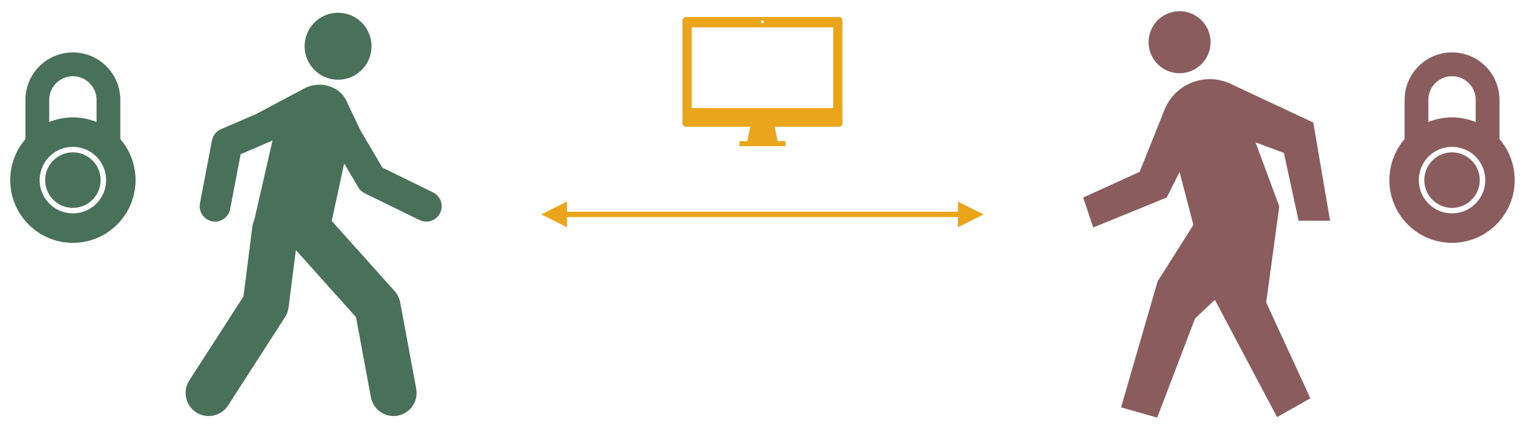
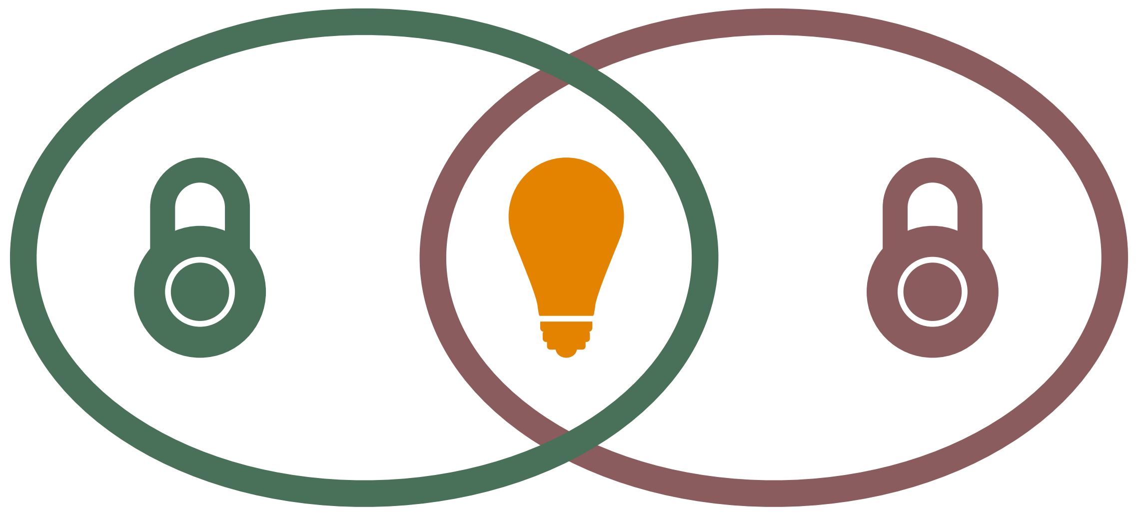
The Concrete Security of Two-Party Computation

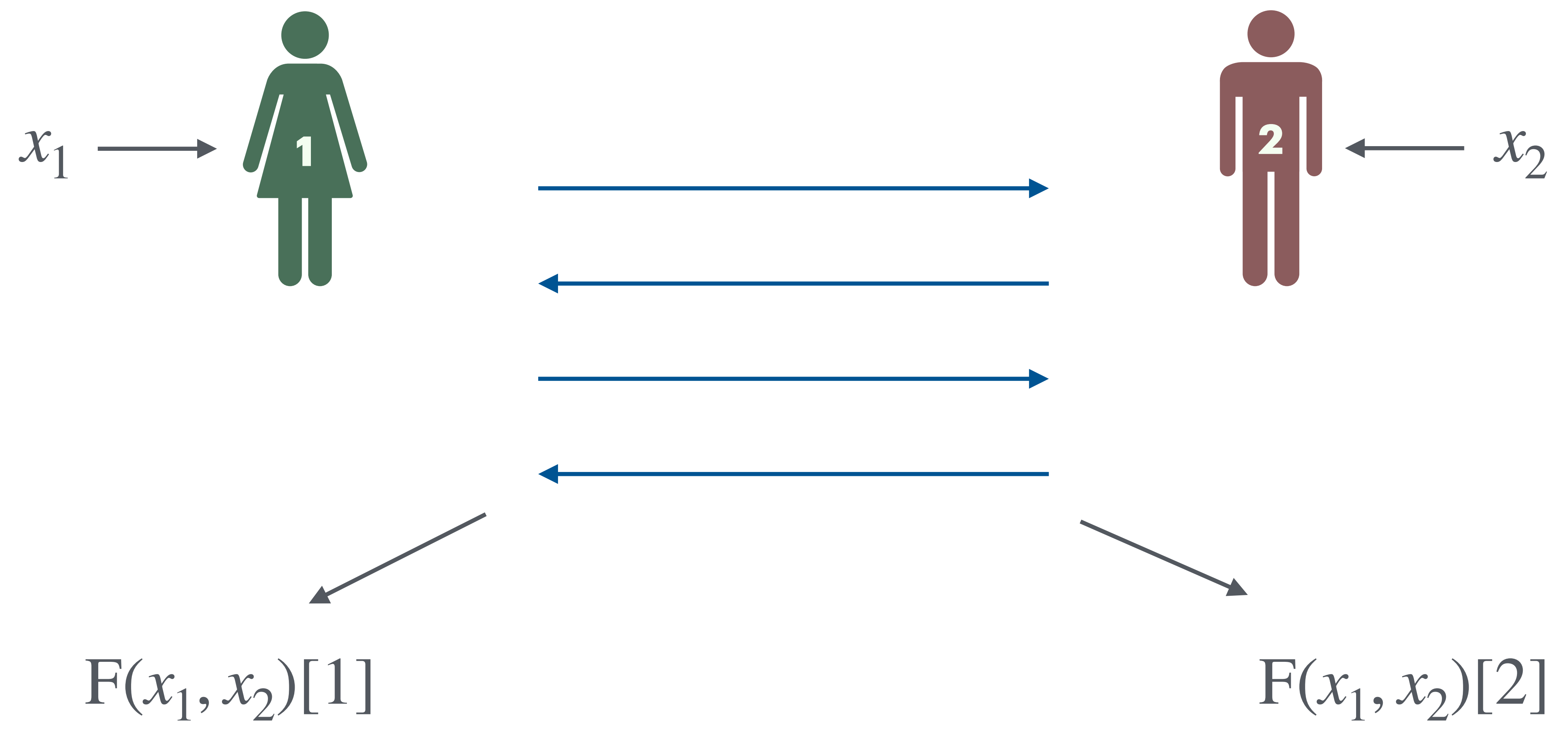
Simple Definitions, and Tight Proofs for PSI and OPRFs

Mihir Bellare, University of California San Diego
Rishabh Ranjan, University of California San Diego
Doreen Riepel, CISPA Helmholtz Center for Information Security
Ali Aldakheel, King Abdulaziz City for Science and Technology





What is two party computation (2PC)?



x_i : Private input of party $i \in \{1,2\}$

F : The 2PC functionality

Example: Private Set Intersection ($\mathrm{F}^{\mathrm{psi}}$)

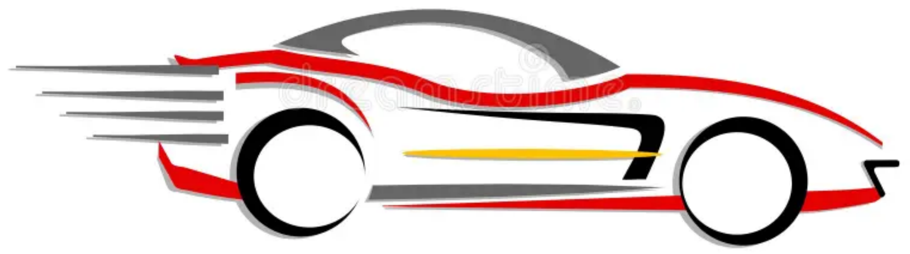
$\mathrm{F}^{\mathrm{psi}}(x_1, x_2)[1]$	$\mathrm{F}^{\mathrm{psi}}(x_1, x_2)[2]$
$(x_1 \cap x_2, \lvert x_2 \rvert)$	$\lvert x_1 \rvert$

Π : Protocol to compute F

Security: Party $i \in \{1,2\}$ should not learn more about x_{3-i} than it could compute from $\mathrm{F}(x_1, x_2)[i]$.

2PC research

Theory



Practice

Protocols for arbitrary functionalities

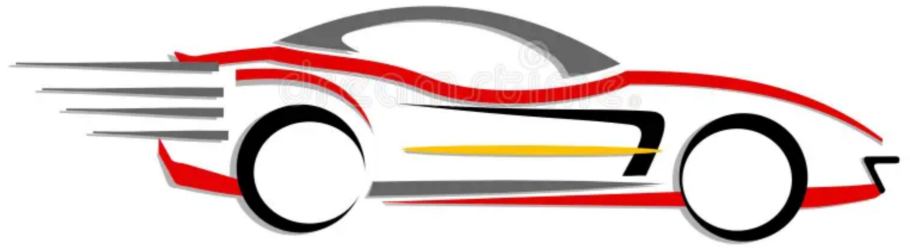
Security Proofs based on general assumptions like OT, OWFs etc.

Polynomial time protocols

Asymptotic Security

2PC research

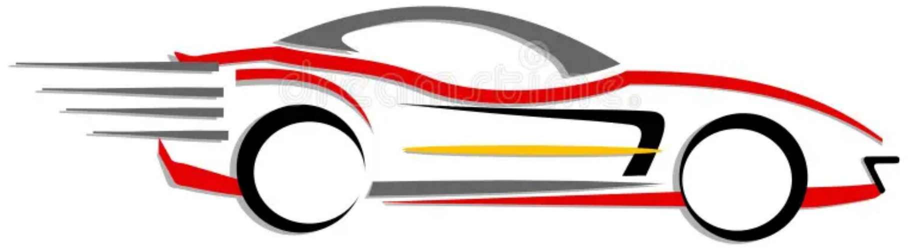
Theory



Practice

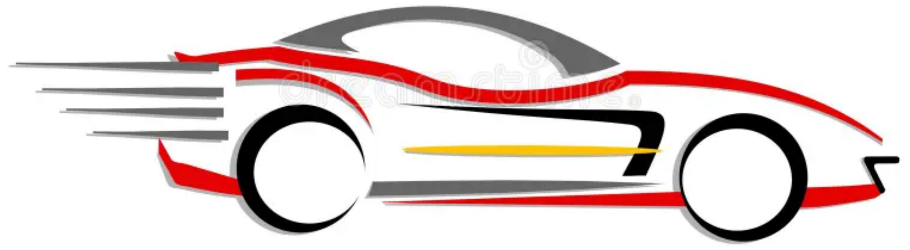
Theory	Practice
Protocols for arbitrary functionalities	✓ Protocols for particular Functionalities (eg. PSI, OPRF)
Security Proofs based on general assumptions like OT, OWFs etc.	✓ Security Proofs in the Random Oracle Model, based on particular computational assumptions (eg. Discrete log)
Polynomial time protocols	✓ Fast protocols
Asymptotic Security	

2PC research

Theory		Practice
Protocols for arbitrary functionalities	✓	Protocols for particular Functionalities (eg. PSI, OPRF)
Security Proofs based on general assumptions like OT, OWFs etc.	✓	Security Proofs in the Random Oracle Model, based on particular computational assumptions (eg. Discrete log)
Polynomial time protocols	✓	Fast protocols
Asymptotic Security	✗	Concrete Security

2PC research

Theory		Practice
Protocols for arbitrary functionalities	✓	Protocols for particular Functionalities (eg. PSI, OPRF)
Security Proofs based on general assumptions like OT, OWFs etc.	✓	Security Proofs in the Random Oracle Model, based on particular computational assumptions (eg. Discrete log)
Polynomial time protocols	✓	Fast protocols
Asymptotic Security	✗	Concrete Security

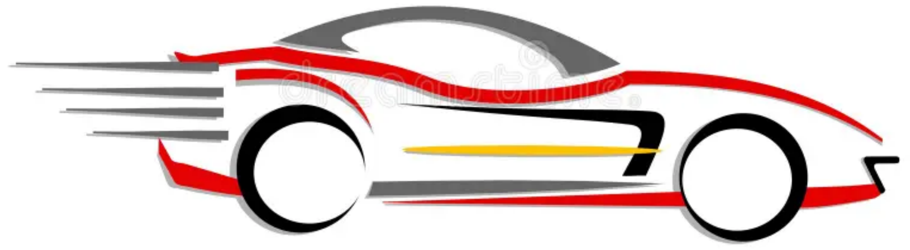


Can't pick parameters to guarantee a desired level of proven security.

Unclear how many bits of security an implementation provides.

2PC research

Theory



Practice

Theory		Practice
Protocols for arbitrary functionalities	✓	Protocols for particular Functionalities (eg. PSI, OPRF)
Security Proofs based on general assumptions like OT, OWFs etc.	✓	Security Proofs in the Random Oracle Model, based on particular computational assumptions (eg. Discrete log)
Polynomial time protocols	✓	Fast protocols
Asymptotic Security	✓	Concrete Security

We fill this gap

Now can pick parameters to guarantee a desired level of proven security for an implementation.

Our contributions in brief

Initiate the study of concrete security
for Two Party Computation

1. **Definitions**

   Input Indistinguishability (InI): A 2PC security definition that

   - Is indistinguishability based
   - Yet equivalent to simulation for PSI and friends
   - Concrete security and cryptanalysis friendly

Our contributions in brief

Initiate the study of concrete security
for Two Party Computation

1. Definitions

Input Indistinguishability (InI): A 2PC security definition that

- Is indistinguishability based
- Yet equivalent to simulation for PSI and friends
- Concrete security and cryptanalysis friendly

Definitions explicitly incorporate ROM and
surface subtleties in this regard

Our contributions in brief

Initiate the study of concrete security for Two Party Computation

1. **Definitions**

   Input Indistinguishability (InI): A 2PC security definition that

   - Is indistinguishability based
   - Yet equivalent to simulation for PSI and friends
   - Concrete security and cryptanalysis friendly

Definitions explicitly incorporate ROM and surface subtleties in this regard

2. **Concrete security results for PSI and OPRFs**

Our contributions in brief

Initiate the study of concrete security for Two Party Computation

1. **Definitions**

   Input Indistinguishability (InI): A 2PC security definition that

   - Is indistinguishability based
   - Yet equivalent to simulation for PSI and friends
   - Concrete security and cryptanalysis friendly

   Definitions explicitly incorporate ROM and surface subtleties in this regard

2. **Concrete security results for PSI and OPRFs**

   - CDH, V-CDH, CDH-MUC —non-tight→ 2H-DH OPRF
   - V-CDH-MUC, DDH —tight→ 2H-DH OPRF
   - 2H-DH OPRF —tight→ DH-PSI

3. **Salted DH-PSI**

   New PSI protocol, as efficient as DH-PSI, but

   - CDH, CDH-MUC —more tight→ Salted DH-PSI
   - V-CDH, V-CDH-MUC, DDH —tight→ Salted DH-PSI

Our contributions in brief

Initiate the study of concrete security for Two Party Computation

1. **Definitions**

   Input Indistinguishability (InI): A 2PC security definition that

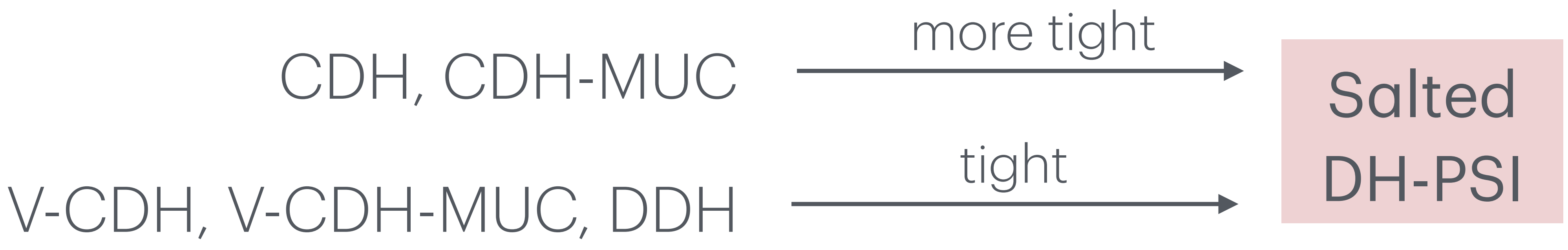
   - Is indistinguishability based
   - Yet equivalent to simulation for PSI and friends
   - Concrete security and cryptanalysis friendly

Definitions explicitly incorporate ROM and surface subtleties in this regard

2. **Concrete security results for PSI and OPRFs**

3. **Salted DH-PSI**

   New PSI protocol, as efficient as DH-PSI, but



Our definitions and results are for the semi-honest (honest-but-curious) setting

Remarks

Concrete security started with Bellare and Rogaway in the 1990s.

It is the norm in proofs for symmetric cryptography, applied public-key cryptography and authenticated key exchange.

Large body of work on proof/reduction tightness in these areas.

Work on concrete security of garbling schemes [BHKR13, ZRE14, GKWWY19, GLNP23,...].

We are bringing this to 2PC and PSI.

Opens up new research directions:

- Give concrete security results for existing 2PC protocols
- Give new protocols with tight security

Allows sound choices of parameters (groups) in practice for a desired number of bits of security.

Plan

- ☐ Background: Asymptotic and Concrete security
- ☐ Definitions and Relations
- ☐ Results for DH PSI
- ☐ Salted DH-PSI

Asymptotic Security

Given: A protocol or scheme Π
That targets achieving a security notion T
Based on the assumption that problem P is hard

Concrete Security

Asymptotic Security

Given: A protocol or scheme Π

That targets achieving a security notion T

Based on the assumption that problem P is hard

A —Reduction→ A'

Adversary attacking T-security of Π

Adversary attacking P

Concrete Security

Asymptotic Security

Concrete Security

Given: A protocol or scheme Π

That targets achieving a security notion T

Based on the assumption that problem P is hard

A —Reduction→ A'

Adversary attacking T-security of Π

Adversary attacking P

If A runs in polynomial time and has advantage that is **not negligible** then A' runs in polynomial time and has advantage that is **not negligible**

Asymptotic Security

Concrete Security

Given: A protocol or scheme Π

That targets achieving a security notion T

Based on the assumption that problem P is hard

A —Reduction→ A'

Adversary attacking T-security of Π

Adversary attacking P

If A runs in polynomial time and has advantage that is **not negligible** then A' runs in polynomial time and has advantage that is **not negligible**

If A runs in time t and has advantage that is $\epsilon = \mathbf{Adv}_{\Pi}^{\mathrm{T}}(\mathrm{A})$ then A′ runs in time about t and has advantage ϵ' such that $\epsilon \leq \mathrm{B}(\epsilon')$

The Bound, eg. $B(\epsilon') = 2\epsilon'$

Asymptotic Security

Concrete Security

Given: A protocol or scheme Π
That targets achieving a security notion T
Based on the assumption that problem P is hard

A —Reduction→ A'

Adversary attacking T-security of Π

Adversary attacking P

If A runs in polynomial time
and has advantage that is **not negligible**
then A' runs in polynomial time
and has advantage that is **not negligible**

If A runs in time
t
and has advantage that is
$\epsilon = \mathbf{Adv}_{\Pi}^{\mathrm{T}}(\mathrm{A})$
then A' runs in time about
t
and has advantage
ϵ' such that $\epsilon \leq \mathrm{B}(\epsilon')$

The Bound, eg. $B(\epsilon') = 2\epsilon'$

How many bits **s** of security does Π have on a 256 bit curve with $\mathrm{P} = \mathrm{DL}$?

?

s could be close to 0.

!

s = 127 bits

Plan

- ☑ Background: Asymptotic and Concrete security
- ☐ Definitions and Relations
- ☐ Results for DH PSI
- ☐ Salted DH-PSI

Single-quantifier definitions

$$\forall A \;\; \mathrm{Adv}^{T}_{\Pi}(A) \leq \epsilon$$

IND-CPA, IND-CCA, UF-CMA, AKE, ...

All indistinguishability-based definitions

Concrete-security friendly.

This is the type assumed in the prior discussion of concrete security.

Double-quantifier definitions

$$\exists S \, \forall A \;\; \mathrm{Adv}^{T}_{\Pi,S}(A) \leq \epsilon$$

All simulation-based definitions.

Single-quantifier definitions

$\forall A \;\; \mathrm{Adv}^{T}_{\Pi}(A) \leq \epsilon$

IND-CPA, IND-CCA, UF-CMA, AKE, ...
All indistinguishability-based definitions

Concrete-security friendly.
This is the type assumed in the prior discussion of concrete security.

Double-quantifier definitions

$\exists S \forall A \;\; \mathrm{Adv}^{T}_{\Pi,S}(A) \leq \epsilon$

All simulation-based definitions.

Concrete-security unfriendly.
Concrete security would need to bring in simulator and its running time.

Intuitively capture strong security.
Traditional in 2PC.
General composition theorems.

Single-quantifier definitions

$$\forall A \;\; \mathrm{Adv}^{T}_{\Pi}(A) \leq \epsilon$$

IND-CPA, IND-CCA, UF-CMA, AKE, ...
All indistinguishability-based definitions

Concrete-security friendly.
This is the type assumed in the prior discussion of concrete security.

Can we have the best of both worlds?

Double-quantifier definitions

$$\exists S \forall A \;\; \mathrm{Adv}^{T}_{\Pi,S}(A) \leq \epsilon$$

All simulation-based definitions.

Concrete-security unfriendly.
Concrete security would need to bring in simulator and its running time.

Intuitively capture strong security.
Traditional in 2PC.
General composition theorems.

Single-quantifier definitions

$$\forall \mathrm{A} \;\; \mathrm{Adv}_{\Pi}^{\mathrm{T}}(\mathrm{A}) \leq \epsilon$$

Double-quantifier definitions

$$\exists \mathrm{S} \forall \mathrm{A} \;\; \mathrm{Adv}_{\Pi,\mathrm{S}}^{\mathrm{T}}(\mathrm{A}) \leq \epsilon$$

Can we have the best of both worlds?

History says YES for encryption

Single-quantifier definitions

$\forall \mathrm{A} \;\; \mathrm{Adv}_{\Pi}^{\mathrm{T}}(\mathrm{A}) \leq \epsilon$

Double-quantifier definitions

$\exists \mathrm{S} \forall \mathrm{A} \;\; \mathrm{Adv}_{\Pi,\mathrm{S}}^{\mathrm{T}}(\mathrm{A}) \leq \epsilon$

Indistinguishability for public-key and symmetric encryption

EQUIVALENT!

⟷

[GM,Go,BDJR]

Semantic security for public-key and symmetric encryption

Can we have the best of both worlds?

History says YES for encryption

Single-quantifier definitions

$\forall A \;\; \mathrm{Adv}^{T}_{\Pi}(A) \leq \epsilon$

Double-quantifier definitions

$\exists S \forall A \;\; \mathrm{Adv}^{T}_{\Pi,S}(A) \leq \epsilon$

Indistinguishability for public-key and symmetric encryption

EQUIVALENT!

⟷

[GM,Go,BDJR]

Semantic security for public-key and symmetric encryption

Can we have something like this **for 2PC**?

Single-quantifier definitions

$\forall A \;\; \mathrm{Adv}^{T}_{\Pi}(A) \leq \epsilon$

Double-quantifier definitions

$\exists S \forall A \;\; \mathrm{Adv}^{T}_{\Pi,S}(A) \leq \epsilon$

EQUIVALENT!

For a class of functionalities including PSI and friends

InI ⟷ SIM

[BRRA]

Can we have something like this **for 2PC**?

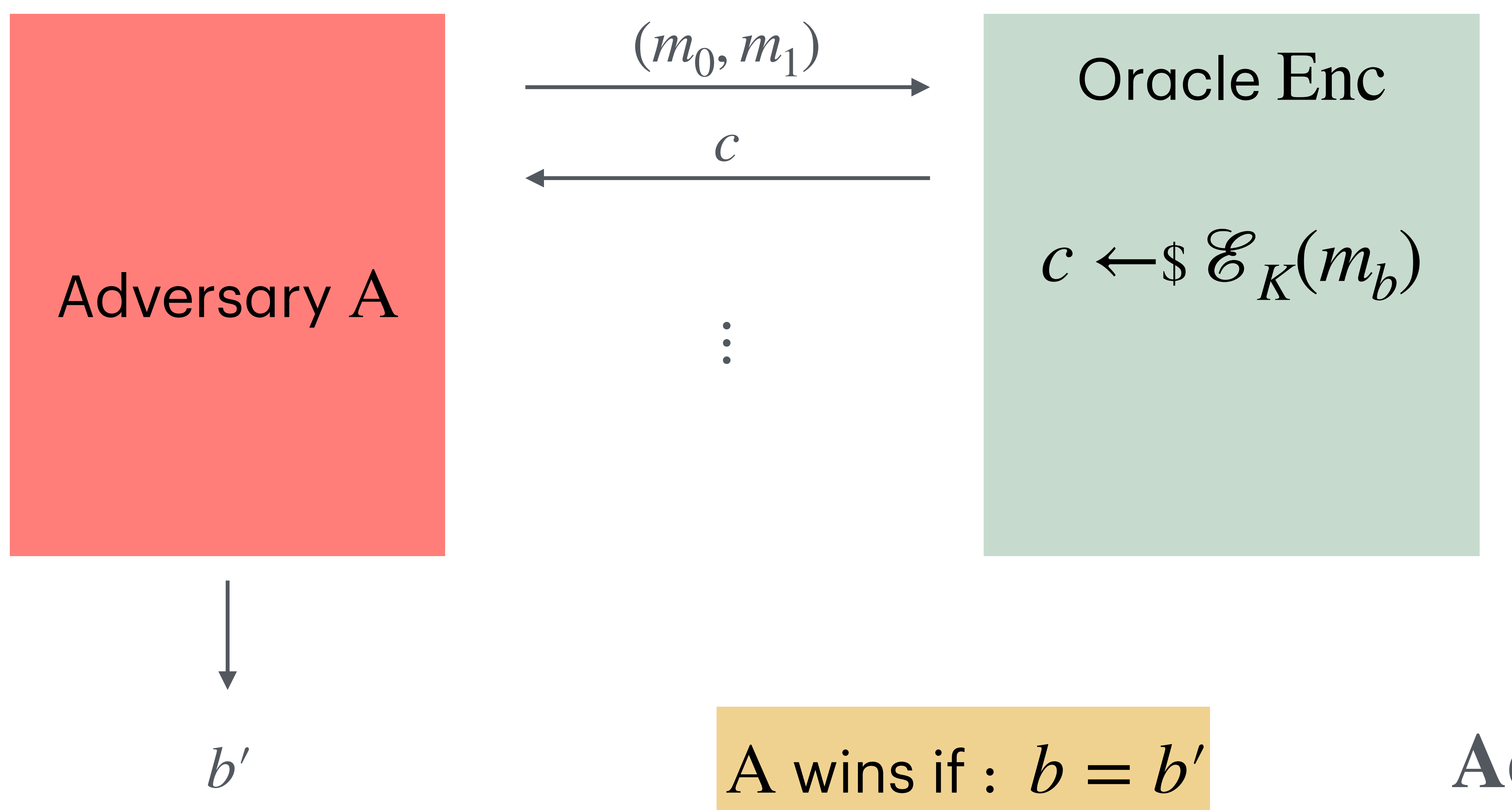
We say YES for 2PC

Recall: Indistinguishability for (randomized, symmetric) encryption [BDJR97]

Given: Symmetric encryption scheme $\mathscr{E}$ with key-space Keys

$b \xleftarrow{\$} \{0,1\}$ ← Challenge bit

$K \xleftarrow{\$} \text{Keys}$



A wins if : $b = b'$

$$\mathbf{Adv}^{\mathrm{ind}}_{\mathscr{E}}(A) = 2\Pr[b' = b] - 1$$

Games $\mathbf{G}^{\mathrm{ind}}_{\mathcal{E}}$

INITIALIZE():
$b \leftarrow\$ \{0,1\}$; $K \leftarrow\$ \text{Keys}$

ENC(m_0, m_1):
$c \leftarrow\$ \mathcal{E}_K(m_b)$
Return c

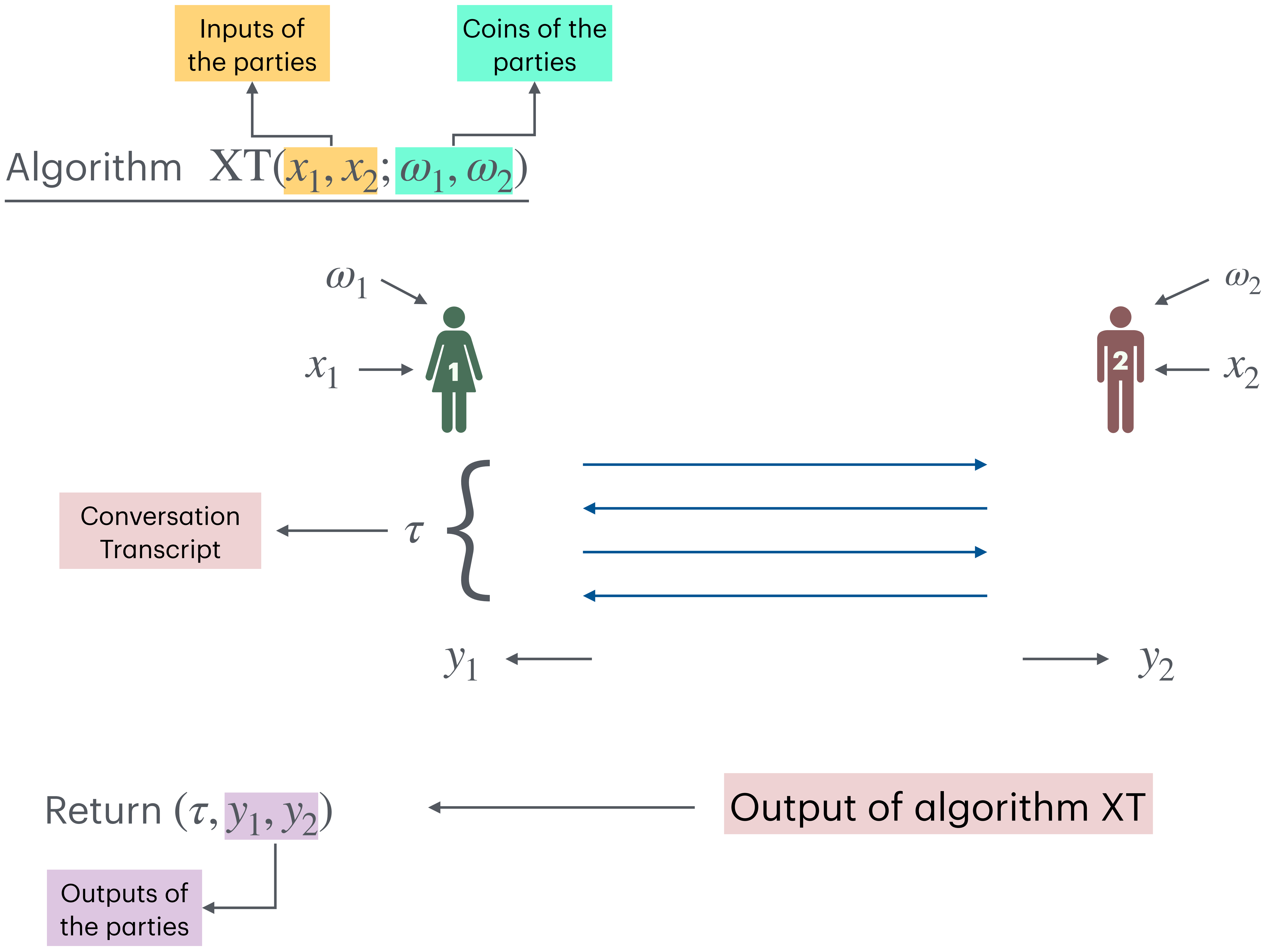
FINALIZE(b'):
Return $[[b = b']]$

From encryption to 2PC

	Encryption	2PC
Adversary provides	Messages m_0, m_1	Inputs $x_{2,0}, x_{2,1}$ for the honest party (say party 2) Also an input x_1 for the dishonest party
Adversary receives	Ciphertext $C \leftarrow\$ \mathscr{E}_K(m_b)$	Conversation transcript, output and coins of dishonest party from execution of Π on $x_1, x_{2,b}$
Restriction to avoid trivial win	Lengths of m_0, m_1 must be equal	$\mathrm{F}(x_1, x_{2,0})[1]$ and $\mathrm{F}(x_1, x_{2,1})[1]$ must be equal

Given: Protocol Π for functionality F

We first define **algorithm XT** that takes
the parties inputs and coins,
and returns
the conversation transcript and party outputs
from the execution of protocol Π

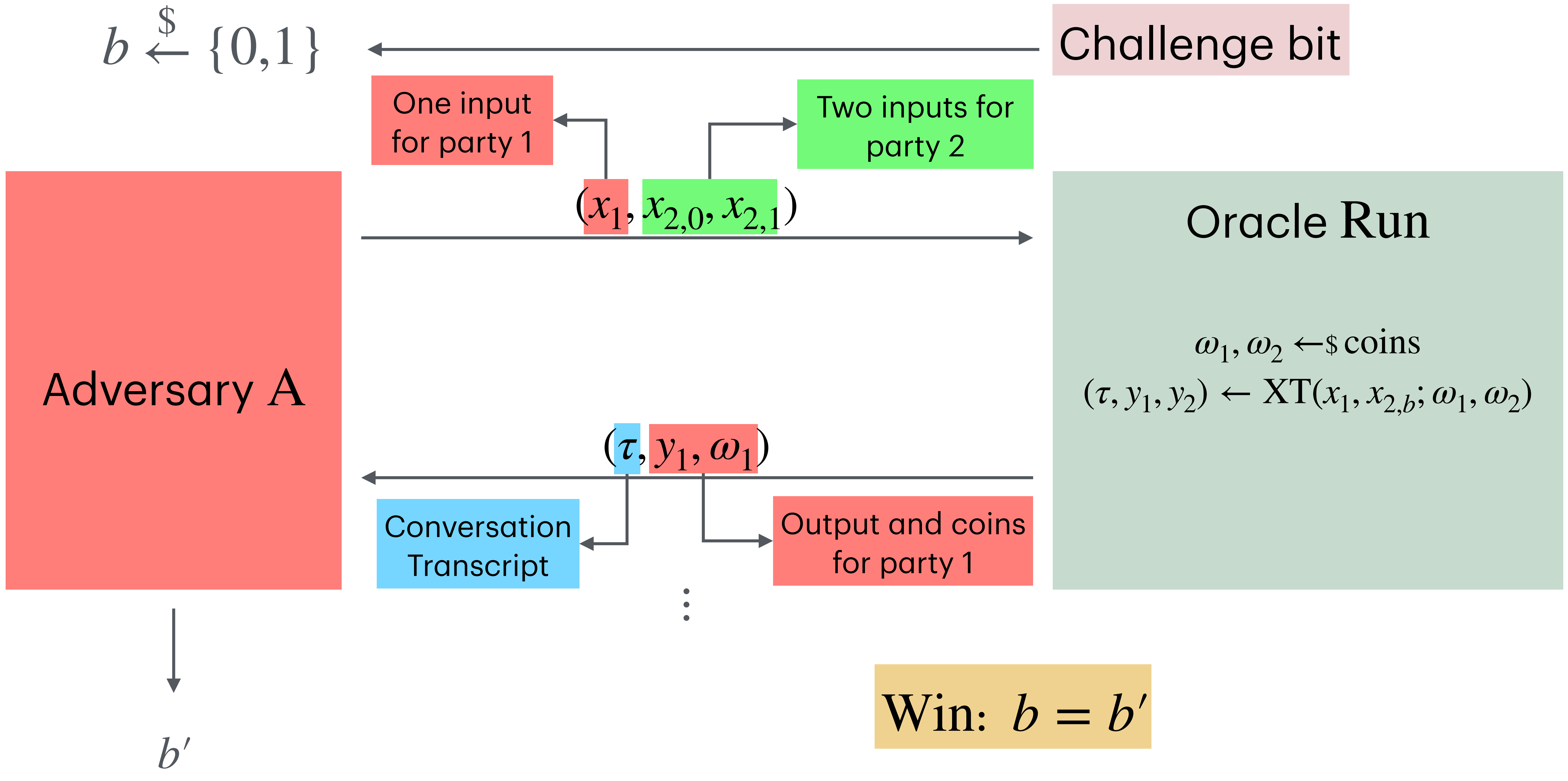


Our Input Indistinguishability (InI) definition for 2PC

Given: Protocol Π for functionality F

Let party 2 be the honest party.

Adversary plays party 1



Advantage of adversary A:

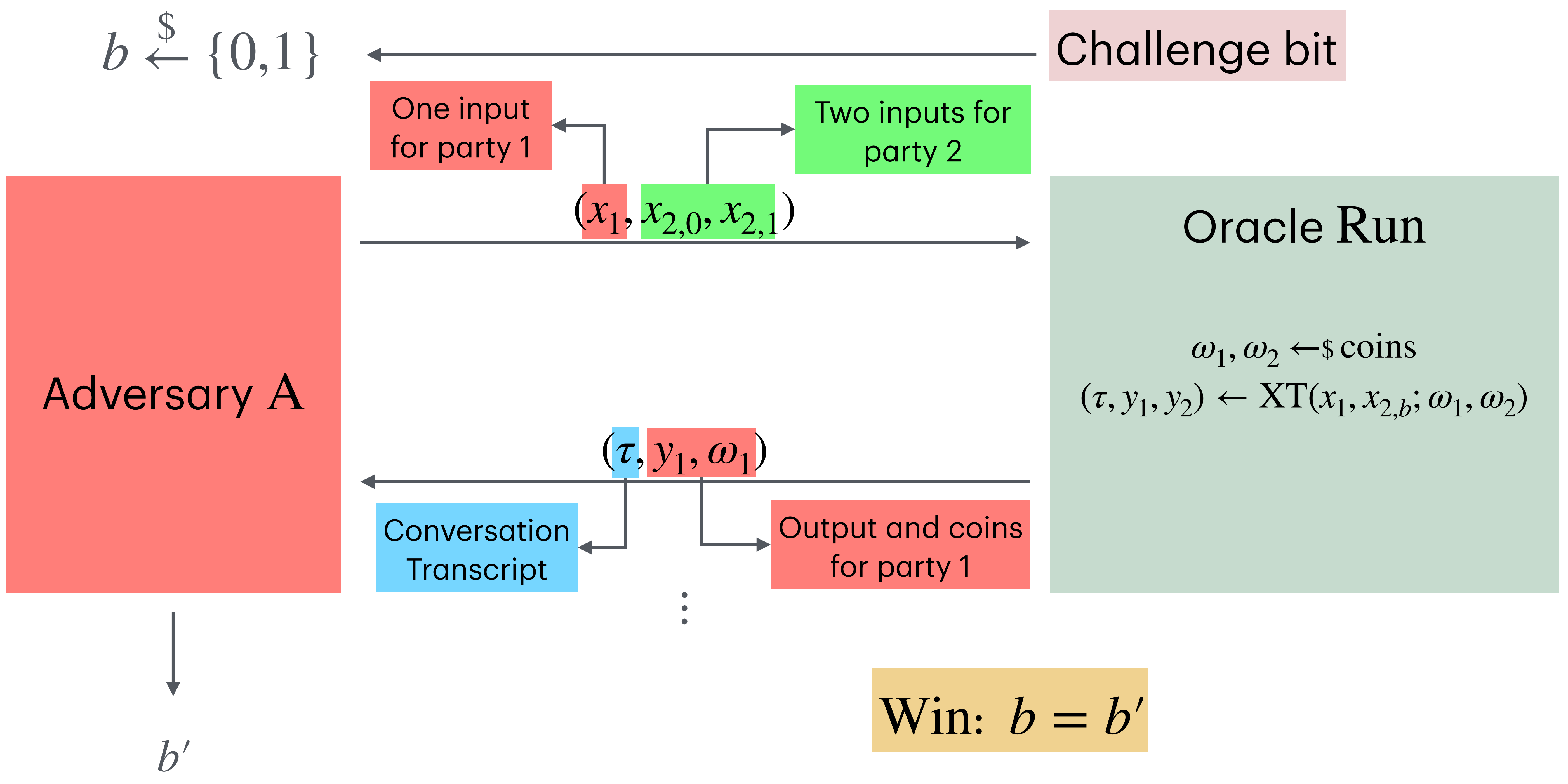
$$\mathrm{Adv}^{\mathrm{ini}}_{\mathrm{F},\Pi,2}(\mathrm{A}) = 2 \cdot \Pr[\mathrm{Win}] - 1$$

Our Input Indistinguishability (InI) definition for 2PC

Given: Protocol Π for functionality F

Let party 2 be the honest party.

Adversary plays party 1



Problem!

We know that $y_1 = \mathrm{F}(x_1, x_{2,b})[1]$

So if $\mathrm{F}(x_1, x_{2,0})[1] \neq \mathrm{F}(x_1, x_{2,1})[1]$

then A can trivially win.

Advantage of adversary A:

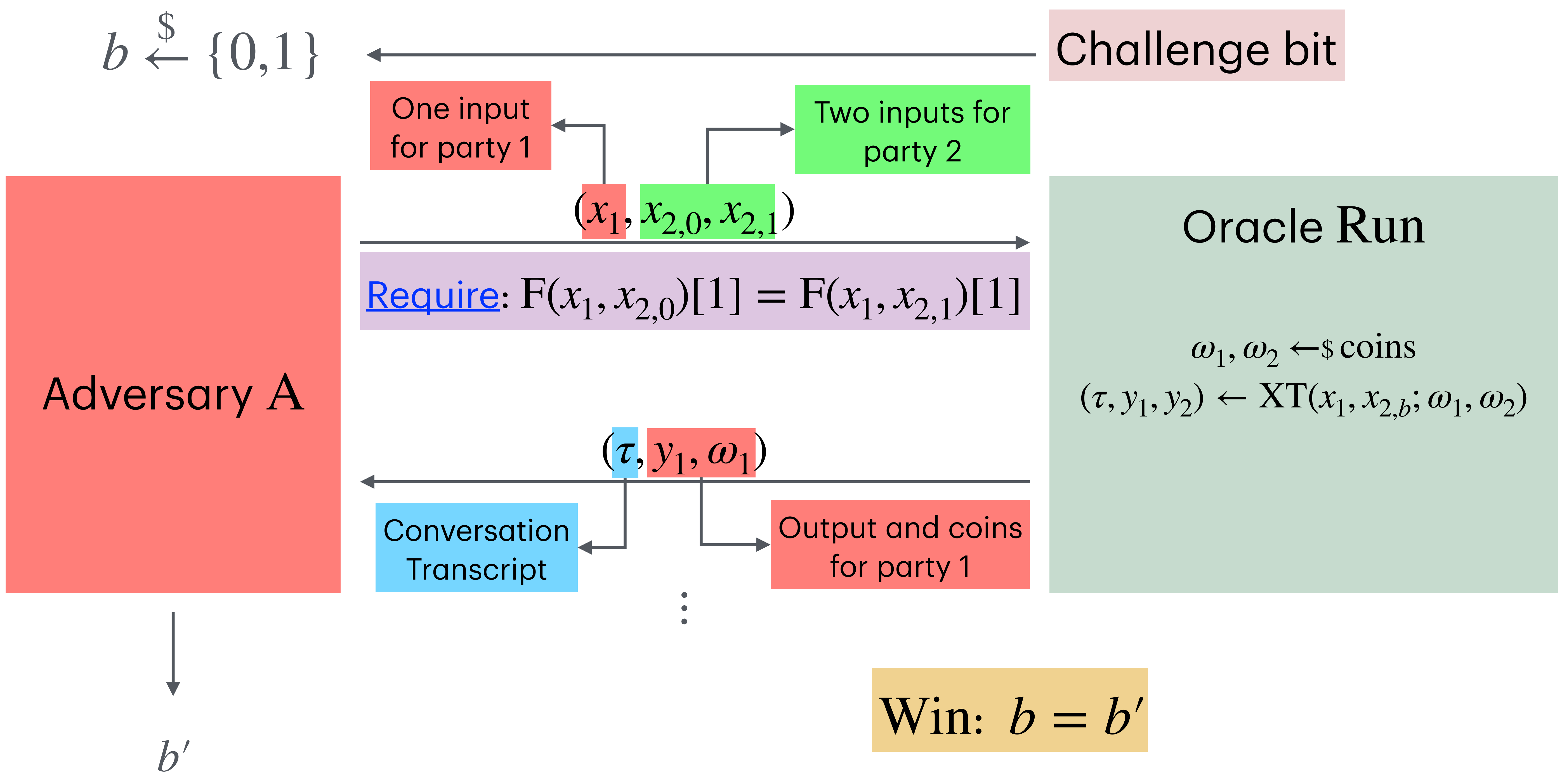
$$\mathrm{Adv}^{\mathrm{ini}}_{\mathrm{F},\Pi,2}(\mathrm{A}) = 2 \cdot \Pr[\mathrm{Win}] - 1$$

Our Input Indistinguishability (InI) definition for 2PC

Given: Protocol Π for functionality F

Let party 2 be the honest party.

Adversary plays party 1



Advantage of adversary A**:**

$$\mathrm{Adv}^{\mathrm{ini}}_{\mathrm{F},\Pi,2}(\mathrm{A}) = 2 \cdot \Pr[\mathrm{Win}] - 1$$

Problem!

We know that $y_1 = \mathrm{F}(x_1, x_{2,b})[1]$

So if $\mathrm{F}(x_1, x_{2,0})[1] \neq \mathrm{F}(x_1, x_{2,1})[1]$

then A can trivially win.

Solution

The "Require" check ensures this does not happen.

Input Indistinguishability (InI)

Pick random oracle H from a scheme-prescribed space OS.
Pick challenge bit b.

Game $\mathbf{G}^{\mathrm{ini}}_{\mathsf{F},\Pi,2}$

INITIALIZE:
$\mathsf{H} \leftarrow_{\$} \mathsf{OS}$; $b \leftarrow_{\$} \{0,1\}$

Adversary calls Run oracle with a pair of inputs $x_{2,0}, x_{2,1}$ for the honest party and a single input x_1 for the dishonest party. Multiple queries to Run allowed!

RUN$(x_1, x_{2,0}, x_{2,1})$:
$(y_{1,0}, y_{2,0}) \leftarrow \mathsf{F}[\mathsf{H}](x_1, x_{2,0})$
$(y_{1,1}, y_{2,1}) \leftarrow \mathsf{F}[\mathsf{H}](x_1, x_{2,1})$
If $(y_{1,0} \neq y_{1,1})$ then return $\perp$
$\omega_1, \omega_2 \leftarrow_{\$} \Omega$
$(\tau, y_1, y_2) \leftarrow \mathbf{XT}_{\Pi}[\mathsf{H}](x_1, x_{2,b}; \omega_1, \omega_2)$
Return (τ, y_1, ω_1)

Avoid trivial attack by ensuring that $x_{2,0}, x_{2,1}$ result in the same functionality outputs for the dishonest party.

Compute conversation transcript and protocol outputs for protocol execution with inputs x_1 and $x_{2,b}$.

Return conversation transcript, and output and coins of dishonest party, to adversary.

RO(X):
Return $\mathsf{H}(X)$

Random Oracle

FINALIZE(b'):
Return $[[b' = b]]$

Takes adversary guess b' and returns true iff $b' = b$.

Advantage of adversary A: $\mathrm{Adv}^{\mathrm{ini}}_{\mathrm{F},\Pi,2}(\mathrm{A}) = 2 \cdot \Pr[\mathrm{G}^{\mathrm{ini}}_{\mathrm{F},\Pi,2}(\mathrm{A})] - 1$

Our Simulation-based (SIM, SIM-np) definitions for 2PC

- We specify these using games.
- The games are parameterized by a simulator S.
- Similar to InI, the game randomly picks a challenge bit b.
- Oracle Run takes inputs x_1, x_2 for the parties and returns the view of the dishonest party (party 1), generated as follows
  * Case $b = 1$: via execution of the protocol Π on inputs x_1, x_2
  * Case $b = 0$: by the simulator S given the functionality output $\mathrm{F}[\mathrm{H}](x_1, x_2)[1]$.
- Difference between SIM and SIM-np is in the output of the random oracle when $b = 0$:
  * SIM: simulator programs the output of random oracle
  * SIM-np: same, honest random oracle used for both values of b

Advantage of adversary A: $\mathrm{Adv}^{\mathrm{X}}_{\mathrm{F},\mathrm{S},\Pi,2}(\mathrm{A}) = 2 \cdot \Pr[\mathrm{G}^{\mathrm{X}}_{\mathrm{F},\Pi,\mathrm{S},2}(\mathrm{A})] - 1, \quad$ for $\mathrm{X} \in \{\mathrm{sim}, \mathrm{sim\text{-}np}\}$

Subtle point about RO in SIM

Some functionalities use the random oracle RO.
For example, the functionality F underlying the 2H-DH OPRF.

RO queries are thus made by the adversary, protocol and functionality.
In a programmable-ROM simulation-based definition, we would expect ALL these queries to be answered by the simulator.

But we show this to be WRONG for functionality queries.
If functionality queries are answered by the simulator, obviously insecure protocols can be proven secure.
In the paper, we give a counterexample to show this.

Our SIM definition handles this via a new definitional approach.
The game picks an honest random function H which is used to answer functionality queries.
The simulator can access H and must then itself answer adversary and protocol RO queries.

Remarks on our definitions

Multiple queries to Run oracle allowed to capture multiple executions of protocol on different inputs.

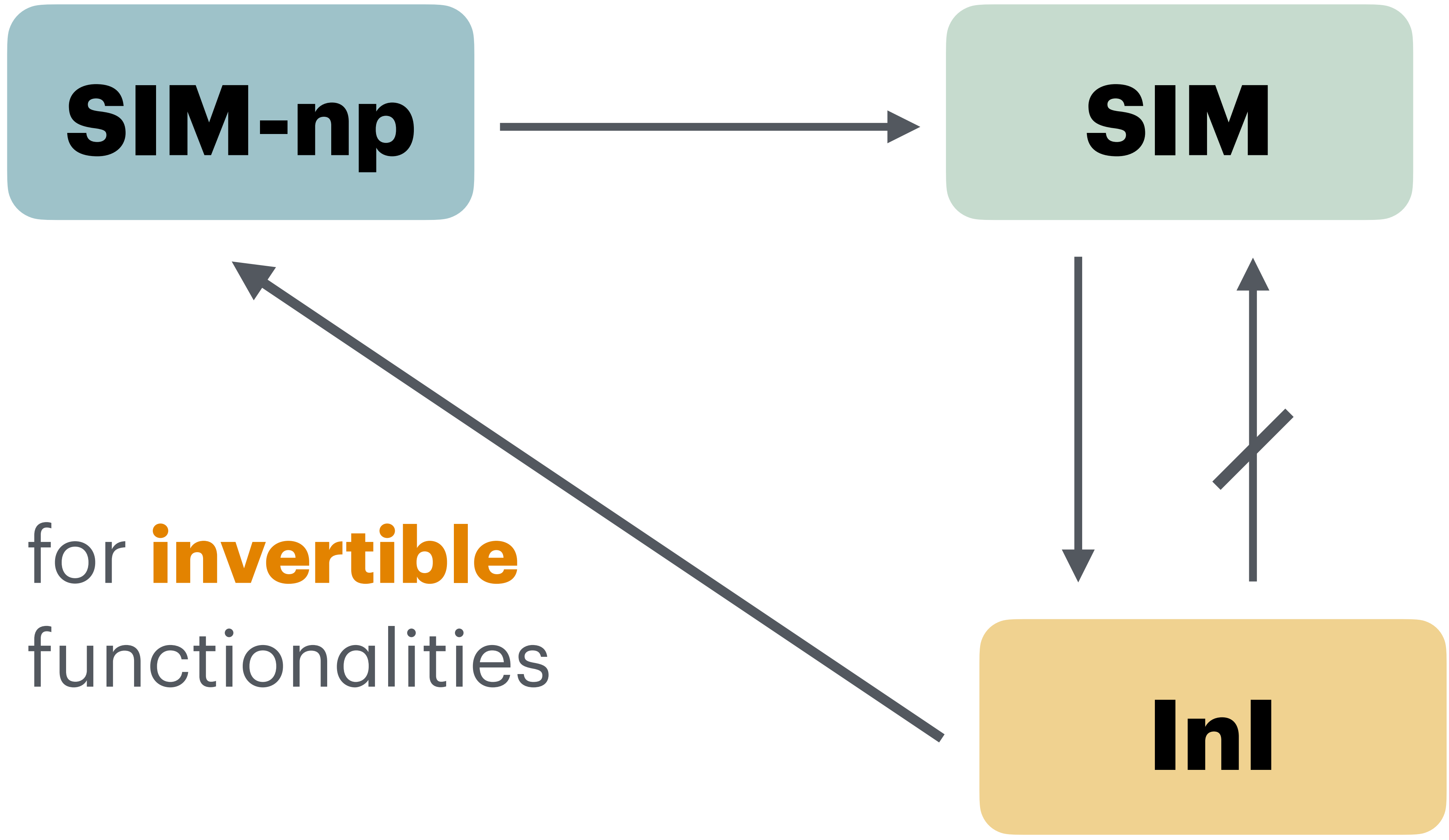
We want to see how adversary advantage degrades concretely as a function of the number q_{Run} of queries it makes to Run.

ROM explicitly incorporated in the games.

Schemes name space OS from which their RO H is drawn to allow scheme-dependent ranges for H.

RO is not programmed in InI and SIM-np. It is programmed in SIM.

Relations between definitions



$\mathbf{A} \longrightarrow \mathbf{B}$: An Implication

For any protocol Π for any functionality F:

If Π is $\mathbf{A}$-secure then it is also $\mathbf{B}$-secure.

$\mathbf{B} \not\longrightarrow \mathbf{A}$: A separation

There exists a protocol Π for some functionality F such that:

Π is $\mathbf{B}$-secure but NOT $\mathbf{A}$-secure.

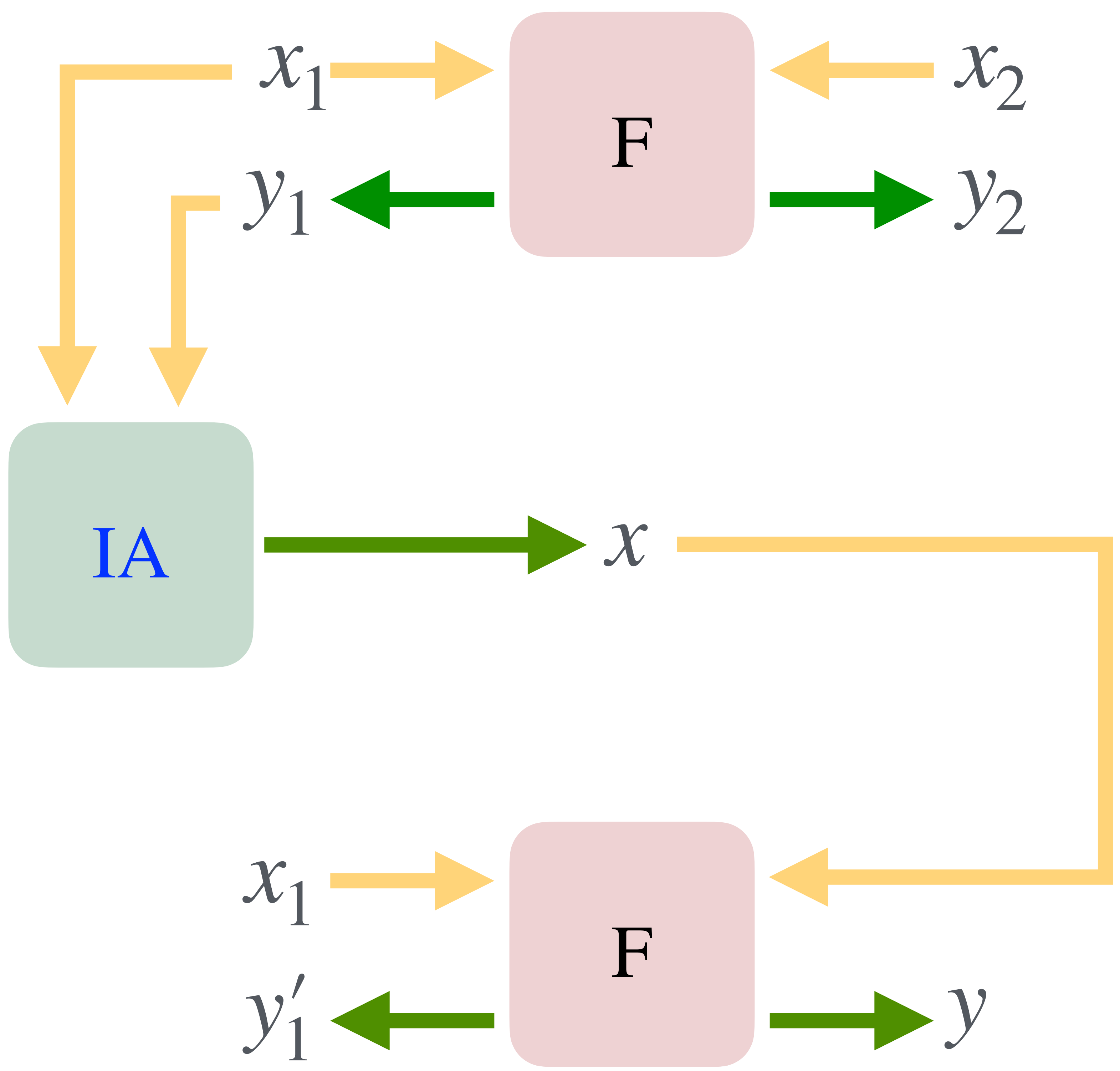
SIM, SIM-np always imply InI

Main Result: InI implies SIM-np and SIM whenever the functionality F satisfies a condition, called **invertibility**, that we define.

We show that PSI and related functionalities are **invertible**.
So for these we have the best-of-both-worlds.

Invertibility

A functionality F is **invertible** with respect to party h (here we let $h = 2$) if there exists an efficient algorithm IA, called the **inverter**, such that for every input x_1, x_2 the check below is always true:



The check: $y_1 = y_1'$

Invertibility with respect to party 2

Given the input and output for party 1

x_1, y_1,

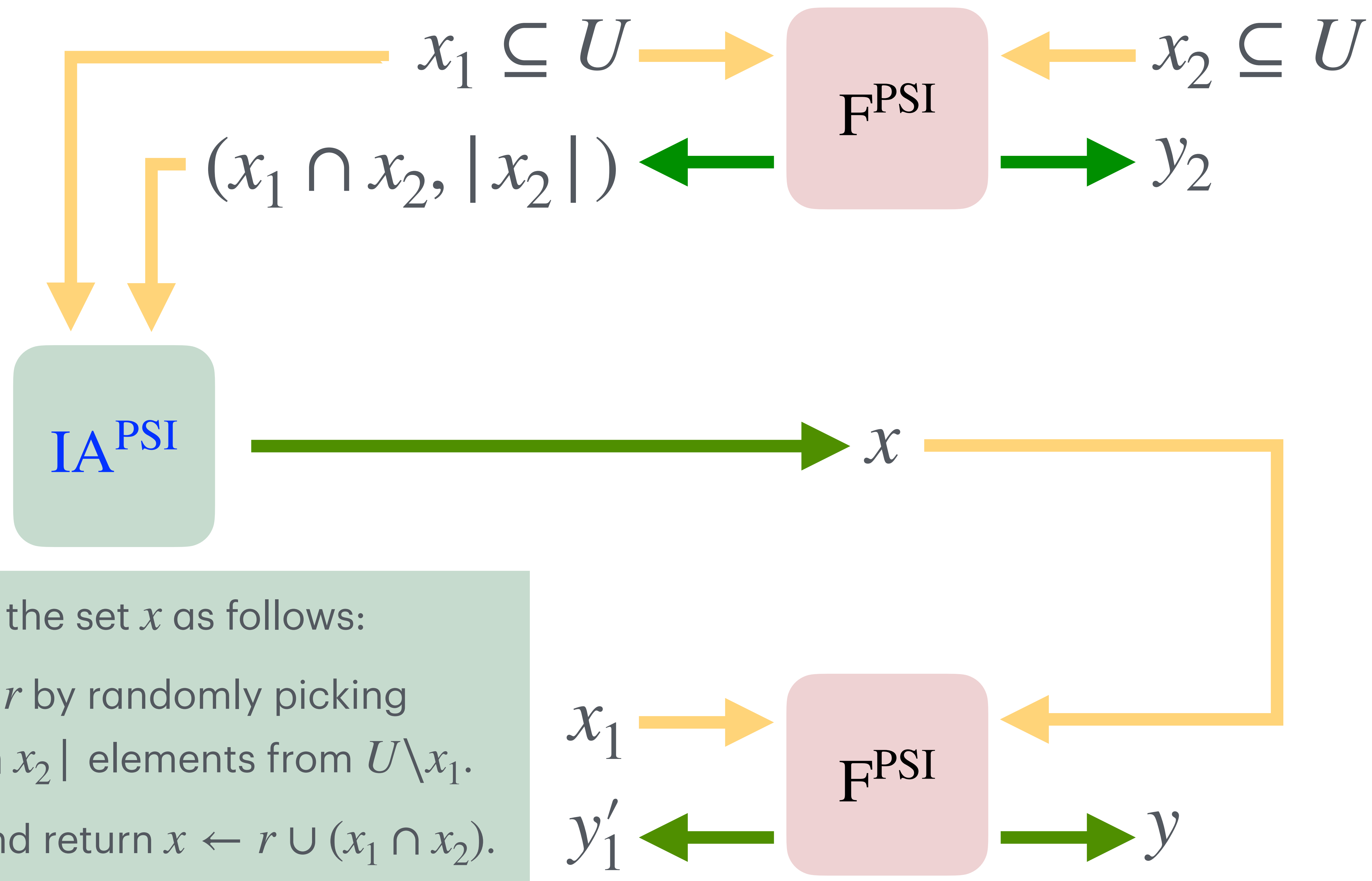
the inverter IA produces an input for party 2,

x

such that

$\mathrm{F}(x_1, x)[1] = y_1$.

Invertibility for PSI



Inverter creates the set x as follows:

1. Create a set r by randomly picking $|x_2| - |x_1 \cap x_2|$ elements from $U \backslash x_1$.
2. Construct and return $x \leftarrow r \cup (x_1 \cap x_2)$.

An inverter with respect to party 1 also exists.

Invertibility for PSI and friends

Our paper similarly shows invertibility for numerous PSI-related functionalities

Friends

Threshold Private Set Intersection (F_t^{tpsi})

$F_t^{tpsi}(x_1, x_2)[1]$	$F_t^{tpsi}(x_1, x_2)[2]$
$I \leftarrow \begin{cases} x_1 \cap x_2 & \text{if } \lvert x_1 \cap x_2 \rvert \geq t \\ \bot & \text{otherwise} \end{cases}$ $(I, \lvert x_2 \rvert)$	$\lvert x_1 \rvert$

Cardinality Private Set Intersection (F^{cpsi})

$F^{cpsi}(x_1, x_2)[1]$	$F^{cpsi}(x_1, x_2)[2]$
$(\lvert x_1 \cap x_2 \rvert, \lvert x_2 \rvert)$	$\lvert x_1 \rvert$

Conclusion: For PSI and friends

the simple single-quantifier, concrete-security-friendly InI definition

is equivalent to

the double-quantifier, strong SIM definition

This allows us to safely target InI for concrete security

Plan

- ☑ Background: Asymptotic and Concrete security
- ☑ Definitions and Relations
- ☐ Results for DH PSI
- ☐ Salted DH-PSI

The DH PSI protocol

- Hazay and Lindell [HL08] gave a PSI protocol (HL-PSI) using Oblivious Pseudorandom Functions (OPRFs).
- Jarecki et. al. [JKK14] give a very efficient and widely used OPRF called 2H-DH.
- We denote by DH-PSI the PSI protocol one gets when HL-PSI is instantiated with 2H-DH. This is a very efficient and canonical protocol for PSI.
- We give the first concrete-security analysis of DH-PSI.

Note: Our paper arrives at this in a modular way. We:
- Show that HL-PSI is secure if the OPRF is secure, with a tight reduction
- Give concrete security proofs for 2H-DH
- Deduce concrete security results for DH-PSI

In this presentation however we discuss only the DH-PSI results.

We prove InI security of the DH-PSI protocol under a few different DL-related assumptions to showcase the variations in tightness.

Our Assumptions in group $\mathbb{G}$ underlying the protocol:

- CDH : Regular Computational Diffie-Hellman
- DDH : Regular Decision Diffie-Hellman
- CDH-MUC : CDH in multi-user setting with corruptions
- V-CDH : Verifiable CDH
- V-CDH-MUC : Verifiable CDH-MUC

Our results showing concrete InI security of the DH-PSI protocol

Given: Adversary A attacking InI security of DH-PSI with resources:

- q_{Run} queries to its RUN oracle
- q_{RO} queries to its random oracle

and achieving advantage $\epsilon = \mathbf{Adv}^{\mathrm{ini}}_{\mathsf{F},\Pi,2}(\mathrm{A})$

Our results showing concrete InI security of the DH-PSI protocol

Given: Adversary A attacking InI security of DH-PSI with resources:

- q_{Run} queries to its RUN oracle
- q_{RO} queries to its random oracle

and achieving advantage $\epsilon = \mathbf{Adv}^{\mathrm{ini}}_{\mathsf{F},\Pi,2}(\mathrm{A})$

We build: Adversary A' attacking problem **P** that has about same running time as A and achieves advantage $\epsilon' = \mathbf{Adv}^{\mathrm{P}}_{\mathbb{G}}(\mathrm{A}')$

Our results showing concrete InI security of the DH-PSI protocol

Given: Adversary A attacking InI security of DH-PSI with resources:

- q_{Run} queries to its RUN oracle
- q_{RO} queries to its random oracle

and achieving advantage $\epsilon = \mathbf{Adv}^{\mathrm{ini}}_{\mathsf{F},\Pi,2}(\mathrm{A})$

We build: Adversary A′ attacking problem **P** that has about same running time as A and achieves advantage $\epsilon' = \mathbf{Adv}^{\mathrm{P}}_{\mathbb{G}}(\mathrm{A}')$

Such that: $\epsilon \leq \mathbf{B}(\epsilon', \{q_{\mathrm{Run}}, q_{\mathrm{RO}}\})$

Our results showing concrete InI security of the DH-PSI protocol

Given: Adversary A attacking InI security of DH-PSI with resources:

- q_{Run} queries to its RUN oracle
- q_{RO} queries to its random oracle

and achieving advantage $\epsilon = \mathbf{Adv}^{\text{ini}}_{\text{F},\Pi,2}(\text{A})$

We build: Adversary A′ attacking problem **P** that has about same running time as A and achieves advantage $\epsilon' = \mathbf{Adv}^{\text{P}}_{\mathbb{G}}(\text{A}')$

Such that: $\epsilon \leq \mathbf{B}(\epsilon', \{q_{\text{run}}, q_{\text{RO}}\})$

Problem P	**Bound** $\mathbf{B}(\epsilon', \{q_{\text{run}}, q_{\text{RO}}\})$
CDH	$4 \cdot (q_{\text{RO}}^2 \cdot q_{\text{Run}} \cdot \epsilon' + \alpha)$
V-CDH	$4 \cdot (q_{\text{RO}} \cdot q_{\text{Run}} \cdot \epsilon' + \alpha)$
CDH-MUC	$4 \cdot (q_{\text{RO}} \cdot \epsilon' + \alpha)$
V-CDH-MUC	$4 \cdot (\epsilon' + \alpha)$ ✓
DDH	$4 \cdot (\epsilon' + \alpha)$ ✓

$$\alpha = \frac{(q_{\text{RO}} \cdot q_{Run}) + q_{RO} + 1}{p}$$

p : order of the group $\mathbb{G}$ underlying the problems

Tight reductions!

Plan

- ☑ Background: Asymptotic and Concrete security
- ☑ Definitions and Relations
- ☑ Results for DH PSI
- ☐ Salted DH-PSI

Salted DH-PSI protocol

- We present a new PSI protocol that we call Salted DH-PSI. It is as asymptotically as efficient as DH-PSI but achieves tighter security. So in practice it can be implemented in smaller groups, improving concrete efficiency.
- The idea behind Salted DH-PSI is similar to the one used in PSS [BR96] which is a RSA based signature scheme that is as efficient as FDH-RSA [BR93,BR96] but uses salting to get a tight reduction to the one-wayness of RSA.
- With the addition of a salt, there's also a parameter, the salt-length, sl, which appears in our security results.

Bounds for Salted DH-PSI versus DH-PSI

p : order of the group $\mathbb{G}$ underlying the problems

DH-PSI

$$\alpha = \frac{(q_{\mathrm{RO}} \cdot q_{Run}) + q_{RO} + 1}{p}$$

Salted DH-PSI

$$\beta = \frac{q_{Run} \cdot (q_{Run} + q_{\mathrm{RO}})}{2^{sl}} + \frac{(q_{RO} + 1)}{p}$$

sl : length of salt used in Salted DH-PSI

Problem P	**Bound $\mathbb{B}$ for DH-PSI**	**Bound $\mathbb{B}$ Salted DH-PSI**
CDH	$4 \cdot (q_{\mathrm{RO}}^2 \cdot q_{\mathrm{Run}} \cdot \epsilon' + \alpha)$	$2 \cdot (q_{\mathrm{RO}} \cdot \epsilon' + \beta)$
V-CDH	$4 \cdot (q_{\mathrm{RO}} \cdot q_{\mathrm{Run}} \cdot \epsilon' + \alpha)$	$2 \cdot (\epsilon' + \beta)$
CDH-MUC	$4 \cdot (q_{\mathrm{RO}} \cdot \epsilon' + \alpha)$	$2 \cdot (q_{\mathrm{RO}} \cdot \epsilon' + \beta)$
V-CDH-MUC	$4 \cdot (\epsilon' + \alpha)$	$2 \cdot (\epsilon' + \beta)$
DDH	$4 \cdot (\epsilon' + \alpha)$	$2 \cdot (\epsilon' + \beta)$

Plan

- ☑ Background: Asymptotic and Concrete security
- ☑ Definitions and Relations
- ☑ Results for DH PSI
- ☑ Salted DH-PSI

Summary and Conclusions

Initiate the study of concrete security for Two Party Computation

1. **Definitions**

   Input Indistinguishability (InI): A 2PC security definition that

   - Is indistinguishability based
   - Yet equivalent to simulation for PSI and friends
   - Concrete security and cryptanalysis friendly

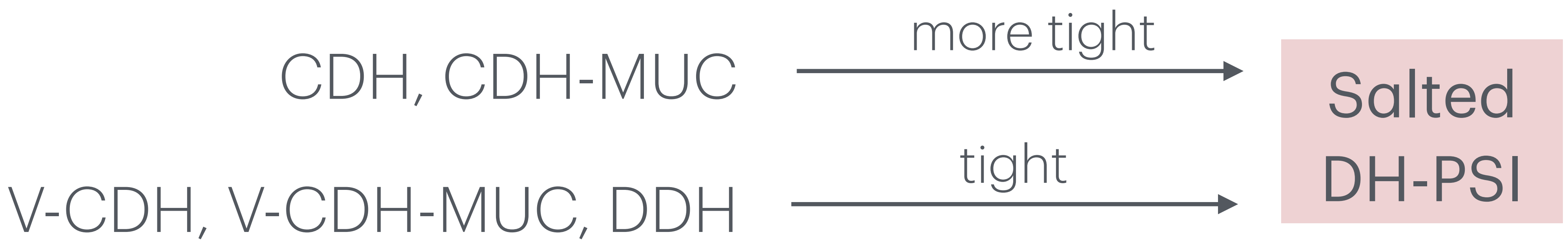
   tight

Definitions explicitly incorporate ROM and surface subtleties in this regard

2. **Concrete security results for PSI and OPRFs**

3. **Salted DH-PSI**

   New PSI protocol, as efficient as DH-PSI, but



Our definitions and results are for the semi-honest (honest-but-curious) setting