Black-Box Constructions of Concurrently Secure Protocols

Huijia (Rachel) Lin
MIT & BU

Rafael Pass
Cornell
Secure MPC
Secure MPC

**Goal:** Allow a set of distrustful parties to compute *any* function $f$ on their own
Secure MPC

**Goal:** Allow a set of *distrustful* parties to compute *ANY* function $f$ on their own.
Secure MPC

**Goal:** Allow a set of *distrustful* parties to compute *ANY* function $f$ on their own

**Correctness**  
What to get---the outputs

**Privacy**  
What to hide---the private inputs
Secure MPC

**Goal:** Allow a set of distrustful parties to compute ANY function $f$ on their own

**Correctness**
What to get---the outputs

**Privacy**
What to hide---the private inputs

Even when no honest majority
Simulation Paradigm
Simulation Paradigm

REAL  ≈  IDEAL

“as correct & private as”
Simulation Paradigm

REAL

≈

IDEAL

“as correct & private as”
Simulation Paradigm

REAL

IDEAL

∀AR

“as correct & private as”
Simulation Paradigm

REAL

\[ \forall A_R \]

IDEAL

\[ \exists A_I \]

“as correct & private as”
Simulation Paradigm

REAL

IDEAL

∀AR

≈

∃AI

“as correct & private as”
Simulation Paradigm

REAL \[ \forall A_R \] \[ x_1 y_1 \] \[ x_2 y_2 \] \[ \approx \] \[ \exists A_I \]

IDEAL \[ x_1 y_1 \] \[ x_2 y_2 \]

“as correct & private as”

Correctness: The output of every player in ideal is the same as in real
“as correct & private as”

Correctness: The output of every player in ideal is the same as in real

Privacy: The simulator can learn whatever the adv learns
Simulation Paradigm

REAL

IDEAL

∀A_R

∃A_I

"as correct & private as"

Correctness: The output of every player in ideal is the same as in real
Privacy: The simulator can learn whatever the adv learns
Simulation Paradigm

**REAL**

\[ x_1 y_1 \quad \leftrightarrow \quad \cdots \quad \leftrightarrow \quad x_2 y_2 \]

\[ \forall A_R \]

**IDEAL**

\[ x_1 y_1 \approx \exists A_I \]

\[ x_2 y_2 \]

**Simulator**

“as correct & private as”

**Correctness:** The output of every player in ideal is the same as in real

**Privacy:** The simulator can learn whatever the adv learns

In this talk, we focus on static malicious corruption
The Concurrent Model
The Concurrent Model

**MANY** sets of players executing **MANY different** protocols all at once

[DDN, DNS, GK, Fe, KPR, RK, CKPR, KP, PRS, C...and many others]
The Concurrent Model

**MANY** sets of players executing **MANY different** protocols all at once
[DDN, DNS, GK, Fe, KPR, RK, CKPR, KP, PRS, C...and many others]
Concurrent Security (informally)

Many executions of different protocols

Many executions with INDEPENDENT trusted parties
Concurrent Security (informally)

Many executions of different protocols

Many executions with INDEPENDENT trusted parties

Universal Composibility (UC) [Can00]
Concurrent Security (informally)

Universal Composibility (UC) [Can00]

Impossible [CF01, CKF03]

Many executions of different protocols

Many executions with INDEPENDENT trusted parties
Super Polynomial Time Simulation (SPS)
Super Polynomial Time Simulation (SPS)
Super Polynomial Time Simulation (SPS)

REAL

IDEAL

— SPS [Pas03, BS05, LPV09, GGJS12]
Super Polynomial Time Simulation (SPS)

- SPS [Pas03, BS05, LPV09, GGJS12]
- Angel-based Security Model [PS04, MMY06]
- UC with super-poly helpers [CLP10]
Super Polynomial Time Simulation (SPS)

Feasibility Results Only

— SPS [Pas03, BS05, LPV09, GGJS12]
— Angel-based Security Model [PS04, MMY06]
— UC with super-poly helpers [CLP10]
Super Polynomial time (SPS) Security

Feasibility Results Only

Due to the Non-Black-Box constructions
(Lots of Karp reductions)
Super Polynomial time (SPS) Security

Feasibility Results Only

Naturally,
Solution: Black-box Constructions
(No Karp reductions)
Super Polynomial time (SPS) Security

Feasibility Results Only

Naturally,
Solution: Black-box Constructions
(No Karp reductions)

Efficient Protocols
BB MPC Protocols
BB MPC Protocols

In the stand alone setting---Solved!

$O(1)$ round BB MPC, $f$/ minimal assumption semi-honest OT
[Kil88,IPS08,IKLP06,Hai08,Wee10,Goy11]
BB MPC Protocols

*In the stand alone setting---Solved!*

O(1) round BB MPC, f/ minimal assumption semi-honest OT
[Kil88,IPS08,IKLP06,Hai08,Wee10,Goy11]

*In the concurrent setting*

Only **unconditionally secure** UC protocols f/ strong set-ups
e.g. Ideal OT [Kil88,IPS08], hardware tokens [GISVW10]
BB MPC Protocols

*In the stand alone setting---Solved!*

O(1) round BB MPC, f/ minimal assumption semi-honest OT [Kil88,IPS08,IKLP06,Hai08,Wee10,Goy11]

*In the concurrent setting*

Only *unconditionally secure* UC protocols f/ strong set-ups e.g. Ideal OT [Kil88,IPS08], hardware tokens [GISVW10]

Can we have

**BB concurrently secure** protocols

in the **plain** model?
Yes!

Our Result (informal):

*BB construction of concurrently secure MPC protocols*

- In the **plain model**
- Based on minimal assumption **Semi-Honest OT**
- Security in the **UC with super-poly helper** model
  - Implies super-polynomial time simulation security
  - Closed under universal composition
Our Result (informal):

**BB construction of concurrently secure MPC protocols**

- In the plain model
- Based on minimal assumption Semi-Honest OT
- Security in the UC with super-poly helper model
  - Implies super-polynomial time simulation security
  - Closed under universal composition

Yes!

How?
Any Functionality

[Kil88, IPS08, GMW87, BGW88]: Unconditional UC-security

Ideal Oblivious Transfer Box $F_{OT}$
Any Functionality

Ideal Oblivious Transfer Box $F_{OT}$

Unconditional UC-security

Stand-alone Semi-honest OT $SH-OT$
Any Functionality

Ideal Oblivious Transfer Box $F_{OT}$

Unconditional UC-security

Stand-Alone Security

Stand-alone Semi-honest OT $SH-OT$
Any Functionality

Ideal Oblivious Transfer Box $F_{OT}$

Any Functionality

[Kil88, IPS08, GMW87, BGW88]:
Unconditional UC-security

Ideal Oblivious Transfer Box $F_{OT}$

[IKLP06, Hai08, Wee10, Goy11]
Stand-Alone Security

This work
UC with Super-Poly Helper

Stand-alone Semi-honest OT $SH-OT$
Any Functionality

Ideal Oblivious Transfer Box $F_{OT}$

Unconditional UC-security

This work

UC with Super-Poly Helper

Stand-Alone Security

Stand-alone Semi-honest OT $SH-OT$

The main tool: BB CCA-Secure Commitments [CLP10]
CCA-Secure Commitments
CCA-Secure Commitments

The commitment analogue of CCA2 encryption.
CCA-Secure Commitments

The commitment analogue of CCA2 encryption.

\[ \text{CCA-Secure Commitments} \]

\[ \text{The commitment analogue of CCA2 encryption.} \]

\[ C(x) \rightarrow A \rightarrow C(y_1) \rightarrow C(y_2) \rightarrow C(y_3) \]
CCA-Secure Commitments

The commitment analogue of CCA2 encryption.

\[ \text{O is a committed-value oracle} \]

If valid com, \( y = \text{the committed value} \)

Else if invalid com, \( y = \text{bot} \)
CCA-Secure Commitments

The commitment analogue of CCA2 encryption.

O is a committed-value oracle

If valid com, $y = \text{the committed value}$

Else if invalid com, $y = \text{bot}$

Note: Original definition in [CLP10] considers a decommitment oracle. (with black-box construction, we can only achieve the weaker notion.)
CCA-Secure Commitments

The commitment analogue of CCA2 encryption.

Chosen-Commitment-Attack (CCA) security:

Either $A$ forwards the left commitment to the right
Or LHS is hiding --- view of $A$ indistinguishable
Concurrent Non-Malleable Commitments

\[ C(x) \rightarrow A \rightarrow C(y_1) \rightarrow C(y_2) \rightarrow C(y_3) \]
Concurrent Non-Malleable Commitments

Non-Malleability

Either \( A \) copies the left commitment to the right
Or \( x \) and \((y_1, y_2, y_3)\) independent

--- view of \( A + (y_1, y_2, y_3) \) indistinguishable
Concurrent Non-Malleable Commitments

Non-Malleability

Either $A$ copies the left commitment to the right
Or $x$ and $(y_1, y_2, y_3)$ independent
--- view of $A + (y_1, y_2, y_3)$ indistinguishable
Concurrent Non-Malleable Commitments

Non-Malleability

Either \( A \) copies the left commitment to the right
Or \( x \) and \((y_1, y_2, y_3)\) independent
--- view of \( A \) + \((y_1, y_2, y_3)\) indistinguishable

CCA security \(\rightarrow\) Non-Malleability
Theorem 1: OWF $\Rightarrow$ BB construction of CCA commitments
Theorem 1: OWF \(\rightarrow\) BB construction of CCA commitments

Theorem 2: CCA commitments + SH-OT \(\rightarrow\) BB implementation of \(F_{OT}\)
Theorem 1: **OWF \(\rightarrow\)** BB construction of CCA commitments

**Proof:** [CLP10]---Non-BB CCA commitments
+ [PW08]---BB trapdoor commitments
+ [CDMW08,09]---Cut & choose for consistency

Theorem 2: **CCA commitments + SH-OT**

\(\rightarrow\) BB implementation of \(F_{OT}\)
Theorem 1: OWF $\Rightarrow$ BB construction of CCA commitments

Proof: [CLP10]---Non-BB CCA commitments
+ [PW08]---BB trapdoor commitments
+ [CDMW08,09]---Cut & choose for consistency

Theorem 2: CCA commitments + SH-OT
$\Rightarrow$ BB implementation of $F_{\text{OT}}$
Theorem 1: **OWF → BB construction of CCA commitments**

**Proof:** [CLP10]---Non-BB CCA commitments
+ [PW08]---BB trapdoor commitments
+ [CDMW08,09]---Cut & choose for consistency

Theorem 2: **CCA commitments + SH-OT → BB implementation of $F_{OT}$**
Theorem 1: **OWF $\rightarrow$ BB construction of CCA commitments**

**Proof:** [CLP10]---Non-BB CCA commitments  
+ [PW08]---BB trapdoor commitments  
+ [CDMW08,09]---Cut & choose for consistency

Theorem 2: **CCA commitments + SH-OT**  
$\rightarrow$ **BB implementation of $F_{OT}$**

1. CCA is the right notion for BB concurrent MPC protocols
Theorem 1: OWF $\rightarrow$ BB construction of CCA commitments

Proof: [CLP10]---Non-BB CCA commitments
+ [PW08]---BB trapdoor commitments
+ [CDMW08,09]---Cut & choose for consistency

Theorem 2: CCA commitments + SH-OT

$\rightarrow$ BB implementation of $F_{OT}$

1. CCA is the right notion for BB concurrent MPC protocols
2. Assuming “AES” is a CCA commitment

$\rightarrow$ Efficient Constant-round BB concurrent MPC protocols
Theorem 1: **OWF $\Rightarrow$ BB construction of CCA commitments**

**Proof:** 
- [CLP10]---Non-BB CCA commitments
- [PW08]---BB trapdoor commitments
- [CDMW08,09]---Cut & choose for consistency

Theorem 2: **CCA commitments + SH-OT**

$\Rightarrow$ **BB implementation of $F_{OT}$**

1. CCA is the right notion for BB concurrent MPC protocols
2. Assuming “AES” is a CCA commitment

$\Rightarrow$ **Efficient Constant-round** BB concurrent MPC protocols
Theorem 2: CCA + SH-OT $\Rightarrow$ BB implementation of $F_{OT}$,
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver
Theorem 2: $\text{CCA} + \text{mS-OT} \implies \text{BB implementation of } F_{\text{OT}}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver

$S(m_0m_1)$  $R(b)$
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver

$S(m_0, m_1)$

$R(b)$

2n ms-OT executions with random inputs

$OT_1$  $\cdots$  $OT_k$  $\cdots$  $OT_{2n}$
Theorem 2: **CCA + mS-OT** $\Rightarrow$ BB implementation of $F_{\text{OT}}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver

$S^{(m_0m_1)}$ $\quad$ $R^{(b)}$

2n ms-OT executions with random inputs

$\text{OT}_1$ $\quad$ $\text{OT}_k$ $\quad$ $\text{OT}_{2n}$

Want: Enforce R behave honestly in OTs
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

Malicious Sender OT (ms-OT) --- OT secure for malicious sender & SH receiver

$S(m_0, m_1)$

$R(b)$

2n ms-OT executions with random inputs

$OT_1 \cdots OT_k \cdots OT_{2n}$

Non-BB Solution

ZK proof $R$ acts honestly

Want: Enforce $R$ behave honestly in OTs
Theorem 2: CCA + mS-OT $\rightarrow$ BB implementation of $F_{OT}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver

$$S_{(m_0m_1)}$$

$2n$ ms-OT executions with random inputs

BB Solution: Cut & Choose

$$\mathcal{R}_{(b)}$$

Want: Enforce $R$ behave honestly in OTs
Malicious Sender OT (ms-OT) --- OT secure for malicious sender & SH receiver

Want: Enforce R behave honestly in OTs
Malicious Sender OT (ms-OT) --- OT secure for malicious sender & SH receiver

Theorem 2: CCA + mS-OT \( \rightarrow \) BB implementation of \( F_{OT} \)

Want: Enforce \( R \) behave honestly in OTs

2n ms-OT executions with random inputs

BB Solution: Cut & Choose

\[ S(m_0m_1) \quad \text{and} \quad R(b) \]

\[ \begin{align*}
\text{OT}_1 & \quad \text{--} \quad \text{OT}_k & \quad \text{--} \quad \text{OT}_{2n}
\end{align*} \]

\( T \subset [2n], \ |T| = n \)

Open Randomness in \( \text{OT}_k \) for \( k \in T \)
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{\text{OT}}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver

$S (m_0, m_1)$

2n ms-OT executions with random inputs

BB Solution: Cut & Choose

Open Randomness in $OT_k$ for $k \in T$

Cut & Choose $\Rightarrow$ R behave honestly in most OTs [IKLP06, Wee10]
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

Malicious Sender OT (ms-OT) --- OT secure for malicious sender & SH receiver

$S$ $(m_0, m_1)$

$R(b)$

2n ms-OT executions with random inputs

OT$_1$ - - OT$_k$ - - OT$_{2n}$

BB Solution: Cut & Choose

Cut & Choose $\Rightarrow$ R behave honestly in most OTs [IKLP06, Wee10]
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

Malicious Sender OT (ms-OT)---OT secure for malicious sender & SH receiver

$S(m_0m_1)$

$\mathcal{R}(b)$

2n ms-OT executions with random inputs

OT$_1$ $\quad$ $\cdots$ $\quad$ OT$_k$ $\quad$ $\cdots$ $\quad$ OT$_{2n}$

BB Solution:
Cut & Choose

$T \subset [2n], \ |T| = n$

Open Randomness in OT$_k$ for $k \in T$

To prove security against a malicious sender, Simulator needs to bias the set T to be cut
To prove security against a malicious sender, Simulator needs to bias the set $T$ to be cut.

**Theorem 2:** $\text{CCA} + \text{mS-OT} \rightarrow \text{BB implementation of } F_{\text{OT}}$

$S_{(m_0m_1)}$  \hspace{2cm} $R_{(b)}$

2$n$ ms-OT executions with random inputs

**BB Solution:** Cut & Choose

OT$^1$  ---  OT$^k$  ---  OT$^{2n}$

ExtCom($r$)

$r'$

Open to $r$

Open Randomness in OT$^k$ for $k \in T$
To prove security against a malicious sender, Simulator needs to bias the set $T$ to be cut.

**Theorem 2:** CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

$$S_{(m_0m_1)} \quad R_{(b)}$$

2n ms-OT executions with random inputs

BB Solution: Cut & Choose

OT$_1$ -- OT$_k$ -- OT$_{2n}$

ExtCom($r$)

$T = r \ XOR \ r'$

Open Randomness in OT$_k$ for $k \in T$
Theorem 2: CCA + mS-OT $\Rightarrow$ BB implementation of $F_{OT}$

$S(m_0m_1)$

2n ms-OT executions with random inputs

BB Solution: Cut & Choose

OT$_1$ -- OT$_k$ -- OT$_{2n}$

ExtCom(r)

$T = r \ XOR \ r'$

Open Randomness in OT$_k$ for $k \in T$

Using **Coin Tossing**, Simulator can bias the set T to be cut
Informally, SH-OT + Coin-Tossing
→ Ideal OT in stand-alone setting [IKLP06,Wee10]
In the concurrent setting,
Main issue: simulation-sound coin tossing
In the concurrent setting,
Main issue: *simulation-sound* coin tossing

No adv can bias the coin tossing results,
even when the *simulator* is doing so
In the concurrent setting,
Main issue: simulation-sound coin tossing

No adv can bias the coin tossing results,
even when the simulator is doing so
In the concurrent setting,
Main issue: simulation-sound coin tossing
No adv can bias the coin tossing results,
even when the simulator is doing so
In the concurrent setting, Main issue: simulation-sound coin tossing
No adv can bias the coin tossing results, even when the simulator is doing so
In the concurrent setting,
Main issue: **simulation-sound** coin tossing

No adv can bias the coin tossing results,
even when the **simulator** is doing so
In the concurrent setting,
Main issue: *simulation-sound* coin tossing

No adv can bias the coin tossing results, even when the *simulator* is doing so

**Random!**
In the concurrent setting, Main issue: simulation-sound coin tossing

No adv can bias the coin tossing results, even when the simulator is doing so

Informally, SH-OT + simulation sound coin tossing

⇒ Ideal OT in concurrent setting
In the concurrent setting, Main issue: simulation-sound coin tossing

No adv can bias the coin tossing results, even when the simulator is doing so

Informally, SH-OT + simulation sound coin tossing
⇒ Ideal OT in concurrent setting
In the concurrent setting, Main issue: simulation-sound coin tossing

No adv can bias the coin tossing results, even when the simulator is doing so.

Informally, SH-OT + simulation sound coin tossing
→ Ideal OT in concurrent setting
Concurrent Coin Tossing from CCA
Concurrent Coin Tossing from CCA

- $\text{ExtCom}(r)$
- $r'$
- open to $r$
Concurrent Coin Tossing from CCA

CCACom(r)

r'

open to r
Concurrent Coin Tossing from CCA
Simulator can bias coins, by using oracle to break CCACom from adv
Simulator can bias coins, by using oracle to break CCACom from adv

The adv cannot bias coins, as the CCACom from honest player is still hiding
Simulator can bias coins, by using oracle to break CCACom from adv

The adv cannot bias coins, as the CCACom from honest player is still hiding

**Theorem 2:** CCA + SH-OT $\rightarrow$ BB implementation of $F_{OT}$
Our Result (informal):

**BB construction of concurrently secure MPC protocols**

- In the plain model
- Assuming Semi-Honest Oblivious Transfer protocols
- Security in the UC with super-poly helper model [CLP10]
  - Implies SPS security
  - Closed under universal composition
Our Result (informal): 

**BB construction of concurrently secure MPC protocols**

- In the plain model
- Assuming Semi-Honest Oblivious Transfer protocols
- Security in the UC with super-poly helper model [CLP10]
  - Implies SPS security
  - Closed under universal composition

BB CCA Commitments
Our Result (informal):  

**BB construction of concurrently secure MPC protocols**

- In the plain model
- Assuming Semi-Honest Oblivious Transfer protocols
- Security in the UC with super-poly helper model [CLP10]
  - Implies SPS security
  - Closed under universal composition

KEY Notion

BB CCA Commitments
Our Result (informal):

**BB construction of concurrently secure MPC protocols**

- In the plain model
- Assuming Semi-Honest Oblivious Transfer protocols
- Security in the UC with super-poly helper model [CLP10]
  - Implies SPS security
  - Closed under universal composition

**BB CCA Commitments**
O(n)-round, better round-complexity?
Thank you!