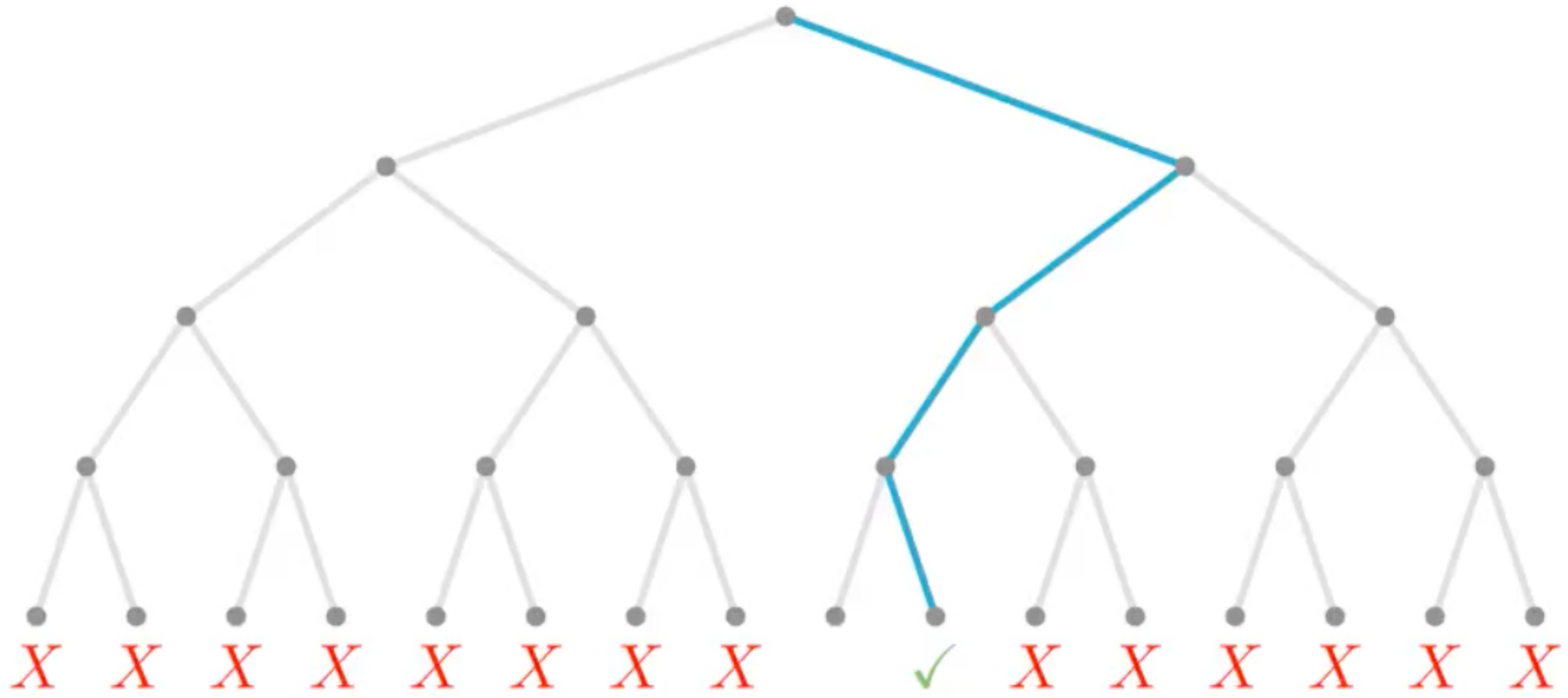
Example. $n = 4$

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000

Fast Exhaustive Search

Gray code

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000



$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

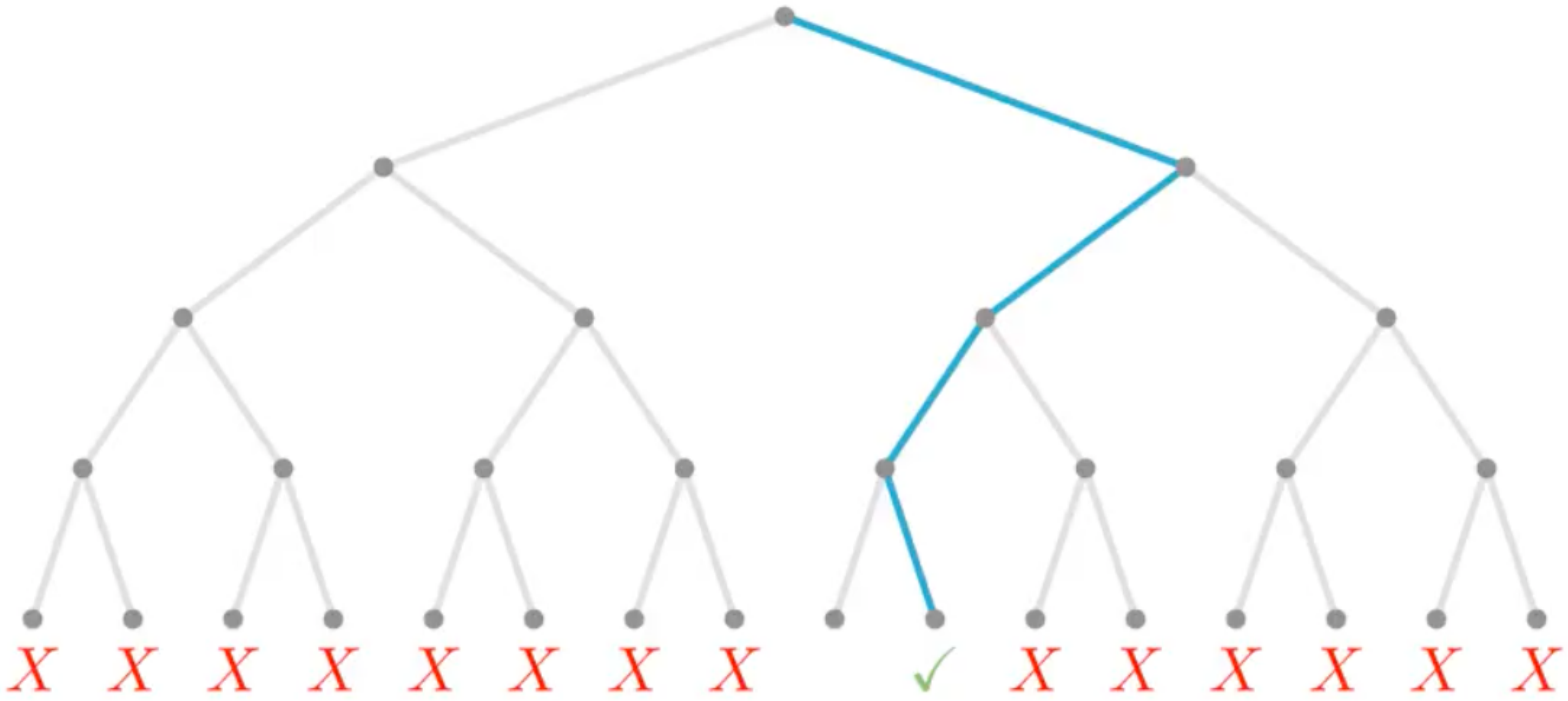
Fast Exhaustive Search

Worst-case complexity: $\mathcal{O}(2^n)$

! But, it differs from the depth-first traversal in the polynomial factors

Gray code

0000	1100
0001	1101
0011	1111
0010	1110
0110	1010
0111	1011
0101	1001
0100	1000



$$1 \cdot 0 + 1 \cdot 0 + 0 \cdot 1 + 0 = 0$$

$$0 \cdot 0 + 0 \cdot 1 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot 0 + 0 \cdot 1 + 1 + 1 = 0$$

$$1 \cdot 1 + 0 \cdot 0 + 0 + 0 + 1 = 0$$

Macaulay matrix

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

↳ Linearisation: for each nonlinear monomial, replace all of its occurrences by a new variable.

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

→ Linearisation: for each nonlinear monomial, replace all of its occurrences by a new variable.

Example.

$$
\begin{aligned}
f_1 &: x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0\\
f_2 &: x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0\\
f_3 &: x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0\\
f_4 &: x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0\\
f_5 &: x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0\\
f_6 &: x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
$$

→

$$
\begin{aligned}
f_1 &: y_2 + y_5 + x_1 + x_3 + x_4 = 0\\
f_2 &: y_4 + y_3 + y_6 + x_1 + x_2 + x_4 = 0\\
f_3 &: y_5 + y_6 + x_1 + x_3 + 1 = 0\\
f_4 &: y_1 + y_2 + y_4 + x_3 + x_4 + 1 = 0\\
f_5 &: y_1 + y_4 + y_3 + x_3 = 0\\
f_6 &: y_2 + y_3 + y_6 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
$$

Linearisation

Linear systems are easy to solve, nonlinear systems are hard.

Linearisation: for each nonlinear monomial, replace all of its occurrences by a new variable.

Example.

$$
\begin{aligned}
f_1 &: x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0\\
f_2 &: x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0\\
f_3 &: x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0\\
f_4 &: x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0\\
f_5 &: x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0\\
f_6 &: x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
\quad\longrightarrow\quad
\begin{aligned}
f_1 &: y_2 + y_5 + x_1 + x_3 + x_4 = 0\\
f_2 &: y_4 + y_3 + y_6 + x_1 + x_2 + x_4 = 0\\
f_3 &: y_5 + y_6 + x_1 + x_3 + 1 = 0\\
f_4 &: y_1 + y_2 + y_4 + x_3 + x_4 + 1 = 0\\
f_5 &: y_1 + y_4 + y_3 + x_3 = 0\\
f_6 &: y_2 + y_3 + y_6 + x_1 + x_2 + x_3 + x_4 = 0
\end{aligned}
$$

Linearisation

Linearisation adds solutions: a *random* quadratic system of m equations in n variables, when $n = m$, is expected to have one solution (probability is $\sim \frac{1}{q}$ for systems over $\mathbb{F}_q$). The corresponding linearised system has a solution space of dimension $\binom{n+1}{2} - m$.

$\binom{n}{2}$ quadratic plus n linear monomials

Linearisation

Linearisation adds solutions: a *random* quadratic system of m equations in n variables, when $n = m$, is expected to have one solution (probability is $\sim \frac{1}{q}$ for systems over $\mathbb{F}_q$). The corresponding linearised system has a solution space of dimension $\binom{n+1}{2} - m$.

$\binom{n}{2}$ quadratic plus n linear monomials

Loss of information: e.g. assignment $x_1 = 1; x_2 = 0; y_1 = 1;$ is part of a valid solution to the linearised system, but $x_1 x_2 \neq y_1$.

Macaulay matrix

Monomials →

Equations ↓

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1											
f_2											
f_3											
f_4											
f_5											
f_6											

$$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$$
$$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$$
$$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$$
$$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$$
$$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$$
$$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$$

Macaulay matrix

Monomials →

Equations ↓

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$

$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$

$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$

$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$

$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$

$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

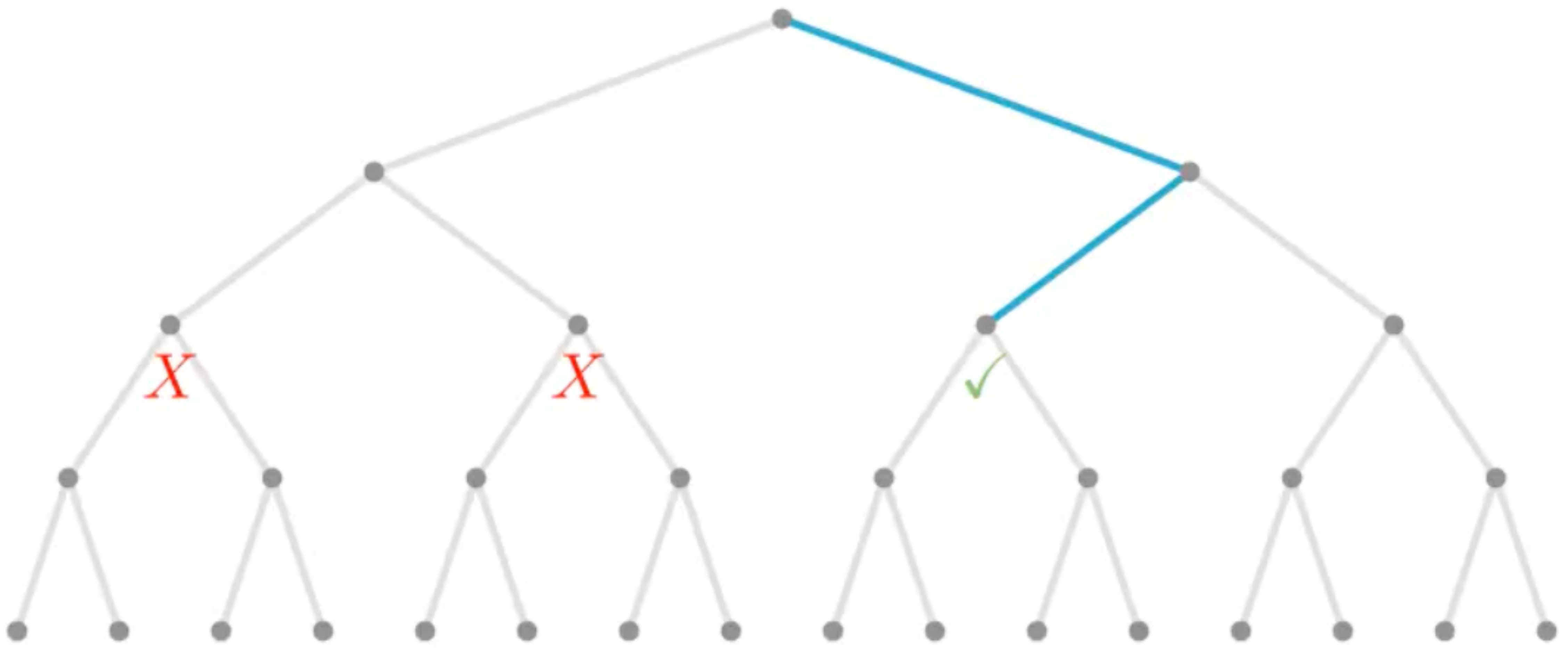
SAT solvers

CryptoMiniSat [Soos, Nohl, Castelluccia, 2009], WDSat [T., Dequen, Ionica, 2020]

Simple algorithm

[Bouillaguet, Delaplace, T., 2021]

Partial assignment and conflicts



$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$

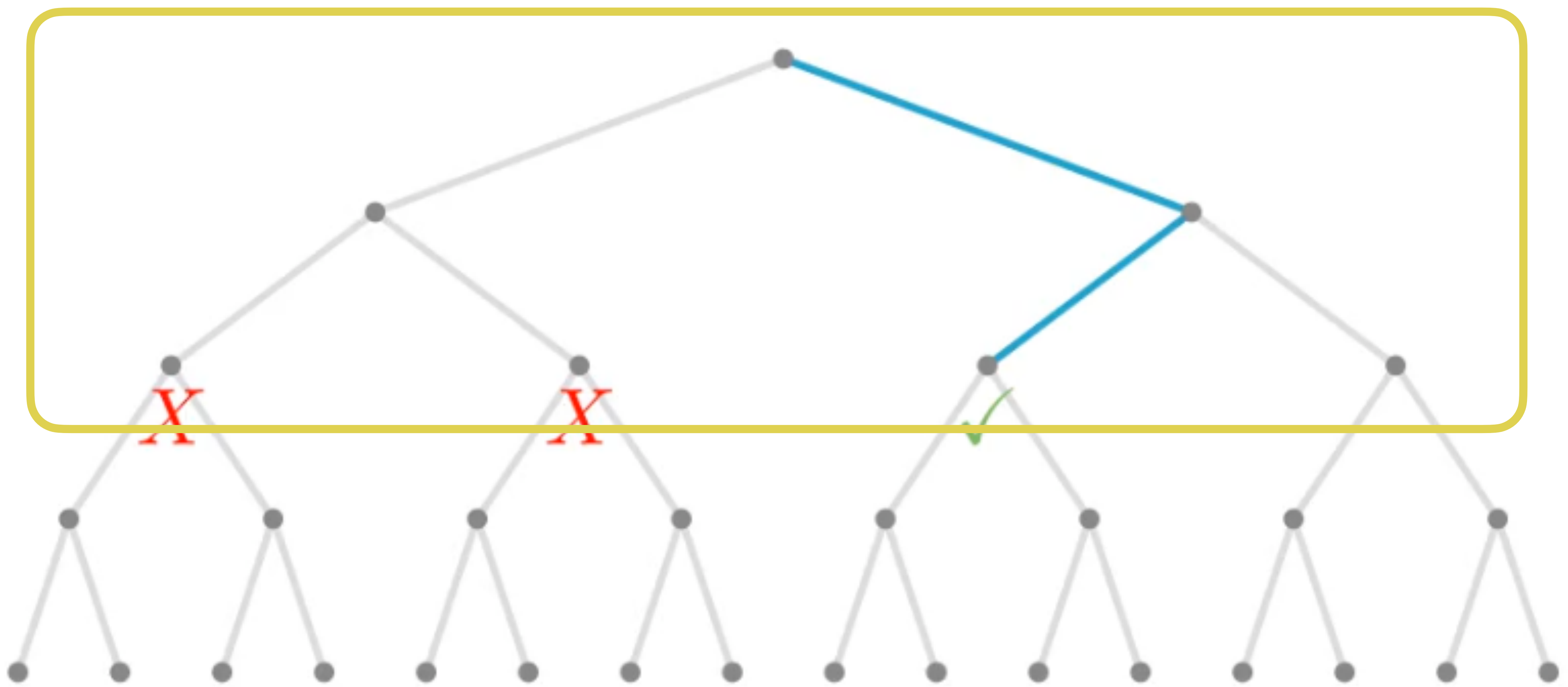
$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$

$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$

$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

Simple algorithm

- Partial assignment
  - Gaussian elimination

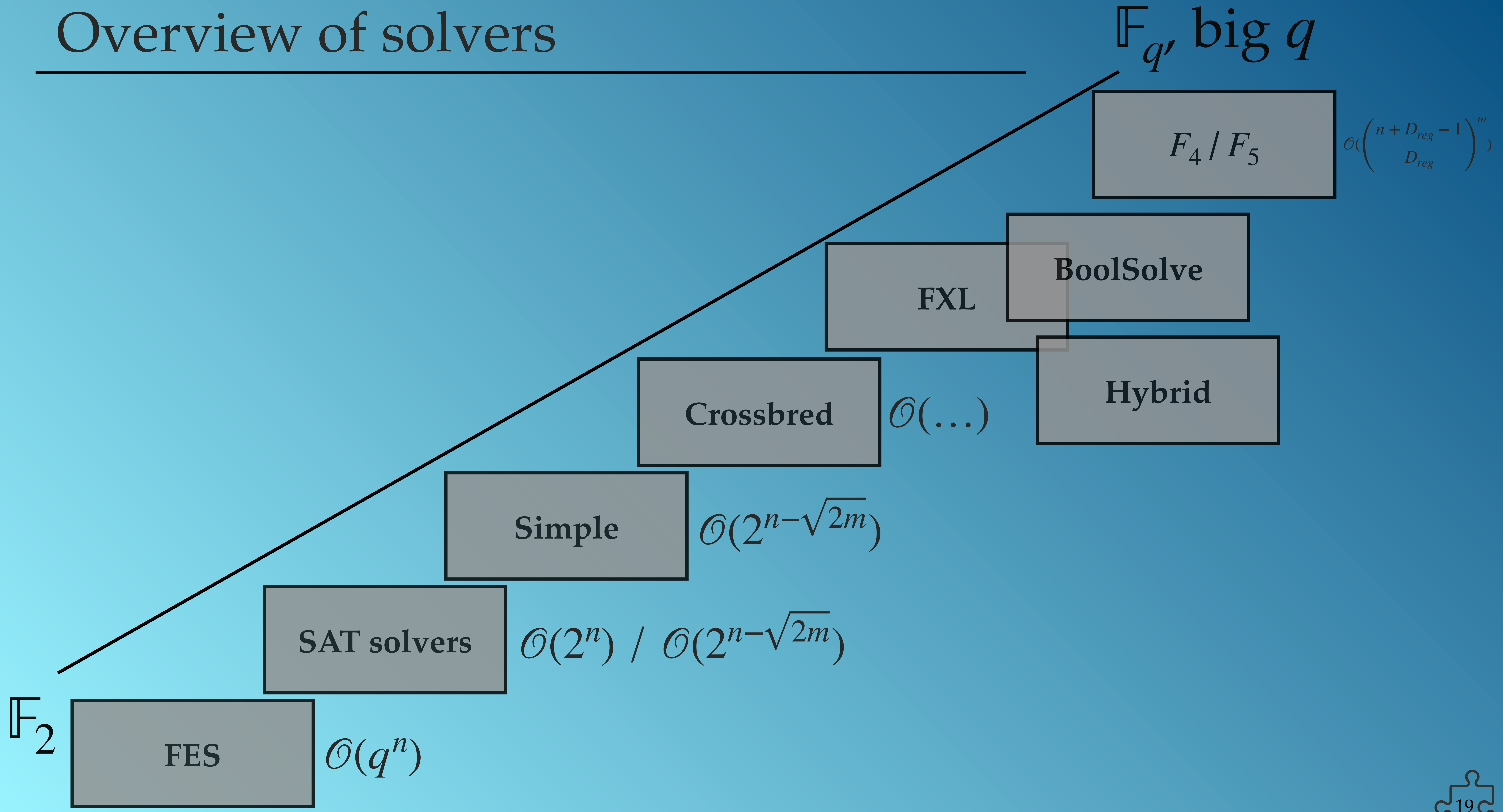


$$1 \cdot 0 + 1 \cdot x_3 + x_3 \cdot x_4 + x_3 = 0$$
$$0 \cdot x_3 + 0 \cdot x_4 + 1 + 0 + 1 = 0$$
$$1 \cdot 0 + 0 \cdot x_3 + 0 \cdot x_4 + 1 + x_4 = 0$$
$$1 \cdot x_4 + 0 \cdot x_3 + 0 + x_3 + x_4 = 0$$

Simple algorithm

Guess sufficiently many variables so that the remaining polynomial system can be solved by linearization.

Overview of solvers



Gröbner basis algorithms

[Buchberger, 1965]

[Lazard, 1983]

F_4/F_5 [Faugère, 1999/2002]

(XL [Courtois, Klimov, Patarin, Shamir, 2000])

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1
f_1	0	1	0	1	0	1	0	0	1	1	0
f_2	0	0	1	1	1	0	1	1	0	1	0
f_3	0	0	0	1	0	1	0	1	1	0	1
f_4	1	1	0	1	1	0	0	0	1	1	1
f_5	1	0	1	1	1	0	0	0	1	0	0
f_6	0	1	1	1	0	0	1	1	1	1	0

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

$D = 3$

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$
f_1	0	1	0	1	0	1	0	0	1	1	0				
f_2	0	0	1	1	1	0	1	1	0	1	0				
f_3	0	0	0	1	0	1	0	1	1	0	1				
f_4	1	1	0	1	1	0	0	0	1	1	1				
f_5	1	0	1	1	1	0	0	0	1	0	0				
f_6	0	1	1	1	0	0	1	1	1	1	0				
x_1f_1															
x_2f_1															
$\dots$															

Gröbner basis algorithms (intuition)

*We are essentially describing the XL algorithm.

$D = 4$

$f_1 : x_1x_3 + x_2x_4 + x_1 + x_3 + x_4 = 0$
$f_2 : x_2x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_4 = 0$
$f_3 : x_2x_4 + x_3x_4 + x_1 + x_3 + 1 = 0$
$f_4 : x_1x_2 + x_1x_3 + x_2x_3 + x_3 + x_4 + 1 = 0$
$f_5 : x_1x_2 + x_2x_3 + x_1x_4 + x_3 = 0$
$f_6 : x_1x_3 + x_1x_4 + x_3x_4 + x_1 + x_2 + x_3 + x_4 = 0$

	x_1x_2	x_1x_3	x_1x_4	x_1	x_2x_3	x_2x_4	x_2	x_3x_4	x_3	x_4	1	$x_1x_2x_3$	$x_1x_2x_4$	$x_1x_3x_4$	$x_2x_3x_4$	$x_1x_2x_3x_4$
f_1	0	1	0	1	0	1	0	0	1	1	0					
f_2	0	0	1	1	1	0	1	1	0	1	0					
f_3	0	0	0	1	0	1	0	1	1	0	1					
f_4	1	1	0	1	1	0	0	0	1	1	1					
f_5	1	0	1	1	1	0	0	0	1	0	0					
f_6	0	1	1	1	0	0	1	1	1	1	0					
x_1f_1																
x_2f_1																
…																
$x_1x_2f_1$																
$x_1x_3f_1$																

XL/Gröbner basis algorithms: complexity

XL / Gröbner basis algorithms: complexity

$$\mathcal{O}\left(mD_{reg}\binom{n+D_{reg}-1}{D_{reg}}^{\omega}\right)$$

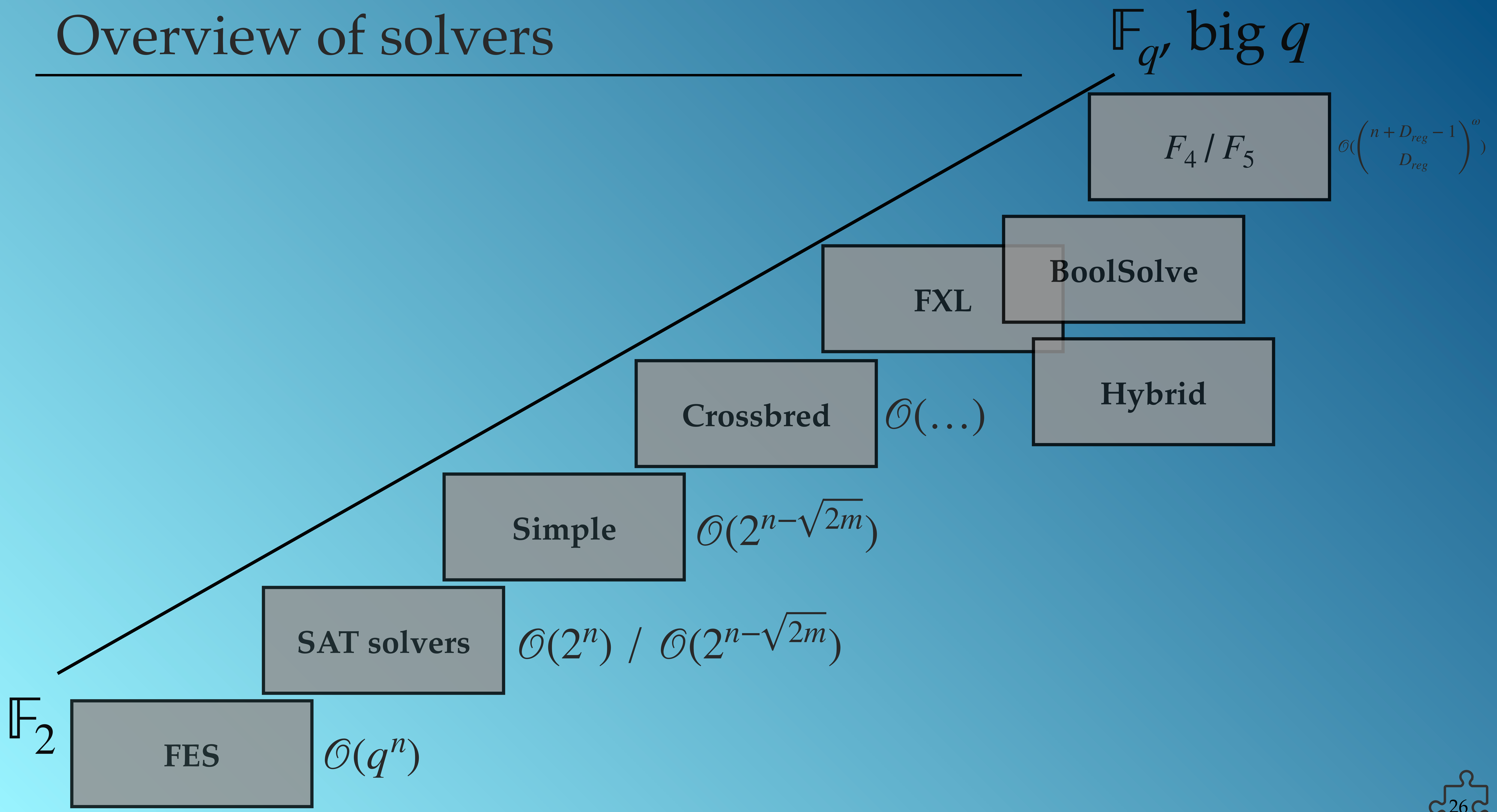
XL/Gröbner basis algorithms: complexity

$$\mathcal{O}\left(mD_{reg} \binom{n + D_{reg} - 1}{D_{reg}}^{\omega} \right)$$

D_{reg}: degree of regularity

↳ the power of the first non-positive coefficient in the expansion of $$\frac{(1-t^2)^m}{(1-t)^n}$$

Overview of solvers



Algebraic cryptanalysis: try it yourself !

Example.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

Algebraic cryptanalysis: try it yourself !

Example.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

⟶ Demo

Algebraic cryptanalysis: try it yourself !

Example.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

→ Demo

→ In the assignment:

- Write down the equations;
- Find a better modelisation for this problem;

Modelisation

A motivating example: a better idea for modelisation.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

Modelisation

A motivating example: a better idea for modelisation.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

→ Demo

Modelisation

A motivating example: a better idea for modelisation.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

→ Demo

→ Results in a linear system with the same number of variables and equations.

Modelisation

A motivating example: a better idea for modelisation.

Given matrices $\mathbf{C}_1, \mathbf{C}_2, \mathbf{D}_1, \mathbf{D}_2 \in \mathscr{M}_{n,n}(\mathbb{F}_q)$ (the space of matrices over $\mathbb{F}_q$ of size $n \times n$), find $\mathbf{A}, \mathbf{B} \in \mathrm{GL}_n(\mathbb{F}_q)$ (the space of invertible matrices over $\mathbb{F}_q$ of size $n \times n$), such that

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

- Demo
- 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 $\mathbf{A}^{-1}$.

Multivariate digital signature schemes

Multivariate signatures

Fiat-Shamir construction

trapdoor construction

Examples.

MQDSS

SOFIA

Examples.

HFEv-

UOV

The MQ problem (recall)

A quadratic system of m equations in n variables over a finite field $\mathbb{F}_q$:

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

The MQ problem

Given m multivariate quadratic polynomials $f^{(1)}, \ldots, f^{(m)}$ of n variables over a finite field $\mathbb{F}_q$, find a tuple $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{F}_q^n$, such that $f^{(1)}(\mathbf{x}) = \ldots = f^{(m)}(\mathbf{x}) = 0$.

The MQ problem (recall)

A quadratic system of m equations in n variables over a finite field $\mathbb{F}_q$:

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

The MQ problem

Given m multivariate quadratic polynomials $f^{(1)}, \ldots, f^{(m)}$ of n variables over a finite field $\mathbb{F}_q$, find a tuple $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{F}_q^n$, such that $f^{(1)}(\mathbf{x}) = \ldots = f^{(m)}(\mathbf{x}) = 0$.

→ Hard in general (should be hard for randomly generated instances).

The MQ problem (recall)

A quadratic system of m equations in n variables over a finite field $\mathbb{F}_q$:

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

The MQ problem

Given m multivariate quadratic polynomials $f^{(1)}, \ldots, f^{(m)}$ of n variables over a finite field $\mathbb{F}_q$, find a tuple $\mathbf{x} = (x_1, \ldots, x_n)$ in $\mathbb{F}_q^n$, such that $f^{(1)}(\mathbf{x}) = \ldots = f^{(m)}(\mathbf{x}) = 0$.

→ Hard in general (should be hard for randomly generated instances).

→ Can become easy if we have some structure (a trapdoor).

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 $\mathbf{T} \in \mathrm{GL}_m(\mathbb{F}_q)$

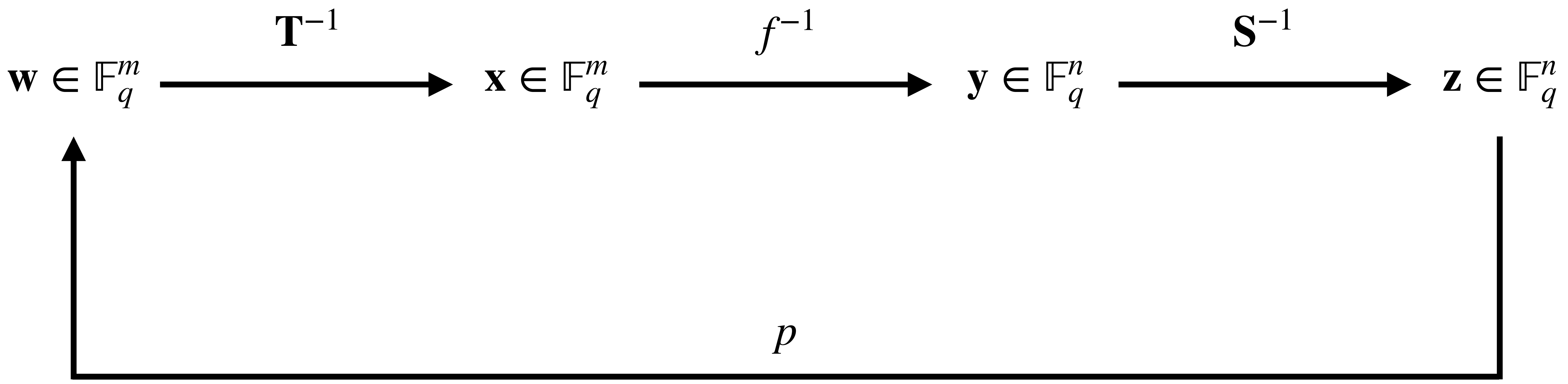
- Public map:
  $p = \mathbf{T} \circ f \circ \mathbf{S}$

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 $\mathbf{x}$.

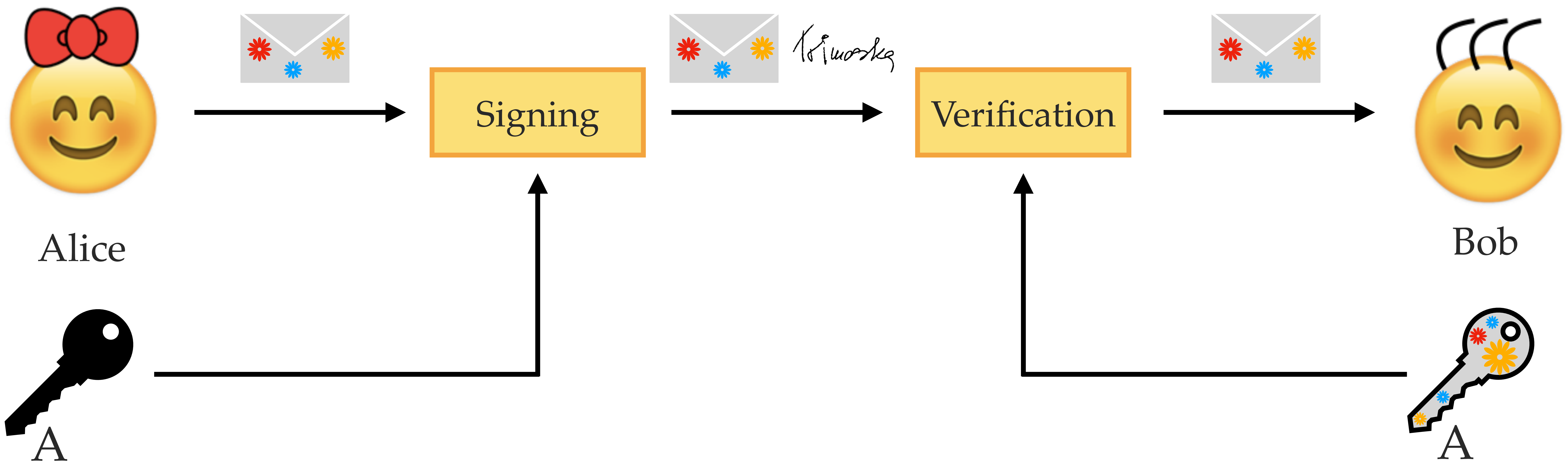
- The linear transformations hide the structure of the central map.

The trapdoor construction

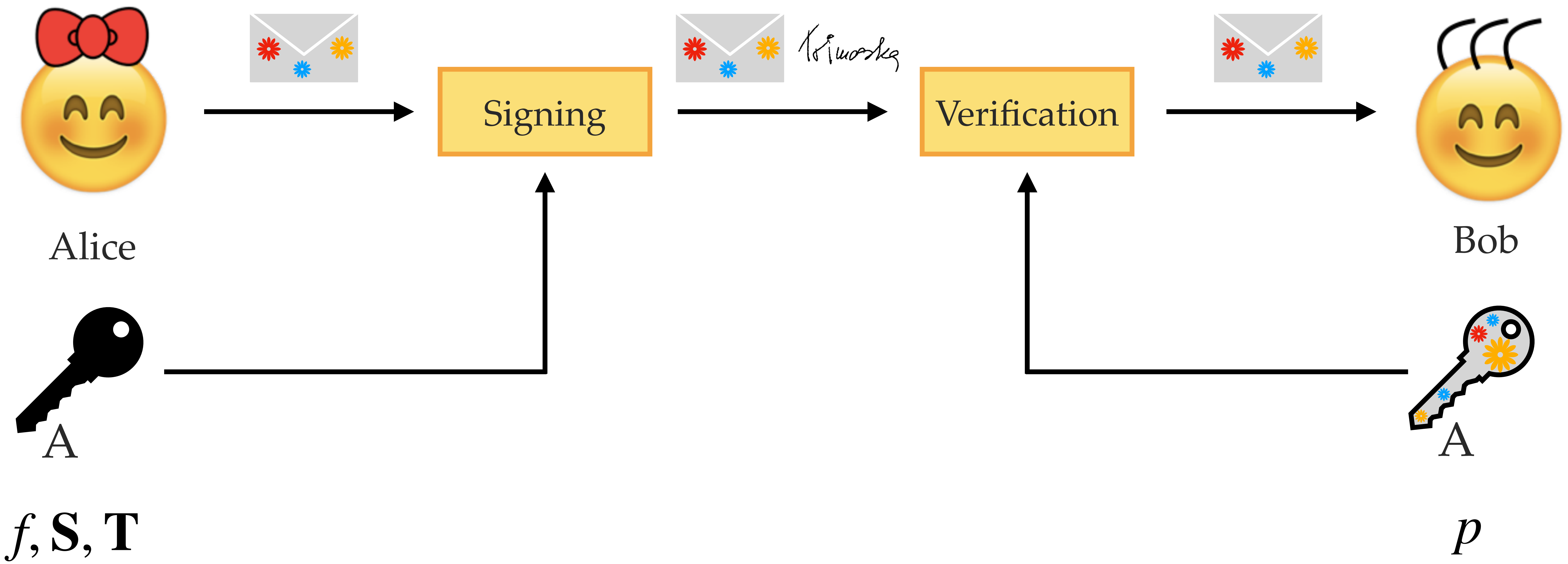


General workflow

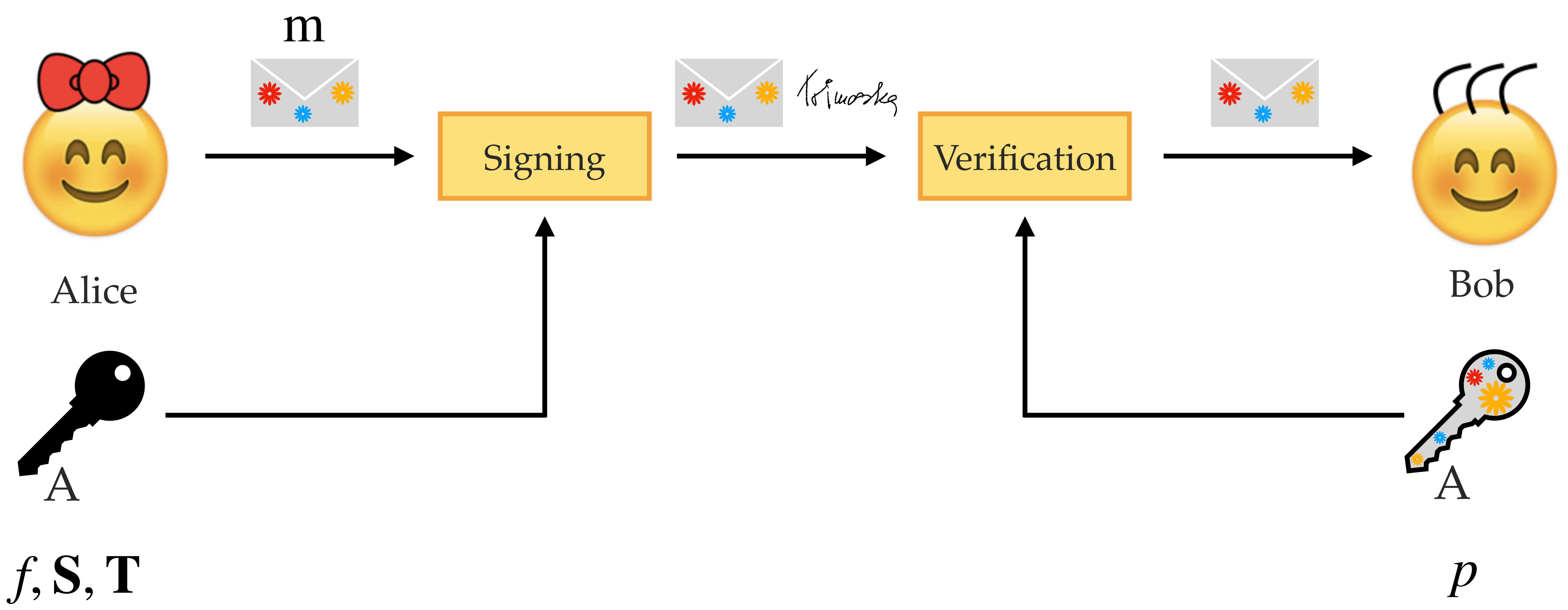
The trapdoor construction



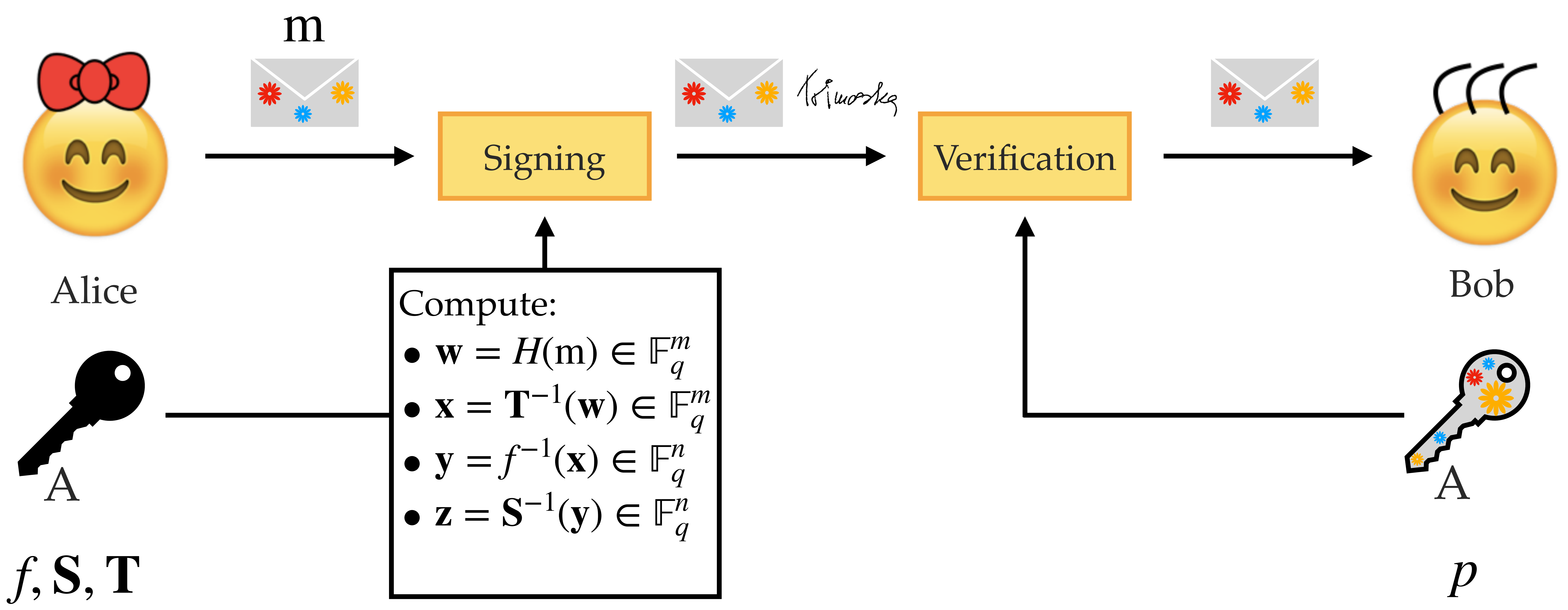
The trapdoor construction



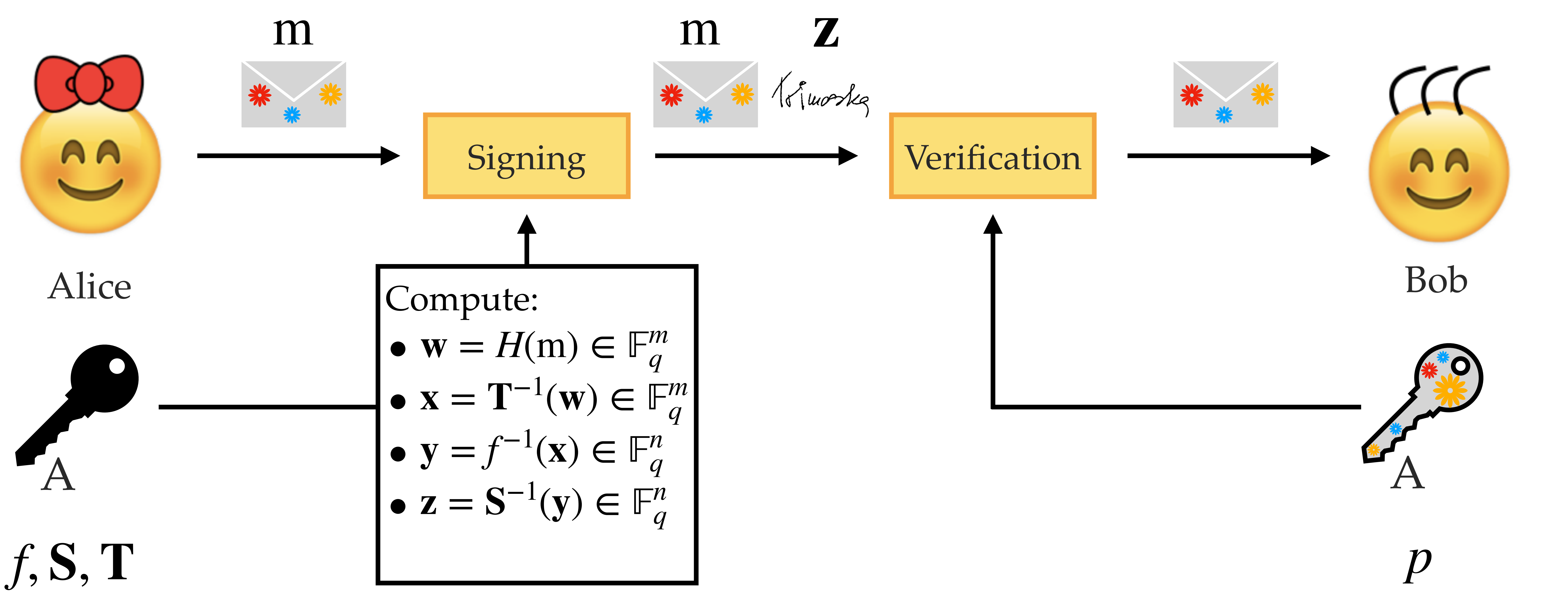
The trapdoor construction



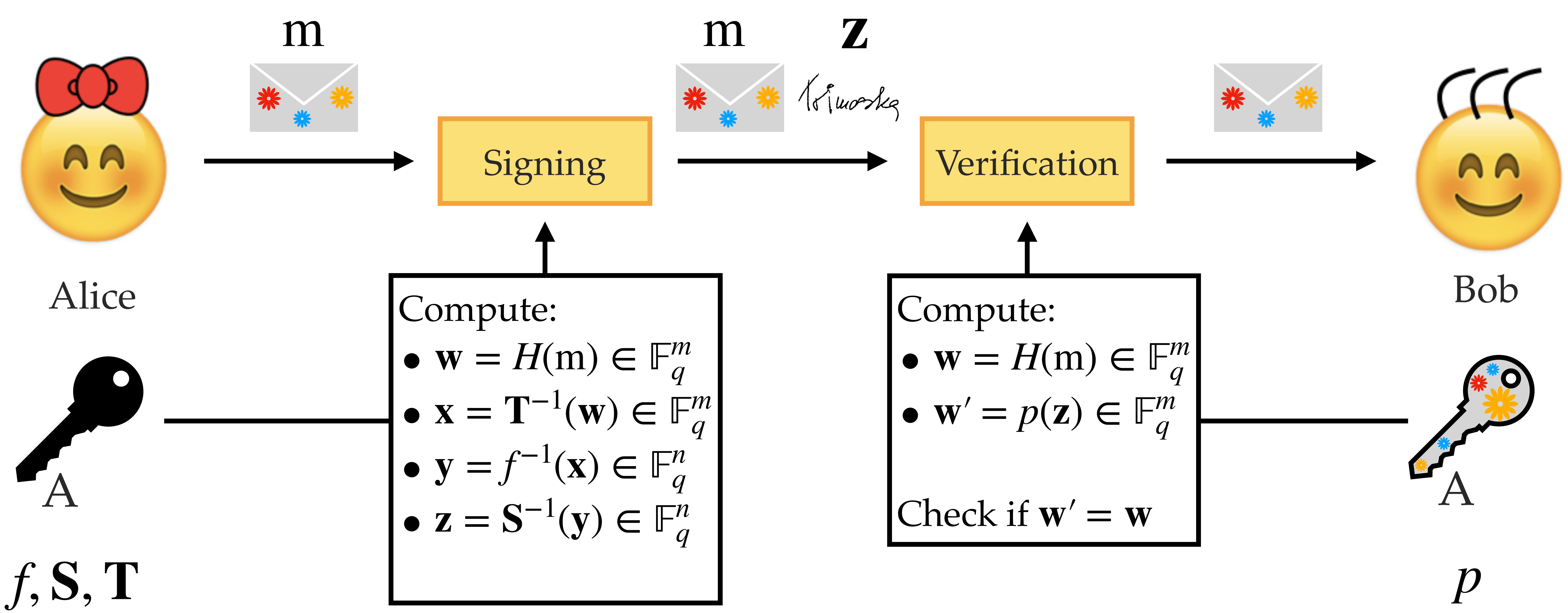
The trapdoor construction



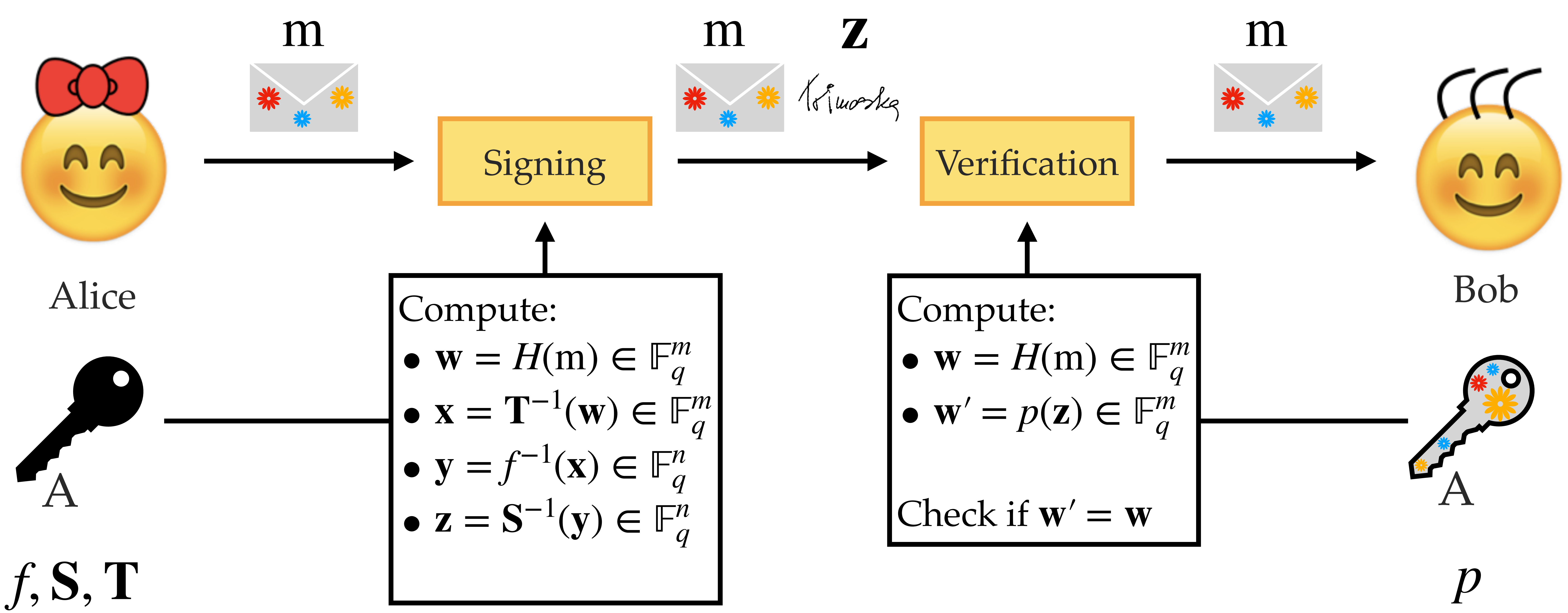
The trapdoor construction



The trapdoor construction



The trapdoor construction



Unbalanced Oil and Vinegar (UOV)

[Kipnis, Patarin, Goubin, 1999]

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

$$\mathbf{x}^\top \quad \begin{bmatrix} x_1 & x_2 & x_3 & x_4 \end{bmatrix}$$

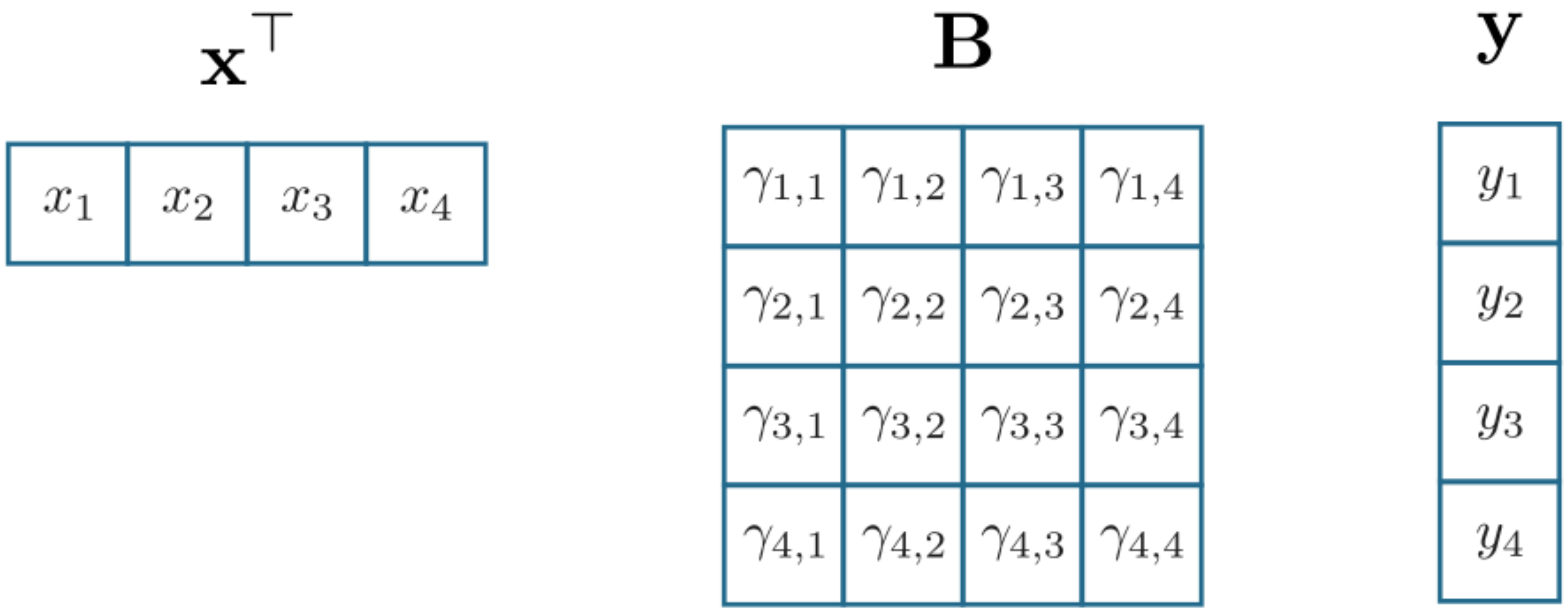
$$\mathbf{F} \quad \begin{bmatrix} \gamma_{1,1} & \frac{\gamma_{1,2}}{2} & \frac{\gamma_{1,3}}{2} & \frac{\gamma_{1,4}}{2} \\ \frac{\gamma_{2,1}}{2} & \gamma_{2,2} & \frac{\gamma_{2,3}}{2} & \frac{\gamma_{2,4}}{2} \\ \frac{\gamma_{3,1}}{2} & \frac{\gamma_{3,2}}{2} & \gamma_{3,3} & \frac{\gamma_{3,4}}{2} \\ \frac{\gamma_{4,1}}{2} & \frac{\gamma_{4,2}}{2} & \frac{\gamma_{4,3}}{2} & \gamma_{4,4} \end{bmatrix}$$

$$\mathbf{x} \quad \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

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

Matrix representation of bilinear forms

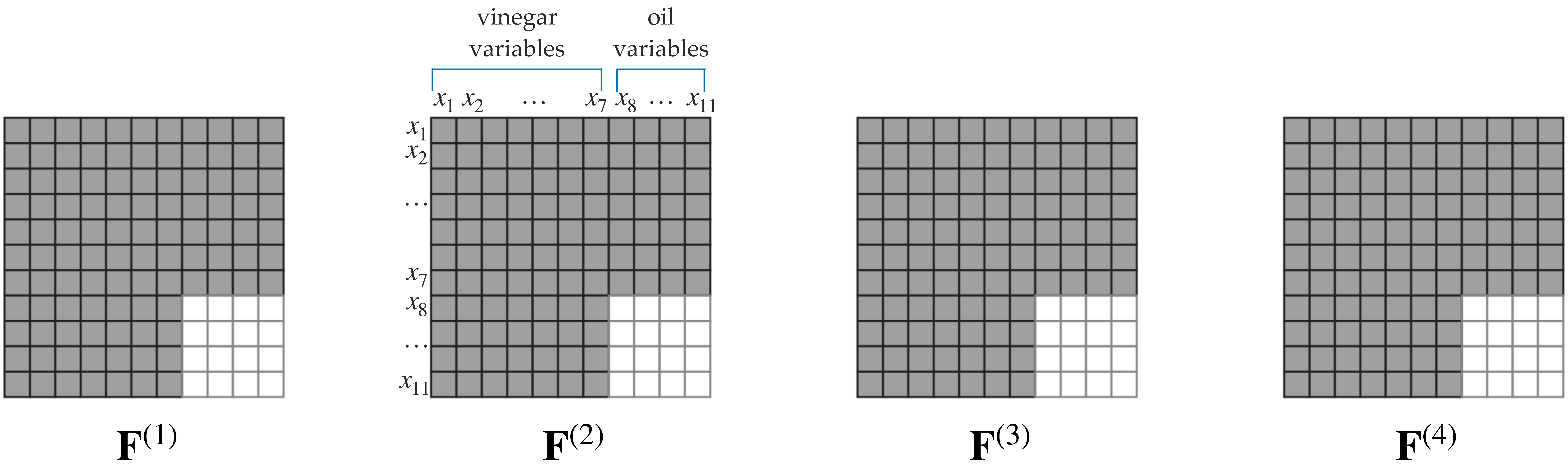
Bilinear form: $f(\mathbf{x}, \mathbf{y}) = \sum \gamma_{ij} x_i y_j$



so with $\mathbf{x} = (x_1, \ldots, x_n)$ and $\mathbf{y} = (y_1, \ldots, y_n)$, we get $\mathbf{x}^\top \mathbf{B} \mathbf{y}$.

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;

UOV key generation

In matrix representation

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

UOV 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 ?

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

UOV 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:

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

UOV 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:

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

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

UOV 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:

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

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

UOV in the NIST competition

UOV
TUOV
PROV
MAYO
VOX
QR-UOV
SNOVA

UOV in the NIST competition

- UOV
- TUOV
- PROV
- MAYO
- VOX
- QR-UOV
- SNOVA
- MQ-Sign (in KpqC)

Example.

	NIST SL	n	m	$\mathbb{F}_q$	|pk| (bytes)	|sk| (bytes)	|cpk| (bytes)	|sig+salt| (bytes)
ov-Ip	1	112	44	$\mathbb{F}_{256}$	278 432	237 896	43 576	128
ov-Is	1	160	64	$\mathbb{F}_{16}$	412 160	348 704	66 576	96
ov-III	3	184	72	$\mathbb{F}_{256}$	1 225 440	1 044 320	189 232	200
ov-V	5	244	96	$\mathbb{F}_{256}$	2 869 440	2 436 704	446 992	260

UOV in the NIST competition

- UOV
- TUOV
- PROV
- MAYO
- VOX
- QR-UOV
- SNOVA
- MQ-Sign (in KpqC)

Example.

	NIST SL	n	m	$\mathbb{F}_q$	$\lvert pk \rvert$ (bytes)	$\lvert sk \rvert$ (bytes)	$\lvert cpk \rvert$ (bytes)	$\lvert sig+salt \rvert$ (bytes)
ov-Ip	1	112	44	$\mathbb{F}_{256}$	278 432	237 896	43 576	128
ov-Is	1	160	64	$\mathbb{F}_{16}$	412 160	348 704	66 576	96
ov-III	3	184	72	$\mathbb{F}_{256}$	1 225 440	1 044 320	189 232	200
ov-V	5	244	96	$\mathbb{F}_{256}$	2 869 440	2 436 704	446 992	260

- We choose $n \sim 2.5m$ (slightly bigger than)

UOV-like schemes have:

- Big public keys
- Small signatures

Attacks on UOV

Attacks on UOV

- Direct attack
- Reconciliation attack
- Kipnis-Shamir attack
- Intersection attack

Direct attack

Direct attack

Try to forge a signature with only the knowledge of the public key.

Direct attack

Try to forge a signature with only the knowledge of the public key.

Constraint for modelisation

For a target $\mathbf{w}$, find $\mathbf{z}$ such that $p(\mathbf{z}) = \mathbf{w}$.

Direct attack

Try to forge a signature with only the knowledge of the public key.

Constraint for modelisation

For a target $\mathbf{w}$, find $\mathbf{z}$ such that $p(\mathbf{z}) = \mathbf{w}$.

Equations:

$$\mathbf{z}^\top \mathbf{P}^{(1)} \mathbf{z} = w_1$$
$$\mathbf{z}^\top \mathbf{P}^{(2)} \mathbf{z} = w_2$$
$$\dots$$
$$\mathbf{z}^\top \mathbf{P}^{(m)} \mathbf{z} = w_m$$

Direct attack

Try to forge a signature with only the knowledge of the public key.

Constraint for modelisation

For a target $\mathbf{w}$, find $\mathbf{z}$ such that $p(\mathbf{z}) = \mathbf{w}$.

Equations:

$$\mathbf{z}^\top \mathbf{P}^{(1)} \mathbf{z} = w_1$$
$$\mathbf{z}^\top \mathbf{P}^{(2)} \mathbf{z} = w_1$$
$$\dots$$
$$\mathbf{z}^\top \mathbf{P}^{(m)} \mathbf{z} = w_m$$

Reconciliation attack

[Ding, Yang, Chen, Chen, Cheng, 2008]
(using description from [Samardjiska, Gligoroski, 2014])

The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

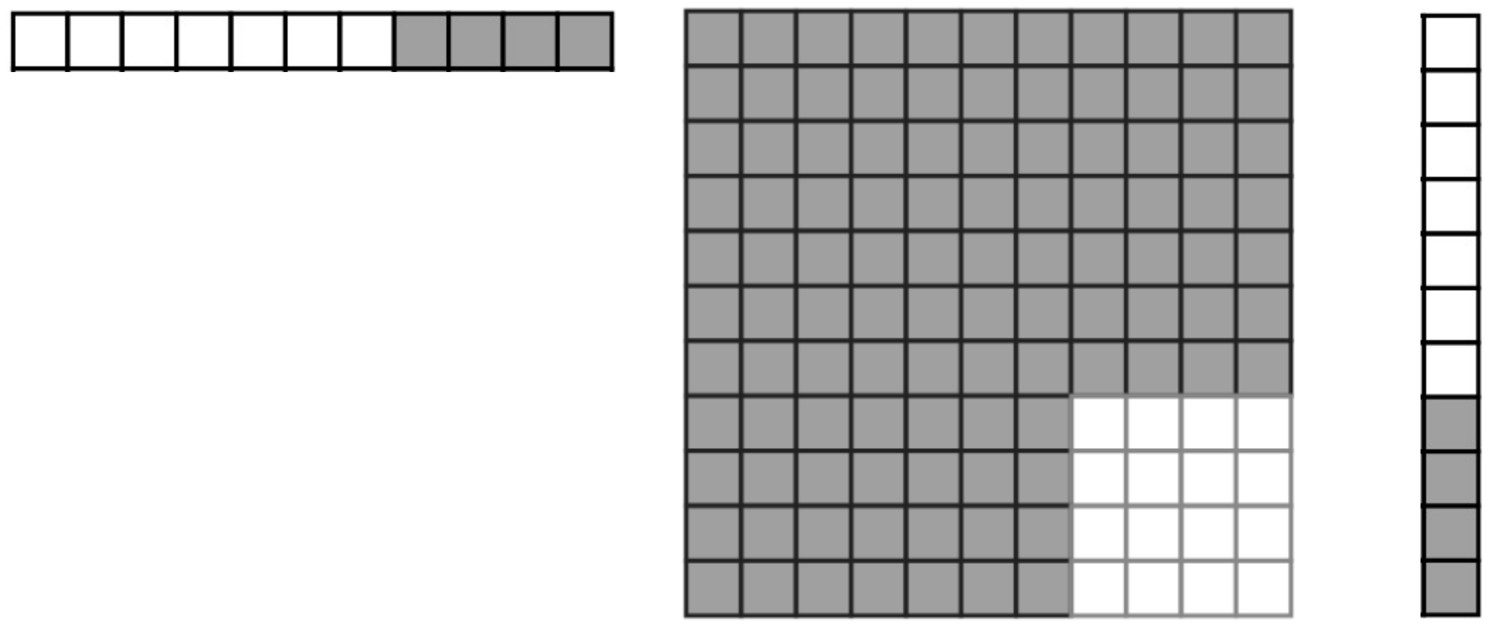
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n - m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n - m\}$.



$$= 0$$

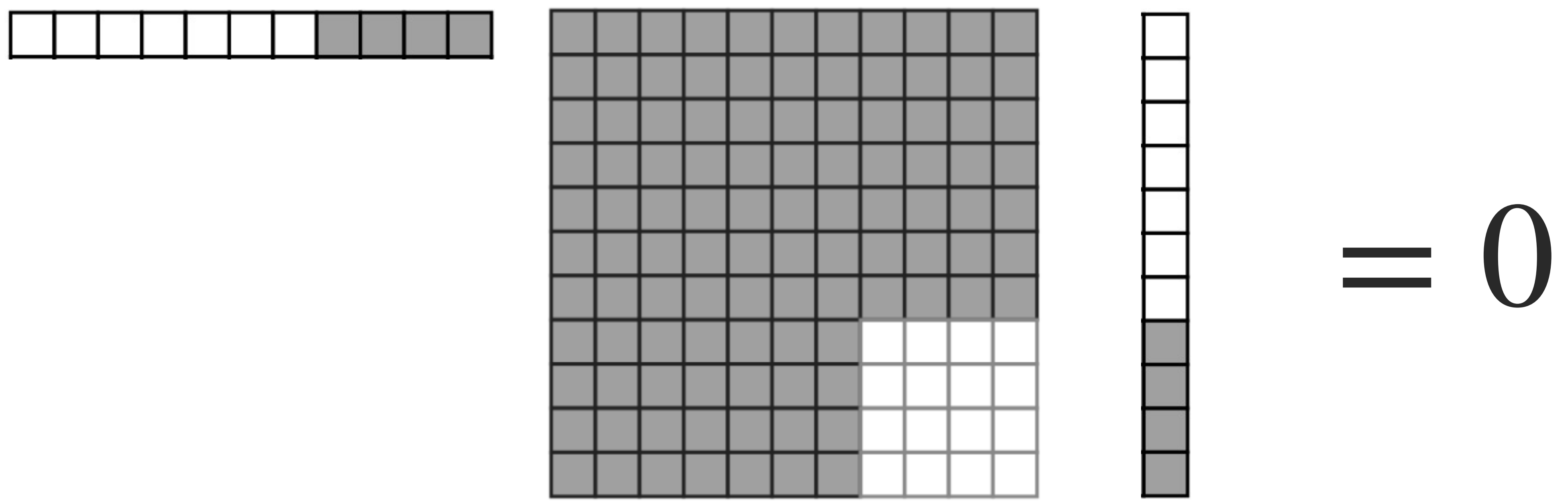
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n - m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n - m\}$.



f vanishes on O'.

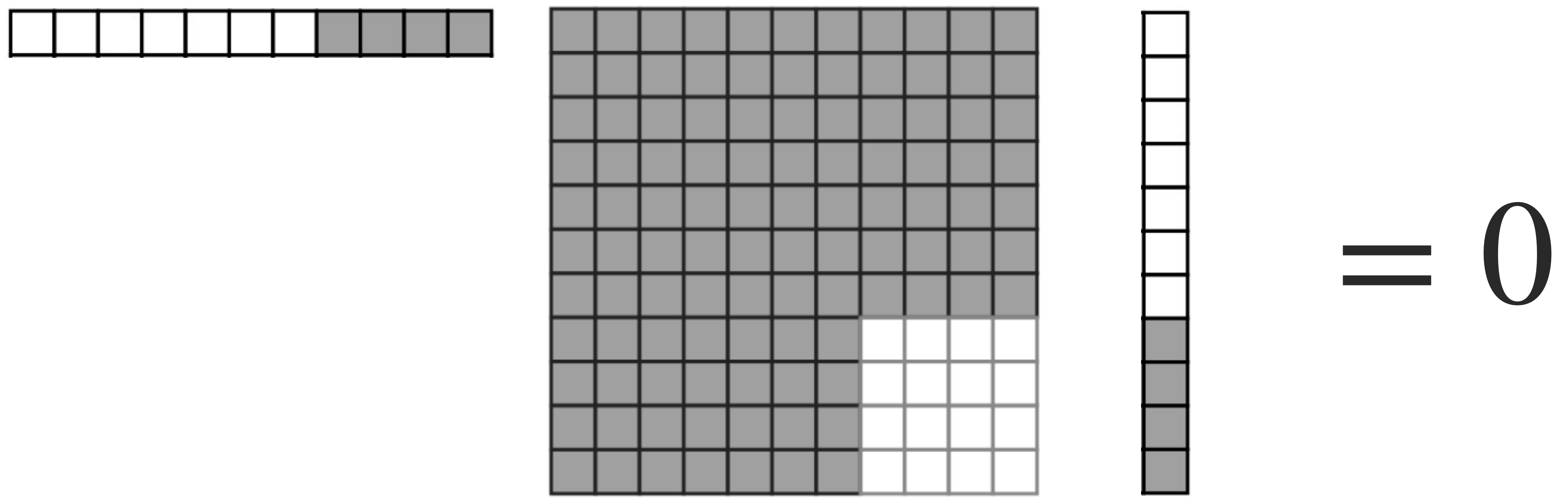
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n - m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n - m\}$.



f vanishes on O'.

Let $O = \mathbf{S}^{-1}(O')$.

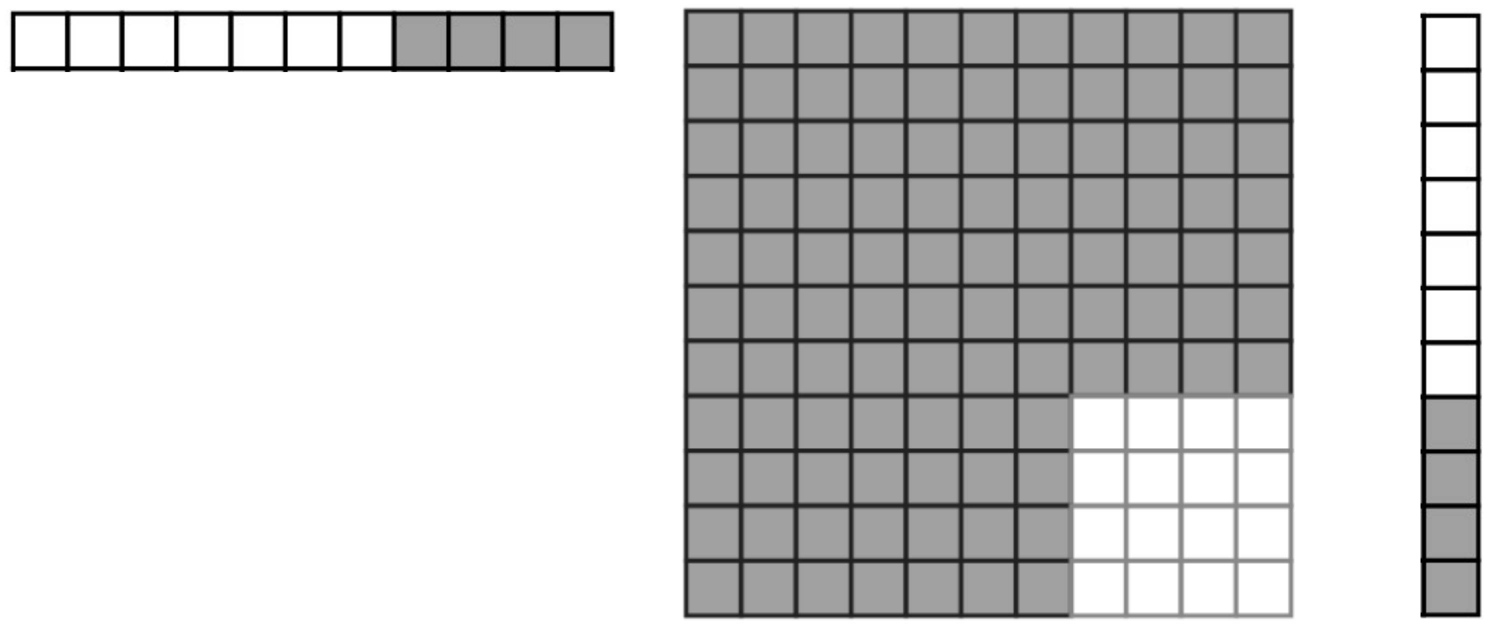
The secret subspace O

The map p with a UOV trapdoor vanishes on a linear subspace $O \subset \mathbb{F}_q^n$ of $\dim(O) = m$:

$$p(\mathbf{o}) = 0, \text{ for all } \mathbf{o} \in O.$$

Why ?

Let $O' \in \mathbb{F}_q^n$ be the m-dimensional space that consists of all the vectors whose first $n - m$ entries (corresponding to the vinegar variables) are zero: $O' = \{\mathbf{v} \mid v_i = 0 \text{ for all } i \leq n - m\}$.



$= 0$

f vanishes on O'.

Let $O = \mathbf{S}^{-1}(O')$.

p vanishes on O.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p^{'(1)}, \ldots, p^{'(m)})$ such that

$$p^{'(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\} .$$

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p'^{(1)}, \ldots, p'^{(m)})$ such that

$$p'^{(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\}\,.$$

What does $p'^{(k)}(\mathbf{x}, \mathbf{y})$ look like ?

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p^{'(1)}, \ldots, p^{'(m)})$ such that

$$p^{'(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\} .$$

What does $p^{'(k)}(\mathbf{x}, \mathbf{y})$ look like ?

Let $\tilde{\mathbf{P}}^{(k)}$ be the upper triangular representation of $p^{(k)}$.

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p^{'(1)}, \ldots, p^{'(m)})$ such that

$$p^{'(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\}\,.$$

What does $p^{'(k)}(\mathbf{x}, \mathbf{y})$ look like ?

Let $\tilde{\mathbf{P}}^{(k)}$ be the upper triangular representation of $p^{(k)}$.

$$\begin{aligned} p^{'(k)}(\mathbf{x}, \mathbf{y}) &= p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}) \\ &= (\mathbf{x} + \mathbf{y})^\top \tilde{\mathbf{P}}^{(k)}(\mathbf{x} + \mathbf{y}) - x^\top \tilde{\mathbf{P}}^{(k)}\mathbf{x} - y^\top \tilde{\mathbf{P}}^{(k)}\mathbf{y} \\ &= x^\top \tilde{\mathbf{P}}^{(k)}\mathbf{y} + y^\top \tilde{\mathbf{P}}^{(k)}\mathbf{x} \\ &= x^\top (\tilde{\mathbf{P}}^{(k)} + \tilde{\mathbf{P}}^{(k)\top})\mathbf{y} = x^\top \mathbf{B}^{(k)}\mathbf{y} \end{aligned}$$

The polar form

The polar form of a quadratic map $p = (p^{(1)}, \ldots, p^{(m)})$ is the bilinear form $p' = (p'^{(1)}, \ldots, p'^{(m)})$ such that

$$p'^{(k)}(\mathbf{x}, \mathbf{y}) = p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}), \text{ for all } k \in \{1, \ldots, m\}.$$

What does $p'^{(k)}(\mathbf{x}, \mathbf{y})$ look like ?

Let $\tilde{\mathbf{P}}^{(k)}$ be the upper triangular representation of $p^{(k)}$.

$$\begin{aligned} p'^{(k)}(\mathbf{x}, \mathbf{y}) &= p^{(k)}(\mathbf{x} + \mathbf{y}) - p^{(k)}(\mathbf{x}) - p^{(k)}(\mathbf{y}) \\ &= (\mathbf{x} + \mathbf{y})^\top \tilde{\mathbf{P}}^{(k)} (\mathbf{x} + \mathbf{y}) - x^\top \tilde{\mathbf{P}}^{(k)} \mathbf{x} - y^\top \tilde{\mathbf{P}}^{(k)} \mathbf{y} \\ &= x^\top \tilde{\mathbf{P}}^{(k)} \mathbf{y} + y^\top \tilde{\mathbf{P}}^{(k)} \mathbf{x} \\ &= x^\top (\tilde{\mathbf{P}}^{(k)} + \tilde{\mathbf{P}}^{(k)\top}) \mathbf{y} = x^\top \mathbf{B}^{(k)} \mathbf{y} \end{aligned}$$

→ So, p' is bilinear and symmetric.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1, \ldots, m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1, \ldots, m\}$.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1,\ldots,m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1,\ldots,m\}$.

→ Equations:

For $i \in \{1,\ldots,m\}$ **do**

$\mathbf{o}_i = (o_1, \ldots, o_v, 0, \ldots, 1_{n-i+1}, 0, \ldots, 0)$

Solve:

$$\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0, \text{ for } k \in \{1,\ldots,m\} \text{ and } j < i$$

$$\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0, \text{ for } k \in \{1,\ldots,m\}$$

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1,\ldots,m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1,\ldots,m\}$.

Equations:

For $i \in \{1,\ldots,m\}$ **do**

$\mathbf{o}_i = (o_1, \ldots, o_v, 0, \ldots, 1_{n-i+1}, 0, \ldots, 0)$

Solve:

$$\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0, \text{ for } k \in \{1,\ldots,m\} \text{ and } j < i$$

$$\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0, \text{ for } k \in \{1,\ldots,m\}$$

In the first iteration, we have only quadratic equations, so this is the bottleneck. Linear constraints facilitate the resolution of a system.

Reconciliation attack

Find the secret oil subspace O : find m linearly independent vectors in O.

Constraint for modelisation

For any vector $\mathbf{o}_i \in O$, we have that $\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0$ for all $k \in \{1, \ldots, m\}$.
For any pair of vectors $\mathbf{o}_i, \mathbf{o}_j \in O$, we have that $\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0$ for all $k \in \{1, \ldots, m\}$.

Equations:

For $i \in \{1, \ldots, m\}$ **do**

$$\mathbf{o}_i = (o_1, \ldots, o_v, 0, \ldots, 1_{n-i+1}, 0, \ldots, 0)$$

Solve:

$$\mathbf{o}_i^\top \mathbf{B}^{(k)} \mathbf{o}_j = 0, \text{ for } k \in \{1, \ldots, m\} \text{ and } j < i$$

$$\mathbf{o}_i^\top \mathbf{P}^{(k)} \mathbf{o}_i = 0, \text{ for } k \in \{1, \ldots, m\}$$

In the first iteration, we have only quadratic equations, so this is the bottleneck. Linear constraints facilitate the resolution of a system.

Kipnis-Shamir attack

[Kipnis, Shamir, 1998]

The orthogonal complement of a subspace

Let $V \subset \mathbb{F}_q^n$. The orthogonal complement of V is $V^\perp$ such that

$$V^\perp = \{\tilde{\mathbf{v}}_i \in \mathbb{F}_q^n \,|\, \langle \mathbf{v}_j, \tilde{\mathbf{v}}_i \rangle = 0, \text{ for all } \mathbf{v}_j \in V\}.$$

If V is m-dimensional, then $V^\perp$ is $(n-m)$-dimensional.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)} O \subset O^{\perp}$.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

$$
\begin{aligned}
\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\
&= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\
&= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0
\end{aligned}
$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)} O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)} O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)} O = O^{\perp} = \mathbf{B}^{(k_2)} O$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)} \mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top} \mathbf{B}^{(k)} \mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^{\perp} = \mathbf{B}^{(k_2)}O$.

Hence, we have that $\mathbf{B}^{(k_1)-1}\mathbf{B}^{(k_2)}O = O$, for all pairs $\mathbf{B}^{(k_1)}, \mathbf{B}^{(k_2)}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p^{'(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^{\perp} = \mathbf{B}^{(k_2)}O$.

Hence, we have that $\mathbf{B}^{(k_1)-1}\mathbf{B}^{(k_2)}O = O$, for all pairs $\mathbf{B}^{(k_1)}, \mathbf{B}^{(k_2)}$.

$$
\begin{aligned}
\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\
&= p'^{(k)}(\mathbf{o}_1, \mathbf{o}_2) \\
&= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0
\end{aligned}
$$

Finding a common invariant subspace of a large number of linear maps is easy.

Kipnis-Shamir attack

Find the secret oil subspace O. Works well for the balanced case ($n = 2m$) - the original proposal of OV.

Constraint for modelisation

For each $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k)}O \subset O^{\perp}$.

Since $\dim(O^{\perp}) = n - m = m$, we have that $\mathbf{B}^{(k)}O = O^{\perp}$.

Since this is true for all $\mathbf{B}^{(k)}$, we have that $\mathbf{B}^{(k_1)}O = O^{\perp} = \mathbf{B}^{(k_2)}O$.

Hence, we have that $\mathbf{B}^{(k_1)-1}\mathbf{B}^{(k_2)}O = O$, for all pairs $\mathbf{B}^{(k_1)}, \mathbf{B}^{(k_2)}$.

$$\begin{aligned}\langle \mathbf{o}_2, \mathbf{B}^{(k)}\mathbf{o}_1 \rangle &= \mathbf{o}_2^{\top}\mathbf{B}^{(k)}\mathbf{o}_1 \\ &= p'^{(k)}(\mathbf{o}_1, \mathbf{o}_2) \\ &= p^{(k)}(\mathbf{o}_1 + \mathbf{o}_2) - p^{(k)}(\mathbf{o}_1) - p^{(k)}(\mathbf{o}_2) = 0\end{aligned}$$

→ Finding a common invariant subspace of a large number of linear maps is easy.

→ Oil and Vinegar becomes Unbalanced Oil and Vinegar because of this attack.

Intersection attack

[Beullens, 2021]

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

If $\mathbf{x}$ is in the intersection $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O$, then both $\mathbf{B}^{(k_1)-1}\mathbf{x}$ and $\mathbf{B}^{(k_2)-1}\mathbf{x}$ are in O.

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^\perp) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^\perp$ and $\mathbf{B}^{(k_2)}O \subset O^\perp$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

If $\mathbf{x}$ is in the intersection $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O$, then both $\mathbf{B}^{(k_1)-1}\mathbf{x}$ and $\mathbf{B}^{(k_2)-1}\mathbf{x}$ are in O.

→ Equations:

$$p(\mathbf{B}^{(k_1)-1}\mathbf{x}) = 0$$
$$p(\mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$
$$p'(\mathbf{B}^{(k_1)-1}\mathbf{x}, \mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$

Intersection attack

Find the secret oil subspace O. Use the ideas of the Kipnis-Shamir attack, but for the unbalanced case ($n > 2m$).

Constraint for modelisation

Since $n > 2m$, $\dim(O^{\perp}) > m$. We still have $\mathbf{B}^{(k_1)}O \subset O^{\perp}$ and $\mathbf{B}^{(k_2)}O \subset O^{\perp}$, but they are not (necessarily) the same subspace.

Idea: assuming that $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O \neq \emptyset$, try to find a vector $\mathbf{x}$ in this intersection.

If $\mathbf{x}$ is in the intersection $\mathbf{B}^{(k_1)}O \cap \mathbf{B}^{(k_2)}O$, then both $\mathbf{B}^{(k_1)-1}\mathbf{x}$ and $\mathbf{B}^{(k_2)-1}\mathbf{x}$ are in O.

→ Equations:

$$p(\mathbf{B}^{(k_1)-1}\mathbf{x}) = 0$$
$$p(\mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$
$$p'(\mathbf{B}^{(k_1)-1}\mathbf{x}, \mathbf{B}^{(k_2)-1}\mathbf{x}) = 0$$

→ The attack can be generalised to find a vector in the intersection of more than two subspaces.

Recap

- The MQ problem is (usually) hard.
- We have a variety of solvers for (over)determined systems.
- Modelisation can be crucial to how efficient an attack is.
- The MQ problem can be easy for some structured systems. We use this to build trapdoors in crypto.
- We saw three different ways to model the recovery of the UOV trapdoor.

Recap

- The MQ problem is (usually) hard.
- We have a variety of solvers for (over)determined systems.
- Modelisation can be crucial to how efficient an attack is.
- The MQ problem can be easy for some structured systems. We use this to build trapdoors in crypto.
- We saw three different ways to model the recovery of the UOV trapdoor.

Resources at https://mtrimoska.com/QSI-multivariate/

Credits: created with ♥, and Manim.