SIMULTANEOUS DIAGONALIZATION OF INCOMPLETE MATRICES AND APPLICATIONS

Jean-Sebastien Coron [1]
Luca Notarnicola [2]
Gabor Wiese [3]

[1,2,3] University of Luxembourg

14th Algorithmic Number Theory Symposium • July 2020

uni.lu
UNIVERSITÉ DU LUXEMBOURG

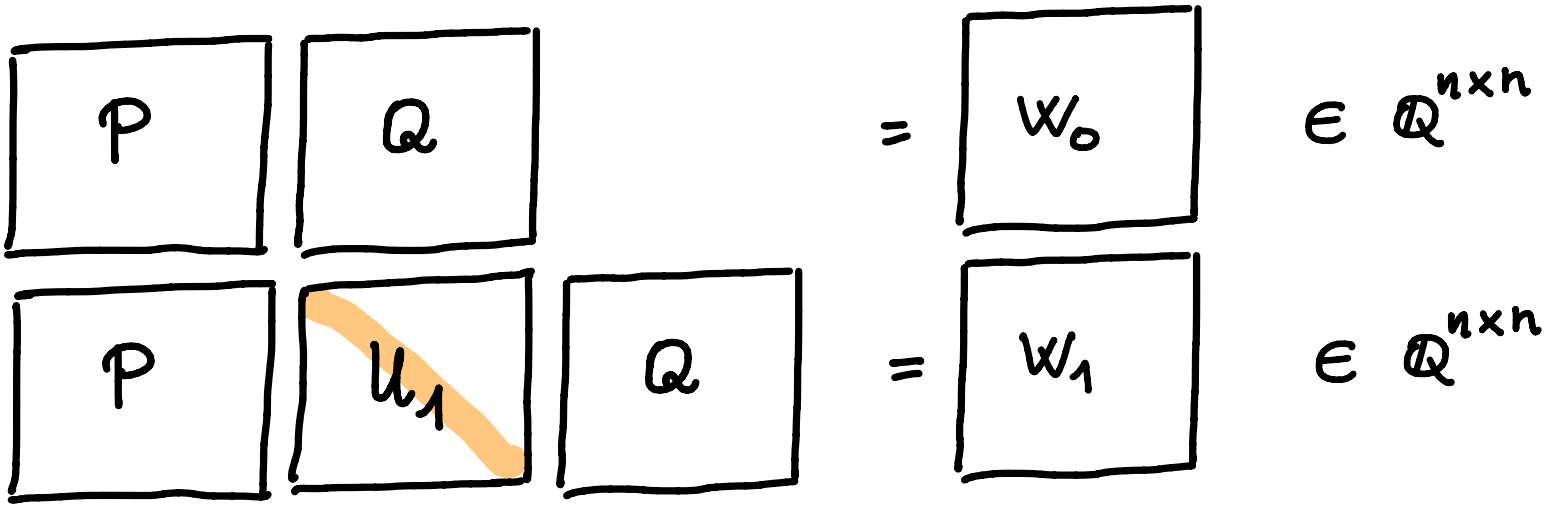
Introduction

Introduction

Let
- $n \in \mathbb{N}_{\geq 2}$
- $P, Q \in \mathbb{Q}^{n \times n}$ of full rank (then PQ of full rank)
- $U_1 \in \mathbb{Q}^{n \times n}$ diagonal matrix

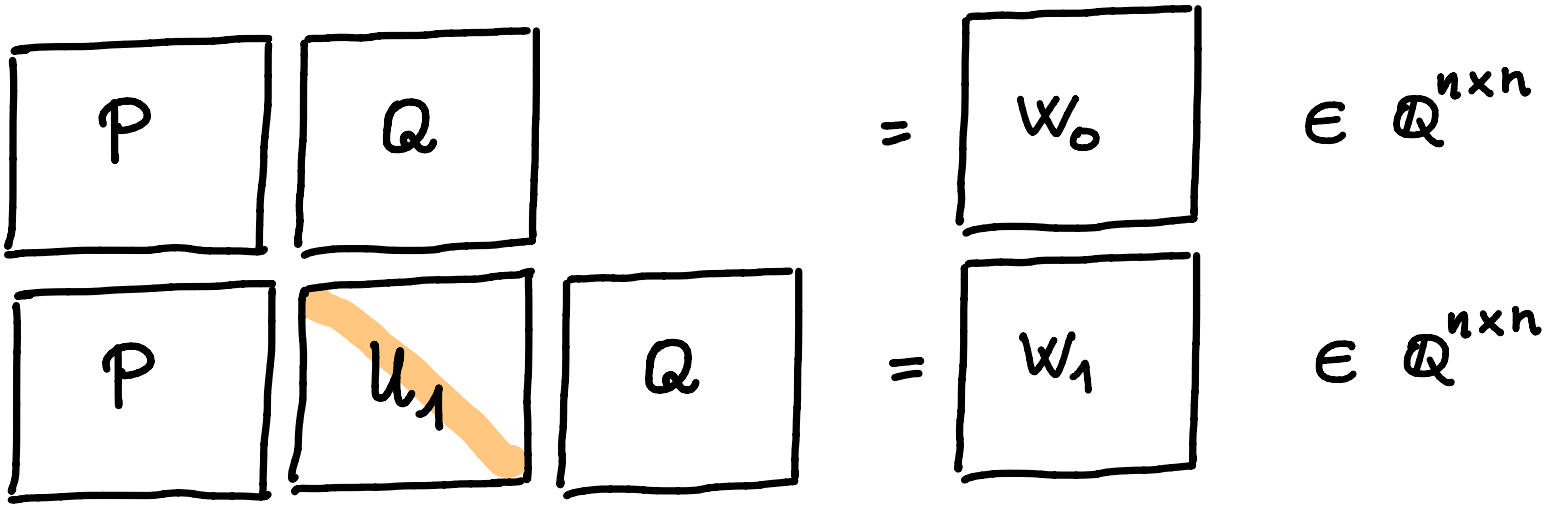
Introduction

Let
- $n \in \mathbb{N}_{\geq 2}$
- $P, Q \in \mathbb{Q}^{n \times n}$ of full rank (then PQ of full rank)
- $U_1 \in \mathbb{Q}^{n \times n}$ diagonal matrix



Introduction

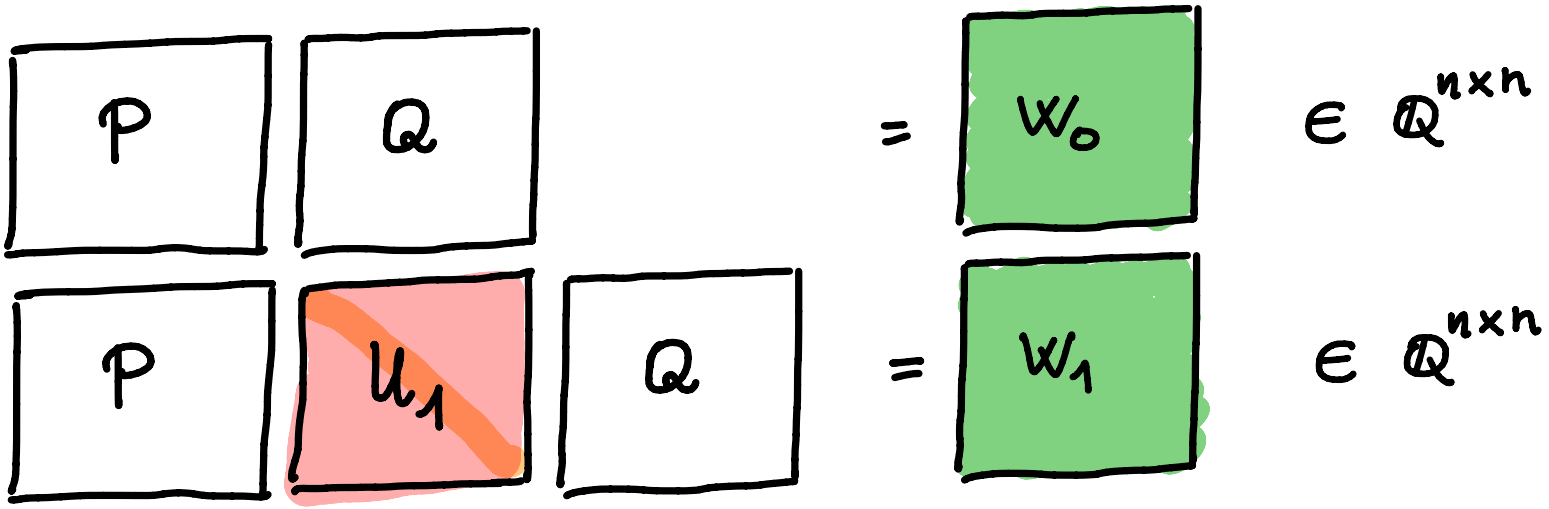
Let
- $n \in \mathbb{N}_{\geq 2}$
- $P, Q \in \mathbb{Q}^{n \times n}$ of full rank (then PQ of full rank)
- $U_1 \in \mathbb{Q}^{n \times n}$ diagonal matrix



Problem: given the product matrices $W_0, W_1 \in \mathbb{Q}^{n \times n}$ reveal the diagonal entries of $U_1 \in \mathbb{Q}^{n \times n}$

Introduction

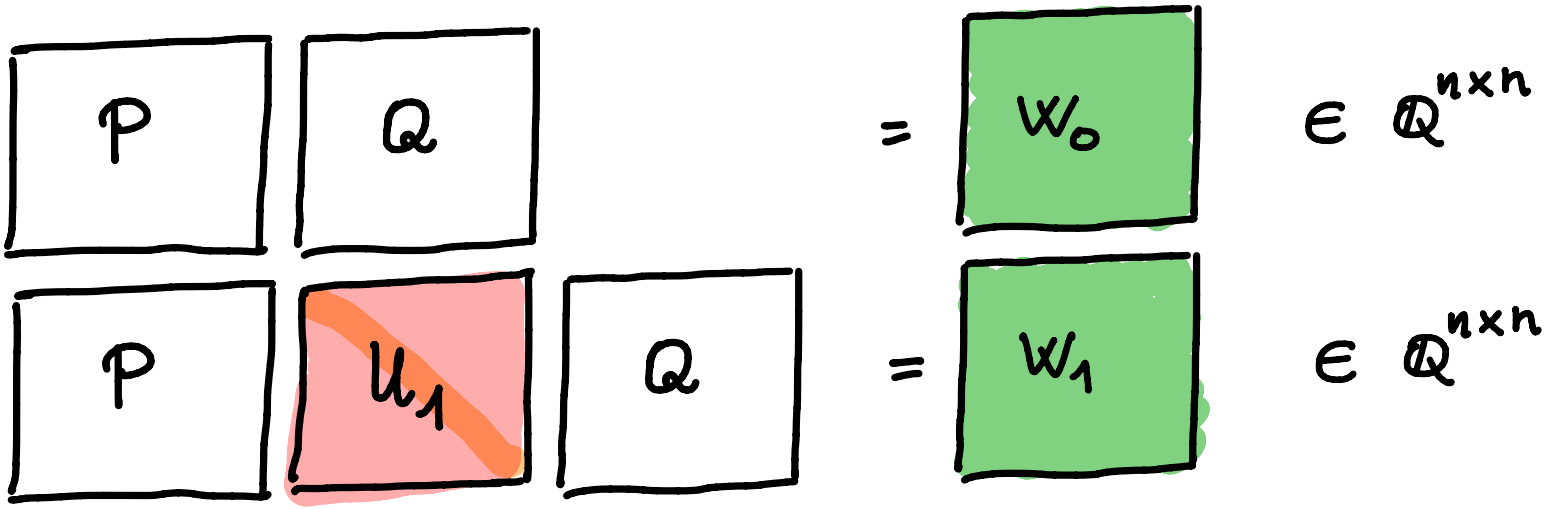
Let
- $n \in \mathbb{N}_{\geq 2}$
- $P, Q \in \mathbb{Q}^{n \times n}$ of full rank (then PQ of full rank)
- $U_1 \in \mathbb{Q}^{n \times n}$ diagonal matrix



Problem: given the product matrices $W_0, W_1 \in \mathbb{Q}^{n \times n}$, reveal the diagonal entries of $U_1 \in \mathbb{Q}^{n \times n}$

Introduction

Let
- $n \in \mathbb{N}_{\geq 2}$
- $P, Q \in \mathbb{Q}^{n \times n}$ of full rank (then PQ of full rank)
- $U_1 \in \mathbb{Q}^{n \times n}$ diagonal matrix

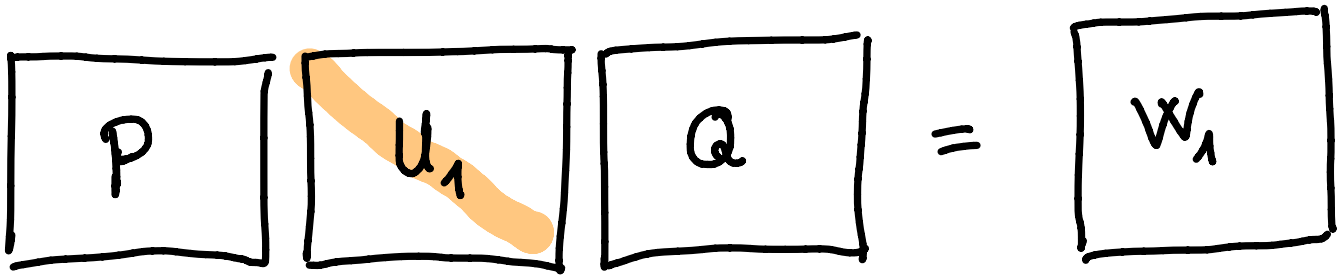


Problem: given the product matrices $W_0, W_1 \in \mathbb{Q}^{n \times n}$, reveal the diagonal entries of $U_1 \in \mathbb{Q}^{n \times n}$

Solution: compute the eigenvalues of $W_0^{-1} W_1 = Q^{-1} U_1 Q$

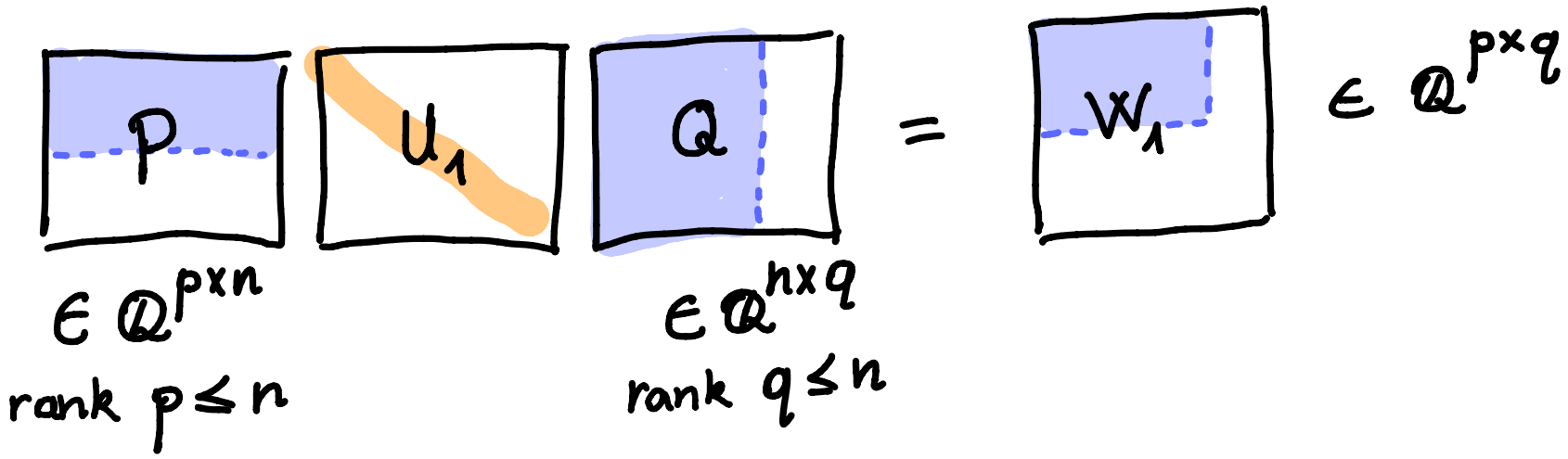
Problem Statement

- We now lower the ranks of P and Q



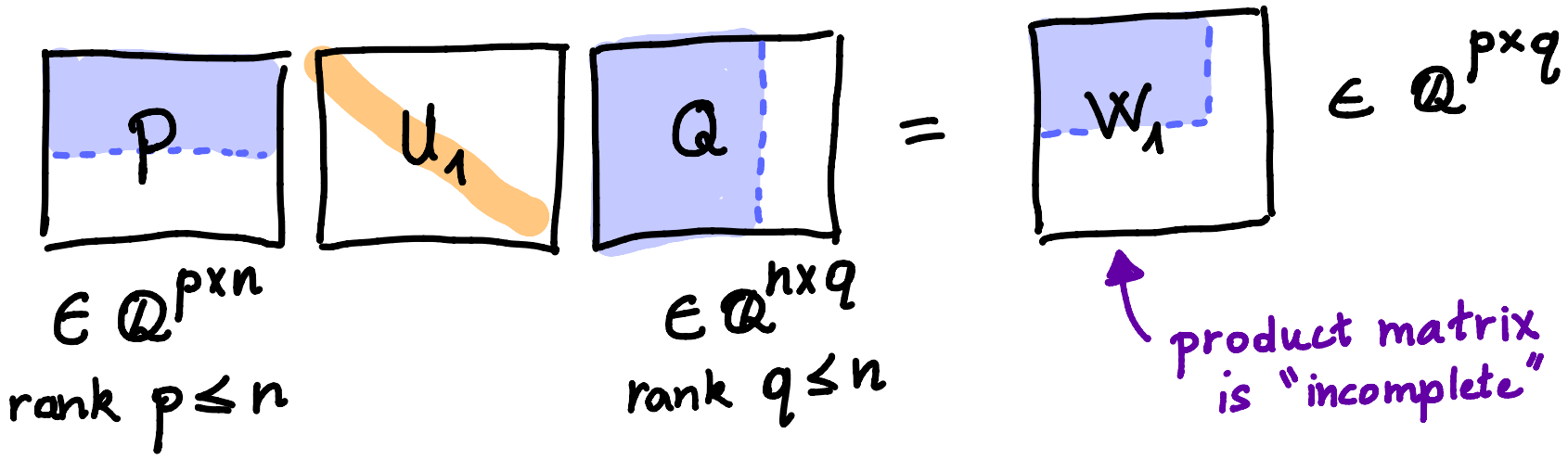
Problem Statement

- We now lower the ranks of P and Q



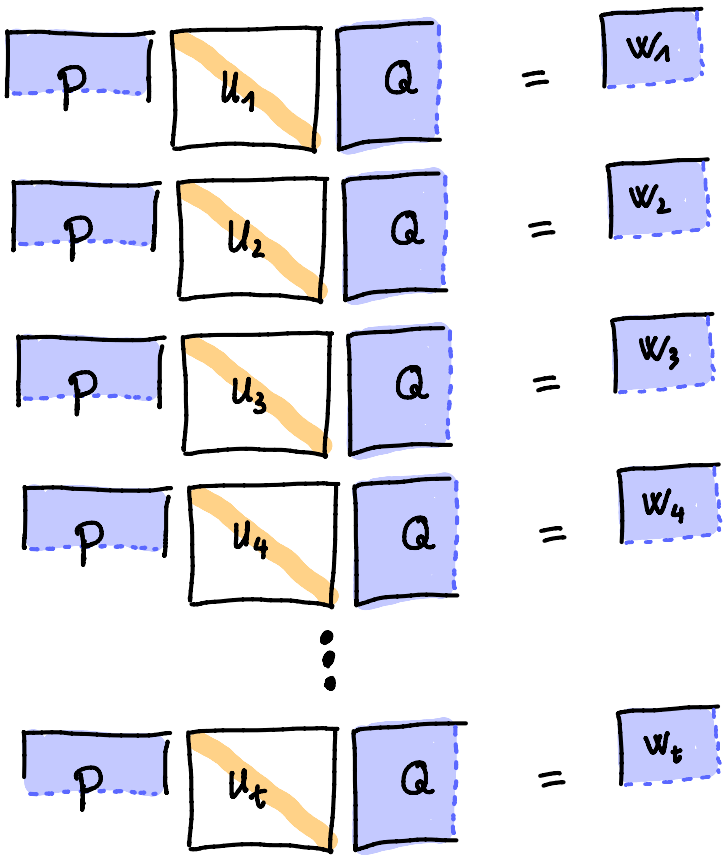
Problem Statement

- We now lower the ranks of P and Q



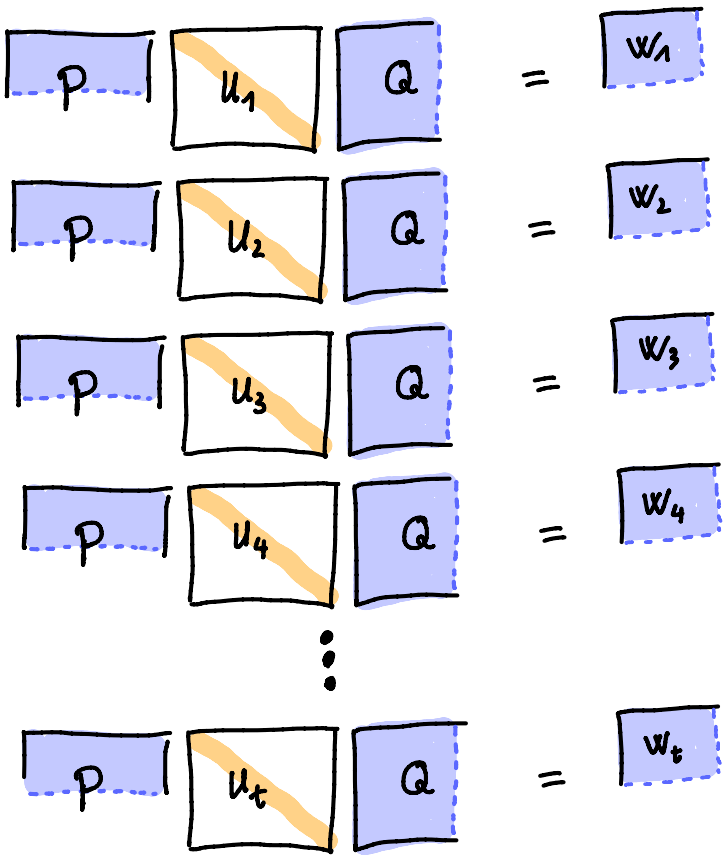
Problem Statement

- We now lower the ranks of P and Q
- In compensation we consider more samples ($i \in I := \{1, \ldots, t\}$, $t \geq 2$)



Problem Statement

- We now lower the ranks of P and Q
- In compensation we consider more samples ($i \in I := \{1, \dots, t\}$, $t \geq 2$)



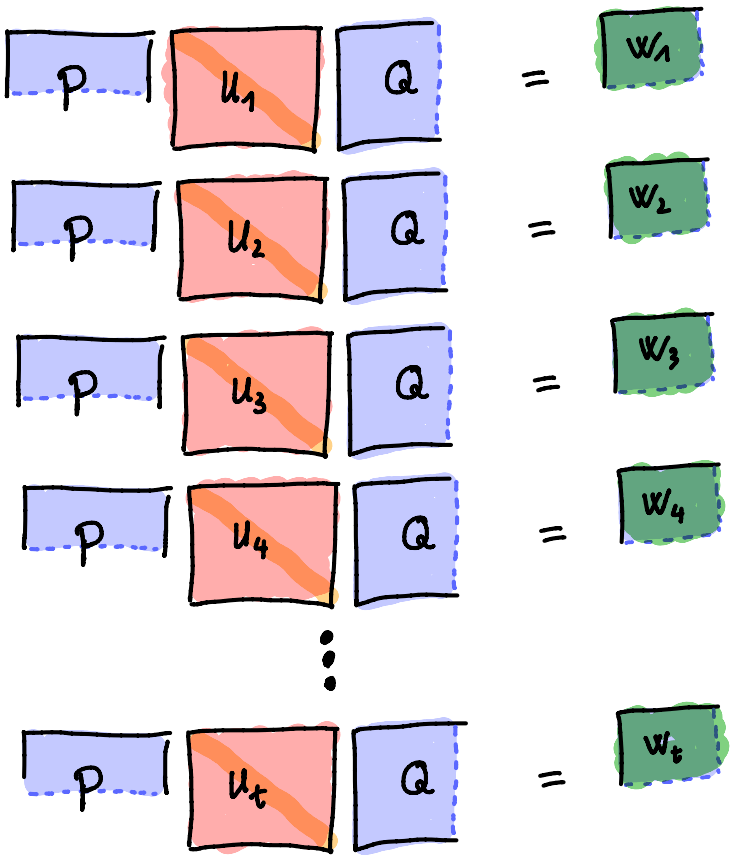
Problem Statement

given the matrices $\{W_i\}_{i \in I \cup \{0\}} \subset \mathbb{Q}^{p \times q}$

compute the diagonal entries of the matrices $\{U_i\}_{i \in I} \subset \mathbb{Q}^{n \times n}$

Problem Statement

- We now lower the ranks of P and Q
- In compensation we consider more samples ($i \in I := \{1, \dots, t\}$, $t \geq 2$)



Problem Statement

given the matrices $\{W_i\}_{i \in I \cup \{0\}} \subset \mathbb{Q}^{p \times q}$

compute the diagonal entries of the matrices $\{U_i\}_{i \in I} \subset \mathbb{Q}^{n \times n}$

Algorithms

- We study 2 cases :

  $(p \leq n, q = n)$ and $(p \leq n, q = p)$, $t \in \mathbb{N}$

- Find range of parameters minimizing p, q, t for fixed n

Algorithms

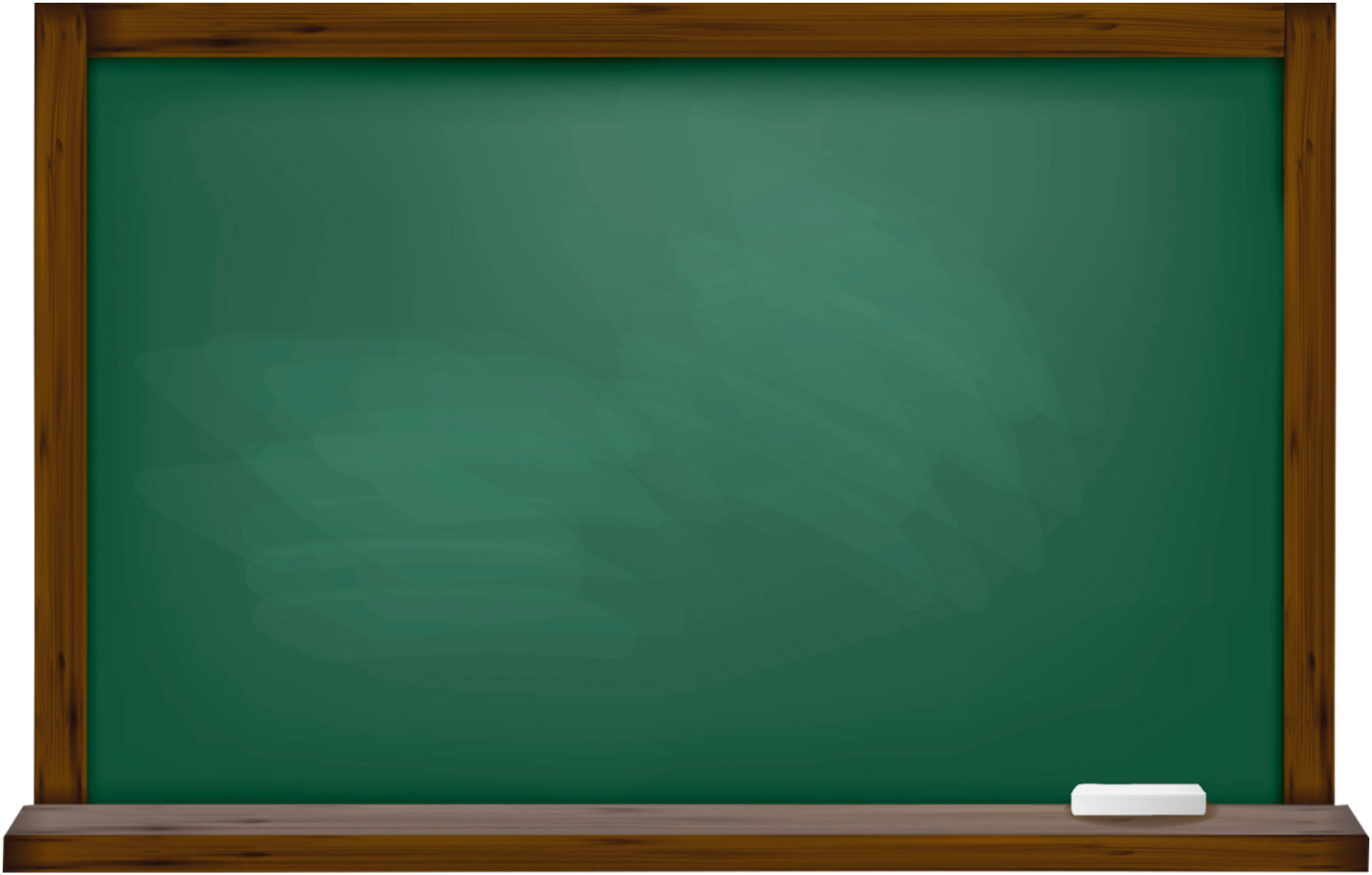
- We study 2 cases :

  $(p \leq n, q = n)$ and $(p \leq n, q = p)$, $t \in \mathbb{N}$

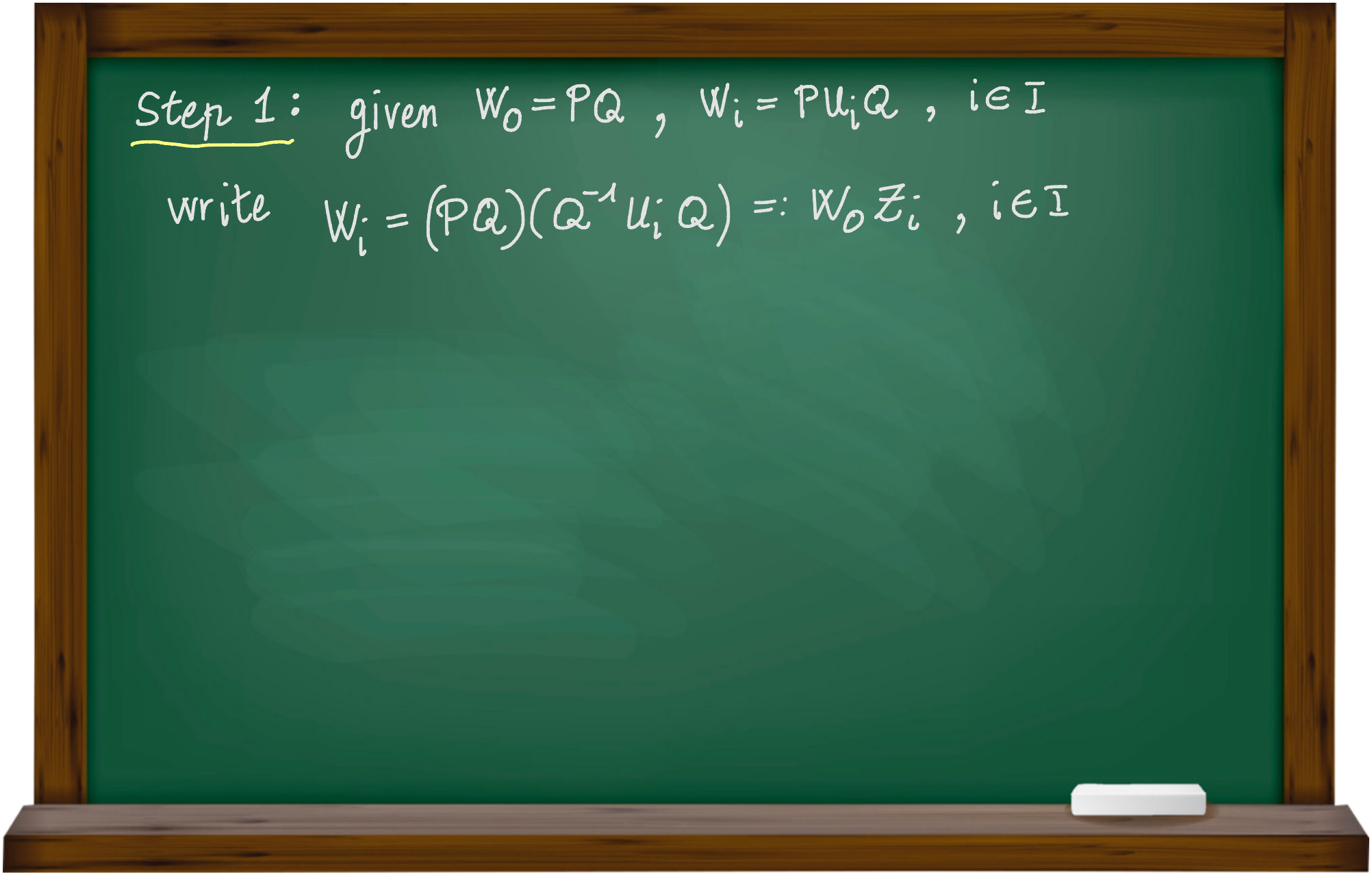
- Find range of parameters minimizing p, q, t for fixed n
- We obtain heuristic algorithms

$p \leq n, q = n$	$p \leq n, q = p$
$p = O(\sqrt{n})$ $q = n$ $t = O(\sqrt{n})$	$p = O(\frac{2}{3}n)$ $q = p$ $t = O(\sqrt{n})$

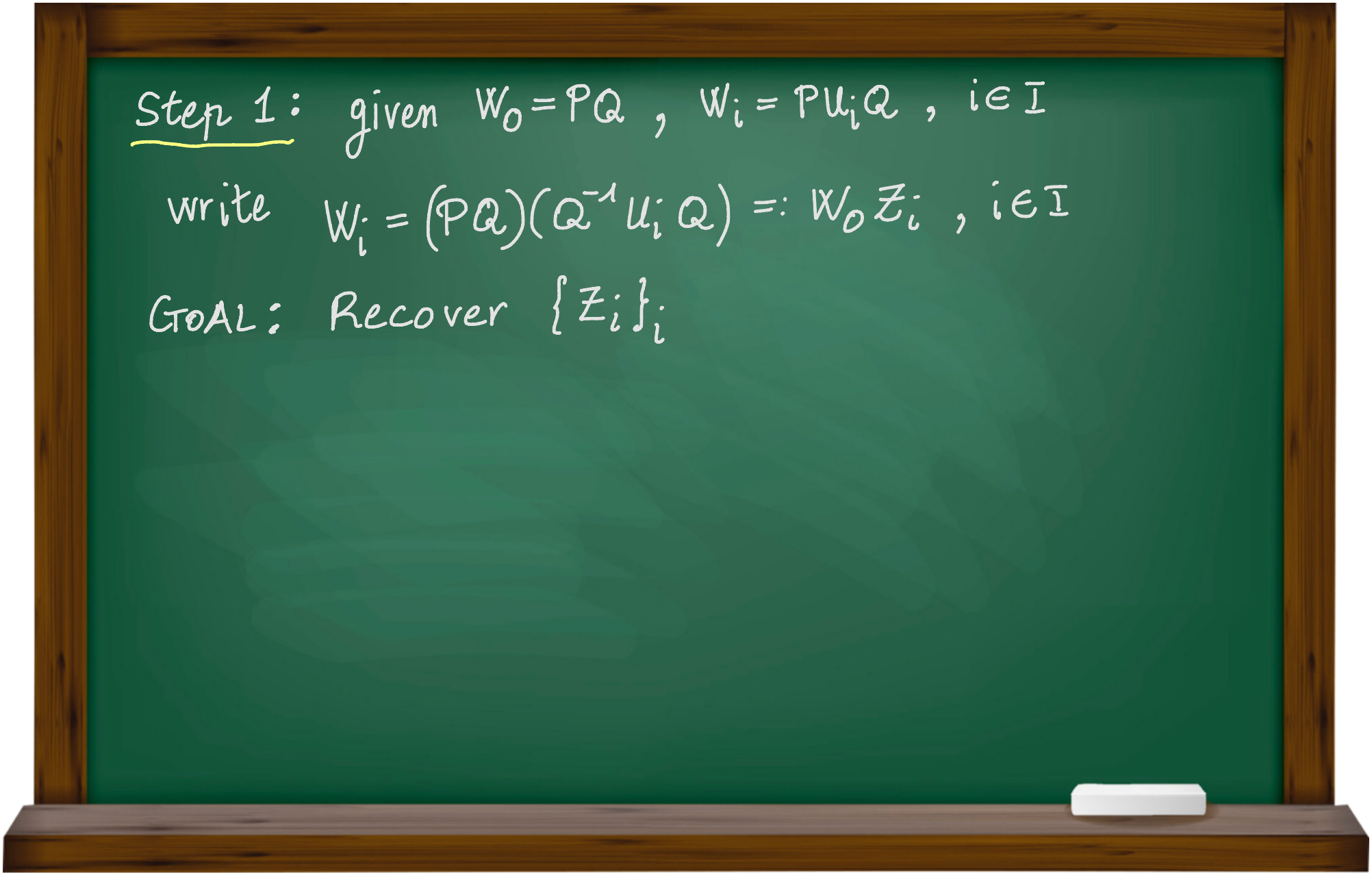
Case ($p \leq n$, $q = n$) - main tools



Case ($p \leq n$, $q = n$) - main tools



Case ($p \leq n$, $q = n$) - main tools



Case ($p \leq n$, $q = n$) - main tools

Step 1: given $W_0 = PQ$, $W_i = PU_iQ$, $i \in I$

write $W_i = (PQ)(Q^{-1}U_iQ) =: W_0 Z_i$, $i \in I$

GOAL: Recover $\{Z_i\}_i$

The matrices $\{Z_i\}_i$ commute

Case ($p \leq n$, $q = n$) - main tools

Step 1: given $W_0 = PQ$, $W_i = PU_iQ$, $i \in I$

write $W_i = (PQ)(Q^{-1}U_iQ) =: W_0 Z_i$, $i \in I$

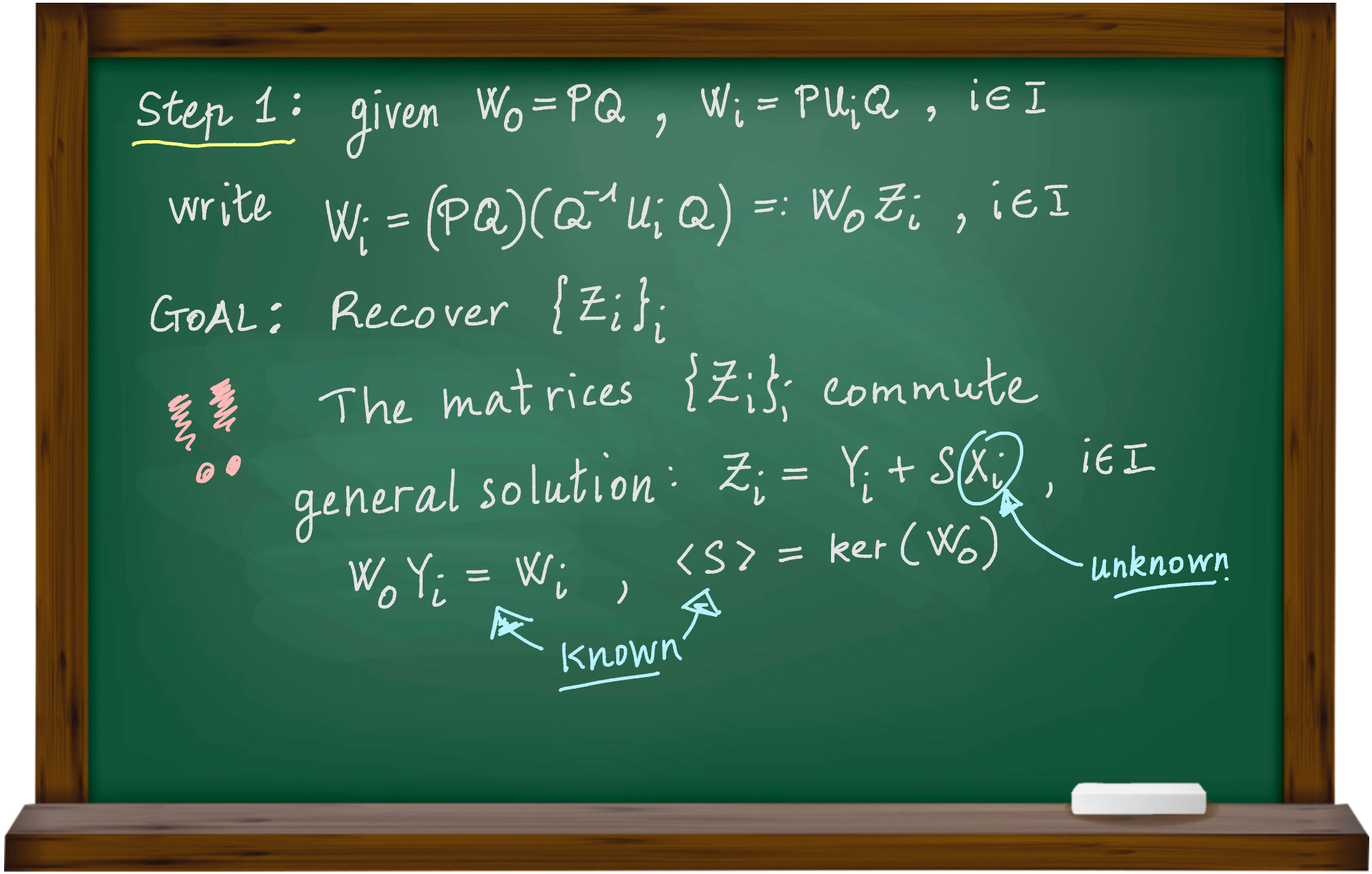
GOAL: Recover $\{Z_i\}_i$

The matrices $\{Z_i\}_i$ commute

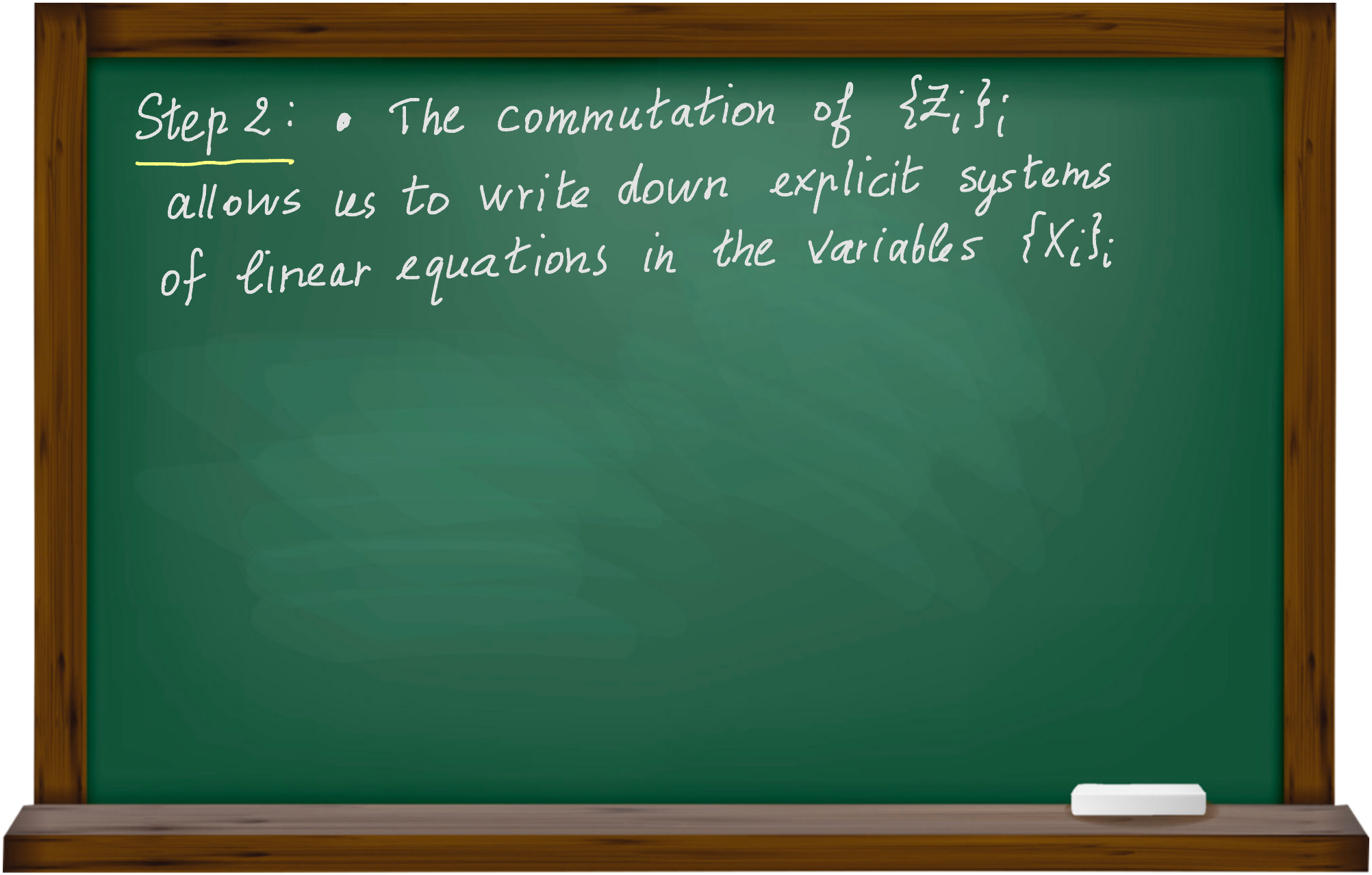
general solution: $Z_i = Y_i + SX_i$, $i \in I$

$W_0 Y_i = W_i$, $\langle S \rangle = \ker(W_0)$

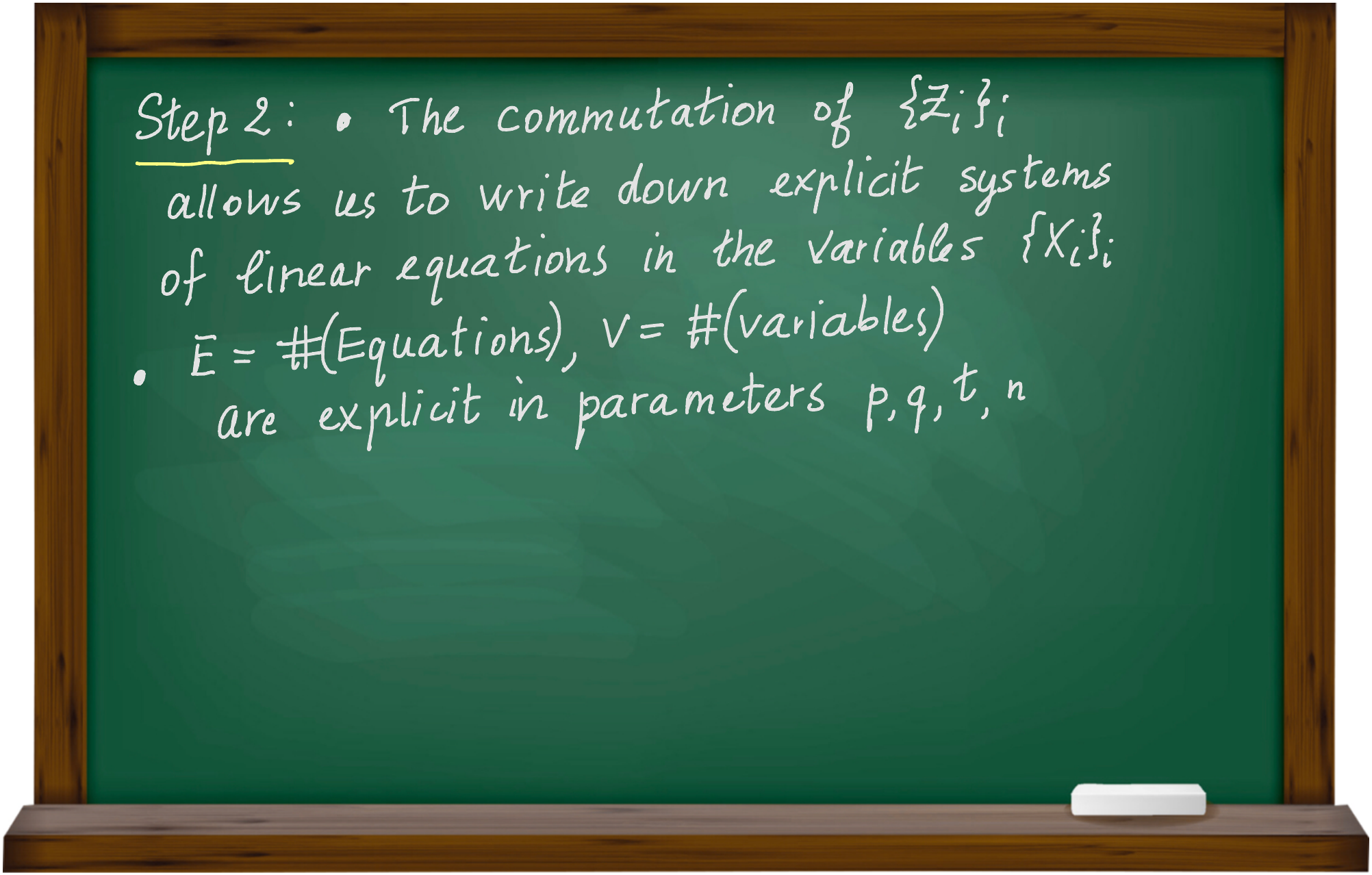
Case ($p \leq n$, $q = n$) - main tools



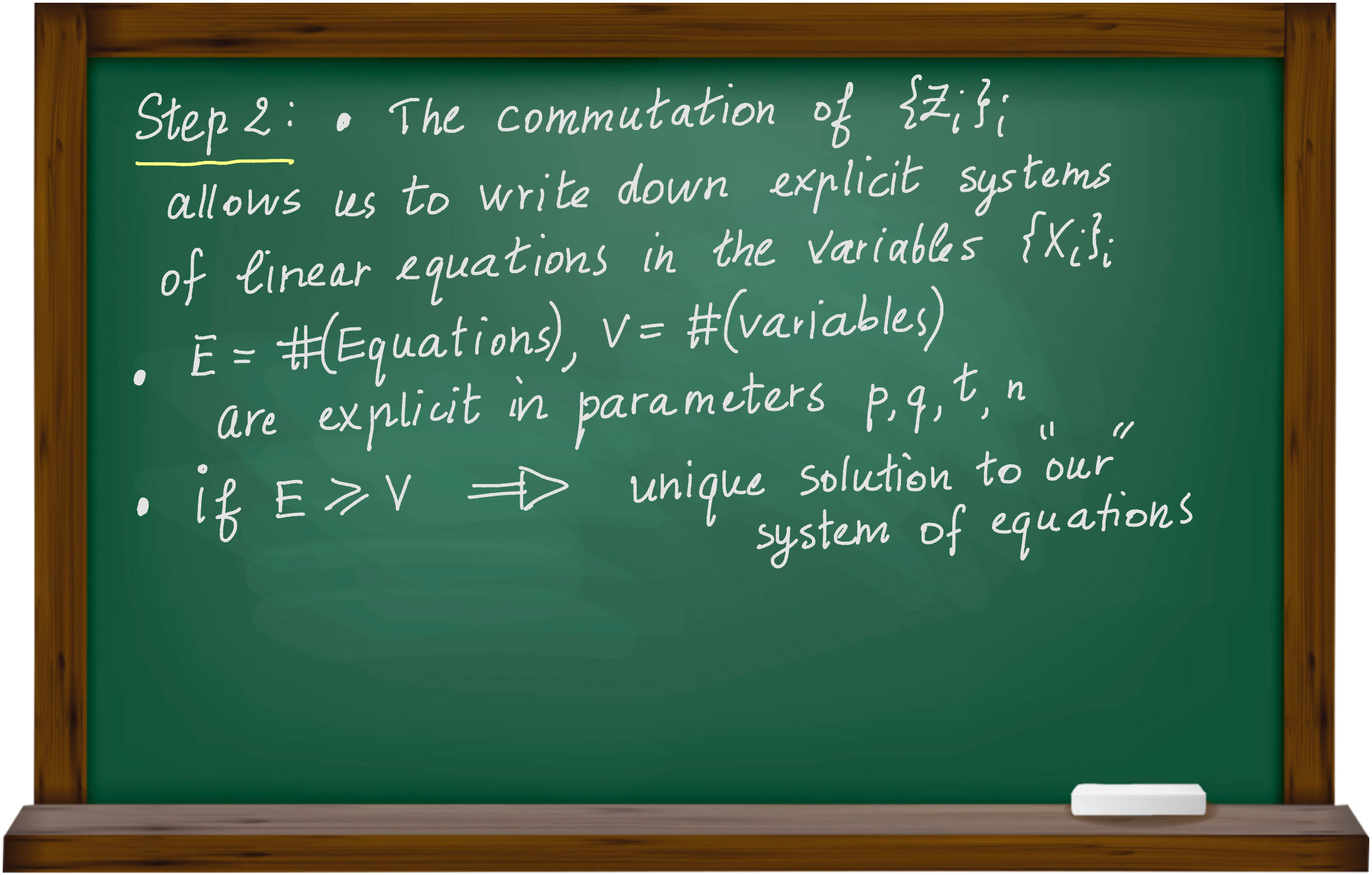
Case ($p \leq n$, $q = n$) - main tools



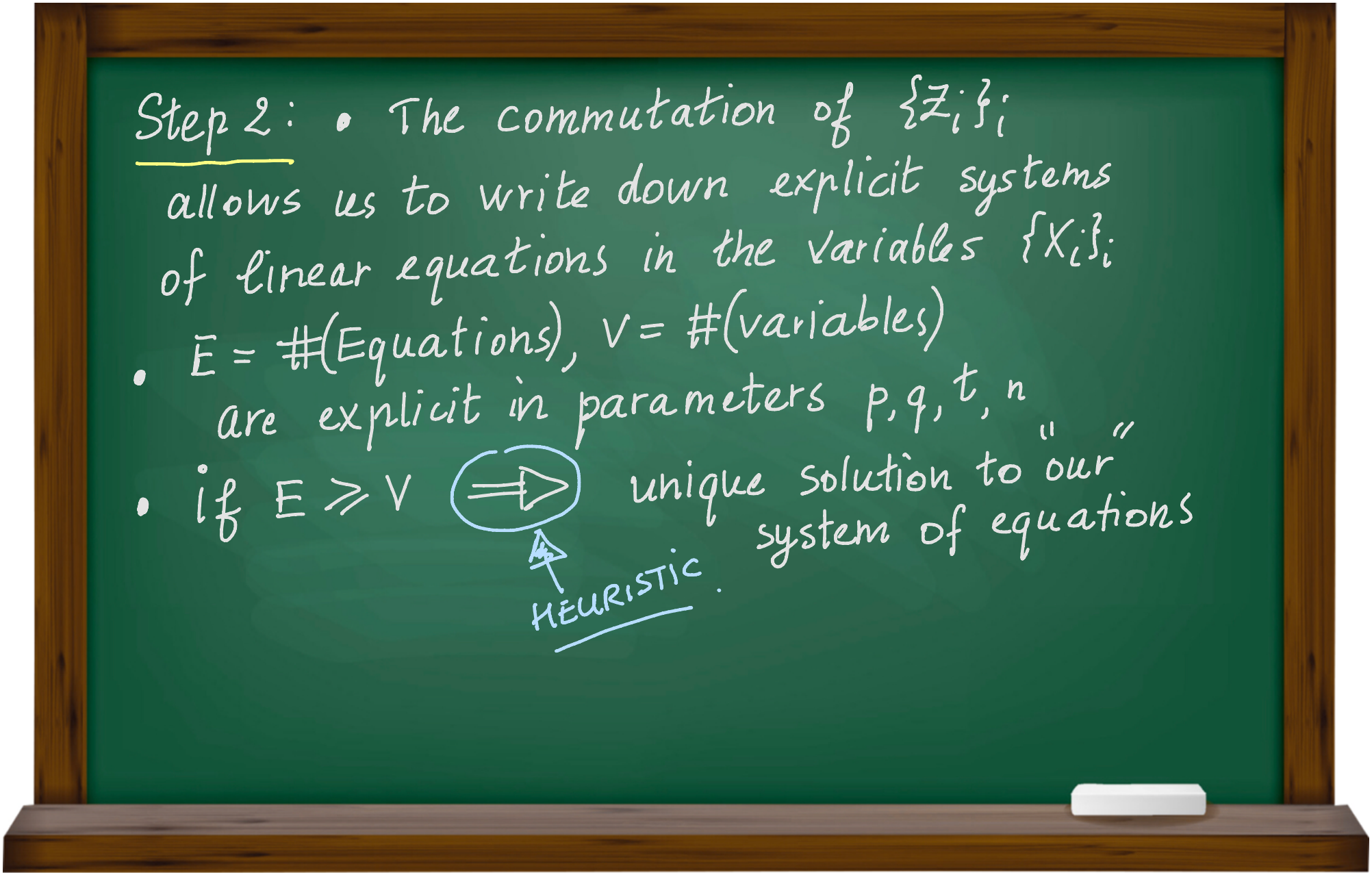
Case ($p \leq n$, $q = n$) - main tools



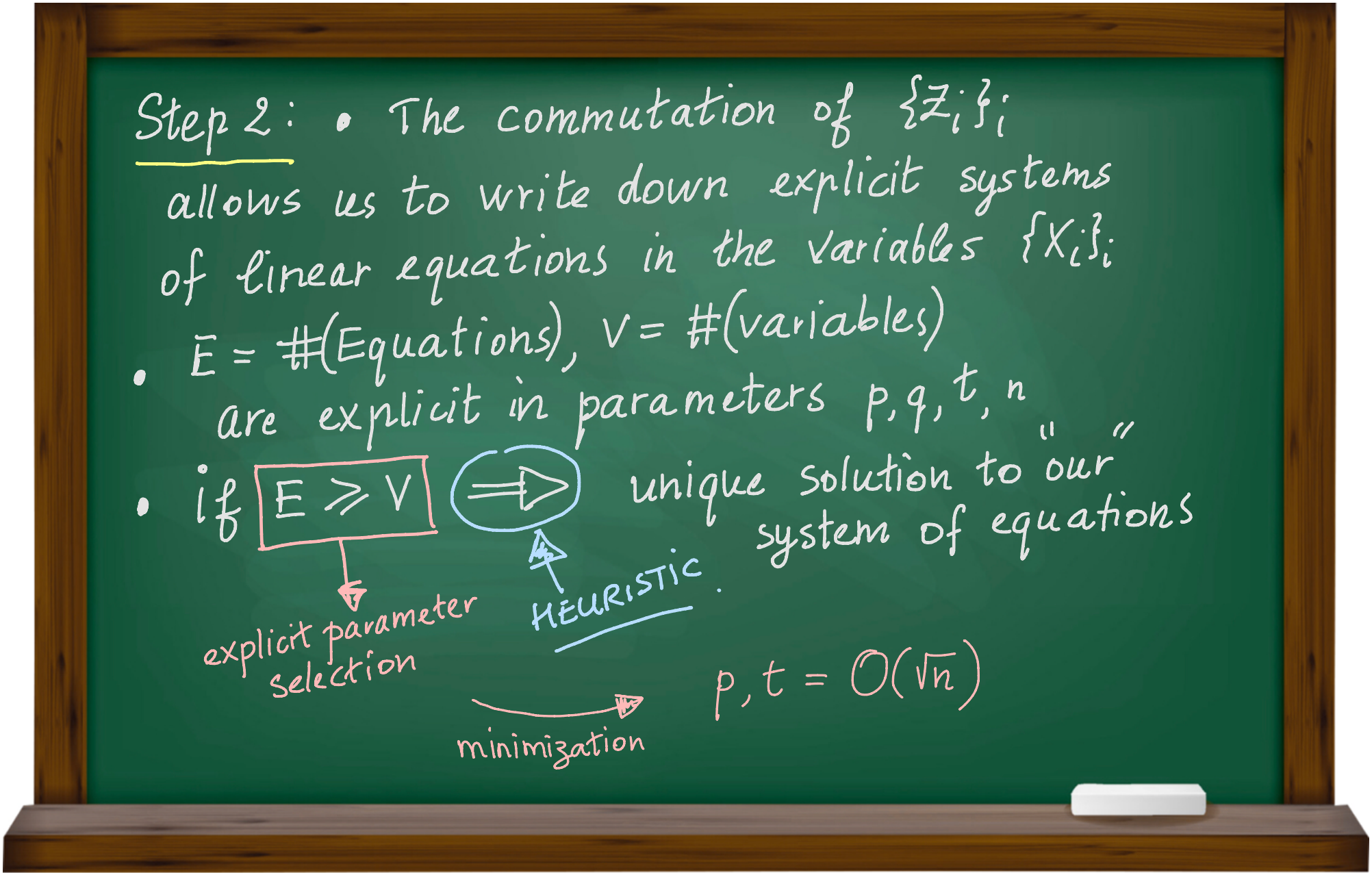
Case ($p \leq n$, $q = n$) - main tools



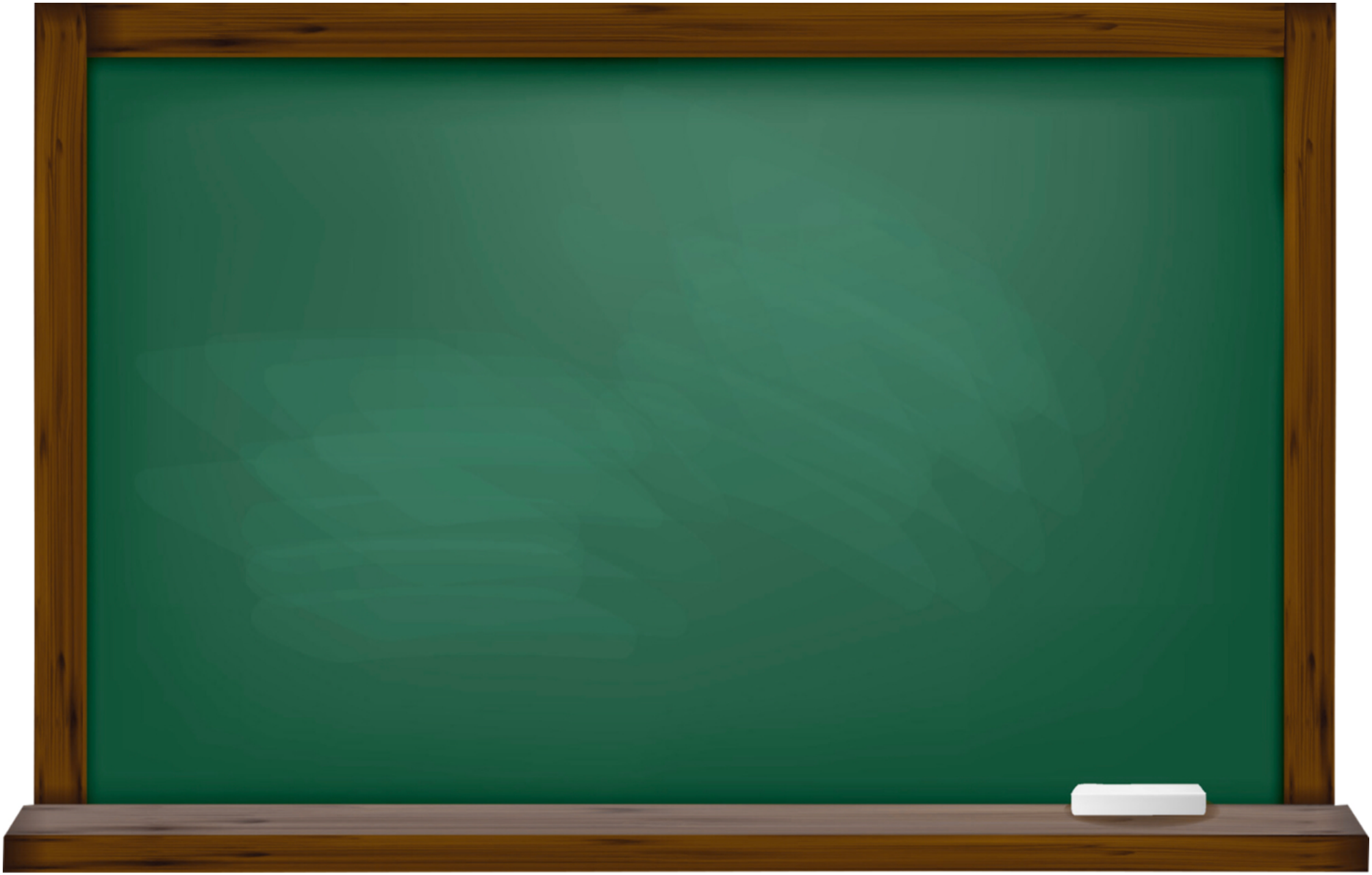
Case ($p \leq n$, $q = n$) - main tools



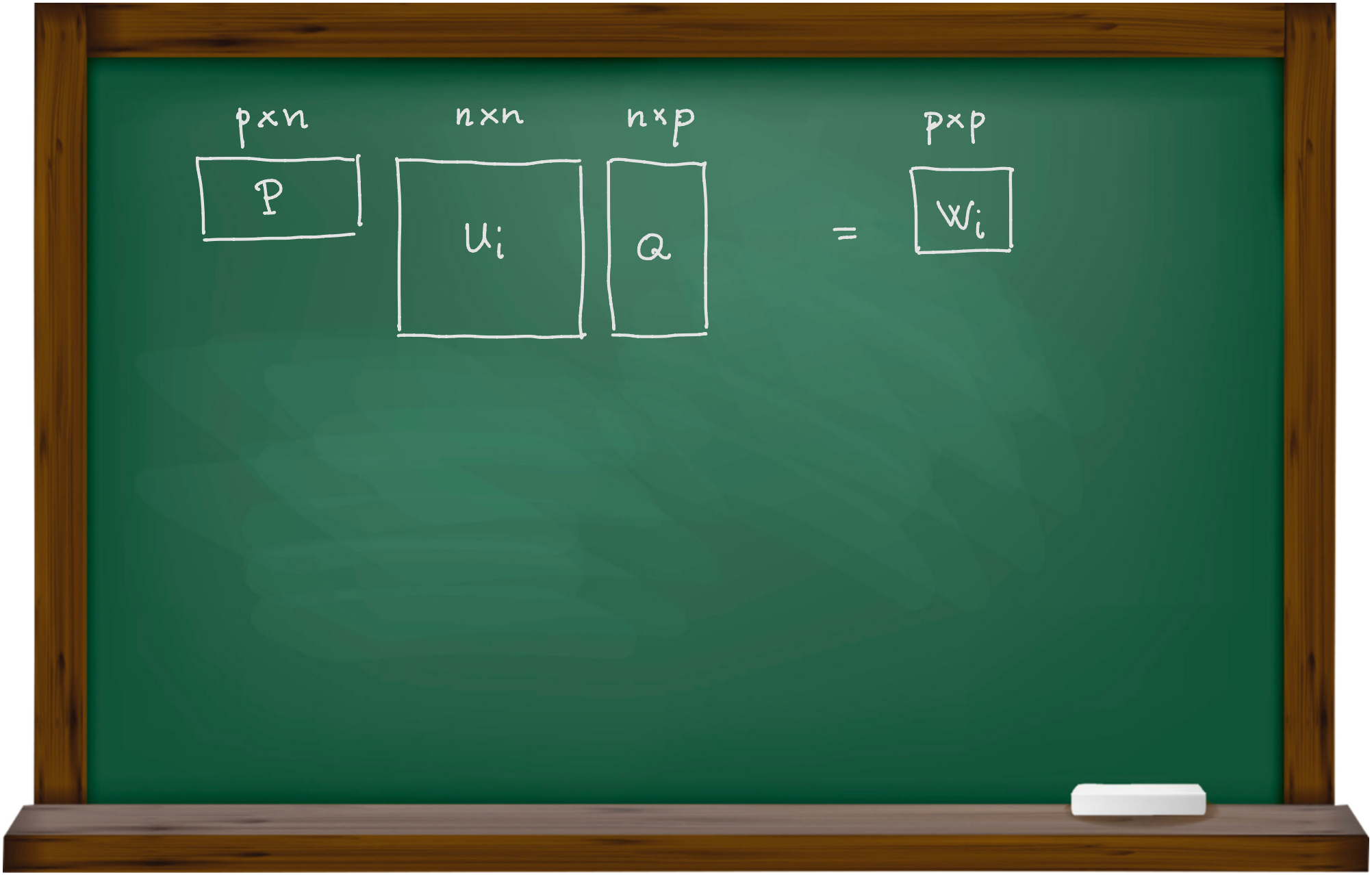
Case ($p \leq n$, $q = n$) - main tools



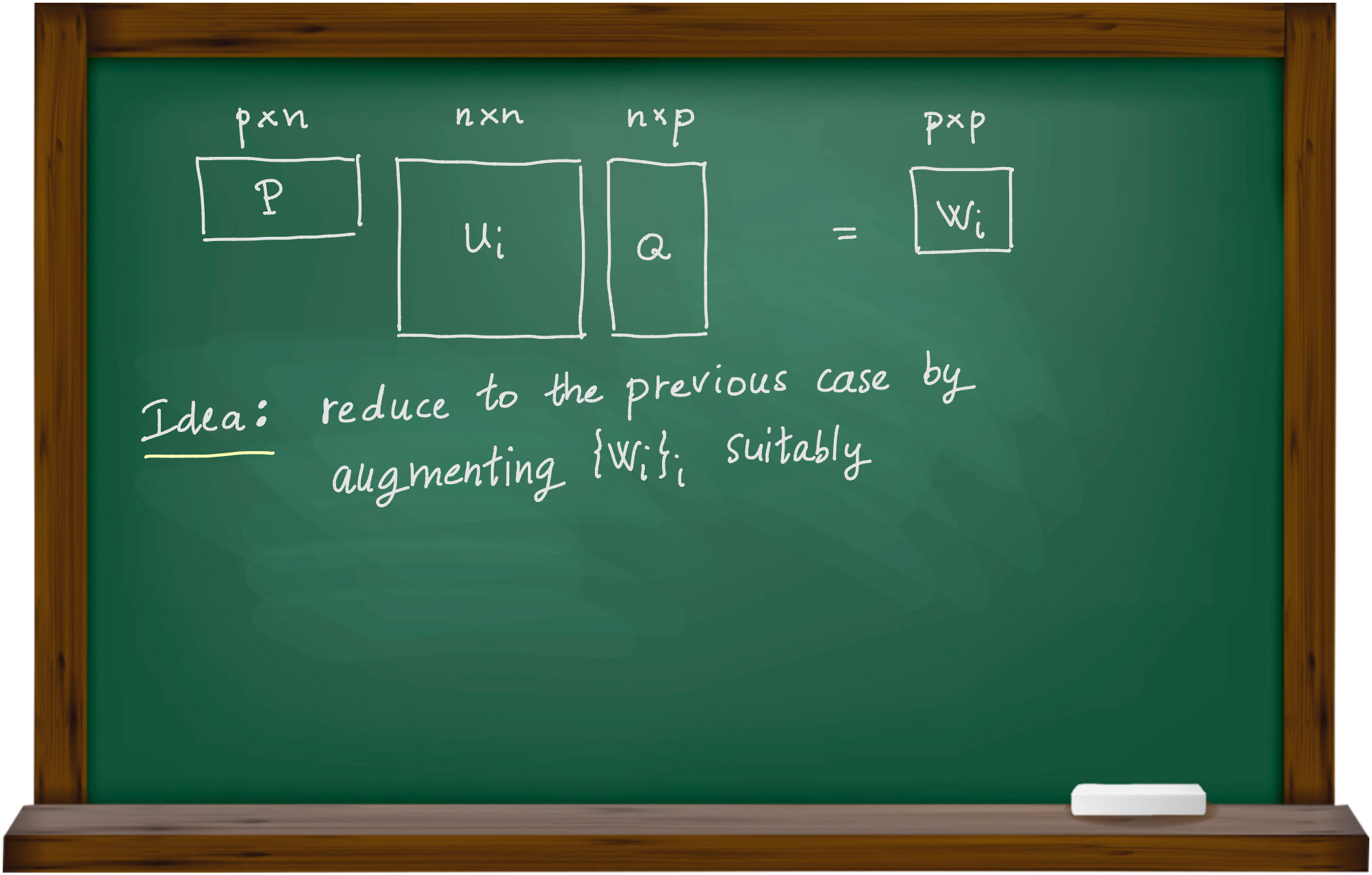
Case ($p = q \leq n$) - main tools



Case ($p = q \leq n$) - main tools

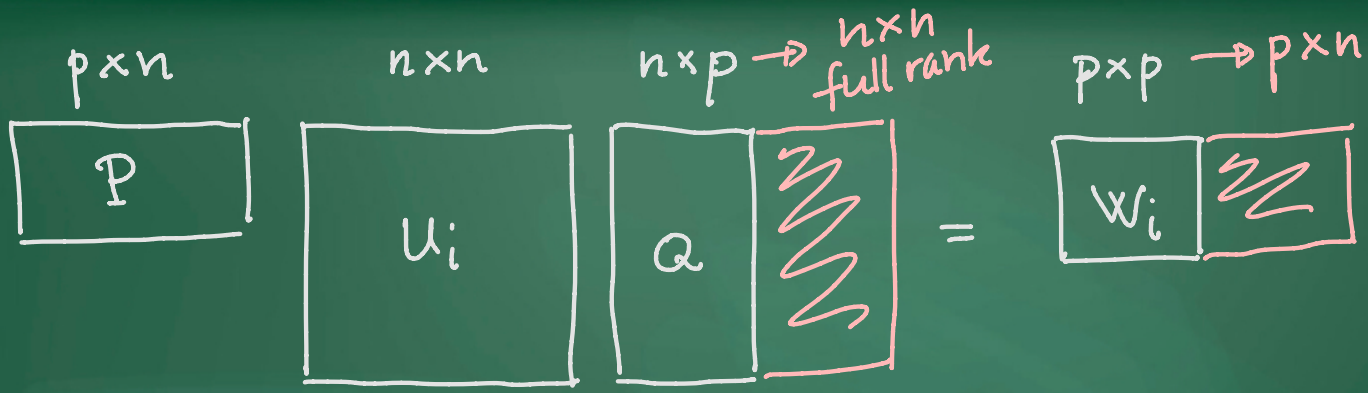


Case ($p = q \leq n$) - main tools



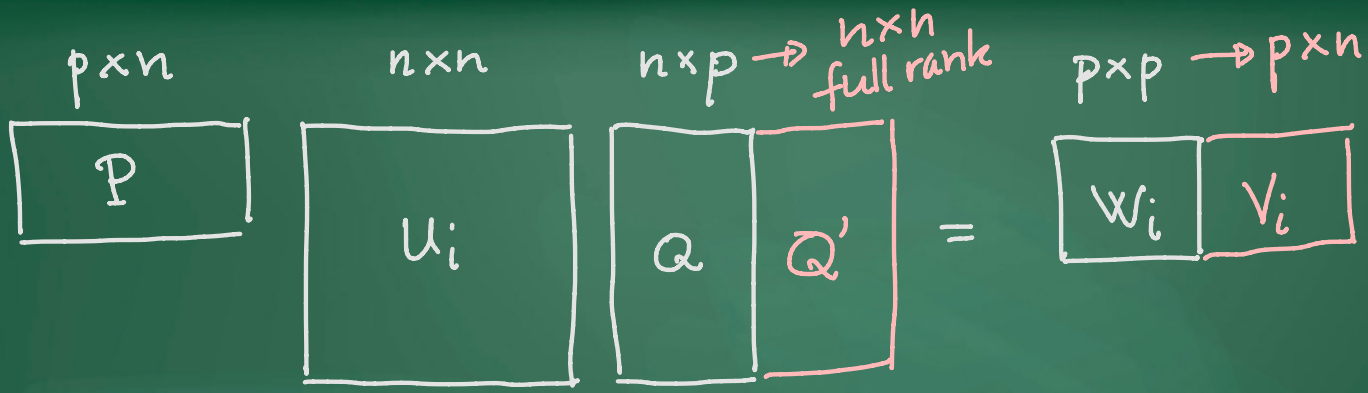
Idea: reduce to the previous case by augmenting $\{W_i\}_i$ suitably

Case ($p = q \leq n$) - main tools



Idea: reduce to the previous case by augmenting $\{W_i\}_i$ suitably

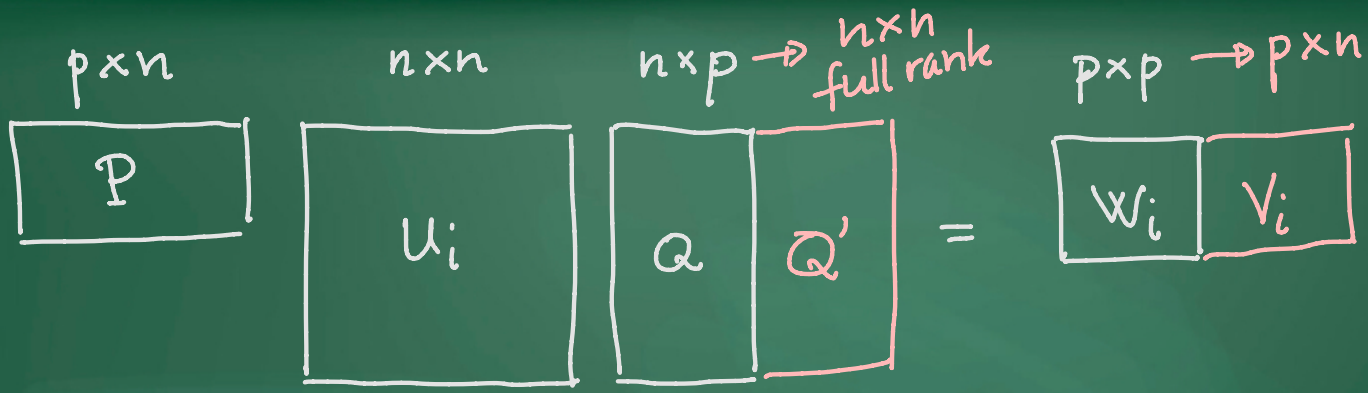
Case ($p = q \leq n$) - main tools



Idea: reduce to the previous case by augmenting $\{W_i\}_i$ suitably

- augmentation process:
  compute matrices $\{V_i\}_i$ s.t. $V_i = P\,U_i\,Q'$ $(i \in I)$
  and $(Q|Q') \in GL_n(\mathbb{Q})$

Case ($p = q \leq n$) - main tools



Idea: reduce to the previous case by augmenting $\{W_i\}_i$ suitably

- computation of $\{V_i\}_i$ needs some extra information

parameters: $p = O(\frac{2}{3}n)$

$t = O(\sqrt{n})$

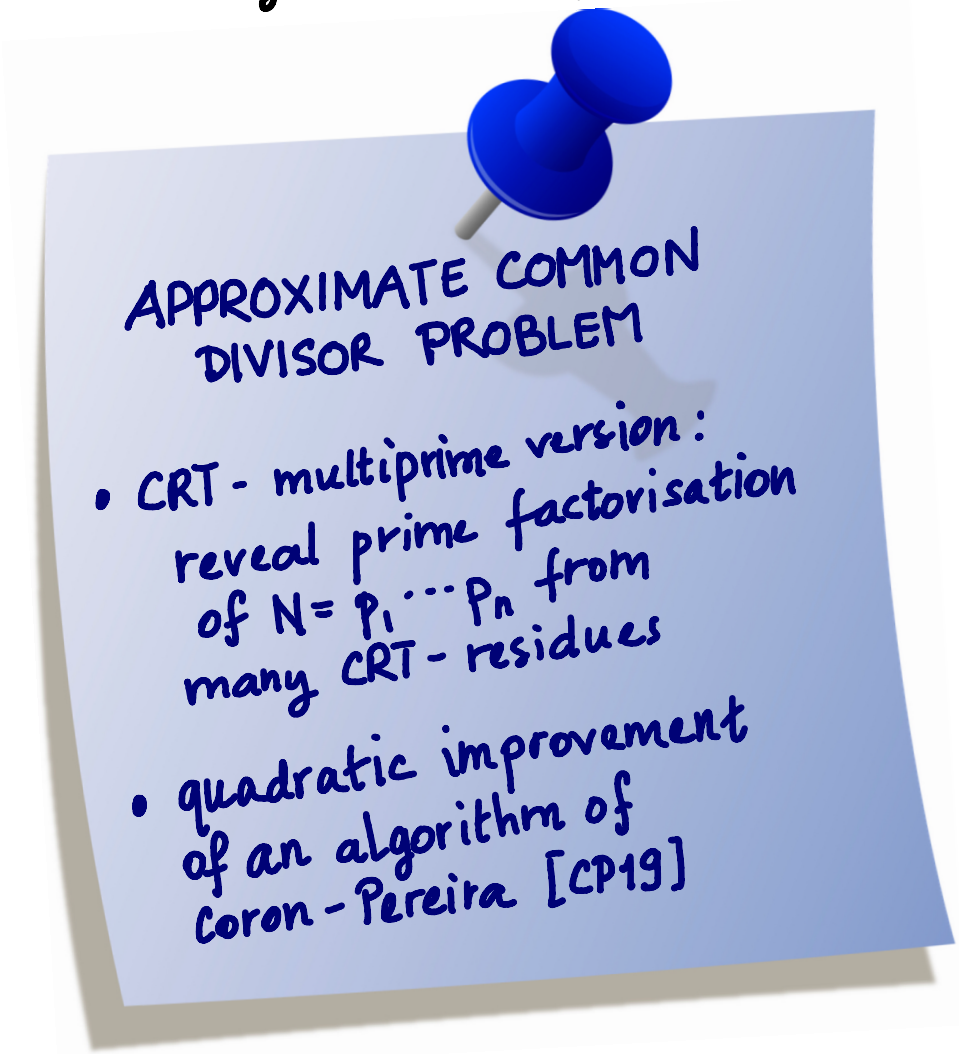
Applications

Applications

- our algorithms apply to

Applications

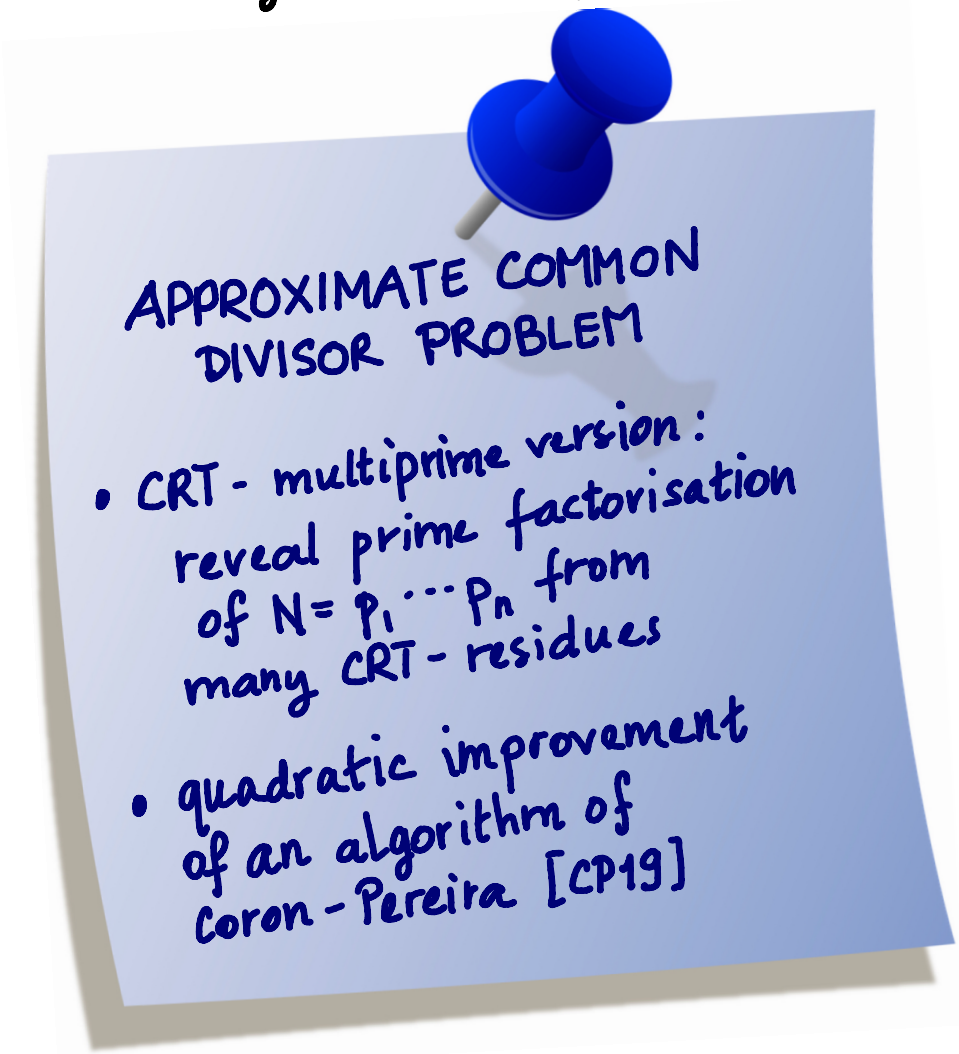
- our algorithms apply to



[CP19] J.S. Coron and H.V.L. Pereira. On Kilian's randomization of multilinear map encodings. ASIACRYPT 2019

Applications

- our algorithms apply to



[CP19] J.S. Coron and H.V.L. Pereira. On Kilian's randomization of multilinear map encodings. ASIACRYPT 2019

[CLT13] J.S. Coron, T. Lepoint, M. Tibouchi. Practical multilinear maps over the integers. CRYPTO 2013

[CHL+15] J.H. Cheon, K. Han, C. Lee, H. Ryu, D. Stehlé. Cryptanalysis of the multilinear map over the integers. EUROCRYPT 2015

CRT - ACD Problem - Main ideas

CRT-ACD Problem - Main ideas

Problem Statement

- $N = \prod_{i=1}^{n} p_i$, p_i large secret primes

  ↑ public modulus

- a finite set

  $S = \{x_1, \dots, x_{|S|}\} \subset \mathbb{Z}$

  with "small" residues mod $\{p_i\}_i$

  i.e. $x_j \equiv x_j^{(i)} \pmod{p_i}$ $\quad j = 1, \dots, |S|$

  and $|x_j^{(i)}|$ "small"

  compared to size of p_i

CRT-ACD Problem - Main ideas

Problem Statement

- $N = \prod_{i=1}^{n} p_i$, p_i: large secret primes
  - ↑ public modulus
- a finite set

  $S = \{x_1, \dots, x_{|S|}\} \subset \mathbb{Z}$

  with "small" residues mod $\{p_i\}_i$:

  i.e. $x_j \equiv x_j^{(i)} \pmod{p_i}$ $\quad j = 1, \dots, |S|$

  and $|x_j^{(i)}|$ "small"

  compared to size of p_i

→ given N and S

factor N completely

CRT-ACD Problem - Main ideas

Problem Statement

- $N = \prod_{i=1}^{n} p_i$, p_i large secret primes
  - ↑ public modulus
- a finite set

  $S = \{x_1, \dots, x_{|S|}\} \subset \mathbb{Z}$

  with "small" residues mod $\{p_i\}_i$:

  i.e. $x_j \equiv x_j^{(i)} \pmod{p_i}$ $\quad j = 1, \dots, |S|$

  and $|x_j^{(i)}|$ "small"

  compared to size of p_i

→ given N and S

factor N completely

[CP19]:

$|S| = n+1$

this work:

$|S| = 2\lceil\sqrt{2n}\rceil$

CRT-ACD Problem - Main ideas

Problem Statement

- $N = \prod_{i=1}^{n} p_i$, p_i large secret primes
  - ↑ public modulus
- a finite set

  $S = \{x_1, \dots, x_{|S|}\} \subset \mathbb{Z}$

  with "small" residues mod $\{p_i\}_i$:

  i.e. $x_j \equiv x_j^{(i)} \pmod{p_i}$ $\quad j = 1, \dots, |S|$

  and $|x_j^{(i)}|$ "small"

  compared to size of p_i

→ given N and S factor N completely

[CP19]:

$|S| = n+1$

this work:

$|S| = 2\lceil\sqrt{2n}\rceil$

quadratic improvement in input length

CRT-ACD Problem - Main ideas

Problem Statement

- $N = \prod_{i=1}^{n} p_i$, p_i large secret primes
  - ↑ public modulus
- a finite set

  $S = \{x_1, \ldots, x_{|S|}\} \subset \mathbb{Z}$

  with "small" residues mod $\{p_i\}_i$:

  i.e. $x_j \equiv x_j^{(i)} \pmod{p_i}$ $\quad j = 1, \ldots, |S|$

  and $|x_j^{(i)}|$ "small"

  compared to size of p_i

→ given N and S factor N completely

[CP19]:

$|S| = n+1$

this work:

$|S| = 2\lceil\sqrt{2n}\rceil$

quadratic improvement in input length

In practice:

e.g. $n = 50$

size$\{p_i\} \sim 2^{800}$

residue size $\sim 2^{100}$

	$\lvert S\rvert$
[CP19] :	51
this work :	19

CRT-ACD Problem - Main ideas

[CP19]

- $|S| = n+1$

this work

CRT-ACD Problem - Main ideas

[CP19]

- $|S| = n+1$
- $S = \{x_1, \dots, x_n, y\}$
  construct (public) vector

$$\underline{b} = \begin{pmatrix} \underline{x} \\ y \cdot \underline{x} \end{pmatrix} \in \mathbb{Z}^{2n}$$

$$\underline{x} = (x_1, \dots, x_n)^T \in \mathbb{Z}^n$$

this work

CRT-ACD Problem - Main ideas

[CP19]

- $|S| = n+1$
- $S = \{x_1, \dots, x_n, y\}$
  construct (public) vector

$$\underline{b} = \begin{pmatrix} \underline{x} \\ y \cdot \underline{x} \end{pmatrix} \in \mathbb{Z}^{2n}$$

$$\underline{x} = (x_1, \dots, x_n)^T \in \mathbb{Z}^n$$

this work

- $|S| = p+t$
  $p, t \in \mathbb{N}$, $p+t \leq n$

CRT-ACD Problem - Main ideas

[CP19]

- $|S| = n+1$
- $S = \{x_1, \dots, x_n, y\}$
  construct (public) vector

$$\underline{b} = \begin{pmatrix} \underline{x} \\ y \cdot \underline{x} \end{pmatrix} \in \mathbb{Z}^{2n}$$

$$\underline{x} = (x_1, \dots, x_n)^T \in \mathbb{Z}^n$$

this work

- $|S| = p+t$
  $p, t \in \mathbb{N}$, $p+t \leq n$
- $S = \{x_1, \dots, x_p, y_1, \dots, y_t\}$
  construct (public) vector

$$\underline{b} = \begin{pmatrix} \underline{x} \\ y_1 \cdot \underline{x} \\ \vdots \\ y_t \cdot \underline{x} \end{pmatrix} \in \mathbb{Z}^{p(t+1)}$$

$$\underline{x} = (x_1, \dots, x_p) \in \mathbb{Z}^p$$

CRT-ACD Problem - Main ideas

[CP19]

- $|S| = n+1$
- $S = \{x_1, \dots, x_n, y\}$

construct (public) vector

$$\underline{b} = \begin{pmatrix} \underline{x} \\ y \cdot \underline{x} \end{pmatrix} \in \mathbb{Z}^{2n}$$

$$\underline{x} = (x_1, \dots, x_n)^T \in \mathbb{Z}^n$$

"fixed" dimension

this work

- $|S| = p+t$

  $p, t \in \mathbb{N}$, $p+t \leq n$
- $S = \{x_1, \dots, x_p, y_1, \dots, y_t\}$

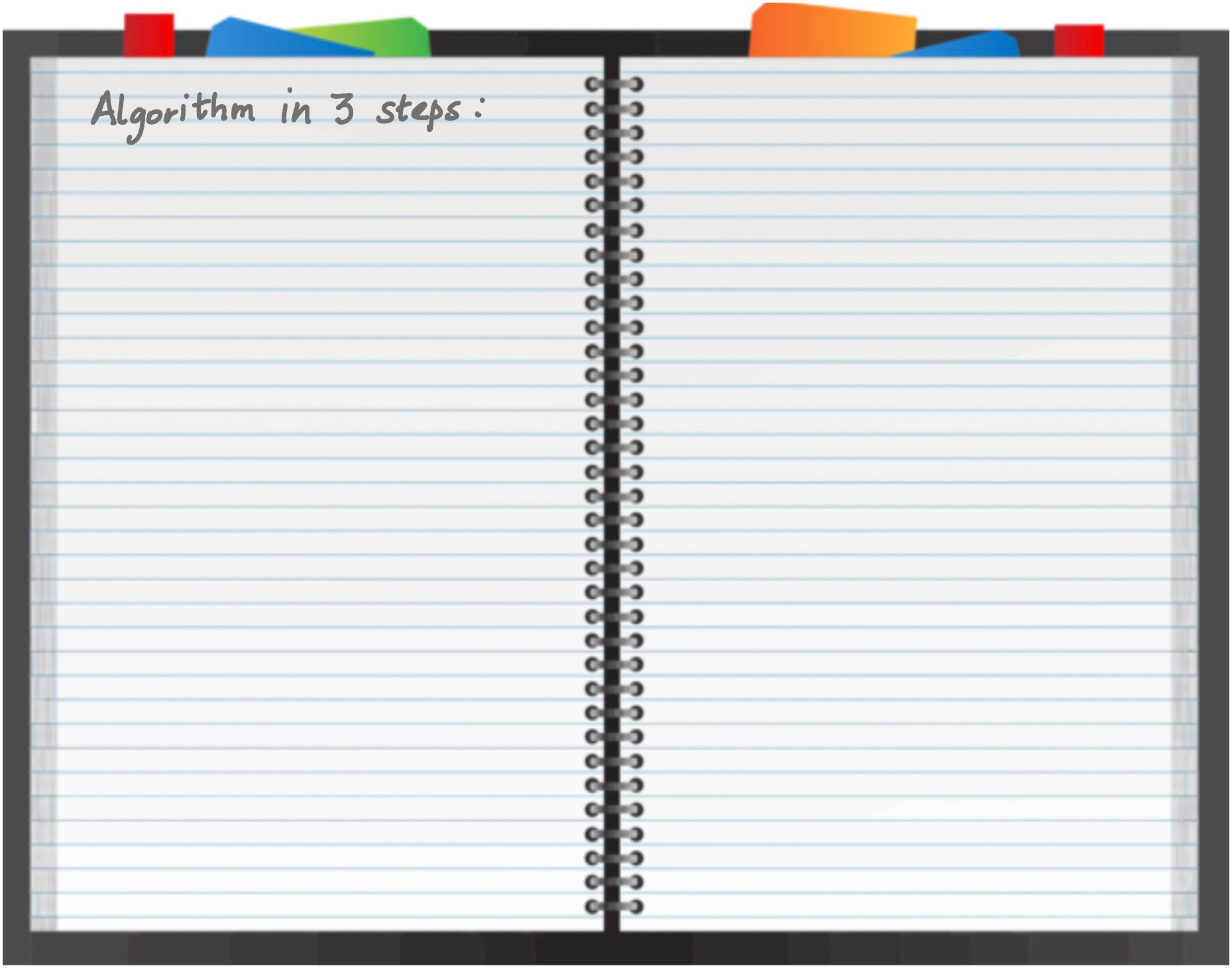
construct (public) vector

$$\underline{b} = \begin{pmatrix} \underline{x} \\ y_1 \cdot \underline{x} \\ \vdots \\ y_t \cdot \underline{x} \end{pmatrix} \in \mathbb{Z}^{p(t+1)}$$

$$\underline{x} = (x_1, \dots, x_p) \in \mathbb{Z}^p$$

"variable" dimension

CRT - ACD Problem - Main ideas



CRT-ACD Problem - Main ideas

Algorithm in 3 steps:

By CRT:

$$\underline{b} = \sum_i \alpha_i \underline{b}^i \pmod N$$

$\alpha_i \in \mathbb{Z}$

$\underline{b}^i \in \mathbb{Z}^{p(t+1)}$ "short"

$\underline{b} \equiv \underline{b}^i \pmod{p_i}$

CRT-ACD Problem - Main ideas

Algorithm in 3 steps:

By CRT:

$$\underline{b} = \sum_i \alpha_i \underline{b}^i \pmod N$$

$\alpha_i \in \mathbb{Z}$

$\underline{b}^i \in \mathbb{Z}^{p(t+1)}$ "short"

$\underline{b} \equiv \underline{b}^i \pmod{p_i}$

- **Step 1: Lattice reduction**

from $\underline{b}$ and N compute with $\underline{x} \equiv \underline{x}^i$, $y_j \equiv y_j^i \pmod{p_i}$

$$\left[\underline{b}^1 \mid \dots \mid \underline{b}^n\right] \cdot Q$$

$$= \left[\begin{array}{c|c|c} \underline{x}^1 & & \underline{x}^n \\ y_1^1 \underline{x}^1 & \cdots & y_1^n \underline{x}^n \\ \vdots & & \vdots \\ y_t^1 \underline{x}^1 & & y_t^n \underline{x}^n \end{array}\right] \cdot Q$$

for $Q \in GL_n(\mathbb{Q})$

CRT-ACD Problem - Main ideas

Algorithm in 3 steps:

By CRT:

$$\underline{b} = \sum_i \alpha_i \underline{b}^i \pmod N$$

$\alpha_i \in \mathbb{Z}$

$\underline{b}^i \in \mathbb{Z}^{p(t+1)}$ "short"

$\underline{b} \equiv \underline{b}^i \pmod{p_i}$

- **Step 1: Lattice reduction**

from $\underline{b}$ and N compute with $\underline{x} \equiv \underline{x}^i$, $y_j \equiv y_j^i \pmod{p_i}$

$$\left[\underline{b}^1 \mid \dots \mid \underline{b}^n\right] \cdot Q$$

$$= \left[\begin{array}{c|c|c} \underline{x}^1 & & \underline{x}^n \\ y_1^1 \underline{x}^1 & \cdots & y_1^n \underline{x}^n \\ \vdots & & \vdots \\ y_t^1 \underline{x}^1 & & y_t^n \underline{x}^n \end{array}\right] \cdot Q$$

for $Q \in GL_n(\mathbb{Q})$

- **Step 2: Diagonalization**

writing out components with

$$P := \left[\underline{x}^1 \mid \dots \mid \underline{x}^n\right] \in \mathbb{Z}^{p \times n}$$

$$v_i := \operatorname{diag}(y_1^i, \dots, y_n^i) \in \mathbb{Z}^{n \times n}$$

CRT-ACD Problem - Main ideas

Algorithm in 3 steps:

By CRT:

$$\underline{b} = \sum_i \alpha_i \underline{b}^i \pmod N$$

$\alpha_i \in \mathbb{Z}$

$\underline{b}^i \in \mathbb{Z}^{p(t+1)}$ "short"

$\underline{b} \equiv \underline{b}^i \pmod{p_i}$

- **Step 1: Lattice reduction**

from $\underline{b}$ and N compute

with $\underline{x} \equiv \underline{x}^i$, $y_j \equiv y_j^i \pmod{p_i}$

$$\left[\underline{b}^1 \,|\, \dots \,|\, \underline{b}^n\right] \cdot Q$$

$$= \left[\begin{array}{c|c|c} \underline{x}^1 & & \underline{x}^n \\ y_1^1 \underline{x}^1 & \cdots & y_1^n \underline{x}^n \\ \vdots & & \vdots \\ y_t^1 \underline{x}^1 & & y_t^n \underline{x}^n \end{array}\right] \cdot Q$$

for $Q \in GL_n(\mathbb{Q})$

- **Step 2: Diagonalization**

writing out components with

$$P := \left[\underline{x}^1 \,|\, \dots \,|\, \underline{x}^n\right] \in \mathbb{Z}^{p \times n}$$

$$U_i := \operatorname{diag}(y_1^i, \dots, y_n^i) \in \mathbb{Z}^{n \times n}$$

this gives public matrices

$$W_0 := P \cdot Q$$

$$W_i := P U_i Q \quad , \; i \in \{1, \dots, t\}$$

our algorithm reveals

diagonal entries of $\{U_i\}_i$

CRT-ACD Problem - Main ideas

Algorithm in 3 steps:

By CRT:

$$\underline{b} = \sum_i \alpha_i \underline{b}^i \pmod N$$

$\alpha_i \in \mathbb{Z}$

$\underline{b}^i \in \mathbb{Z}^{p(t+1)}$ "short"

$\underline{b} \equiv \underline{b}^i \pmod{p_i}$

- **Step 1: Lattice reduction**

from $\underline{b}$ and N compute

with $\underline{x} \equiv \underline{x}^i$, $y_j \equiv y_j^i \pmod{p_i}$

$$\left[\underline{b}^1 \,|\, \dots \,|\, \underline{b}^n\right] \cdot Q$$

$$= \left[\begin{array}{c|c|c} \underline{x}^1 & & \underline{x}^n \\ y_1^1 \underline{x}^1 & \cdots & y_1^n \underline{x}^n \\ \vdots & & \vdots \\ y_t^1 \underline{x}^1 & & y_t^n \underline{x}^n \end{array}\right] \cdot Q$$

for $Q \in GL_n(\mathbb{Q})$

- **Step 2: Diagonalization**

writing out components with

$$P := \left[\underline{x}^1 \,|\, \dots \,|\, \underline{x}^n\right] \in \mathbb{Z}^{p \times n}$$

$$U_i := \operatorname{diag}(y_1^i, \dots, y_n^i) \in \mathbb{Z}^{n \times n}$$

this gives public matrices

$$W_0 := P \cdot Q$$

$$W_i := P U_i Q \quad , \; i \in \{1, \dots, t\}$$

our algorithm reveals

diagonal entries of $\{U_i\}_i$

- **Step 3: gcd computation**

gcd computation reveals

prime factors $\{p_i\}_i$ of N

CRT-ACD Problem - Main ideas

Algorithm in 3 steps:

By CRT:

$$\underline{b} = \sum_i \alpha_i \underline{b}^i \pmod N$$

$\alpha_i \in \mathbb{Z}$

$\underline{b}^i \in \mathbb{Z}^{p(t+1)}$ "short"

$\underline{b} \equiv \underline{b}^i \pmod{p_i}$

- **Step 1: Lattice reduction**

from $\underline{b}$ and N compute

with $\underline{x} \equiv \underline{x}^i$, $y_j \equiv y_j^i \pmod{p_i}$

$$\left[\underline{b}^1 \,|\, \dots \,|\, \underline{b}^n\right] \cdot Q$$

$$= \left[\begin{array}{c|c|c} \underline{x}^1 & & \underline{x}^n \\ y_1^1 \underline{x}^1 & \cdots & y_1^n \underline{x}^n \\ \vdots & & \vdots \\ y_t^1 \underline{x}^1 & & y_t^n \underline{x}^n \end{array}\right] \cdot Q$$

for $Q \in GL_n(\mathbb{Q})$

- **Step 2: Diagonalization**

writing out components with

$$P := \left[\underline{x}^1 \,|\, \dots \,|\, \underline{x}^n\right] \in \mathbb{Z}^{p \times n}$$

$$U_i := \mathrm{diag}(y_1^i, \dots, y_n^i) \in \mathbb{Z}^{n \times n}$$

this gives public matrices

$$W_0 := P \cdot Q$$

$$W_i := P U_i Q \quad , \; i \in \{1, \dots, t\}$$

* our algorithm reveals diagonal entries of $\{U_i\}_i$

- **Step 3: gcd computation**

gcd computation reveals prime factors $\{p_i\}_i$ of N

* with $p, t = O(\sqrt{n})$

$|S| = p + t$

Cryptanalysis of CLT13-maps

Cryptanalysis of CLT13-maps

- CLT13: construction of approximate multilinear maps based on graded encoding schemes

  hardness: factorization and CRT-representations

Cryptanalysis of CLT13-maps

- CLT13: construction of approximate multilinear maps based on graded encoding schemes

  hardness: factorization and CRT-representations

  applications: multiparty DH-key exchange, program obfuscation, ...

Cryptanalysis of CLT13-maps

- CLT13: construction of approximate multilinear maps based on graded encoding schemes

  hardness: factorization and CRT-representations

  applications: multiparty DH-key exchange, program obfuscation, ...

- [CHLT15]: total break of the DH-key exchange on CLT13 with $O(n)$ encodings of zero

Cryptanalysis of CLT13-maps

- CLT13: construction of approximate multilinear maps based on graded encoding schemes

  hardness: factorization and CRT-representations

  applications: multiparty DH-key exchange, program obfuscation, ...

- [CHLT15]: total break of the DH-key exchange on CLT13 with $O(n)$ encodings of zero

  ↳ this work: $O(\sqrt{n})$

Cryptanalysis of CLT13-maps

- CLT13: construction of approximate multilinear maps based on graded encoding schemes

  hardness: factorization and CRT-representations

  applications: multiparty DH-key exchange, program obfuscation, ...

- [CHLT15]: total break of the DH-key exchange on CLT13 with $O(n)$ encodings of zero
  ↳ this work: $O(\sqrt{n})$

- impact: enough low-level encodings (of zero) are not always available to the attacker
  → cryptanalysis with limited number of encodings

Cryptanalysis of CLT13-maps

- CLT13: construction of approximate multilinear maps based on graded encoding schemes

  hardness: factorization and CRT-representations

  applications: multiparty DH-key exchange, program obfuscation, ...

- [CHLT15]: total break of the DH-key exchange on CLT13 with $O(n)$ encodings of zero
  ↳ this work: $O(\sqrt{n})$

- impact: enough low-level encodings (of zero) are not always available to the attacker
  → cryptanalysis with limited number of encodings

- idea: "variable" vs "fixed" number of encodings

Thank you for your attention