Amortized Complexity of Information-Theoretically Secure MPC Revisited

Ignacio Cascudo [1] Ronald Cramer [2,3] Chaoping Xing [4] Chen Yuan [2]

[1]Aalborg University [2]CWI Amsterdam [3]Leiden University [4]NTU Singapore

CRYPTO, 22 August 2018

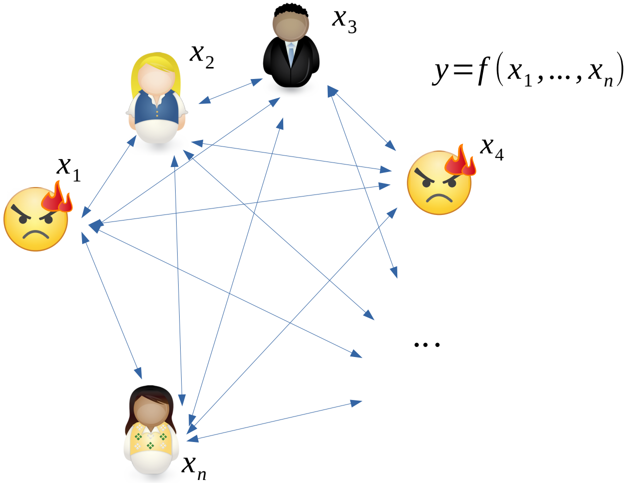
AALBORG UNIVERSITY DENMARK

CWI

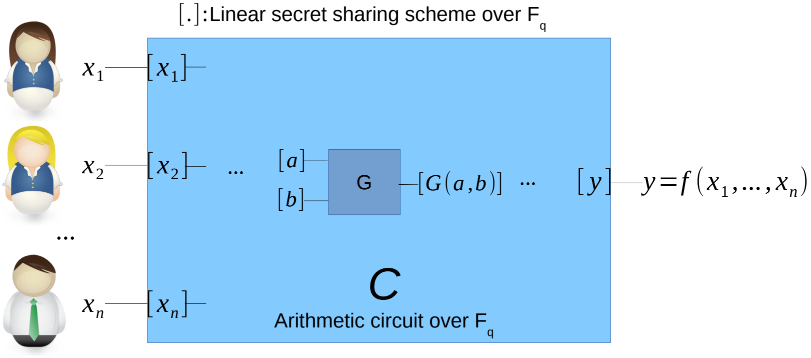
Universiteit Leiden

NANYANG TECHNOLOGICAL UNIVERSITY

Secure multiparty computation (MPC)



Secret-sharing based MPC



- Function represented by arithmetic circuit over some field $\mathbb{F}_q$.
- Parties secret-share inputs.
- Gate-by-gate computation ($[a], [b] \rightarrow [G(a,b)]$)
  - Linear gates: using linearity of secret sharing.
  - Multiplication gates: Dedicated subprotocol.

Motivation

Many secret-sharing-based MPC protocols need large finite fields.

Motivation

Many secret-sharing-based MPC protocols need large finite fields.

For example:

- Use of Shamir's scheme (BGW88 and many others)
- Use of hyperinvertible matrices (Beerliova-Hirt 08)
- Use of message authentication codes (SPDZ)

Motivation

How can we use those protocols for computing arithmetic circuits over **small** fields (e.g. $q = 2$)?.

Motivation

How can we use those protocols for computing arithmetic circuits over **small** fields (e.g. $q = 2$)?.

- **Standard solution:** Consider each input $\in \mathbb{F}_2$ as an element of a large extension field $\mathbb{F}_{2^m}$, use protocol for $\mathbb{F}_{2^m}$.

Motivation

How can we use those protocols for computing arithmetic circuits over **small** fields (e.g. $q = 2$)?.

- **Standard solution:** Consider each input $\in \mathbb{F}_2$ as an element of a large extension field $\mathbb{F}_{2^m}$, use protocol for $\mathbb{F}_{2^m}$.
- **Problem:** Seems wasteful.

Motivation

How can we use those protocols for computing arithmetic circuits over **small** fields (e.g. $q = 2$)?.

- **Standard solution:** Consider each input $\in \mathbb{F}_2$ as an element of a large extension field $\mathbb{F}_{2^m}$, use protocol for $\mathbb{F}_{2^m}$.
- **Problem:** Seems wasteful.
- Can we get more out of this?.

Goal

We want to securely compute $k > 1$ parallel evaluations of the binary circuit...

...by using one execution of the arithmetic MPC protocol over $\mathbb{F}_{2^m}$ plus "cheaper" steps (in terms of communication complexity).

Goal

We want to securely compute $k > 1$ parallel evaluations of the binary circuit...

...by using one execution of the arithmetic MPC protocol over $\mathbb{F}_{2^m}$ plus "cheaper" steps (in terms of communication complexity).

More concretely, we focus on **information-theoretically perfectly** secure MPC.
We consider Beerliova-Hirt 08 as “arithmetic” MPC protocol.

BH08 result / Our result

BH08

There exists an information-theoretically perfectly secure n-party MPC protocol for an arithmetic circuit over $\mathbb{F}_{2^m}$, $2^m > 2n$, which

- Is secure against $\lfloor (n-1)/3 \rfloor$ active corruptions (optimal).
- Has communication complexity of $O(n)$ field elements per gate.

BH08 result / Our result

BH08

There exists an information-theoretically perfectly secure n-party MPC protocol for an arithmetic circuit over $\mathbb{F}_{2^m}$, $2^m > 2n$, which

- Is secure against $\lfloor (n-1)/3 \rfloor$ active corruptions (optimal).
- Has communication complexity of $O(n)$ field elements per gate.

Our main result (Theorem 1:)

There exists a n-party MPC protocol for any **boolean** circuit which

- Is secure against $\lfloor (n-1)/3 \rfloor$ active corruptions (optimal).
- Computes $\Omega(\log n)$ evaluations in parallel.
- Has communication complexity of $O(n)$ **bits** per gate per instance.

Results

- Using packed secret-sharing cannot achieve this, as it can not attain $\lfloor (n-1)/3 \rfloor$ corruption tolerance.
- In fact we can combine our techniques with packed secret sharing and obtain:

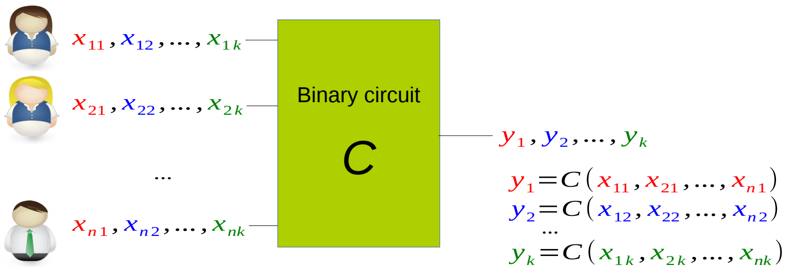
Results

- Using packed secret-sharing cannot achieve this, as it can not attain $\lfloor (n-1)/3 \rfloor$ corruption tolerance.
- In fact we can combine our techniques with packed secret sharing and obtain:

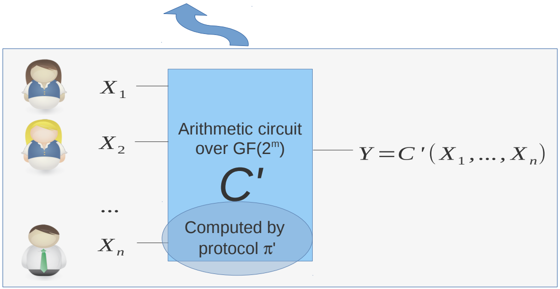
Result 2: for every $\epsilon > 0$, a n-party MPC protocol for any **boolean** circuit

- Secure against $t < (1-\epsilon)n/3$ active corruptions.
- Computes $\Omega(n \log n)$ evaluations in parallel.
- Amortized communication complexity of $O(1)$ **bits** per gate per instance.

Goal



Resource



Obstacle

$(\mathbb{F}_2^k, +)$, $(\mathbb{F}_{2^k}, +)$ isomorphic as $\mathbb{F}_q$-vector spaces, but

$(\mathbb{F}_2^k, +, *)$,$(\mathbb{F}_{2^k}, +, \cdot)$ **not** isomorphic as $\mathbb{F}_q$-algebras for $k > 1$.

(where $*$ is Schur product in $\mathbb{F}_2^k$, and $\cdot$ is field product in $\mathbb{F}_{2^k}$).

Reverse multiplication-friendly embeddings

Next best thing: **reverse multiplication-friendly embeddings** (RMFE)

A $(k, m)_2$-RMFE is a pair (ϕ,ψ) where

- $\phi : \mathbb{F}_2^k \to \mathbb{F}_{2^m}$ is $\mathbb{F}_2$-linear.
- $\psi : \mathbb{F}_{2^m} \to \mathbb{F}_2^k$ is $\mathbb{F}_2$-linear.
- For all $\mathbf{x}, \mathbf{y} \in \mathbb{F}_2^k$,
$$\mathbf{x} * \mathbf{y} = \psi(\phi(\mathbf{x}) \cdot \phi(\mathbf{y}))$$

Remark: ϕ is invertible, **but** $\psi \neq \phi^{-1}$.

History

Multiplication-friendly embeddings ($\mathbb{F}_2^k$ and $\mathbb{F}_{2^m}$ swapped):

- Introduced in MPC in CCCX09
- "Bilinear multiplication algorithms" (Chud 86)

Reverse multiplication-friendly embeddings

- Can be used to improve CCCX09 (unpublished)
- BMN17
- This paper
- BMN18

Constructions

[Remember a $(k, m)_2$-RMFE embeds $\mathbb{F}_2^k$ into $\mathbb{F}_{2^m}$]

- **Asymptotical:**

  There exist families of $(k, O(k))_2$-RMFE.

  Algebraic geometric construction.

Constructions

[Remember a $(k, m)_2$-RMFE embeds $\mathbb{F}_2^k$ into $\mathbb{F}_{2^m}$]

- **Asymptotical:**

  $$\text{There exist families of } (k, O(k))_2\text{-RMFE.}$$

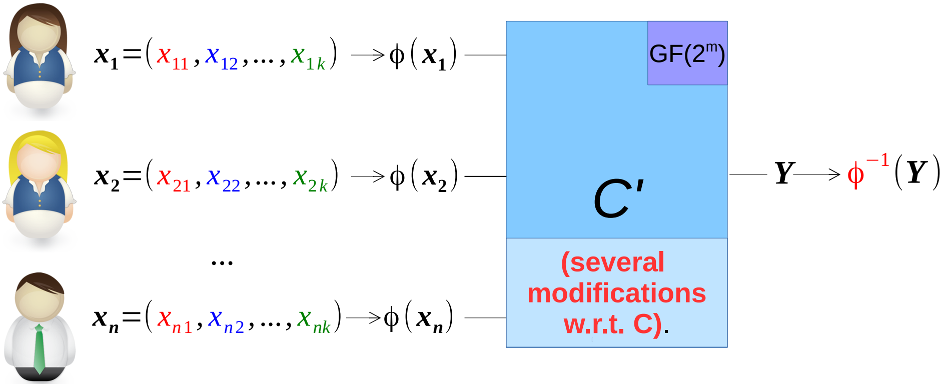
  Algebraic geometric construction.

- **Non-asymptotical:**

  $$\text{For all } r \leq 33 \text{, there exists a } (3r, 10r - 5)_2\text{-RMFE.}$$

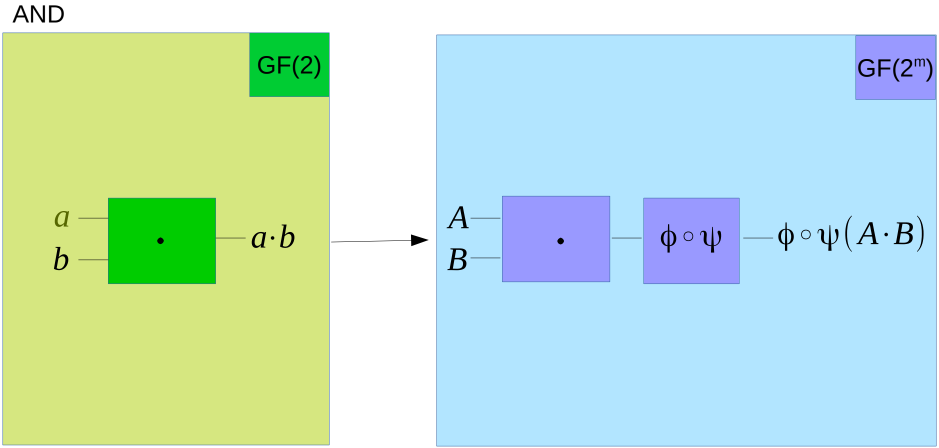
  Polynomial interpolation-based construction (e.g. we can embed $\mathbb{F}_2^{99}$ into $\mathbb{F}_{2^{325}}$).

How to use RMFEs

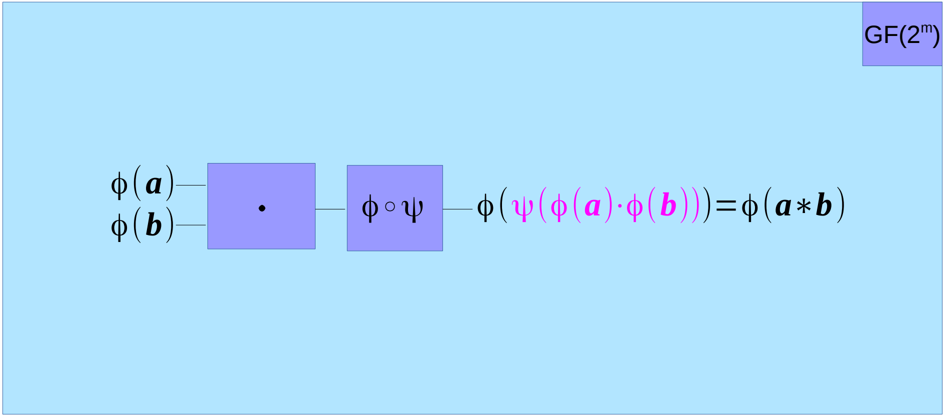


- Invariant: all intermediate values are sharings of ϕ-encodings.
- We decode the output with the inverse ϕ^{-1} (not with ψ).

Main circuit modification

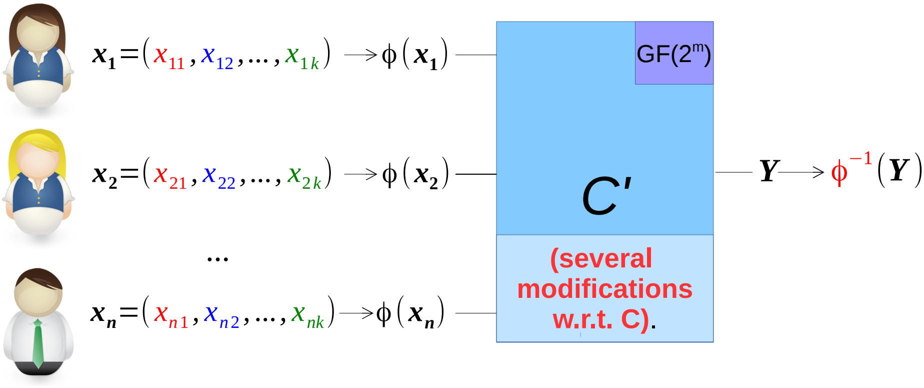


Main circuit modification explained



Obstacles

1. How do we (efficiently) process the $(\phi \circ \psi)$-gates?
2. How do we guarantee that parties input ϕ-encodings?



Random sharings in $\mathbb{F}_2$-linear subspaces

These can be reduced to the following problem:

"Given a $\mathbb{F}_2$-linear subspace $V \subseteq (\mathbb{F}_{2^m})^\ell$,
generate $[R_1], \ldots, [R_\ell]$ for $(R_1, \ldots, R_\ell) \in_R V$."

Random sharings in $\mathbb{F}_2$-linear subspaces

These can be reduced to the following problem:

"Given a $\mathbb{F}_2$-linear subspace $V \subseteq (\mathbb{F}_{2^m})^\ell$,
generate $[R_1], \ldots, [R_\ell]$ for $(R_1, \ldots, R_\ell) \in_R V$."

Hyper-invertible matrices (BH08):

- Would work if V were a $\mathbb{F}_{2^m}$-linear subspace
- But do not work directly for $\mathbb{F}_2$-linear subspaces.

Random sharings in $\mathbb{F}_2$-linear subspaces

These can be reduced to the following problem:

"Given a $\mathbb{F}_2$-linear subspace $V \subseteq (\mathbb{F}_{2^m})^\ell$,
generate $[R_1], \ldots, [R_\ell]$ for $(R_1, \ldots, R_\ell) \in_R V$."

Hyper-invertible matrices (BH08):

- Would work if V were a $\mathbb{F}_{2^m}$-linear subspace
- But do not work directly for $\mathbb{F}_2$-linear subspaces.

Solution: Apply HIM-based protocol to the **tensor product** $\mathbb{F}_{2^m} \otimes V$.

- $\mathbb{F}_{2^m} \otimes V$ is a $\mathbb{F}_{2^m}$-vector space.
- We can see its elements as vectors from V^m.

Conclusions

We present:

- A methodology to securely evaluating several instances in parallel of a circuit over a **small field**, by using a SSS-based MPC for a **large field**.
- An extension of the results from BH08 to small fields (in an amortized sense).
- Main technical handle: Reverse multiplication-friendly embeddings.

Future work:

- Extending these results to other models (e.g. dishonest majority).