Round-Preserving Parallel Composition of Probabilistic-Termination Protocols

Ran Cohen (TAU)
Sandro Coretti (NYU)
Juan Garay (Yahoo Research)
Vassilis Zikas (RPI)
Broadcast is Good

Given a broadcast channel, every function $f$ can be computed with full security (honest majority)

- Round complexity depends only on $f$ (unconditional)
- Constant-round protocols (OWF)
- Optimal three-round protocols (FHE)
Broadcast is Very Good

Parallel composition preserves round complexity

If $r$-round $\pi$ is secure under parallel composition

$\Rightarrow$ poly-many parallel executions of $\pi$ in $r$ rounds
What if Broadcast Doesn’t Exist?
Use Broadcast Protocols

- Trusted setup (PKI/information-theoretic signatures)
- Protocols with **simultaneous termination** require $t + 1$ rounds [FL’82, DRS’90]
- Exp. constant round $\Rightarrow$ **probabilistic termination** [FM’88, FG’03, KK’06, Micali’17]
  - Non-simultaneous termination
  - Termination round not a priori known
- Naïve parallel composition is **not** round preserving
Naïve Parallel Composition

Protocol with *expected* $O(1)$ rounds (geometric dist.)

$\Rightarrow$ $n$ parallel instances take $\Theta(\log n)$ rounds

**Example**: Coin flipping

- Stand-alone coin flip: $\Pr(\text{heads}) = 1/2$
  $\Rightarrow$ output is *heads* in expected 2 rounds

- Flipping in parallel $n$ coins, each coin until *heads*
  $\Rightarrow$ expected $\log n$ rounds
Parallel Composition of Broadcast

• Expected constant round parallel broadcast [BE’03, FG’03, KK’06]

• Composable parallel broadcast [CGZ’16]

⇒ Recipe for MPC:

1) Construct protocol in the broadcast model

2) Instantiate bcast channel using PT parallel bcast
Parallel Composition of Broadcast

• Expected constant round parallel broadcast [BE’03, FG’03, KK’06]

• Composable parallel broadcast [CGZ’16]

⇒ Recipe for MPC:

1) Construct protocol in the broadcast model

2) Instantiate bcast channel using PT parallel bcast

**Problem:**
The MPC protocol has probabilistic termination
(Naïve parallel composition not round preserving)
Main Question

Can parallel composition of arbitrary PT protocols be round-preserving?
Main Question

Can parallel composition of arbitrary PT protocols be round-preserving?
In a black-box way?

BB w.r.t. **functionality**
[Rosulek’12, IKPSY’16]

BB w.r.t. **protocol**
(next-message function)
Common Terminology
(1) Secure Multiparty Computation
(2) Synchronous MPC [KMTZ‘13, CCGZ‘16]

- Ideal world captures round complexity of $\pi$
- Trusted party samples $r_{term} \leftarrow D = D(\pi)$
- Parties continuously ask for output (receive by $r_{term}$)
- $S$ can instruct early delivery for specific parties
(3) Functionally BB Protocols

- Traditional MPC: all parties know $f$
(3) Functionally BB Protocols

- Traditional MPC: all parties know $f$
- FBB protocol is defined for function class $\mathcal{F} = \{f_1, \ldots, f_N\}$
- Parties have oracle access to $f \in \mathcal{F}$ ($\mathcal{Z}, \mathcal{A}, \mathcal{S}$ know $f$)
(3) Functionally BB Protocols

Protocol $\pi$ is **FBB protocol** for $\mathcal{F}$

if $\forall f \in \mathcal{F}$ protocol $\pi^f$ securely computes $f$
Functionally BB Protocols

Theorem [IKPSY’16]:

\[ \exists 2\text{-party function class } \mathcal{F} \text{ such that no FBB protocol computes } \mathcal{F} \text{ facing semi-honest adversary} \]

Proof intuition:

The function class \( \mathcal{F} = \{f_\alpha\}_{\alpha \in \{0, 1\}^\kappa} \) defined as

\[
f_\alpha(x_1, x_2) = \begin{cases} 
1, & x_1 \oplus x_2 = \alpha \\
0, & x_1 \oplus x_2 \neq \alpha
\end{cases}
\]
Given $n$-party functions $f_1, f_2, \ldots, f_m$ denote by $f_1 \parallel f_2 \parallel \cdots \parallel f_m$ the following function:

- Each $P_i$ has input $x_i = (x_i^1, x_i^2, \ldots, x_i^m)$
- Output is $y = (y_1, y_2, \ldots, y_m)$
FBB Parallel Composition
Semi-Honest FBB Protocol

Theorem:

• Let $F_1, \ldots, F_m$ be deterministic function classes

• Consider $(F_1, \ldots, F_m)$-hybrid model that $\forall j$ computes the function $f_j \in F_j$ with expected constant round complexity $\mu$

• Then $\exists$ FBB protocol for $F_1 \parallel \cdots \parallel F_m$ with expected constant round complexity
1) Parties invoke $\ell$ instances of each $f_j$

2) Each $P_i$ sends $x_{i}^{j}$ to all instances of $f_j$ and asks output for $r$ rounds

3) If some $P_i$ received output $y_j$ for each $f_j$ distribute $(y_1, \ldots, y_m)$ and halt, otherwise restart
Semi-Honest FBB Protocol

Proof intuition:

✓ Correctness
✓ Privacy: corrupt parties always use the same input values (semi-honest)
✓ Round complexity: for \( \ell = \Omega(\log m) \) and constant \( r > \mu \), the expected number of “restarts” is constant (Markov)
What About Malicious?

• The previous protocol is not secure for malicious
• The adversary can send different $x_i^j$ and $\tilde{x}_i^j$ to $f_j$ and learn multiple outputs
• This is inherent for **batched-parallel composition protocols**
  - Parties use $\left(x_1^k, \ldots, x_n^k\right)$ as input for two calls to the trusted party
  - Possibly in different rounds $\rho$ and $\rho'$
  - Possibly for computing different $f_j$ and $f_j'$
Malicious FBB Protocol

**Theorem:** Let \( m = O(\kappa) \)

\( \exists n \)-party function classes \( \mathcal{F}_1, \ldots, \mathcal{F}_m \) s.t.

if \( \pi \) computes \( \mathcal{F}_1 \parallel \cdots \parallel \mathcal{F}_m \) in \( (\mathcal{F}_1, \ldots, \mathcal{F}_m) \)-hybrid model (with exp. 2 rounds, geometric dist.)

then, facing a **single** malicious corrupted party:

- \( \pi \) must call each \( \mathcal{F}_i \) at least once
- If \( \pi \) is naïve parallel composition
  \( \Rightarrow \) not round preserving (\( \log \kappa \))
- \( \pi \) is not batched-parallel composition protocol

until some get output

call each \( \mathcal{F}_j \) until all

parties get output

using same inputs in two calls
Proof Intuition

Define $\mathcal{F}_1 = \cdots = \mathcal{F}_m = \{f_\alpha\}_{\alpha \in \{0,1\}^k}$ where

$$f_\alpha(x_1, x_2, \lambda, \ldots, \lambda) = \begin{cases} (x_2, x_1, \alpha, \ldots, \alpha), & x_1 \oplus x_2 = \alpha \\ (0^k, 0^k \ldots, 0^k), & x_1 \oplus x_2 \neq \alpha \end{cases}$$

- Naïve composition fails for geometric dist.
- No FBB protocol (without invoking trusted party) – extending [IKPSY’16]
- No batched-parallel protocol

See the paper for details
Protocol-BB Parallel Composition
Protocol-BB Parallel Composition

Theorem:

• Let PT protocols $\pi_1, \ldots, \pi_m$ realizing $f_1, \ldots, f_m$
• Then $\pi = \text{compiler}(\pi_1, \ldots, \pi_m)$ realizes $f_1 \parallel \cdots \parallel f_m$

➤ Round preserving $\mathbb{E}(\pi) = O\left(\max_i \mathbb{E}(\pi_i)\right)$

➤ Black-box w.r.t. protocols $\pi_1, \ldots, \pi_m$

The compiler doesn’t know the code of $\pi_i$
(oracle access to next-message function)
Protocol Compiler

$\pi_1$

$\pi_2$

$\pi_m$
Protocol Compiler

\[ \pi_1 \quad \cdots \quad \pi_2 \quad \cdots \quad \pi_m \]
Prevent Multiple Inputs

Use **Setup, Commit, then Prove** functionality with a tweak [CLOS’02, IOZ’14]
Prevent Multiple Inputs

Use **Setup, Commit, then Prove** functionality with a tweak [CLOS’02, IOZ’14]
Prevent Multiple Inputs

**Setup (correlated randomness)**

Use **Setup, Commit, then Prove** functionality with a tweak [CLOS’02, IOZ’14]
Prevent Multiple Inputs

Use **Setup, Commit, then Prove** functionality with a tweak [CLOS’02, IOZ’14]
Prevent Multiple Inputs

Use **Setup, Commit, then Prove** functionality with a tweak [CLOS’02, IOZ’14]
Some Challenges

• Recover from invalid ZK proofs without:
  1) Breaching privacy ($\mathcal{A}$ might have learned output)
  2) Blowing up round complexity

• Implement the Setup in constant rounds (use only correlated randomness for broadcast)

• Reactive functionalities with probabilistic termination

See the paper for details
Summary

We study parallel composition of PT protocols

**Functionally black-box (FBB) protocols**
- No round-preserving FBB parallel composition (using known techniques)
- Round-preserving FBB parallel composition with semi-honest security

**Black-box w.r.t. protocols**
- Round-preserving compiler for parallel composition

Thank You