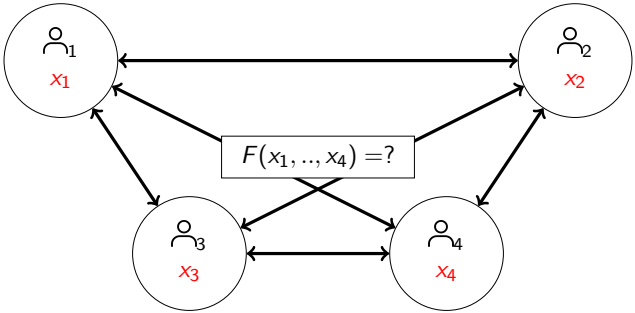
Secure Computations on Shared Polynomials in MPC

Pascal Giorgi[1], Fabien Laguillaumie[1], **Lucas Ottow**[1], Damien Vergnaud[2]

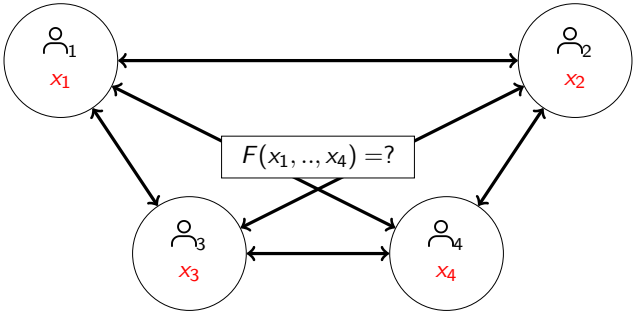
[1] LIRMM, Univ. Montpellier, Montpellier
[2] LIP6, Sorbonne Université, Paris

JNCF - March 2024

Multiparty Computation

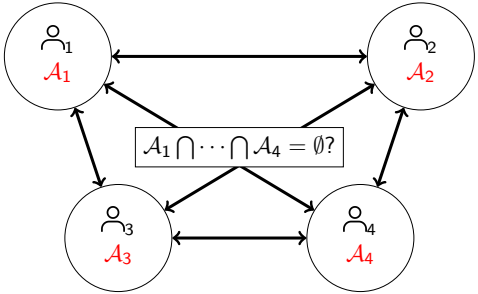


Multiparty Computation



Motivation

Private Disjointess Test

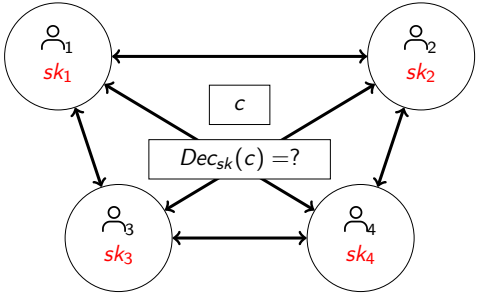


$P_{\mathcal{A}} = \prod_{\mathcal{A}}(X - \alpha)$

$P_{\cap} = \sum r_i P_{\mathcal{A}_i}$

$\Rightarrow$ Polynomial evaluation

Threshold McEliece Scheme

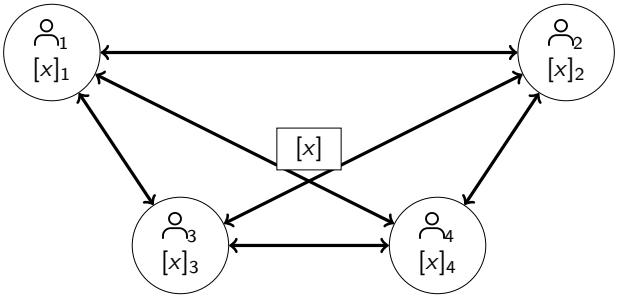


$S_c, g \rightarrow S_c^{-1} \mod g$

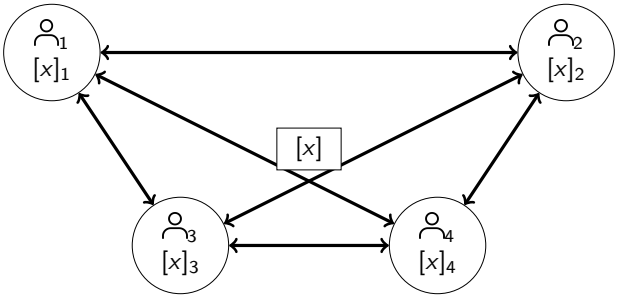
g irreducible

$\Rightarrow$ Modular inversion

Secret Sharing in MPC

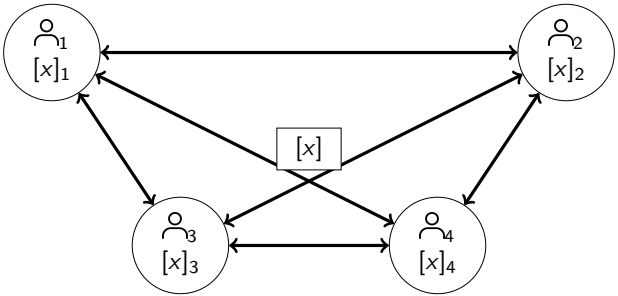


Secret Sharing in MPC



- General sol: share secret input + compute jointly
  [Bar-Ilan,Beaver,1988], [Ben-On Goldwasser Widgerson,1988].

Secret Sharing in MPC



- General sol: share secret input + compute jointly
  [Bar-Ilan,Beaver,1988], [Ben-On Goldwasser Widgerson,1988].
- Goal: Faster solutions for specific problems

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$
  - shared polynomials, shared matrices,...

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$
  - shared polynomials, shared matrices,...
- Basic operations:

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$
  - shared polynomials, shared matrices,...
- Basic operations:
  - Computation on public data: local computations — no communication

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$
  - shared polynomials, shared matrices,...
- Basic operations:
  - Computation on public data: local computations — no communication
  - $[a], [b] \to [a+b]$: $[a+b]_j \leftarrow [a]_j + [b]_j$ — no communication
  - $c, [a] \to [ca]$: $[ca]_j \leftarrow c[a]_j$ — no communication

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$
  - shared polynomials, shared matrices,...
- Basic operations:
  - Computation on public data: local computations — no communication
  - $[a], [b] \to [a+b]$: $[a+b]_j \leftarrow [a]_j + [b]_j$ — no communication
  - $c, [a] \to [ca]$: $[ca]_j \leftarrow c[a]_j$ — no communication
  - $[a], [b] \to [a \cdot b]$ (secure multiplication) — requires communications

Computing in this model

- Two types of data: shared data $[x] \in \mathbb{F}_q$ and public data $y \in \mathbb{F}_q$
  - Ex: $[x] = [x]_1 + [x]_2 + \cdots + [x]_m$
  - shared polynomials, shared matrices,...
- Basic operations:
  - Computation on public data: local computations — no communication
  - $[a], [b] \to [a + b]$: $[a + b]_j \leftarrow [a]_j + [b]_j$ — no communication
  - $c, [a] \to [ca]$: $[ca]_j \leftarrow c[a]_j$ — no communication
  - $[a], [b] \to [a \cdot b]$ (secure multiplication) — requires communications
- Goal: For a given operation, as few sec. mult. as possible and constant round.

Previous work and our contribution

- [Mohassel, Franklin,2006]: specific operations on shared polynomials

	prod.	div.	fan-in prod.		interpol. + multi eval.		mod. inv.	
	[MF,06]		[MF,06]	Ours	[MF,06]	Ours	[MF,06]	Ours
Sec. mult.	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(n^2 d)$	$\mathcal{O}(\tau n^{1+1/\tau} d)$	$\mathcal{O}(d^2)$	$\mathcal{O}(\tau d^{1+1/\tau})$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^2)$
Round	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Previous work and our contribution

- [Mohassel, Franklin,2006]: specific operations on shared polynomials

	prod.	div.	fan-in prod.		interpol. + multi eval.		mod. inv.	
	[MF,06]		[MF,06]	Ours	[MF,06]	Ours	[MF,06]	Ours
Sec. mult.	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(n^2 d)$	$\mathcal{O}(\tau n^{1+1/\tau} d)$	$\mathcal{O}(d^2)$	$\mathcal{O}(\tau d^{1+1/\tau})$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^2)$
Round	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Only secure multiplications!

$$[f], [g], (\mathsf{a_i})_{\leq 2d+1} \xrightarrow{local} [f(a_i)][g(a_i)] \xrightarrow{\mathcal{O}(d)} [f(a_i)g(a_i)] \xrightarrow{local} [fg]$$

Previous work and our contribution

- [Mohassel, Franklin,2006]: specific operations on shared polynomials

	prod.	div.	fan-in prod.		interpol. + multi eval.		mod. inv.	
	[MF,06]		[MF,06]	Ours	[MF,06]	Ours	[MF,06]	Ours
Sec. mult.	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(n^2 d)$	$\mathcal{O}(\tau n^{1+1/\tau} d)$	$\mathcal{O}(d^2)$	$\mathcal{O}(\tau d^{1+1/\tau})$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^2)$
Round	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Fan-in mult. :
$[f_1], \ldots, [f_n] \rightarrow [f_1 \times \cdots \times f_n]$

Previous work and our contribution

- [Mohassel, Franklin,2006]: specific operations on shared polynomials

	prod.	div.	fan-in prod.		interpol. + multi eval.		mod. inv.	
	[MF,06]		[MF,06]	Ours	[MF,06]	Ours	[MF,06]	Ours
Sec. mult.	$\mathcal{O}(d)$	$\mathcal{O}(d)$	$\mathcal{O}(n^2 d)$	$\mathcal{O}(\tau n^{1+1/\tau} d)$	$\mathcal{O}(d^2)$	$\mathcal{O}(\tau d^{1+1/\tau})$	$\mathcal{O}(d^3)$	$\mathcal{O}(d^2)$
Round	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(\tau)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Evaluation on **shared** points :

$$[f], [\alpha_1], \ldots, [\alpha_d] \rightarrow [f(\alpha_1)], \ldots, [f(\alpha_d)]$$

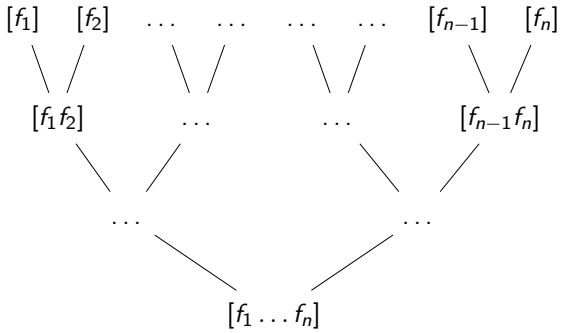
Improved fan-in multiplication

$$\underline{\text{Op:}}\ [f_1], \ldots, [f_n]\ (\textbf{non-zero}\ \text{and}\ \deg < d) \longrightarrow\ [f_1 \times \cdots \times f_n]$$

Improved fan-in multiplication

Op: $[f_1], \ldots, [f_n]$ (**non-zero** and $\deg < d$) $\longrightarrow$ $[f_1 \times \cdots \times f_n]$

Binary tree



2 poly at a time

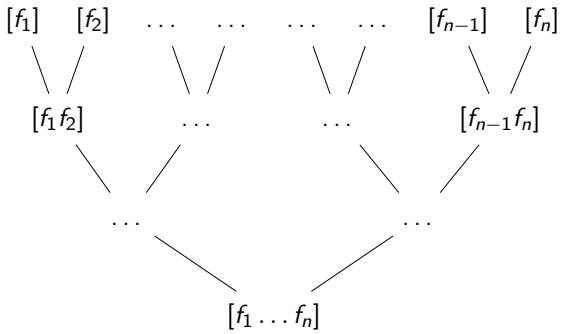
$\log n \times \mathcal{O}(nd)$ sec. mult.

$\mathcal{O}(\log n)$ rounds

Improved fan-in multiplication

Op: $[f_1], \ldots, [f_n]$ (**non-zero** and $\deg < d$) $\longrightarrow$ $[f_1 \times \cdots \times f_n]$

Binary tree



2 poly at a time

$\log n \times \mathcal{O}(nd)$ sec. mult.

$\mathcal{O}(\log n)$ rounds

[Mohassel, Franklin,2006]

$[f_i]$ = *element of* $F_q[X]/P$ ($\deg P \sim nd$)

1. $[p_i] = [r_{i-1}]^{-1}[f_i][r_i]$ *comm.*
   $[r_i] \leftarrow^{\$} (F_q[X]/P)^*$
2. $\prod p_i = \prod f_i \cdot r_n$ *local*
3. $\prod p_i [r_n^{-1}] = [\prod f_i]$ *local*

n poly at a time

$1 \times \mathcal{O}(n^2 d)$ sec. mult.

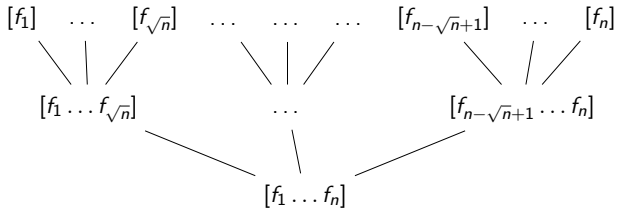
$\mathcal{O}(1)$ rounds

Improved fan-in multiplication

Our sol: "Squished" tree

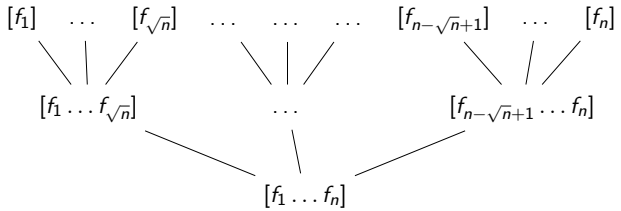
Improved fan-in multiplication

Our sol: "Squished" tree



Improved fan-in multiplication

Our sol: "Squished" tree



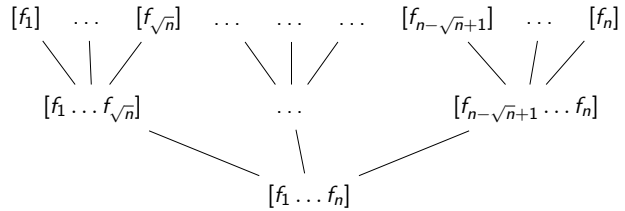
$\sqrt{n}$ poly at a time

$\sqrt{n} \times \mathcal{O}((\sqrt{n})^2 d) = \mathcal{O}(n^{3/2} d)$ sec. mult.

$\mathcal{O}(1)$ rounds (2 steps)

Improved fan-in multiplication

Our sol: "Squished" tree



$\sqrt{n}$ poly at a time

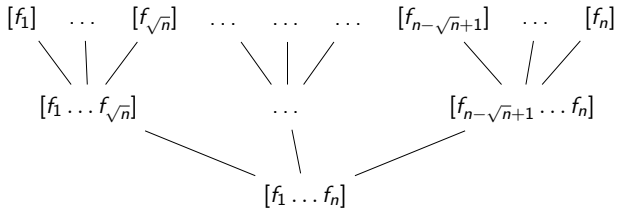
$\sqrt{n} \times \mathcal{O}((\sqrt{n})^2 d) = \mathcal{O}(n^{3/2} d)$ sec. mult.

$\mathcal{O}(1)$ rounds (2 steps)

- Generalization: $\mathcal{O}(\tau)$ rounds and $\mathcal{O}(\tau n^{1+1/\tau} d)$ sec. mult.

Improved fan-in multiplication

Our sol: "Squished" tree



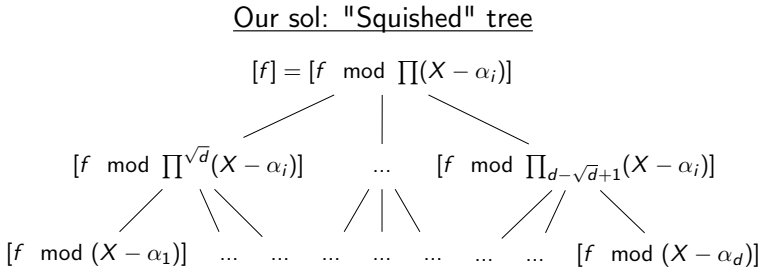
$\sqrt{n}$ poly at a time

$\sqrt{n} \times \mathcal{O}((\sqrt{n})^2 d) = \mathcal{O}(n^{3/2} d)$ sec. mult.

$\mathcal{O}(1)$ rounds (2 steps)

- Generalization: $\mathcal{O}(\tau)$ rounds and $\mathcal{O}(\tau n^{1+1/\tau} d)$ sec. mult.
- With same ideas, interpolation and multi-point evaluation on d shared points in $\mathcal{O}(\tau d^{1+1/\tau})$ sec. mult.

Improved fan-in multiplication



$\sqrt{n}$ poly at a time

$\sqrt{n} \times \mathcal{O}((\sqrt{n})^2 d) = \mathcal{O}(n^{3/2} d)$ sec. mult.

$\mathcal{O}(1)$ rounds (2 steps)

- Generalization: $\mathcal{O}(\tau)$ rounds and $\mathcal{O}(\tau n^{1+1/\tau} d)$ sec. mult.
- With same ideas, interpolation and multi-point evaluation on d shared points in $\mathcal{O}(\tau d^{1+1/\tau})$ sec. mult.

Modular inversion: previously

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}]\ \mod [g]$$

Modular inversion: previously

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}]\ \mod [g]$$

- [Mohassel, Franklin,2006]: Sylvester matrices, costly...

1. $[M] \leftarrow^{\$} (\mathbb{F}_q^{n \times n})^*$
2. $[A] \leftarrow [Syl(f, g)][M]$, reveal A
3. $[M]A^{-1} = [Syl(f, g)^{-1}]$

- $\mathcal{O}(d^3)$ sec. mult.

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}]\ \mod [g]$$

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}] \mod [g]$$

- Rmk: with public modulo h, $[f^{-1}] \mod h$ fast: $\mathcal{O}(d)$ sec. mult.

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}] \mod [g]$$

- Rmk: with public modulo h, $[f^{-1}] \mod h$ fast: $\mathcal{O}(d)$ sec. mult.
- Translate operation from "shared modulo" to "public modulo"?

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}]\ \mod [g]$$

- Rmk: with public modulo h, $[f^{-1}]\ \mod h$ fast: $\mathcal{O}(d)$ sec. mult.
- Translate operation from "shared modulo" to "public modulo"?

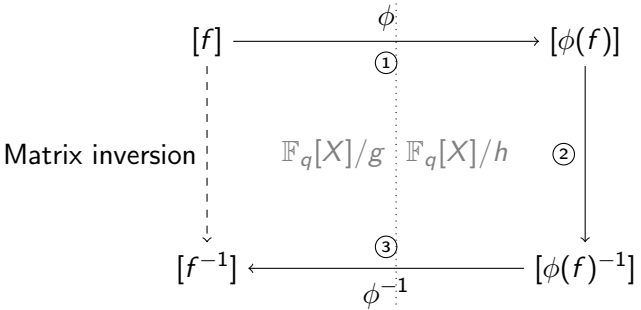
- h irred. public polynomial, degree d. $\Rightarrow\ \exists \phi : \mathbb{F}_q[X]/g \to \mathbb{F}_q[X]/h$ field isomorphism.

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}]\ \bmod [g]$$

- Rmk: with public modulo h, $[f^{-1}]\ \bmod h$ fast: $\mathcal{O}(d)$ sec. mult.
- Translate operation from "shared modulo" to "public modulo"?

- h irred. public polynomial, degree d. $\Rightarrow\ \exists \phi : \mathbb{F}_q[X]/g \to \mathbb{F}_q[X]/h$ field isomorphism.

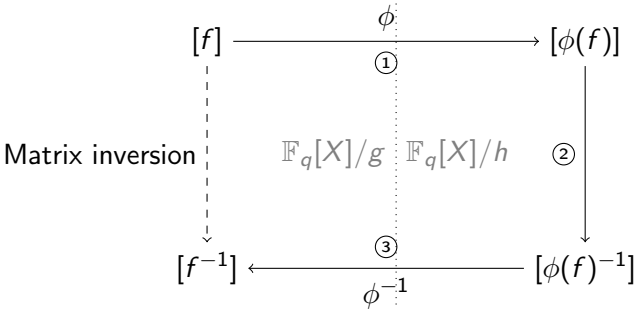


Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}] \mod [g]$$

- Rmk: with public modulo h, $[f^{-1}] \mod h$ fast: $\mathcal{O}(d)$ sec. mult.
- Translate operation from "shared modulo" to "public modulo"?

- h irred. public polynomial, degree d. $\Rightarrow\ \exists \phi : \mathbb{F}_q[X]/g \to \mathbb{F}_q[X]/h$ field isomorphism.



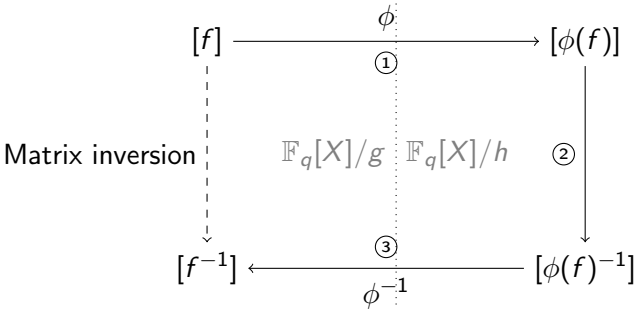
- Step 1: modular composition
  [Brent, Kung,1978]
  $\mathcal{O}(d^2)$ sec. mult., $[\phi(x)]$

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}]\ \mod [g]$$

- Rmk: with public modulo h, $[f^{-1}]\ \mod h$ fast: $\mathcal{O}(d)$ sec. mult.
- Translate operation from "shared modulo" to "public modulo"?

- h irred. public polynomial, degree d. $\Rightarrow\ \exists \phi : \mathbb{F}_q[X]/g \to \mathbb{F}_q[X]/h$ field isomorphism.



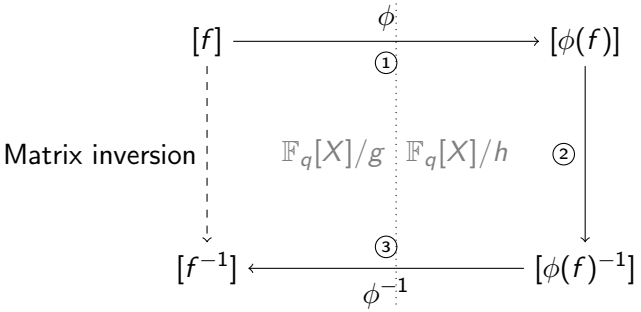
- Step 1: modular composition
  [Brent, Kung,1978]
  $\mathcal{O}(d^2)$ sec. mult., $[\phi(x)]$
- Step 2: simple
  $\mathcal{O}(d)$ sec. mult.

Modular inversion

$$\underline{\text{Op:}}\ [f], [g]\ (g \text{ irred and } \deg f < d, \deg g = d) \longrightarrow\ [f^{-1}] \mod [g]$$

- Rmk: with public modulo h, $[f^{-1}] \mod h$ fast: $\mathcal{O}(d)$ sec. mult.
- Translate operation from "shared modulo" to "public modulo"?

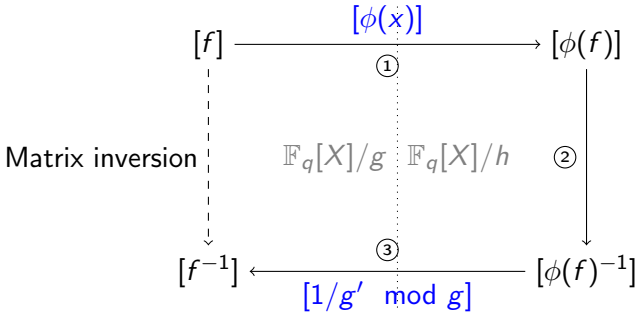
- h irred. public polynomial, degree d. $\Rightarrow$ $\exists \phi : \mathbb{F}_q[X]/g \to \mathbb{F}_q[X]/h$ field isomorphism.



- Step 1: modular composition
  [Brent, Kung,1978]
  $\mathcal{O}(d^2)$ sec. mult., $[\phi(x)]$
- Step 2: simple
  $\mathcal{O}(d)$ sec. mult.
- Step 3: Dual representation
  [De Feo, Doliskani, Schost,2014]
  $\mathcal{O}(d^2)$ sec. mult., $[1/g' \mod g]$

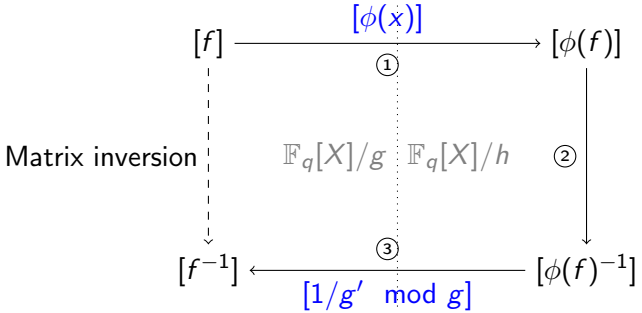
Modular inversion

Op: $[f], [g], h, [\phi(x)], [1/g' \mod g]$ $(\deg f < d, \deg g = d) \longrightarrow [f^{-1} \mod g]$



Modular inversion

Op: $[f], [g], h, [\phi(x)], [1/g' \mod g]$ $(\deg f < d, \deg g = d) \longrightarrow [f^{-1} \mod g]$



- $\mathcal{O}(d^3) \rightarrow \mathcal{O}(d^2)$ sec. mult. (with auxiliary inputs only on $[g]$)

Conclusion

- evaluation $\Rightarrow$ Private Disjointess Test (m people, n elements per set):

	Comm.	Rounds	crypto hyp.
[Sathya Narayanan et al. ,CANS2009]	$\mathcal{O}(mn^2)$	$\mathcal{O}(\log(mn))$	No
[Chandran et. al.,DCC2021]	$\mathcal{O}(mn\log^2(m))$	$\mathcal{O}(log(m\log(n)))$	Yes
Our protocol	$\mathcal{O}(nm + \tau n^{1+1/\tau})$	$\mathcal{O}(\tau)$	No

Conclusion

- evaluation $\Rightarrow$ Private Disjointess Test (m people, n elements per set):

	Comm.	Rounds	crypto hyp.
[Sathya Narayanan et al. ,CANS2009]	$\mathcal{O}(mn^2)$	$\mathcal{O}(\log(mn))$	No
[Chandran et. al.,DCC2021]	$\mathcal{O}(mn\log^2(m))$	$\mathcal{O}(log(m\log(n)))$	Yes
Our protocol	$\mathcal{O}(nm + \tau n^{1+1/\tau})$	$\mathcal{O}(\tau)$	No

- Modular inversion $\Rightarrow$ Threshold McEliece Scheme:
  - Improvements from [Takahashi et al.,DCC2023]: shorter private key and some faster steps in decryption

Conclusion

- evaluation $\Rightarrow$ Private Disjointess Test (m people, n elements per set):

	Comm.	Rounds	crypto hyp.
[Sathya Narayanan et al. ,CANS2009]	$\mathcal{O}(mn^2)$	$\mathcal{O}(\log(mn))$	No
[Chandran et. al.,DCC2021]	$\mathcal{O}(mn\log^2(m))$	$\mathcal{O}(log(m\log(n)))$	Yes
Our protocol	$\mathcal{O}(nm + \tau n^{1+1/\tau})$	$\mathcal{O}(\tau)$	No

- Modular inversion $\Rightarrow$ Threshold McEliece Scheme:
  - Improvements from [Takahashi et al.,DCC2023]: shorter private key and some faster steps in decryption
  - Future work: improvements on key generation and other steps of decryption
    $\Rightarrow$ generating shared ired. polynomial, rational reconstruction

Conclusion

- evaluation $\Rightarrow$ Private Disjointess Test (m people, n elements per set):

	Comm.	Rounds	crypto hyp.
[Sathya Narayanan et al. ,CANS2009]	$\mathcal{O}(mn^2)$	$\mathcal{O}(\log(mn))$	No
[Chandran et. al.,DCC2021]	$\mathcal{O}(mn\log^2(m))$	$\mathcal{O}(log(m\log(n)))$	Yes
Our protocol	$\mathcal{O}(nm + \tau n^{1+1/\tau})$	$\mathcal{O}(\tau)$	No

- Modular inversion $\Rightarrow$ Threshold McEliece Scheme:
  - Improvements from [Takahashi et al.,DCC2023]: shorter private key and some faster steps in decryption
  - Future work: improvements on key generation and other steps of decryption
    $\Rightarrow$ generating shared ired. polynomial, rational reconstruction

Thank you for your attention!