Amortized Complexity of Information-Theoretically Secure MPC Revisited[1]

Ignacio Cascudo, Ronald Cramer, Chaoping Xing, and Chen Yuan

Presenter: Liu Xiaoyi

[1] eprint 2018/429

In the BGW-model…

How to use SIMD to squeeze more “computation” into a single computation pass?

In the BGW-model…

How to use SIMD to squeeze more “computation” into a single computation pass?

With packed Shamir secret sharing.

$\Omega(n)$ parallel instances: $O(n) \to O(1)$ communication per multiplication gate.

In the BGW-model…

How to use SIMD to squeeze more "computation" into a single computation pass?

With packed Shamir secret sharing.

$\Omega(n)$ parallel instances: $O(n) \to O(1)$ communication per multiplication gate.

SSS requires a "large enough" field. Can we encode multiple elements in the base field together? e.g. when the base field is small ($\mathbb{F}_2$)

Squeezing elements together

Namely, can we have a map: $\phi : \mathbb{F}_q^k \to \mathbb{F}_{q^m}$, and some protocol to evaluate arithmetic circuits "through" ϕ:

- Input: All parties receives k inputs $\boldsymbol{x} = (x_1, ..., x_k) \in \mathbb{F}_k$.
- Encode: compute $\phi(\boldsymbol{x}) \in \mathbb{F}_{q^m}$
- Compute: All parties evaluate arithmetic circuit with $\phi(\boldsymbol{x})$ as input, reconstruct output $\phi(\boldsymbol{o}) \in \mathbb{F}_{q^m}$
- Decode: computing $\boldsymbol{o} = \phi^{-1}(\phi(\boldsymbol{o}))$

Main result fo the paper

Yes, we can! And pretty efficiently: $\forall q \forall k \exists m \exists \phi : \mathbb{F}_q^k \to \mathbb{F}_{q^m}$, where $m = O(k)$

Main result fo the paper

Yes, we can! And pretty efficiently: $\forall q \forall k \exists m \exists \phi : \mathbb{F}_q^k \to \mathbb{F}_{q^m}$, where $m = O(k)$

In malicious settings,

- Modified DN protocol with small fields: Do $\Omega(\log n)$ parallel computation, $O(n \log n) \to O(n)$ bit per gate.
- Modified DN protocol with small fields and suboptimal threshold: Combine with Packed SSS, $O(\log n) \to O(1)$ bit per gate.

What is omitted in this presentation

- The concrete construction of such ϕ, and proof of why can $m = O(k)$. Instead, several praticle parameter choices are given.
- The detailed handling of player elimination.

The procedure

$$\phi : \mathbb{F}_q^k \to \mathbb{F}_{q^m}$$

- Input: All parties receives k inputs $\boldsymbol{x} = (x_1, ..., x_k) \in \mathbb{F}_k$.
- Encode: compute $\phi(\boldsymbol{x}) \in \mathbb{F}_{q^m}$
- Compute: All parties evaluate arithmetic circuit with $\phi(\boldsymbol{x})$ as input, reconstruct output $\phi(\boldsymbol{o}) \in \mathbb{F}_{q^m}$
- Decode: computing $\boldsymbol{o} = \phi^{-1}(\phi(\boldsymbol{o}))$

“Just use the bits”

$$\phi : \mathbb{F}_q^k \leftrightarrow \mathbb{F}_{q^k}$$

"Just use the bits"

$$\phi : \mathbb{F}_q^k \leftrightarrow \mathbb{F}_{q^k}$$

... but multiplication does not work.
Notably, $\mathbb{F}_q^k$ contains zero divisors for $k \geq 2$.

Giving up strict inverse for multiplication

Define a pair of $\mathbb{F}_q$-linear maps:

$$\phi : \mathbb{F}_q^k \leftrightarrows \mathbb{F}_{q^m} : \psi$$

where ψ is the **"decode multiplication"** map:

$$\boldsymbol{x} * \boldsymbol{y} = \psi(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}))$$

Giving up

All of the following do not necessarily hold:

- $\phi(\boldsymbol{x} * \boldsymbol{y}) = \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})$
- $\boldsymbol{x} = \psi(\phi(\boldsymbol{x}))$
- $\boldsymbol{x} * \boldsymbol{y} * \boldsymbol{z} = \psi(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}) \cdot \phi(\boldsymbol{z}))$

Define a pair

where ψ is the **"decode multiplication"** map:

$$\boldsymbol{x} * \boldsymbol{y} = \psi(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}))$$

$(k, m)_q$ —RMFE (Reverse multiplication friendly embedding)

Define a pair of $\mathbb{F}_q$-linear maps:

$$\phi : \mathbb{F}_q^k \leftrightarrows \mathbb{F}_{q^m} : \psi$$

where ψ is the **"decode multiplication"** map:

$$\boldsymbol{x} * \boldsymbol{y} = \psi(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y}))$$

Thm. Exists $(k, m)_q$-RMFE for all k, q with $m = O(k)$

Random gates

- (k) uniformly random $\mathbb{F}_q$ elements
-
-

Random gates

- (k) uniformly random $\mathbb{F}_q$ elements
- $\phi(r_1, ..., r_k)$ with uniformly random $r_1, ..., r_k \in \mathbb{F}_q$
-

Random gates

- (k) uniformly random $\mathbb{F}_q$ elements
- $\phi(r_1, ..., r_k)$ with uniformly random $r_1, ..., r_k \in \mathbb{F}_q$
- Uniformly random $r' \in \mathrm{Im}\ \phi$

Random gates

- (k) uniformly random $\mathbb{F}_q$ elements
- $\phi(r_1, \ldots, r_k)$ with uniformly random $r_1, \ldots, r_k \in \mathbb{F}_q$
- Uniformly random $r' \in \operatorname{Im} \phi$ $\longleftarrow$ This is a $\mathbb{F}_q$-linear subspace

The hyper-invertible matrices

$A \in \mathbb{F}^{m \times n}$ $(n < m)$ is *super-invertible* if the matrices formed by selecting any n rows of A is invertible.

The hyper-invertible matrices

$A \in \mathbb{F}^{m \times n}$ is *hyper-invertible* if for all $k \leq \min(m, n)$, the matrices formed by selecting any k rows and k columns of A is invertible.

The hyper-invertible matrices

$A \in \mathbb{F}^{n \times n}$ is *hyper-invertible* if for all $k \leq n$, the matrices formed by selecting any k rows and k columns of A is invertible.

The hyper-invertible matrices

$A \in \mathbb{F}^{n \times n}$ is *hyper-invertible* if for all $k \leq n$, the matrices formed by selecting any k rows and k columns of A is invertible.

Construction: Select $2n$ evaluation points $\alpha_1, ..., \alpha_n, \beta_1, ..., \beta_n$. Consider the $\mathbb{F}$-**linear** map of reconstructing a degree-$(n-1)$ polynomial from n values as evaluations at $\alpha_1, ..., \alpha_n$, and evaluate the polynomial at $\beta_1, ..., \beta_n$.

$$\lambda_{i,j} = \prod_{k \in \{1,..,n\} \setminus j} \frac{\beta_i - \alpha_k}{\alpha_j - \alpha_k}$$

$\Pi_{\mathrm{RandElIm}\phi}$: Generate random elements in $\mathrm{Im}\ \phi$

- Fixed public $n \times n$ hyper-invertible matrix M. $1 \leq T \leq n - 2t$
- Outputs: T correct secret sharings of uniformly random $\mathrm{Im}\ \phi$ elements

- Each party P_i uniformly samples a $s^i \in \mathrm{Im}\ \phi$, shares it.
- Parties locally computes $([r^1], ..., [r^n])^T = M \cdot ([s^1], ..., [s^n])^T$
- For each $T + 1 \leq i \leq n$, P_i opens r^i, and check if it's in $\mathrm{Im}\ \phi$. If not, complains.
- Output unopened $[r^1], ..., [r^T]$

Fact: If all honest parties are happy, then $[r^1], ..., [r^T]$ are correct, and adversary has no information of them besides $r^1, ..., r^T \in \mathrm{Im}\ \phi$

- Outputs: T correct secret sharings of uniformly random $\mathrm{Im}\ \phi$ elements

- Each party P_i uniformly samples a $s^i \in \mathrm{Im}\ \phi$, shares it.
- Parties locally computes $([r^1], ..., [r^n])^T = M \cdot ([s^1], ..., [s^n])^T$
- For each $T + 1 \leq i \leq n$, P_i opens r^i, and check if it's in $\mathrm{Im}\ \phi$. If not, complains.
- Output unopened $[r^1], ..., [r^T]$

$\Pi_{\text{RandElIm}\phi}$: **Generate random elements in** $\text{Im } \phi$

- Fixed public $n \times n$ hyper-invertible matrix M. $1 \leq T \leq n - 2t$
- Outputs: T correct secret sharings of uniformly random $\text{Im } \phi$ elements

- Each party P_i uniformly samples a $s^i \in \text{Im } \phi$, shares it.
- Parties locally computes $([r^1], ..., [r^n])^T = M \cdot ([s^1], ..., [s^n])^T$
- For each $T + 1 \leq i \leq n$, P_i opens r^i, and check if it's in $\text{Im } \phi$. If not, complains.
- Output uno

Where is M and $\text{Im } \phi$ defined upon?

Bundling secret sharings together

Fundmentally, the problem is that the secret space is too small, so the sharing scheme **may not be linear** over the extension field.

$$\mathbb{F}_q \text{ vs. } \mathbb{F}_{q^m}$$

Bundling secret sharings together

Fundmentally, the problem is that the secret space is too small, so the sharing scheme **may not be linear** over the extension field.

$$\mathbb{F}_q \text{ vs. } \mathbb{F}_{q^m}$$

But if we gather m $\mathbb{F}_q$-linear secret sharing together, they can natually form a $\mathbb{F}_{q^m}$-linear secret sharing, while being individually easily accessible.

Bundling secret sharings together

Assume we want to force the secrets to lie in $\mathbb{F}_q$-linear subspace $V \subseteq \mathbb{F}_{q^m}^v$

If we have m of them, we can form a $m \times n$ $\mathbb{F}_q^m$ matrix with everyones' shares.

$$\begin{pmatrix} [x_1] \\ \dots \\ [x_m] \end{pmatrix}$$

$\mathbb{F}_{q^m}$ elements as $\mathbb{F}_q^{m\times m}$ matrices

Fix a basis of $\mathbb{F}_{q^m}$ as a $\mathbb{F}_q$-vector space. Then $\forall\lambda \in \mathbb{F}_{q^m}$:

$$\lambda \cdot (-) : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$$

is a linear map. Thus each λ can be identified with a $\mathbb{F}_q^{m\times m}$.

$\mathbb{F}_{q^m}$ elements as $\mathbb{F}_q^{m\times m}$ matrices

Fix a basis of $\mathbb{F}_{q^m}$ as a $\mathbb{F}_q$-vector space. Then $\forall\lambda \in \mathbb{F}_{q^m}$:

$$\lambda \cdot (-) : \mathbb{F}_{q^m} \to \mathbb{F}_{q^m}$$

is a linear map. Thus each λ can be identified with a $\mathbb{F}_q^{m\times m}$.

This induces a (injective) $\mathbb{F}_q$-algebra morphism $\Phi : \mathbb{F}_{q^m} \to \mathbb{F}_q^{m\times m}$ that fixes $\mathbb{F}_q$:

$$\forall\lambda \in \mathbb{F}_q \subseteq \mathbb{F}_{q^m}, \Phi(\lambda) = \lambda \cdot I_{m\times m}$$

$\mathbb{F}_{q^m}$ elements as $\mathbb{F}_q^{m\times m}$ matrices

$$\begin{pmatrix} [y_1] \\ \dots \\ [y_m] \end{pmatrix} = \lambda \cdot \begin{pmatrix} [x_1] \\ \dots \\ [x_m] \end{pmatrix} \stackrel{\mathbf{def}}{=} \Phi(\lambda) \cdot \begin{pmatrix} [x_1] \\ \dots \\ [x_m] \end{pmatrix}$$

- $x_1, \dots, x_m \in V \Rightarrow y_1, \dots, y_m \in V$
- Compatible with $\mathbb{F}_q$-linear.

$\Pi_{\mathrm{RandElSub}(V)}$: Generate random elements in V

- Fixed $\mathbb{F}_q$-vector subspace $V \subseteq \mathbb{F}_{q^m}^v$.
- Fixed basis for $\mathbb{F}_{q^m}$ as a $\mathbb{F}_q$-vector space.
- Fixed public $n \times n$ hyper-invertible matrix M.
- $1 \leq T \leq n - 2t$.
- Outputs: $T \times m$ correct secret sharings of uniformly random V elements

Exactly the same as $\Pi_{\mathrm{RandElIm}\phi}$

What’s missing

- Multiplication

$$(\phi \circ \psi)(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})) = \phi(\boldsymbol{x} * \boldsymbol{y})$$

-

What's missing

- Multiplication

$$(\phi \circ \psi)(\phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})) = \phi(\boldsymbol{x} * \boldsymbol{y})$$

- Verify input shares

$\Pi_{\mathrm{CorrInput}}$: Checking the consistency of sharings

- Input: A secret sharing $[x]$
- Output: Accepts if $x \in \mathrm{Im}\ \phi$, rejects otherwise.

- Take an unused $[r]$ from $\mathrm{RandElSub}(\mathrm{Im}\ \phi)$
- Computes [x + r], publicly opens it, checks if $x + r \in \mathrm{Im}\ \phi$

Π_{ReEncode}: Computes $\phi \circ \psi$

- Input: A secret sharings $[x]$
- Output: $[\phi(\psi(x))]$

Π_{ReEncode}: Computes $\phi \circ \psi$

- Input: A secret sharings $[x]$
- Output: $[\phi(\psi(x))]$

Notice that $\phi \circ \psi$ is $\mathbb{F}_q$-linear, but not $\mathbb{F}_{q^m}$-linear, i.e. there may not exists a $\lambda \in \mathbb{F}_{q^m}$ s.t. $\phi \circ \psi = \lambda \cdot (-)$

But: $W = \left\{(x, \phi(\psi(x))) : x \in \mathbb{F}_{q^m}\right\} \subseteq \left(\mathbb{F}_{q^m}\right)^2$ is a $\mathbb{F}_q$-linear subspace.

Π_{ReEncode}: Computes $\phi \circ \psi$

- Input: A secret sharings $[x]$
- Output: $[\phi(\psi(x))]$

- Take an unused $([r], [\phi(\psi(r))])$ from $\text{RandElSub}(W)$
- Computes [x + r], publicly opens it
- Locally compute $\phi(\psi(x + r)) - [\phi(\psi(r))] = [\phi(\psi(x))]$

Conclusion

In the BGW-model, there is an efficient MPC protocol for n parties...

- ...secure against the maximal number of active corruptions $\lfloor \frac{n-1}{3} \rfloor$ that computes $\Omega(\log n)$ evaluations of a single binary circuit in parallel with an amortized communication complexity (per instance) of $O(n)$ bits per gate.
- For every $\varepsilon > 0$, ...secure against a submaximal number of active corruptions $t < (1 - \varepsilon)\frac{n}{3}$ that computes $\Omega(n \log n)$ evaluations of a single binary circuit in parallel with an amortized communication complexity (per instance) of $O(1)$ bits per gate.

Concatenation of RMFEs

If (ϕ_1, ψ_1) is an $(k_1, m_1)_{q^{m_2}}$-RMFE, (ϕ_2, ψ_2) is an $(k_2, m_2)_q$-RMFE, then the following pair of map gives a $(k_1 k_2, m_1 m_2)_q$-RMFE:

$$\left(x_1, ..., x_{k_1}\right) \mapsto \left(\phi_2(x_1), ..., \phi_2\left(x_{k_1}\right)\right) \mapsto \phi_1\left(\phi_2(x_1), ..., \phi_2\left(x_{k_1}\right)\right)$$

$$a \mapsto \psi_1(a) = \left(u_1, ..., u_{k_1}\right) \mapsto \left(\psi_2(u_1), ..., \psi_2\left(u_{k_1}\right)\right)$$

For boolean circuits

With $q = 2$, there exists $(3, 5)_2$-RMFE and a family of $(k, m)_{32}$-RMFE where $\frac{m}{k} \to \frac{62}{21}$.

Thus, there exists a family of $(k, m)_2$-RMFE with $\frac{m}{k} \to 4.92...$

Construction for relatively small k

If $1 \leq k \leq q + 1$, there exists a $(k, 2k - 1)_q$-RMFE

Choose any primitive element a of $\mathbb{F}_{q^{2k-1}}/\mathbb{F}_q$, choose k evaluation points $\alpha_1, ..., \alpha_k \in \mathbb{F}_q \cup \{\infty\}$

ϕ is defined as (evaluate at a $\circ$ Langrange interpolation)

Thank you!

Q&A