Polynomial XL: A Variant of the XL Algorithm Using Macaulay Matrices over Polynomial Rings

Hiroki Furue [1] , Momonari Kudo [2]

1. NTT Social Informatics Laboratories, Japan
2. Fukuoka Institute of Technology, Japan

2024/6/12

PQCrypto 2024

Our Contributions

MQ problem：solving quadratic system over finite fields

XL algorithm

・solving the MQ problem

・applying Gaussian elimination on coefficient matrices

Hybrid XL (h-XL)：an efficient variant of XL

Main Result

We proposed an efficient variant of Hybrid XL.

(with matrices over polynomial rings)

Outline

- **XL Algorithm・Hybrid Approach**
- Proposed Algorithm
- Conclusions

MQ Problem

Solving quadratic systems over $\mathbb{F}_q$

($\mathbb{F}_q$: the finite field with q elements)

- q : the number of elements of the finite field
- n : the number of variables
- m : the number of equations

MQ (Multivariate Quadratic equations) Problem

Given $\mathcal{F} = (f_1, \dots, f_m) \in \mathbb{F}_q[x_1, \dots, x_n]^m$ with $\deg f_i = 2$,

find *one* solution $(a_1, \dots, a_n) \in \mathbb{F}_q^n$ to $\mathcal{F}(x_1, \dots, x_n) = \mathbf{0} \in \mathbb{F}_q^m$.

Cryptosystems based on the MQ problem (e.g., UOV) are candidates for **post-quantum cryptosystems** (PQC).

Macaulay Matrices

$F := (f_1, \ldots, f_m)$: an ordered set of polynomials

$T := (t_1, \ldots, t_n)$: an ordered set of monomials

$\mathrm{coeff}(f_i, t_j)$: the coefficient of t_j in f_i

Macaulay Matrices

$$\mathrm{Mac}(F, T) := \begin{pmatrix} \mathrm{coeff}(f_1, t_1) & \cdots & \mathrm{coeff}(f_1, t_n) \\ \vdots & \ddots & \vdots \\ \mathrm{coeff}(f_m, t_1) & \cdots & \mathrm{coeff}(f_m, t_n) \end{pmatrix}$$

(We generally use the lexicographic or graded lexicographic order for T)

(Ex) lexicographic order

$$x^2yz,\ x^3,\ xy^3\ \ (x > y > z) \Rightarrow x^3 > x^2yz > xy^3$$

XL Algorithm

Input

- $\mathcal{F} = (f_1, \ldots, f_m) \in \mathbb{F}_q[x_1, \ldots, x_n]^m$ with $\deg f_i = 2$
- $D \in \mathbb{Z}_{\geq 2}$: a parameter for the degree of monomials

- $T_{\leq d} \coloneqq \left\{ x_1^{\alpha_1} \cdots x_n^{\alpha_n} \,\middle|\, \sum_{i=1}^{n} \alpha_i \leq d \right\}$

(We use a monomial order such that $x_n^D, \ldots, x_n, 1$ are listed last.)

- $I_{\leq d}$: the set of products of monomials with degree $\leq d-2$ and $f_1, \ldots, f_m$

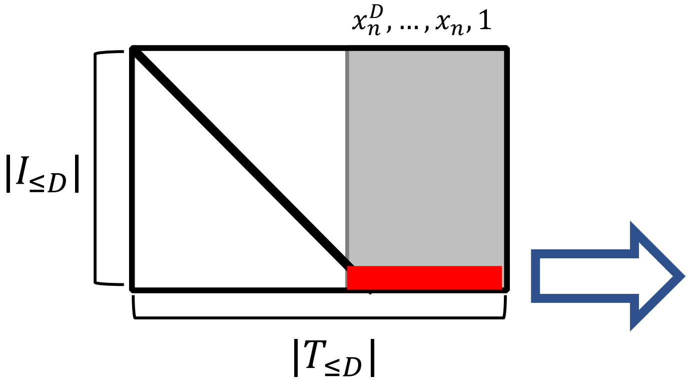
$$I_{\leq d} \coloneqq \bigcup_{i=1}^{m} \{ t \cdot f_i \mid t \in T_{\leq (d-2)} \}$$

(with any monomial order)

XL Algorithm

[Courtois et al., EUROCRYPT 2000]

1. **Multply** ： Generate $\mathrm{Mac}(I_{\leq D}, T_{\leq D})$.
2. **Linearize** ： Perform Gaussian elimination on $\mathrm{Mac}(I_{\leq D}, T_{\leq D})$.



This equation includes only the last $|T_{\leq D}| - \mathrm{rank}\big(\mathrm{Mac}(I_{\leq D}, T_{\leq D})\big) + 1$ monomials.

3. **Solve** ： Solve the univariate equation obtained in step 2 and then find the values of the other variables.

※ We have to choose D such that a univariate equation is found in step2.

Hybrid Approach

[Yang et al., ICICS 2004]
[Bettale et al., J. Math. Cryptol., 2009]

approach for using MQ solvers such as XL more efficiently

Given $f_i(x_1, \ldots, x_n)$ $(1 \le i \le m)$, $k \in \{1, \ldots, n\}$

① Choose $a_1, \ldots, a_k \in \mathbb{F}_q$ randomly.

② Solve

$$f_1(a_1, \ldots, a_k, x_{k+1}, \ldots, x_n) = \cdots$$
$$= f_m(a_1, \ldots, a_k, x_{k+1}, \ldots, x_n) = 0$$

for $x_{k+1}, \ldots, x_n$.

Repeat ①, ② until we find a solution.

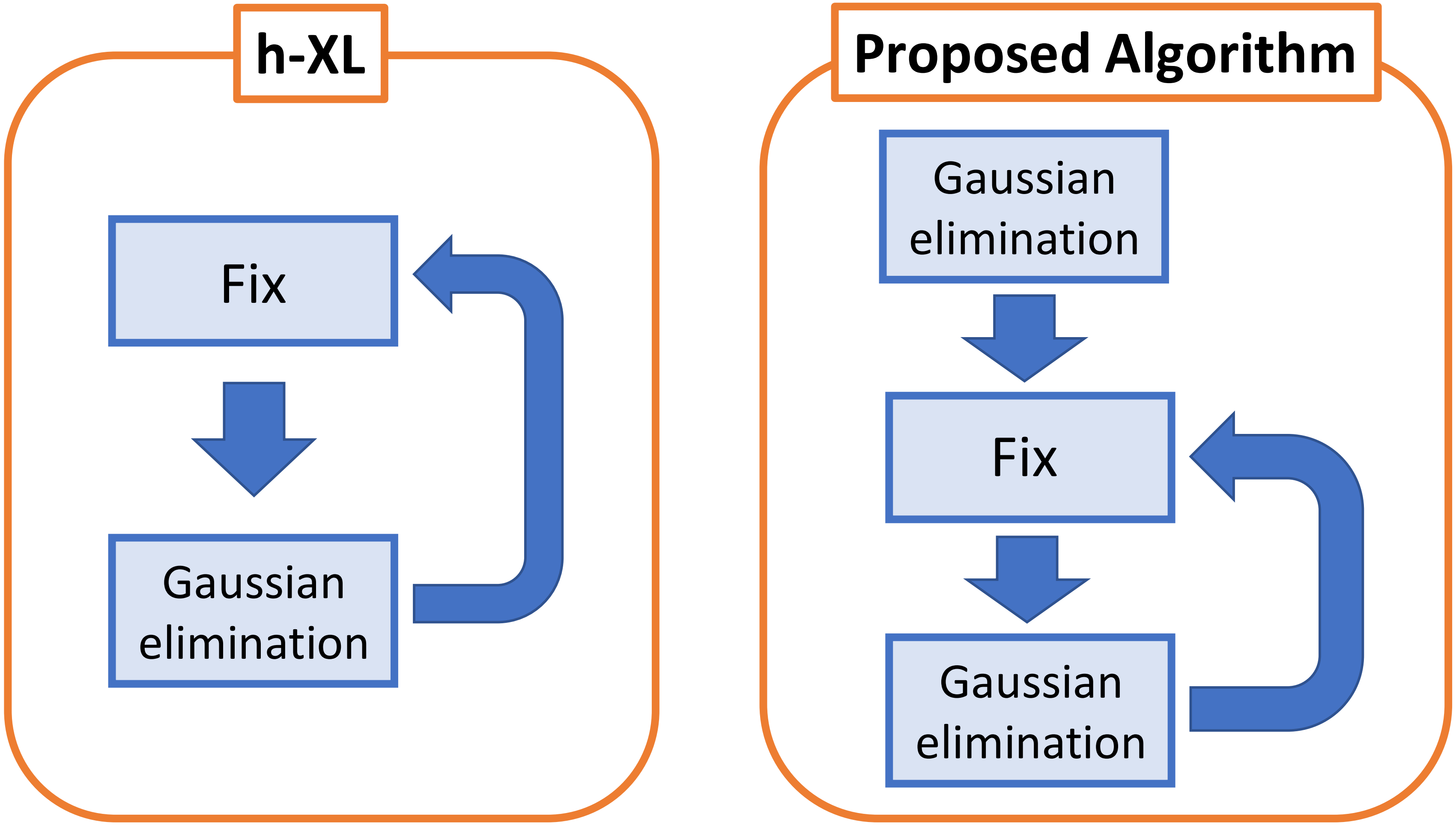
$O(q^k)$ times iteration

- **h-XL** : hybrid approach with XL

Outline

- XL Algorithm ・ Hybrid Approach
- **Proposed Algorithm**
- Conclusions

Main Idea



The amount of operations required for each guessed value can be reduced.

Main Idea

($x_1, \dots, x_k$: variables to be fixed)

$$\mathbb{F}_q[x_1, \dots, x_n] \to \left(\mathbb{F}_q[x_1, \dots, x_k]\right)[x_{k+1}, \dots, x_n]$$

Ex) $q = 7\ (\mathbb{F}_7), n = 3, m = 3, k = 1$

$\mathbb{F}_7[x, y, z]$

$$f_1 = 5x^2 + 6xy + 4xz + yz + 5z^2 + 4x + 5y + 3$$
$$f_2 = 4x^2 + 5xy + 4xz + 3y^2 + 5yz + z^2 + 6x + 2y + 3z + 2$$
$$f_3 = 2x^2 + 4xy + 2y^2 + 6z^2 + 6x + y + 3z + 2$$

$(\mathbb{F}_7[x])[y, z]$

$$f_1 = yz + 5z^2 + (6x + 5)y + 4xz + (5x^2 + 4x + 3)$$
$$f_2 = 3y^2 + 5yz + z^2 + (5x + 2)y + (4x + 3)z + (4x^2 + 6x + 2)$$
$$f_3 = 2y^2 + 6z^2 + (4x + 1)y + 3z + (2x^2 + 6x + 2)$$

Generate Macaulay matrices ※with the graded lex order $y > z$

Main Idea

	y^4	y^3z	y^2z^2	yz^3	z^4	y^3	y^2z	yz^2	z^3	y^2	yz	z^2	y	z	1
y^2f_1		1	5			$6x+5$	$4x$			$5x^2+4x+3$					
y^2f_2	3	5	1			$5x+2$	$4x+3$			$4x^2+6x+2$					
y^2f_3	2		6			$4x+1$	3			$2x$					
yzf_1			1	5			$6x+5$	$4x$							
yzf_2		3	5	1			$5x+2$	$4x+3$							
yzf_3		2		6			$4x+1$	3			$2x^2+6x+2$				
z^2f_1				1	5			$6x+5$	$4x$			$5x^2+4x+3$			
z^2f_2			3	5	1			$5x+2$	$4x+3$						
z^2f_3			2		6			$4x+1$	3						
yf_1							1	5		$6x+5$					
yf_2						3	5	1		$5x+2$	$4x+3$		$4x^2+6x+2$		
yf_3						2		6		$4x+1$	3		$2x^2+6x+2$		
zf_1								1	5		$6x+5$	$4x$		$5x^2+4x+3$	
zf_2							3	5	1		$5x+2$	$4x+3$		$4x^2+6x+2$	
zf_3							2		6		$4x+1$	3		$2x^2+6x+2$	
f_1											1	5	$6x+5$	$4x$	$5x^2+4x+3$
f_2										3	5	1	$5x+2$	$4x+3$	$4x^2+6x+2$
f_3										2		6	$4x+1$	3	$2x^2+6x+2$

We cannot perform the Gaussian elimination over the polynomial ring.

Perform the elimination on submatrices over the finite field.

Preliminaries

Input

- $\mathcal{F} = (f_1, \dots, f_m) \in \mathbb{F}_q[x_1, \dots, x_n]^m$ with $\deg f_i = 2$
- $D \in \mathbb{Z}_{\geq 2}, k \in \mathbb{Z}_{\geq 1}$: parameters

- $x_1, \dots, x_k$: fixed variables
- $T_d \coloneqq \left\{ x_{k+1}^{\alpha_{k+1}} \cdots x_n^{\alpha_n} \middle| \sum_{i=k+1}^{n} \alpha_i = d \right\}$
- $T_{\leq d} \coloneqq T_0 \cup \cdots \cup T_d$
- $I_d \coloneqq \{ t \cdot f_i | 1 \leq i \leq m \text{ and } t \in T_{d-2} \}$
- $I_{\leq d} \coloneqq I_2 \cup \cdots \cup I_d$

Preliminaries

- $\mathcal{PM}$: $\mathrm{Mac}(I_{\leq D}, T_{\leq D})$ over $\mathbb{F}_q[x_1, \dots, x_k]$ (graded lex order)
- $\mathcal{PM}[I_a, T_b]$: the submatrix of $\mathcal{PM}$ corresponding to I_a, T_b

	y^4	y^3z	y^2z^2	yz^3	z^4	y^3	y^2z	yz^2	z^3	y^2	yz	z^2	y	z	1
y^2f_1		1	5			$6x+5$	$4x$			$5x^2+4x+3$					
y^2f_2	3	5	1			$5x+2$	$4x+3$			$4x^2+6x+2$					
y^2f_3	2		6			$4x+1$	3			$2x^2+6x+2$					
yzf_1			1	5			$6x+5$	$4x$			$5x^2+4x+3$				
yzf_2															
yzf_3		2		6			$4x+1$	3			$2x^2+6x+2$				
z^2f_1				1	5			$6x+5$	$4x$			$5x^2+4x+3$			
z^2f_2			3	5	1			$5x+2$	$4x+3$			$4x^2+6x+2$			
z^2f_3			2		6			$4x+1$	3			$2x^2+6x+2$			
yf_1							1	5		$6x+5$	$4x$		$5x^2+4x+3$		
yf_2						3	5	1		$5x+2$	$4x+3$		$4x^2+6x+2$		
yf_3						2				$4x+1$					
zf_1									5			$4x$		$4x$	
zf_2							3	5	1		$5x+2$	$4x+3$		$4x^2+6x+2$	
zf_3							2		6		$4x+1$	3		$2x^2+6x+2$	
f_1											1	5	$6x+5$	$4x$	$5x^2+4x+3$
f_2										3		1		$+3$	
f_3										2		6	$4x+1$	3	$2x^2+6x+2$

$\mathcal{PM}[I_4, T_4]$ $\mathcal{PM}[I_4, T_3]$ $\mathcal{PM}[I_4, T_2]$ $\mathcal{PM}[I_4, T_1]$ $\mathcal{PM}[I_4, T_0]$

$\mathcal{PM}[I_3, T_4]$ $\mathcal{PM}[I_3, T_3]$ $\mathcal{PM}[I_3, T_2]$ $\mathcal{PM}[I_3, T_1]$ $\mathcal{PM}[I_3, T_0]$

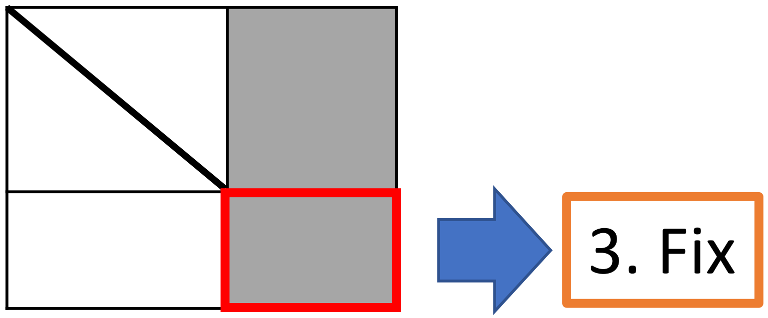
$\mathcal{PM}[I_2, T_4]$ $\mathcal{PM}[I_2, T_3]$ $\mathcal{PM}[I_2, T_2]$ $\mathcal{PM}[I_2, T_1]$ $\mathcal{PM}[I_2, T_0]$

Polynomial XL (PXL)

1. **Multiply**：Generate $\mathcal{PM}$.

2. **Linearize(1)**：

(partial Gaussian elimination)



3. **Fix**：Fix the values of k variables.

4. **Linearize(2)**：Perform Gaussian elimination on the matrix obtained in step 3.

5. **Solve**：Solve the univariate equation obtained in step 4 and then find the values of the other variables.

※ Repeat step 3-5 until we find a solution.

Details of Linearize(1)

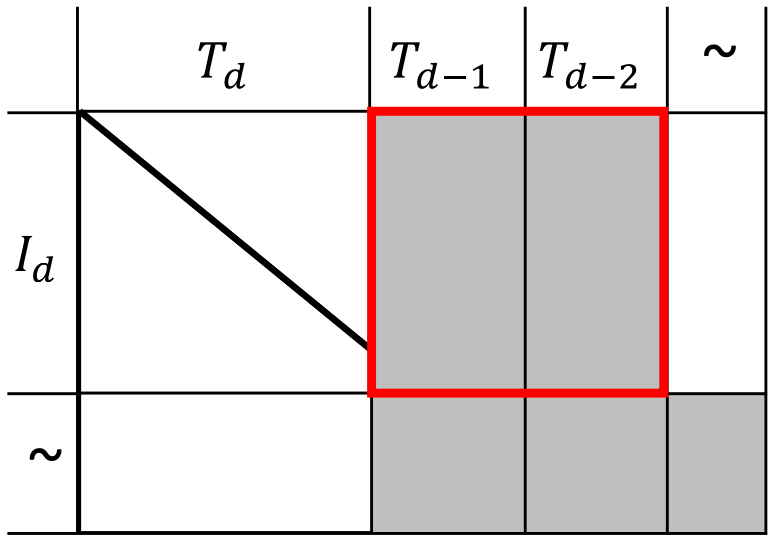
for d in $[D \ldots 2]$ do

① Perform elimination on $\mathcal{PM}[I_d, T_d]$

② Apply the same row operations on $\mathcal{PM}[I_d, T_{d-1}]$ and $\mathcal{PM}[I_d, T_{d-2}]$

Row operations on $\mathcal{PM}[I_d, \sim]$

③ Using the leading coefficients of $\mathcal{PM}[I_d, T_d]$, eliminate corresponding columns.



Comparison

We asymptotically evaluate the time complexity by the formula.

(We experimentally confirmed that the proposed algorithm behaves as our complexity estimation.)

- $q = 2^8, m = n$ (the estimation of the number of operations over $\mathbb{F}_q$)

n	20	40	60	80
h-XL	2^{75}	2^{134}	2^{194}	2^{252}
h-WXL	2^{75}	2^{129}	2^{182}	2^{234}
Crossbred	2^{65}	2^{123}	2^{180}	2^{237}
PXL	$\mathbf{2^{62}}$	$\mathbf{2^{117}}$	$\mathbf{2^{169}}$	$\mathbf{2^{220}}$

$\mathbf{2^{10} \sim 2^{20}}$ times faster in the $m = n$ case

WXL: a variant of XL using the sparsity of Macaulay matrices [Yang et al., FSE 2007]

h-WXL: hybrid approach with WXL

Crossbred: an XL variant with similar construction as PXL [Joux, Vitse, NuTMiC 2017]

Outline

・XL Algorithm・Hybrid Approach

・Proposed Algorithm

・**Conclusions**

Conclusions and Future Works

Conclusions

- We proposed a new variant of h-XL.
- The proposed algorithm is asymptotically more efficient than other algorithms in the case of $n \approx m$.

Future Works

- generalizing the proposed algorithm to higher degree cases
- proposing fast implementation
  + analysis of practical efficiency