Efficient secure two-party computation secure against active adversary using Yao's Garbled Circuit and GMW paradigm

Mohammad Hossein Faghihi Sereshgi

University Of Rochester

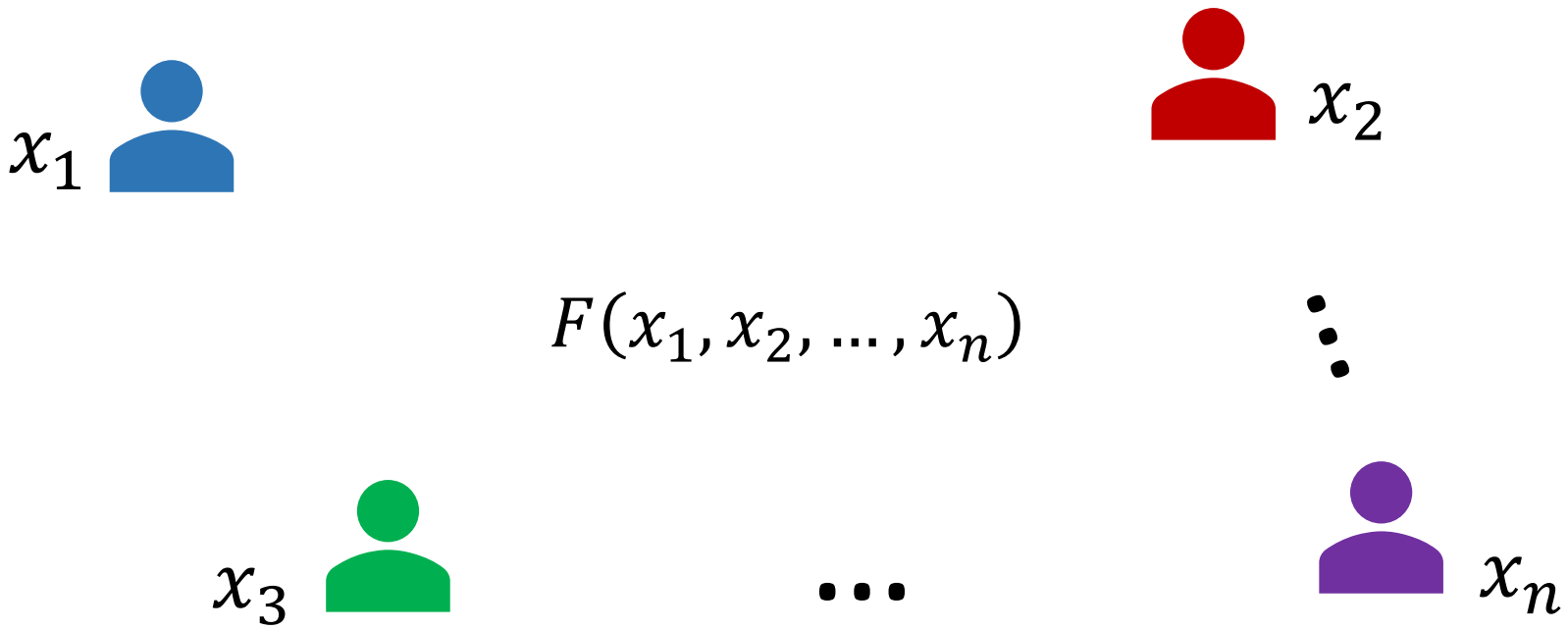
Collaborators: Jackson Abascal, Carmit Hazay, Yuval Ishai, Muthuramakrishnan Venkitasubramaniam

Outline

- What is secure Multi-party Computation
- Yao's Garbled Circuit
- Protocol with Active security
- Proof of Security
- Results

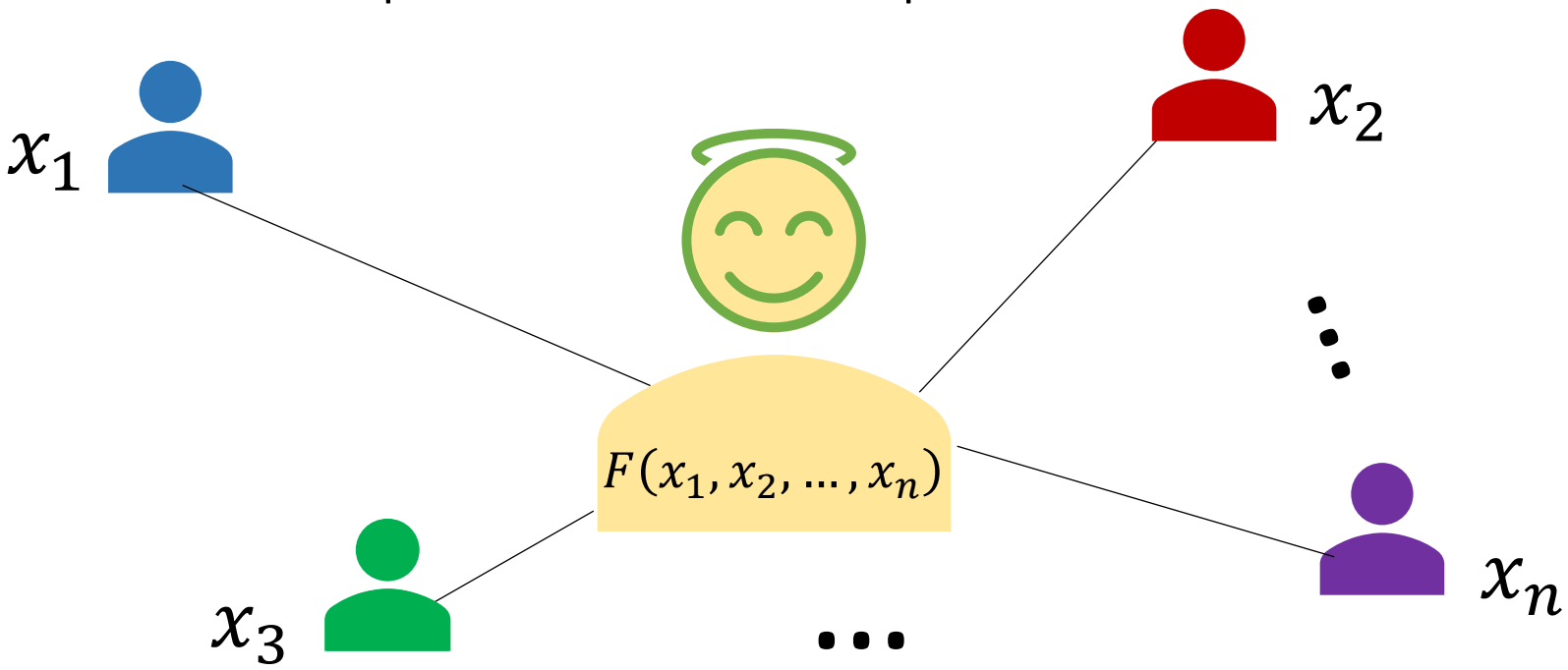
Secure Multi-Party Computation

- n parties want to compute $F(x_1, x_2, \dots, x_n)$
  - Keep the inputs private
    - No one learns anything more than the output of the function



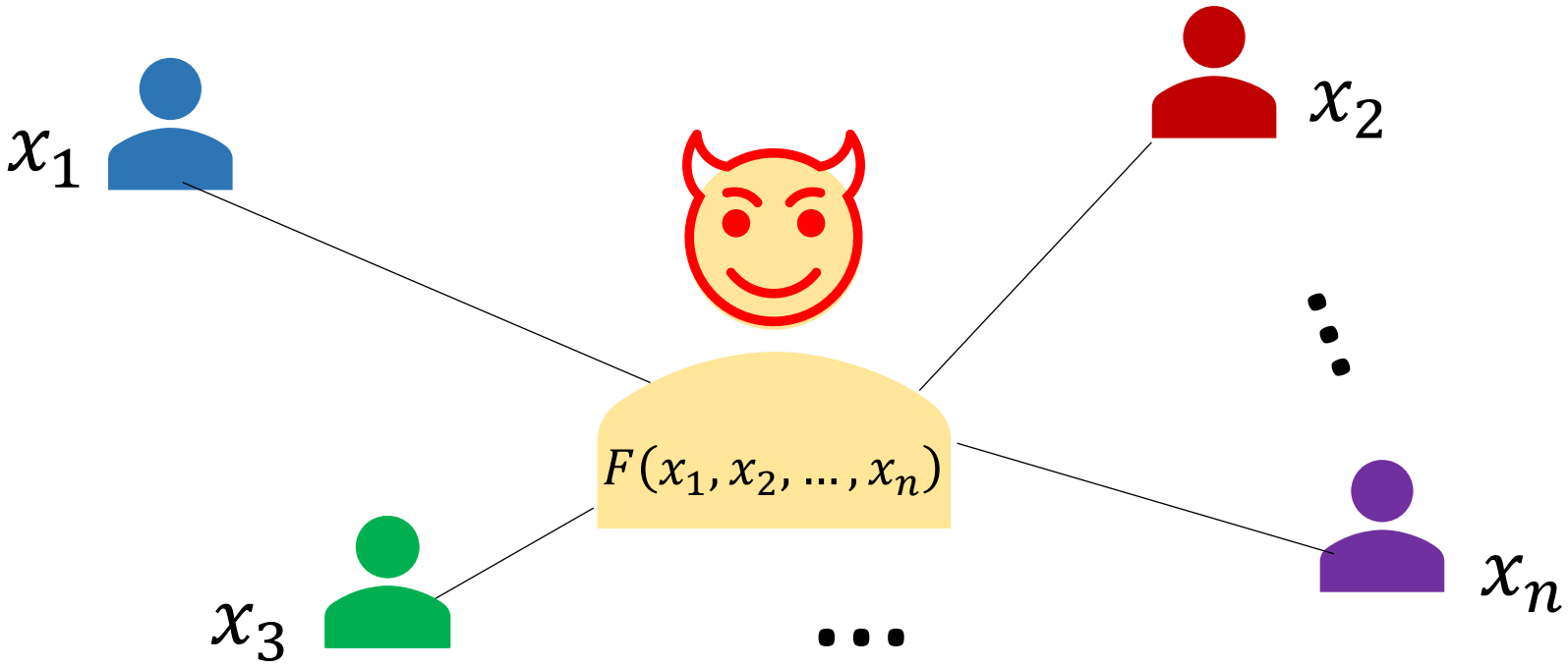
Secure Multi-Party Computation

- Trusted Third Party
  - Receives the inputs and returns the output



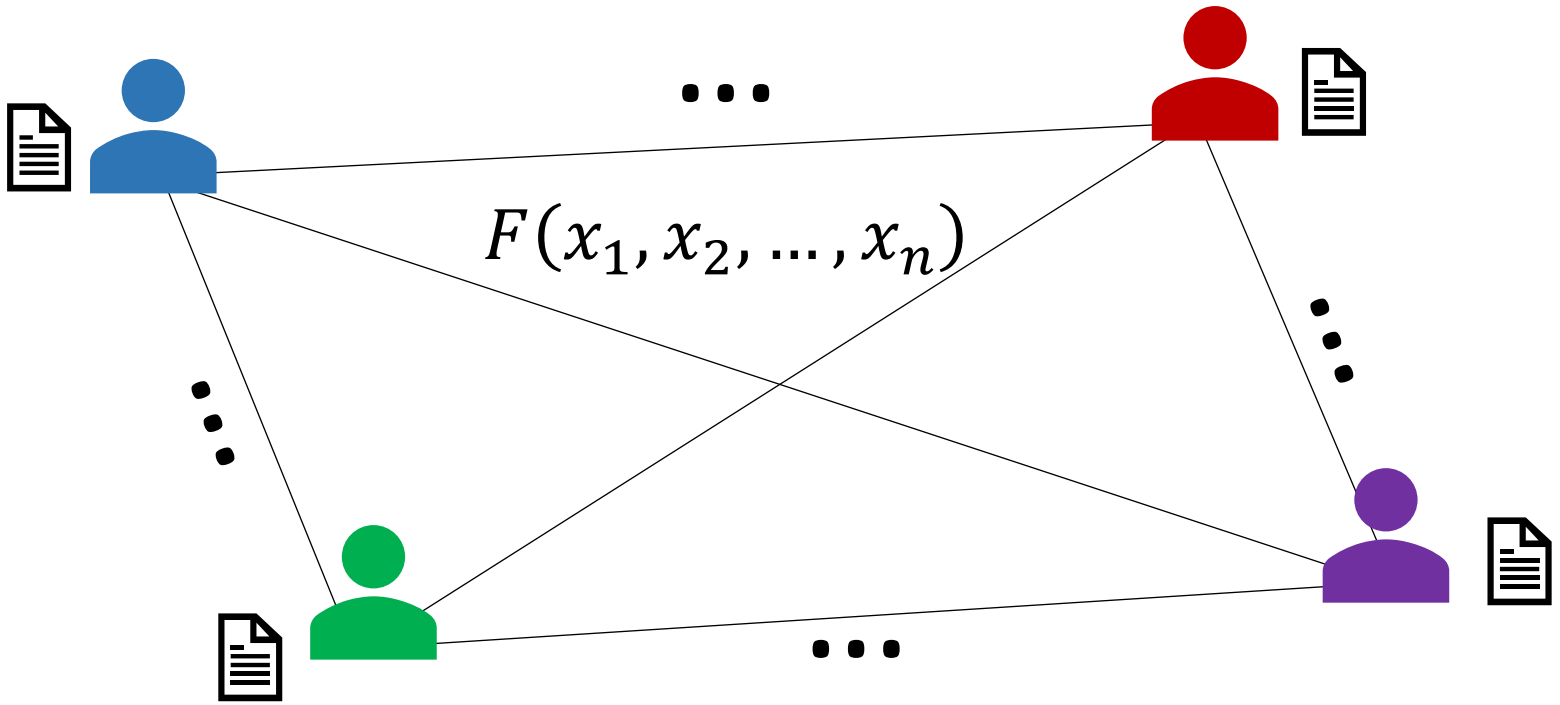
Secure Multi-Party Computation

- It is almost impossible to find an entity trusted by everyone



Secure Multi-Party Computation

- Use a protocol that does not need a TTP.



Secure Multi-Party Computation

- Secure Two-Party Computation

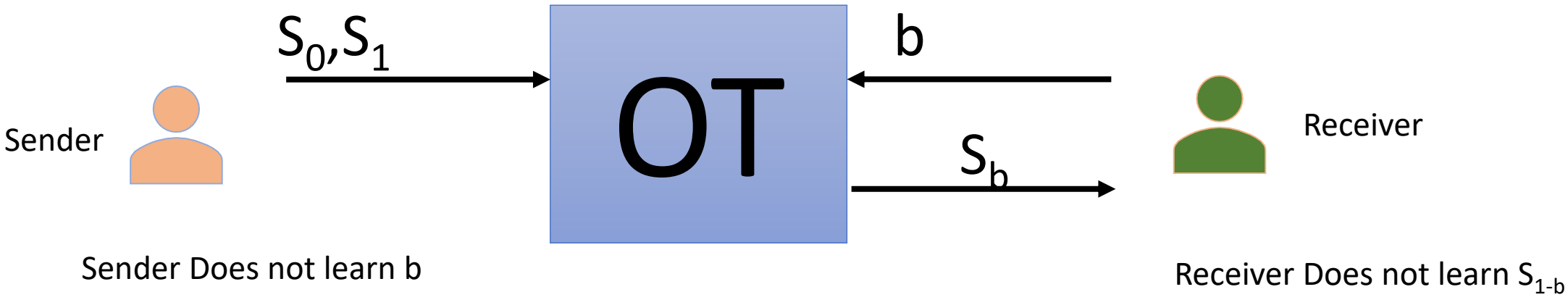
Adversary

- Two types of adversary
  - Semi-Honest (Passive, Honest-but-curious)
    - Follows the protocol
    - Investigates the communications
  - Malicious (Active, Byzantine)
    - Deviates from the protocol
    - Sends bogus messages or goes offline
    - Adversary wants to violate correctness of result and privacy

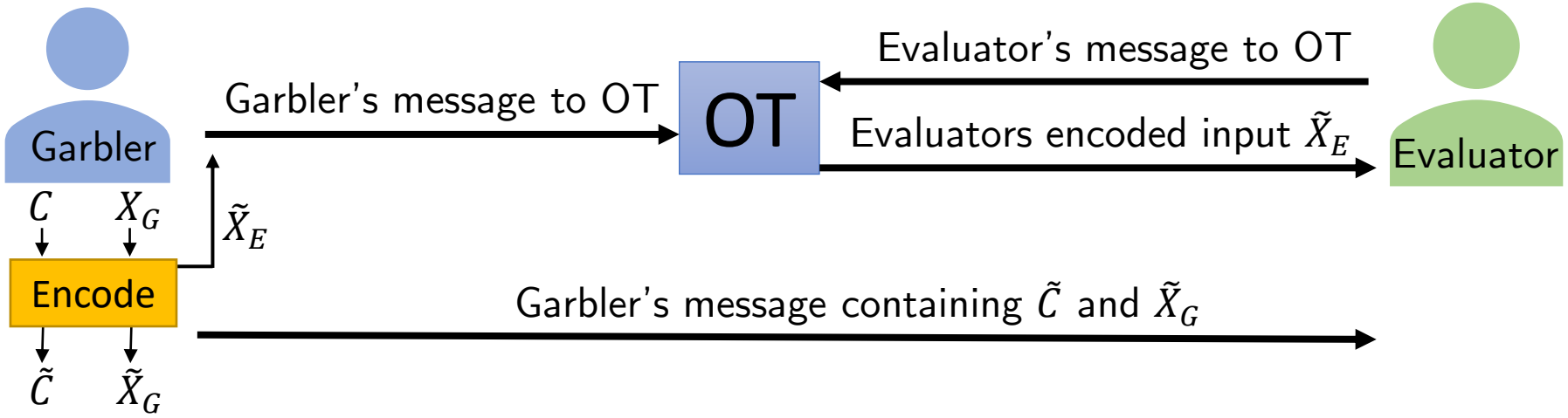
Yao's Garbled Circuit [Yao96]

- One of the first protocol for 2PC
- Passive security
- Assumption:
  - Oblivious Transfer

Oblivious Transfer

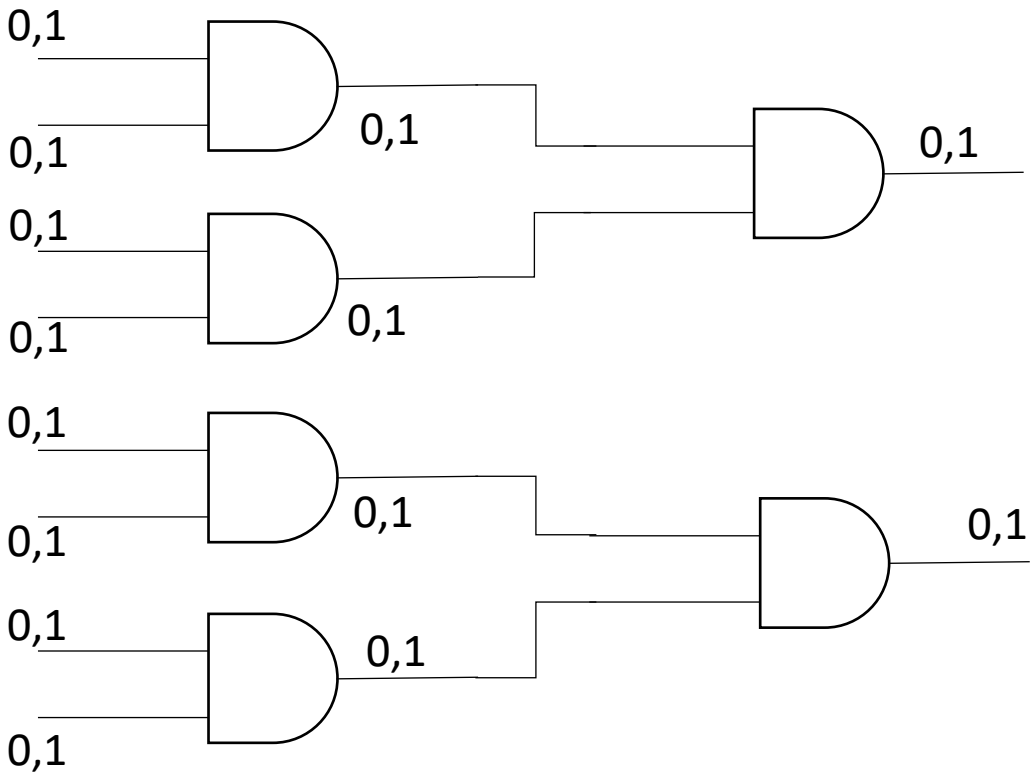


Yao's GC


Garbler
Evaluator
OT
Garbler's message to OT
Evaluator's message to OT
Evaluators encoded input $\tilde{X}_E$
C
X_G
$\tilde{X}_E$
Encode
Garbler's message containing $\tilde{C}$ and $\tilde{X}_G$
$\tilde{C}$
$\tilde{X}_G$

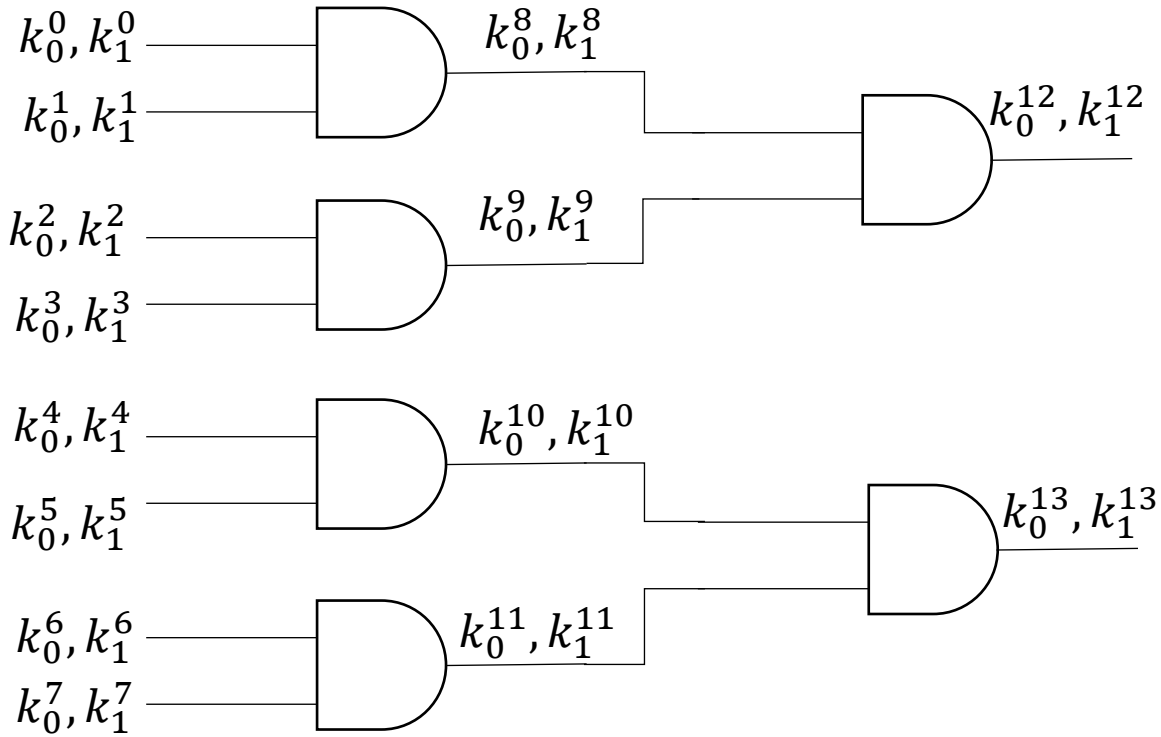
Yao's GC

- Consider a circuit C that computes the function F



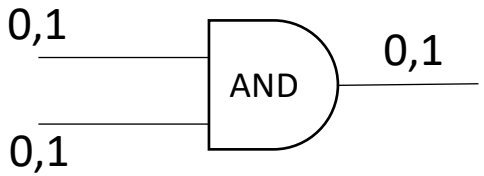
Yao's GC

$k_b^i \in \{0,1\}^\kappa$

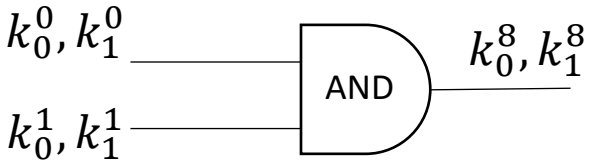


Yao's GC

- Garbling AND gate



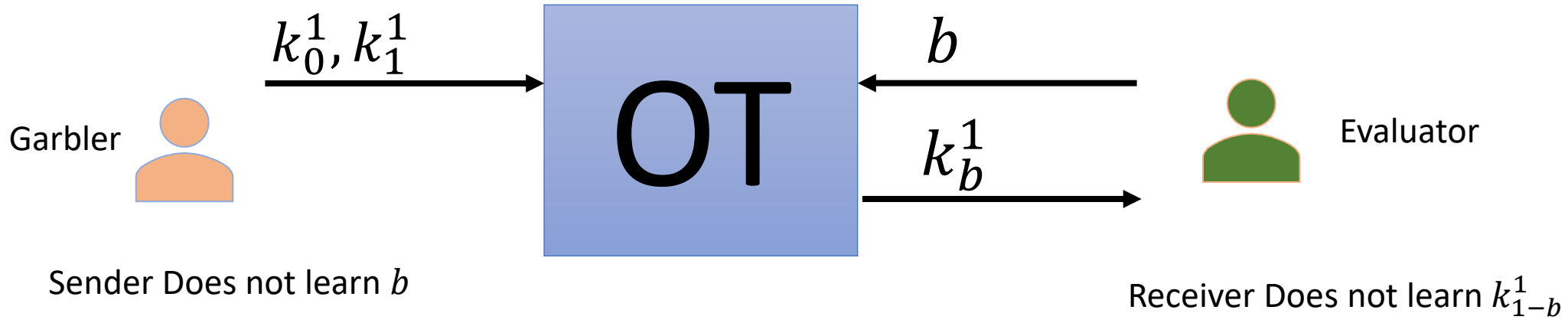
x	y	z
0	0	0
0	1	0
1	0	0
1	1	1



x	y	z
k_0^0	k_0^1	$Enc_{k_0^0}\left(Enc_{k_0^1}(\boldsymbol{k_0^8})\right)$
k_0^0	k_1^1	$Enc_{k_0^0}\left(Enc_{k_1^1}(k_0^8)\right)$
k_1^0	k_0^1	$Enc_{k_1^0}\left(Enc_{k_0^1}(k_0^8)\right)$
k_1^0	k_1^1	$Enc_{k_1^0}\left(Enc_{k_1^1}(k_1^8)\right)$

Yao's GC

- Garbler sends the encoded truth table and his encoded input
- For Evaluator's input, they use OT

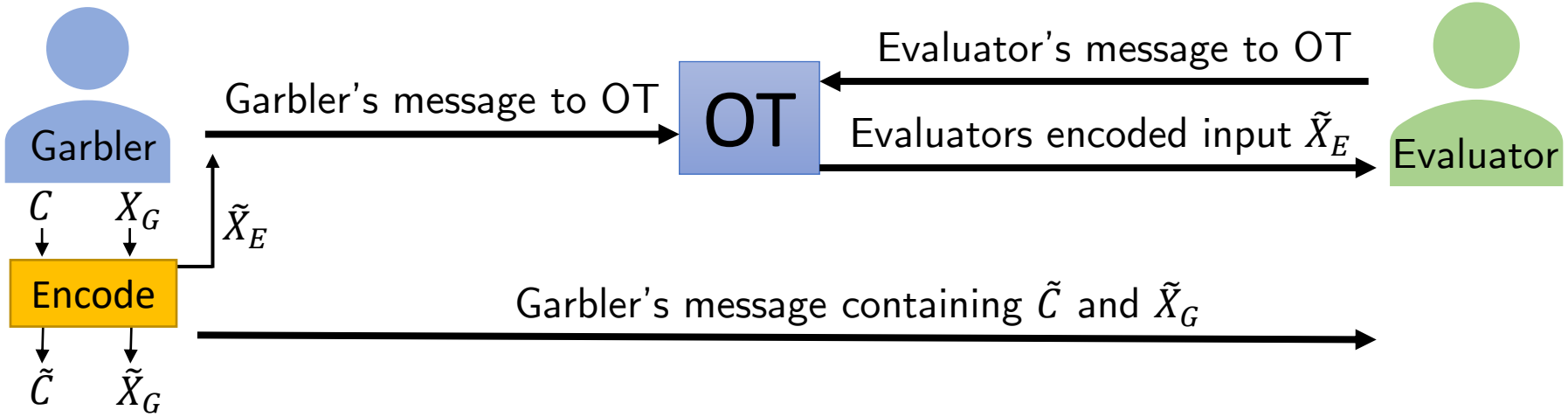


Yao's GC

- Evaluator decrypts all possible outputs
  - Only one of them will be valid
- Assume Garbler's input is 1 and Evaluator's input is 0

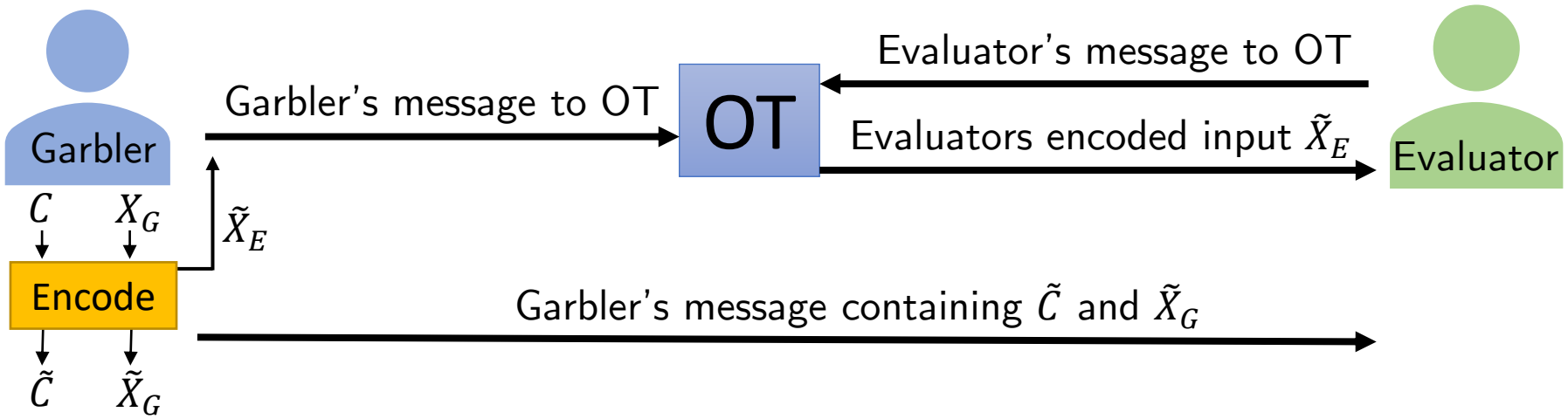
x	y	z		
k_0^0	k_0^1	$Enc_{k_0^0}\left(Enc_{k_0^1}(k_0^8)\right)$	$Dec_{k_1^0}\left(Dec_{k_0^1}(\cdots)\right)$	Invalid!
k_0^0	k_1^1	$Enc_{k_0^0}\left(Enc_{k_1^1}(k_0^8)\right)$	$Dec_{k_1^0}\left(Dec_{k_0^1}(\cdots)\right)$	Invalid!
k_1^0	k_0^1	$Enc_{k_1^0}\left(Enc_{k_0^1}(k_0^8)\right)$	$Dec_{k_1^0}\left(Dec_{k_0^1}(\cdots)\right)$	Valid
k_1^0	k_1^1	$Enc_{k_1^0}\left(Enc_{k_1^1}(k_1^8)\right)$	$Dec_{k_1^0}\left(Dec_{k_0^1}(\cdots)\right)$	Invalid!

Yao's GC



- Complexities:
  - Communication: $O(\kappa|C|)$ bits
  - $O(|C|)$ PRG invocation
  - n Oblivious Transfer on pairs of κ-bit strings
    - n: length of Evaluator's input

Yao's GC



- Secure against passive (honest-but-curious) adversary
- In the OT-hybrid, the protocol is secure against actively corrupted Evaluator
- However, an actively corrupted Garbler can attack the protocol!

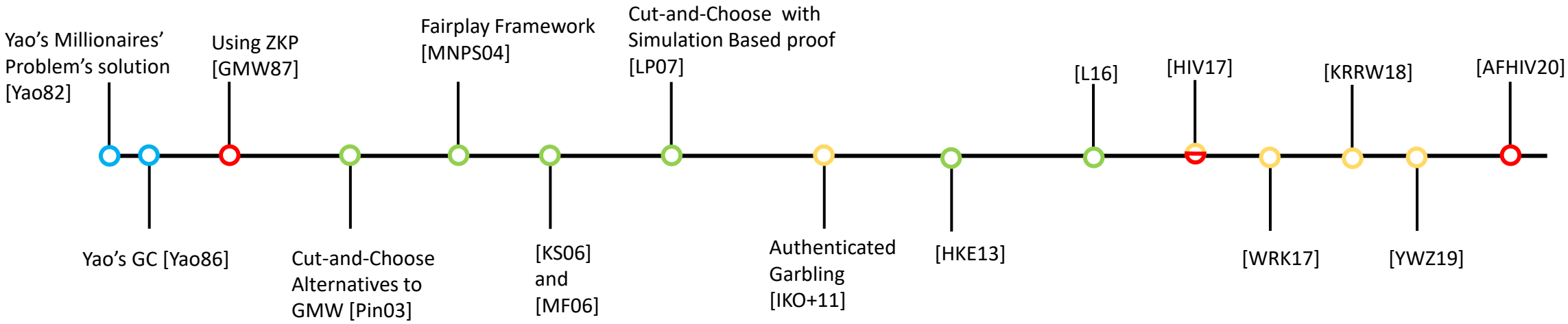
Yao's GC

- Theoretical solution
  - GMW Paradigm [GMW87]: Attach a zero-knowledge proof (ZK) with every message
  - Not considered practical!
- Concretely efficient solutions:
  - Cut-and-Choose [LP07,...]
  - Authenticated Garbling [IKOPS11,WRK17,YWZ19]

This Work: GMW is practical!

Yao's GC

- Timeline of some of the works on 2PC



Zero-knowledge proof

- Prover P has witness w that $x \in L$ and wants to convince V that $x \in L$
- Soundness: if $x \notin L$, a cheating P^* cannot convince V
- Zero Knowledge: The protocol reveals nothing more than $x \in L$

Active Security

GMW Paradigm [GMW87]

- ZKP + passive security = Active security
  - Costly

Comparison

- Asymptotic Complexity

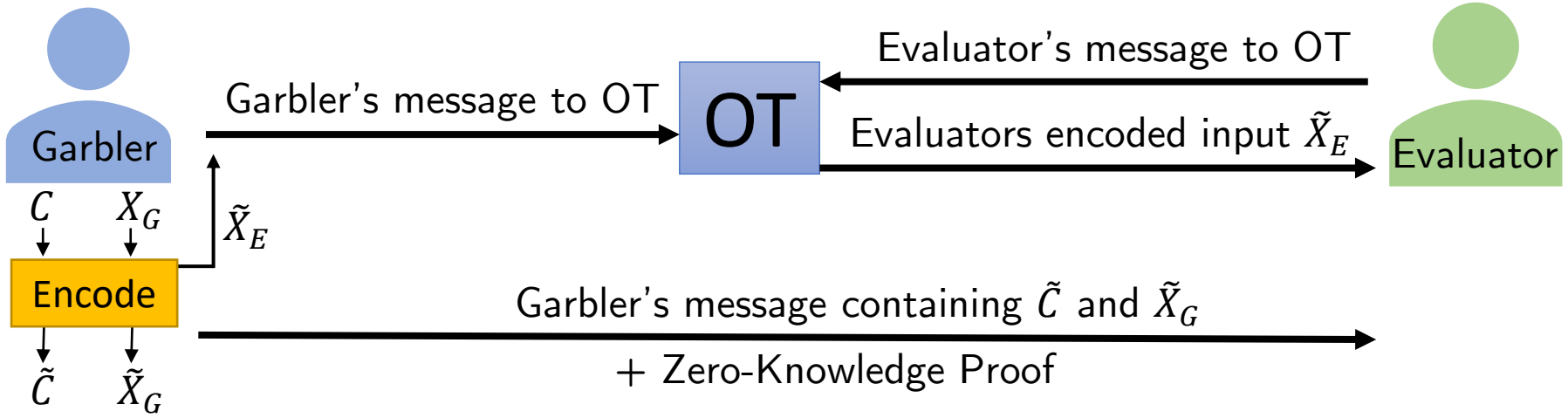
Protocol	Func-ind (Comm./Comp)	Func-dep (Comm./Comp.)	Online (Comm.)
[Yao86]		$O(\|C\|k)$	$O(\|I\|k + \|O\|)$
[HIV17]		$O(\|C\|k)$ (Input dependent)	$O(\|I\|k + \|O\|)$
Authenticated garble[WRK17]	$O\left(\frac{\|C\|\rho k}{\log\tau + \log\|c\|}\right)$	$O(\|C\|)$	$O(\|I\|k + \|O\|)$
[IPS08] in Authenticated garble[WRK17]	$O(\|C\|k)$	$O(\|C\|k)$	$O(\|I\|k + \|O\|)$
[AFHIV20]		$O\left(\|C\|k + \sqrt{\|C\|k}\right)$	$O(\|I\|k + \|O\|)$

k Computational security parameter
ρ Statistical security parameter
τ Number of protocol executions in the amortized setting

Features of the protocol

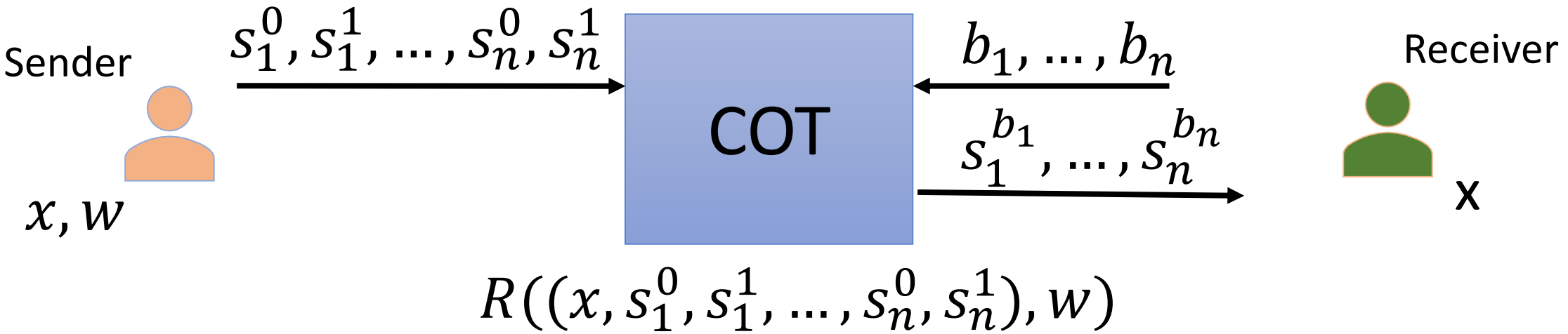
- Boolean operations
  - Based on Yao's GC
- Secure against active Garbler using ZKP
  - Uses Ligero [AHIV17]
- Offline-Online phase
- Offline Phase is non-interactive
  - The two parties do not need to know each other
- Online phase needs only one round

Yao's GC+ZKP



- Garbler proves that:
  - The GC is constructed correctly
  - The Garbler's input is consistent with the GC
  - The Evaluator's encoded input is consistent with the GC
- First Variant: Non-black-box in PRG but black-box in OT
- How? Certified OT [IKOPS11,HIV17]

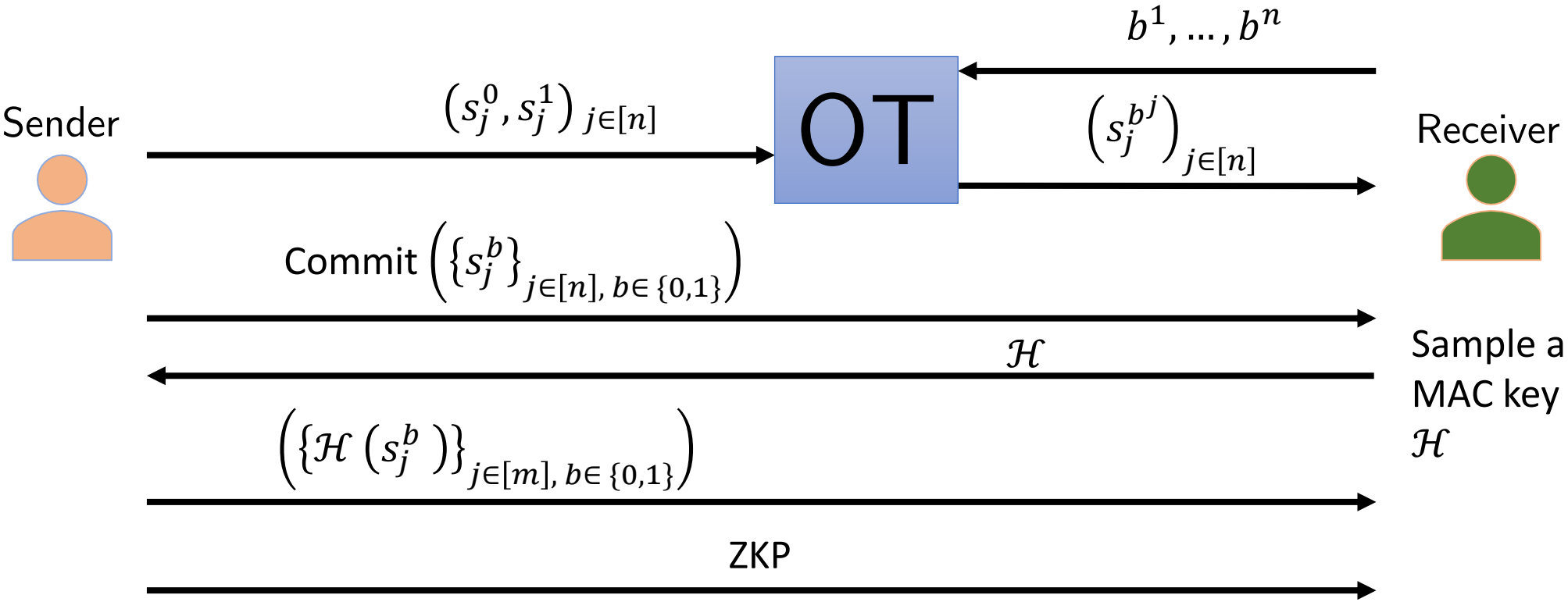
Certified OT



- COT is parameterized with an NP-relation R
- The receiver will receive the output only if the relation is true

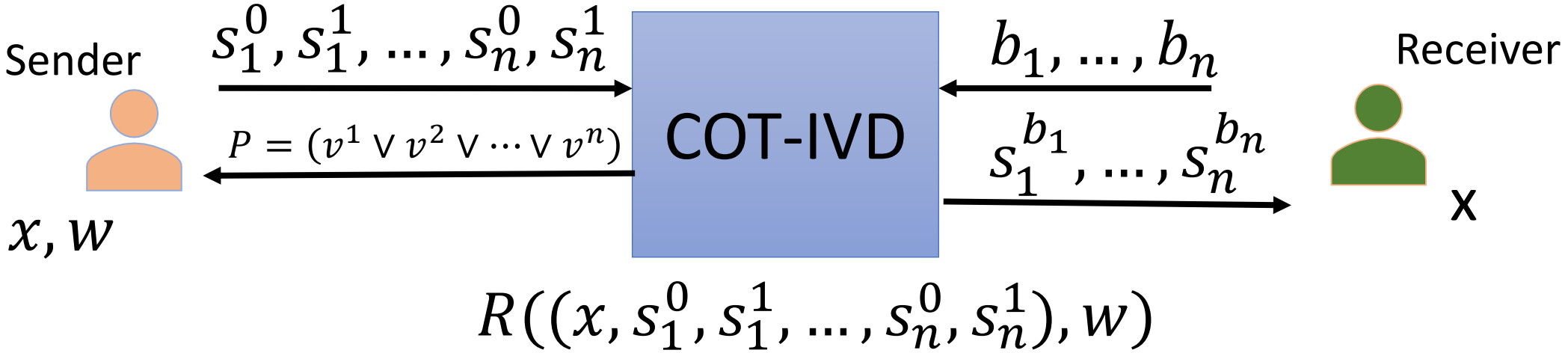
We modularly show how to realize COT using OT in a black-box way

Certified OT



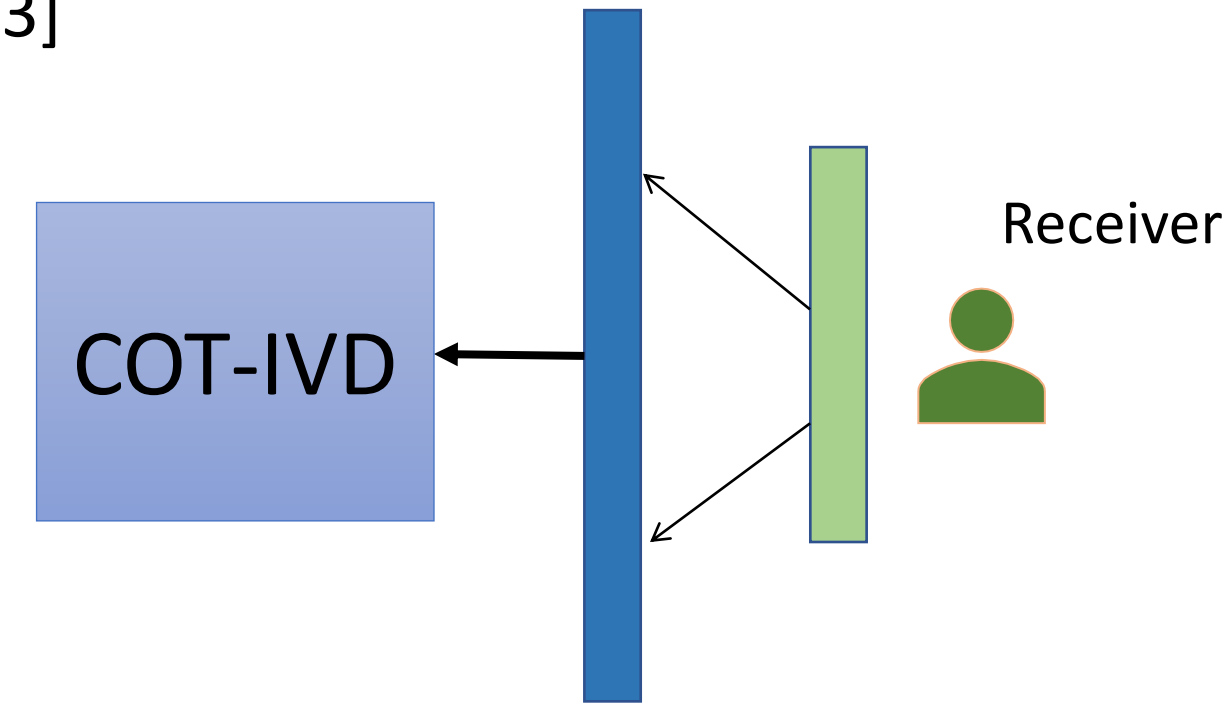
- ZKP shows that
  - NP-relation R on sender's input is satisfied
  - The MAC values are computed correctly
- Can compress rounds using known (Fiat-Shamir's) heuristic

Certified OT-Input-Value Disjunction (IVD)



Certified OT- IVD

- Encode the receiver's input in order to deal with the 1bit leakage [LP07,IKOPS11,SS13]



Proof of Security Using Simulator

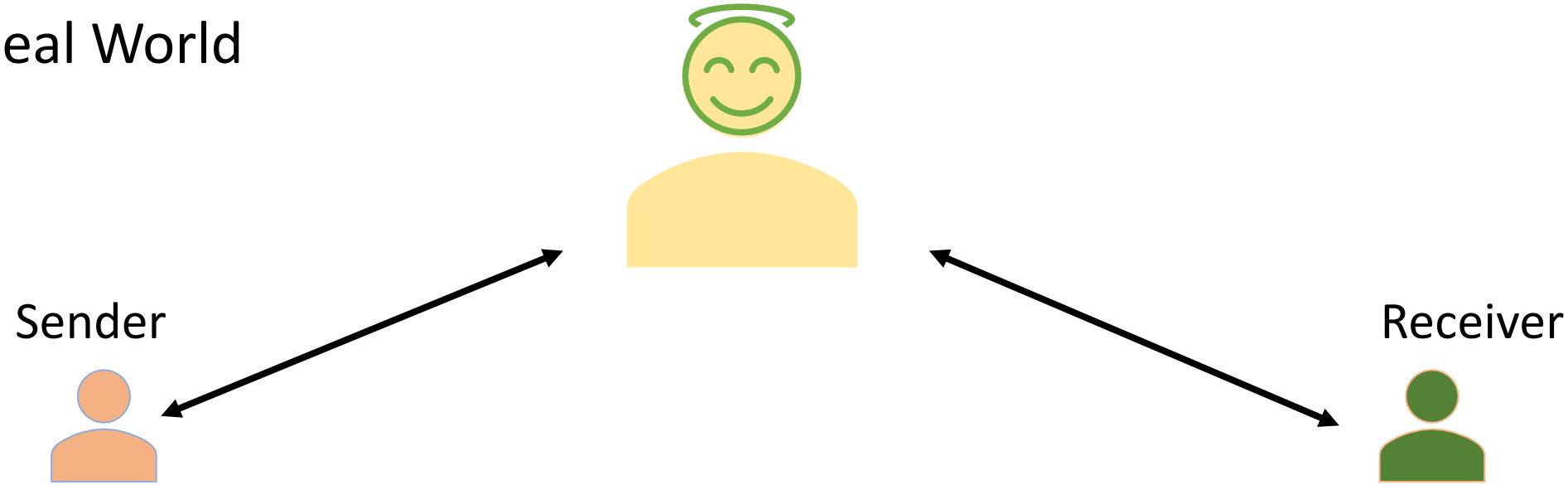
- Probabilistic Polynomial-Time Turing Machine
- Generates (simulates) the view of the adversary
  - View: $\{x, r, m_0, m_1, \dots\}$
  - Given adversary's input and the output

Proof of Security Using Simulator

- Real World

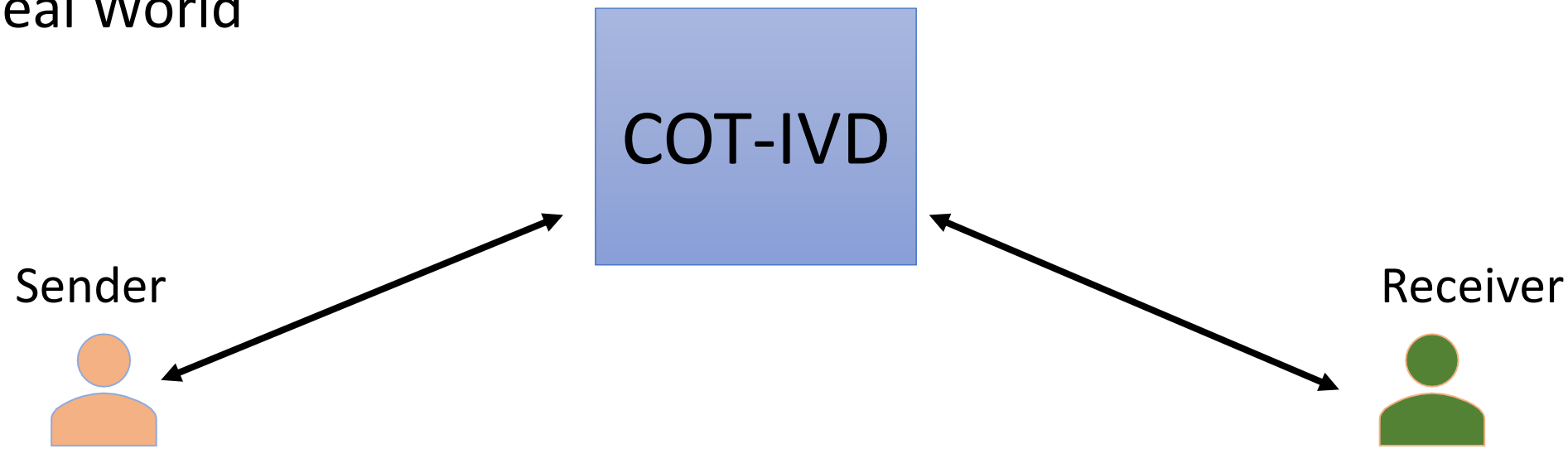
Proof of Security Using Simulator

- Ideal World

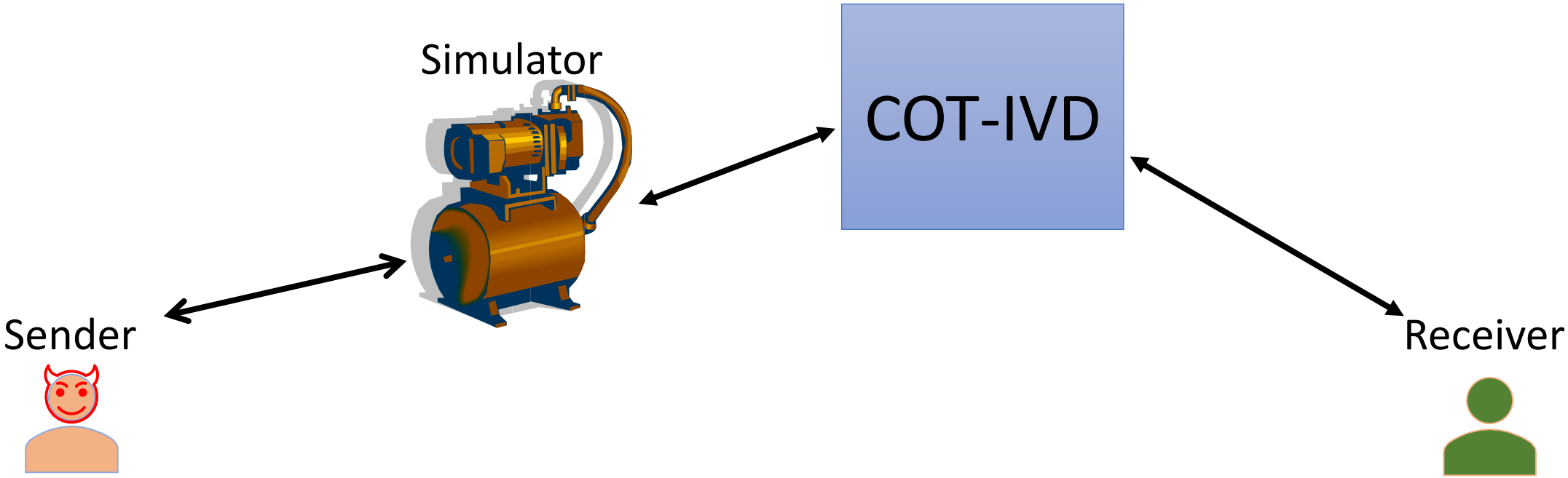


Proof of Security Using Simulator

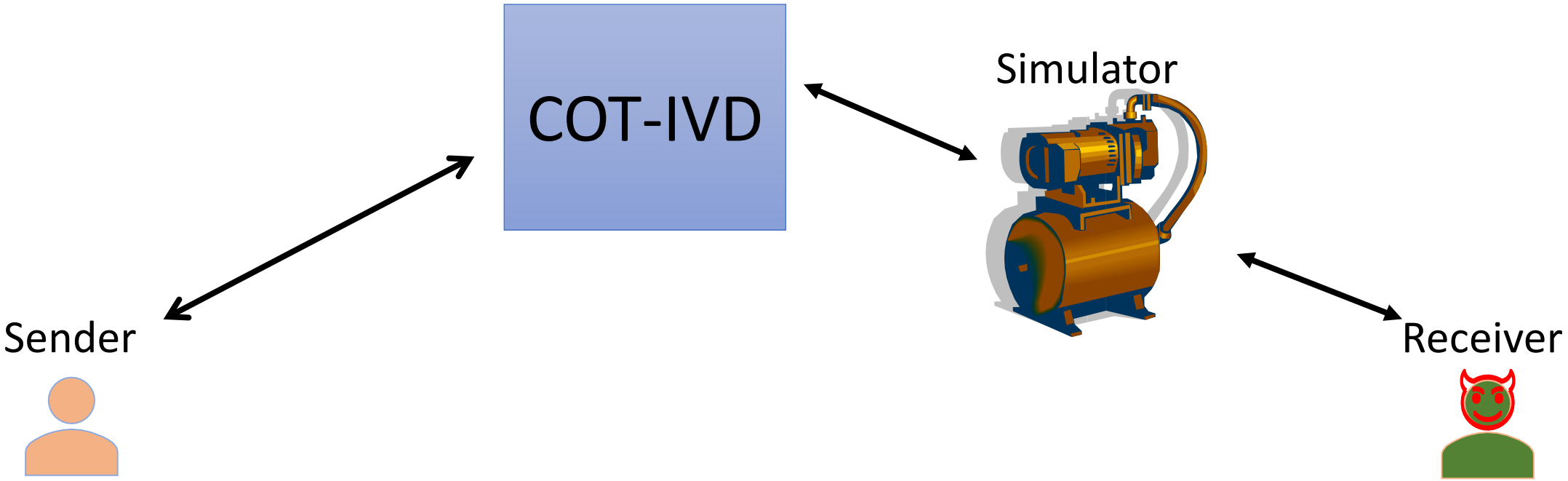
- Ideal World



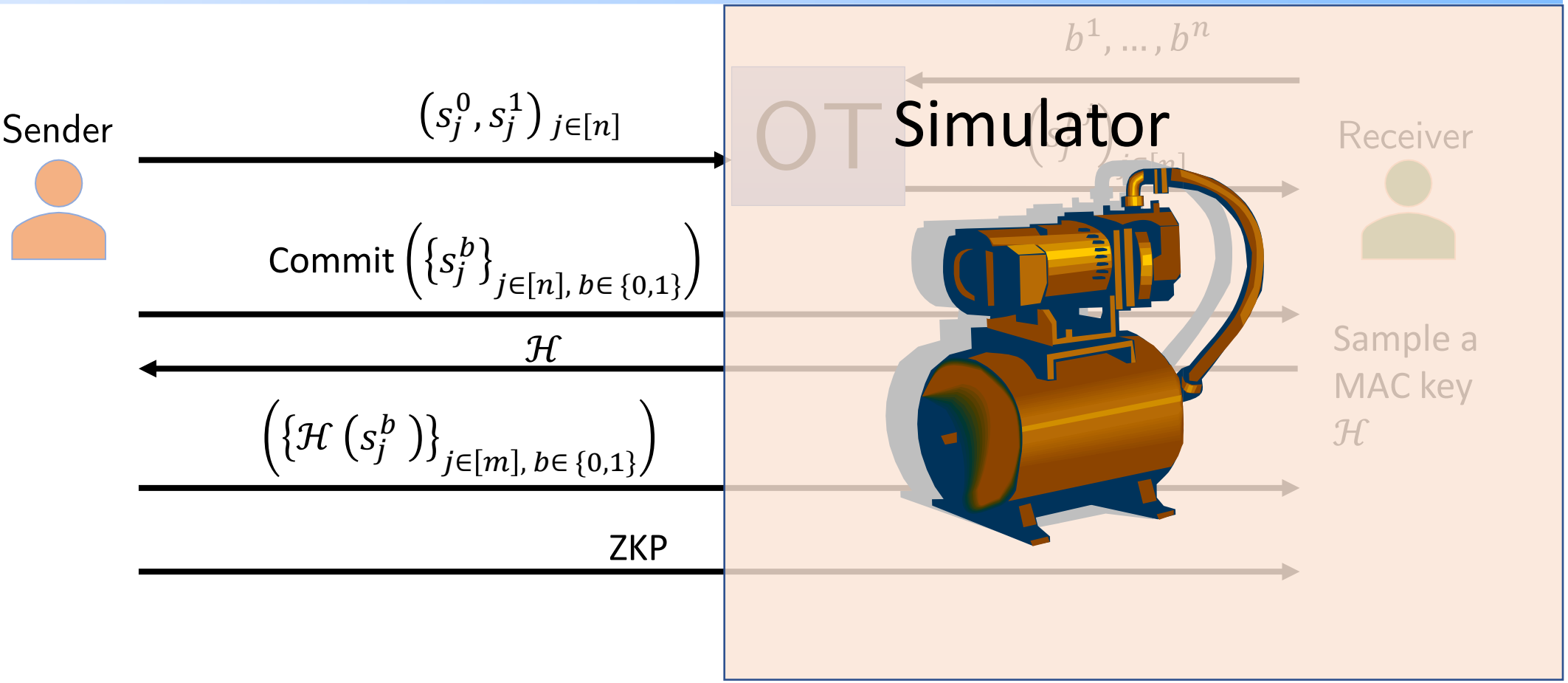
Proof of Security Using Simulator



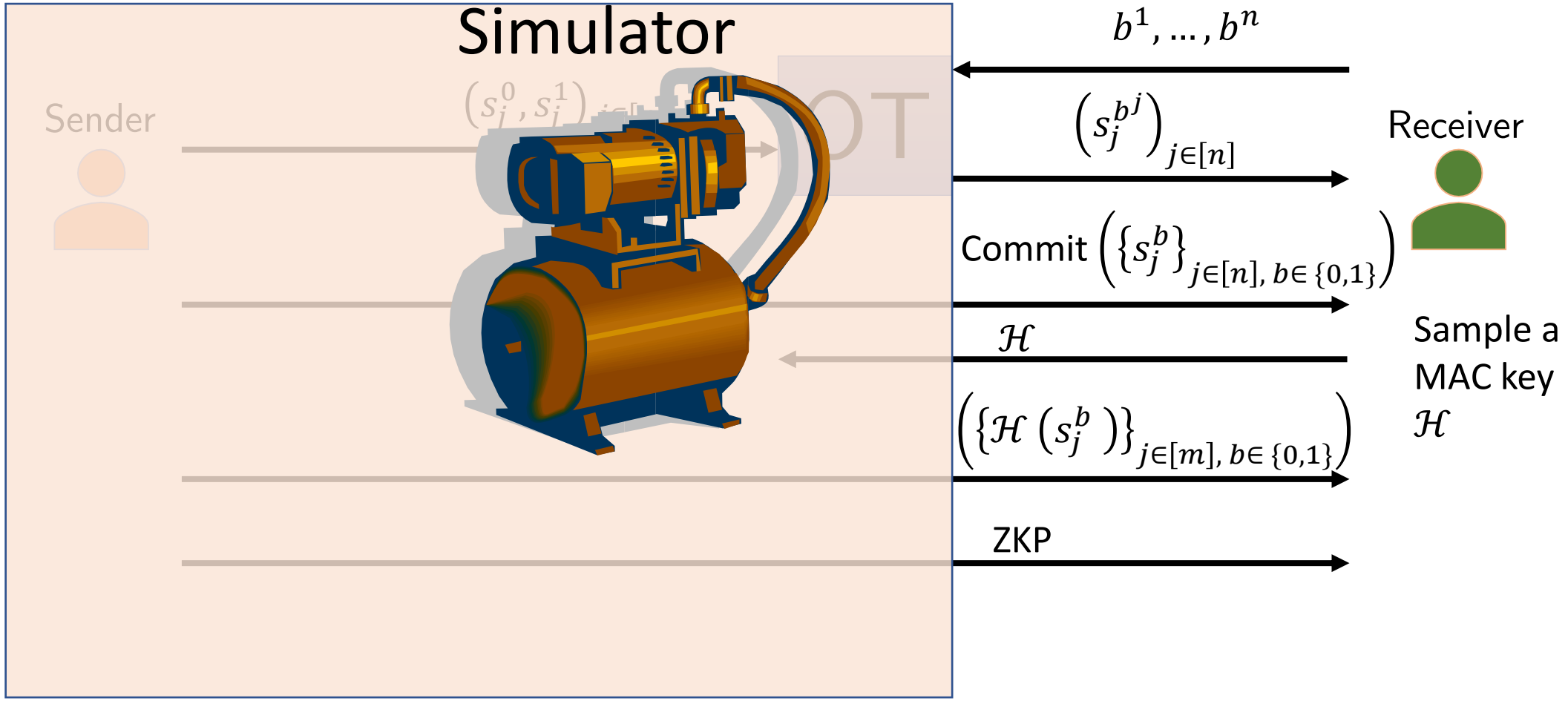
Proof of Security Using Simulator



Certified OT-IVD: Proof of Security



Certified OT-IVD: Proof of Security



Offline-Online setting

- The GC Proof is input independent
  - Can be done offline without interaction (Silent preprocessing)
  - The Garbler can make the GC and the ZKP available on internet.
- The Evaluator’s message for OT protocol does not need Interaction.
  - The Evaluator can make it available on internet before protocol starts.

Offline-Online setting

We split the protocol in Offline phase and Online phase

- Offline phase
  - Garbler publishes the GC and its proof of correctness
  - Evaluator publishes the first message of the OT protocol
- Online phase
  - Garbler sends the response to OT
  - Garbler sends a proof that the labels transmitted are consistent with the GC

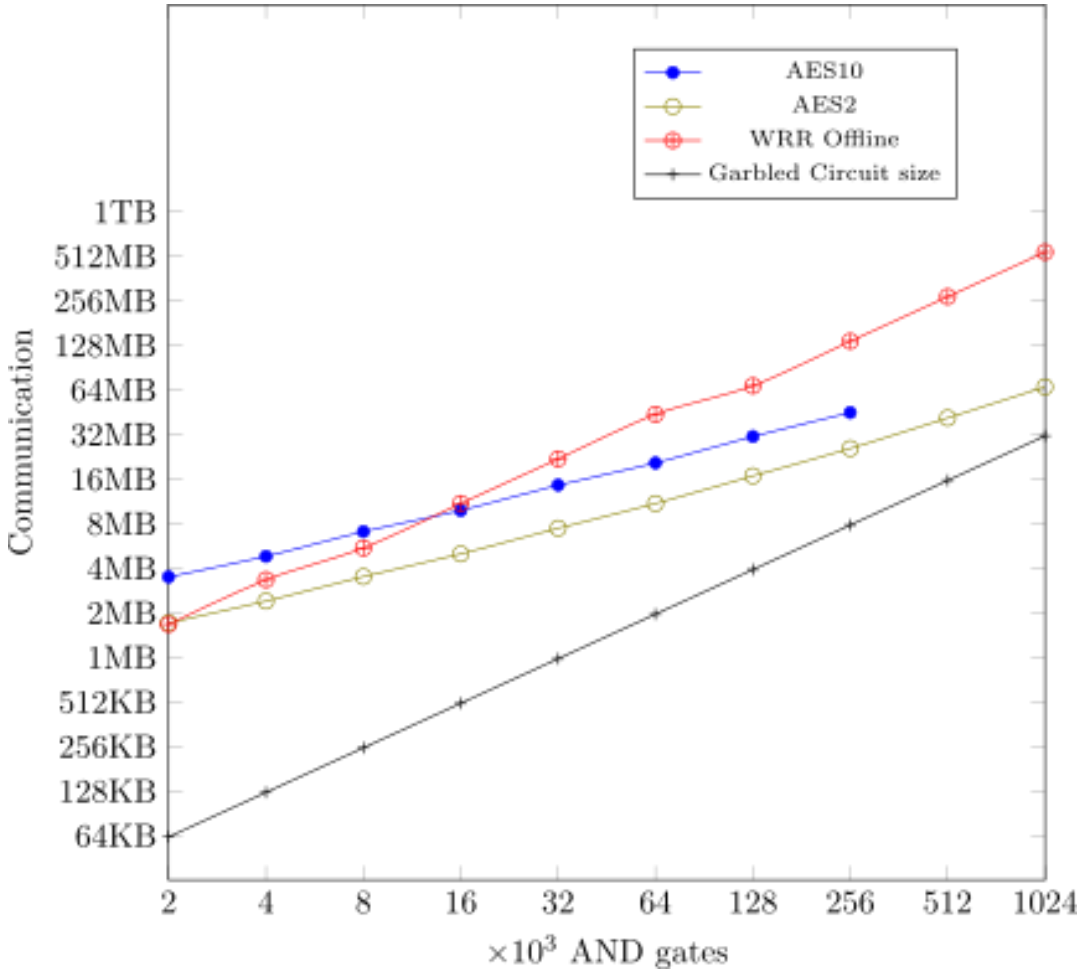
Offline-Online setting

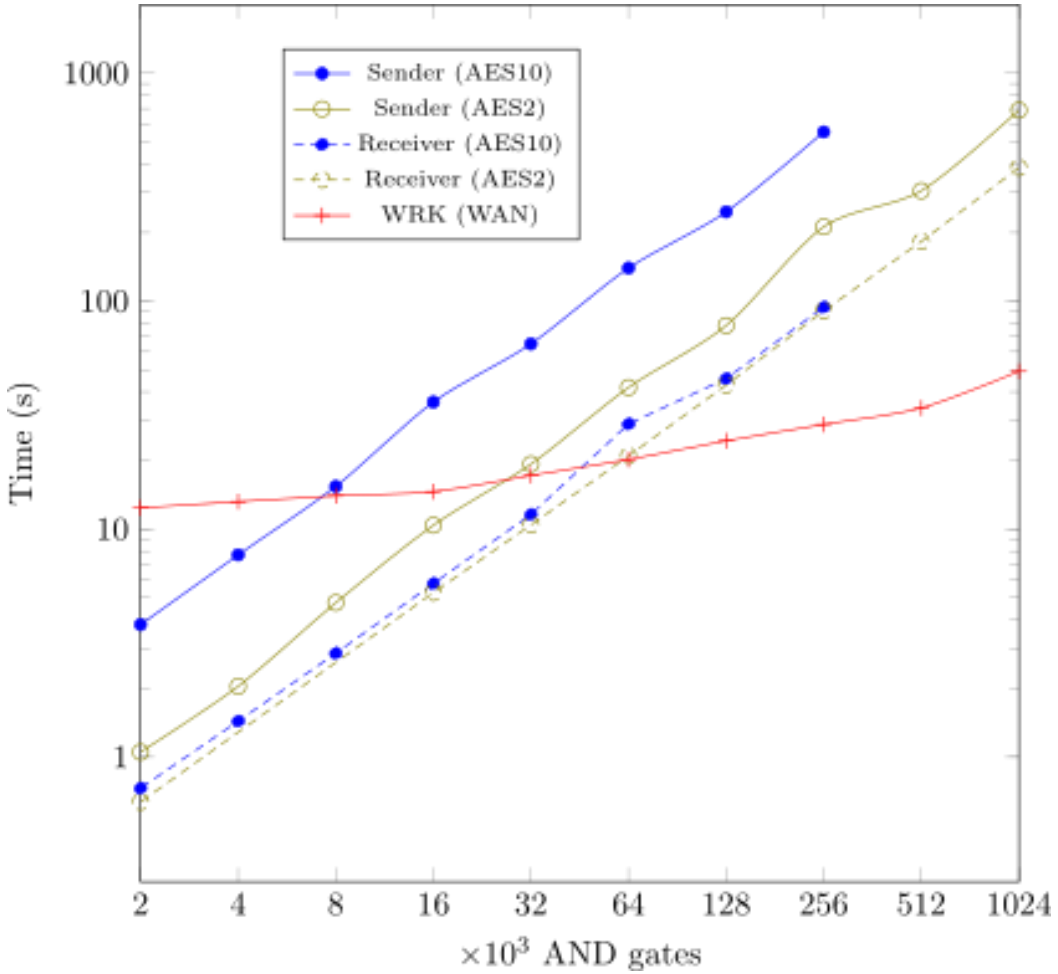
- Split the zero-knowledge proofs into two parts:
  - ZK_{off}
    - GC is constructed correctly
  - ZK_{on}
    - Inputs to OT functionality are consistent with the GC
- Need a commit-and-prove system where we can give multiple proofs on committed values
  - Instantiate using MPC-in-the-head paradigm [IKOS07]
  - Design a concretely efficient variant with sublinear communication complexity (using a variant of Ligero [AHIV17])

Variants of the protocol

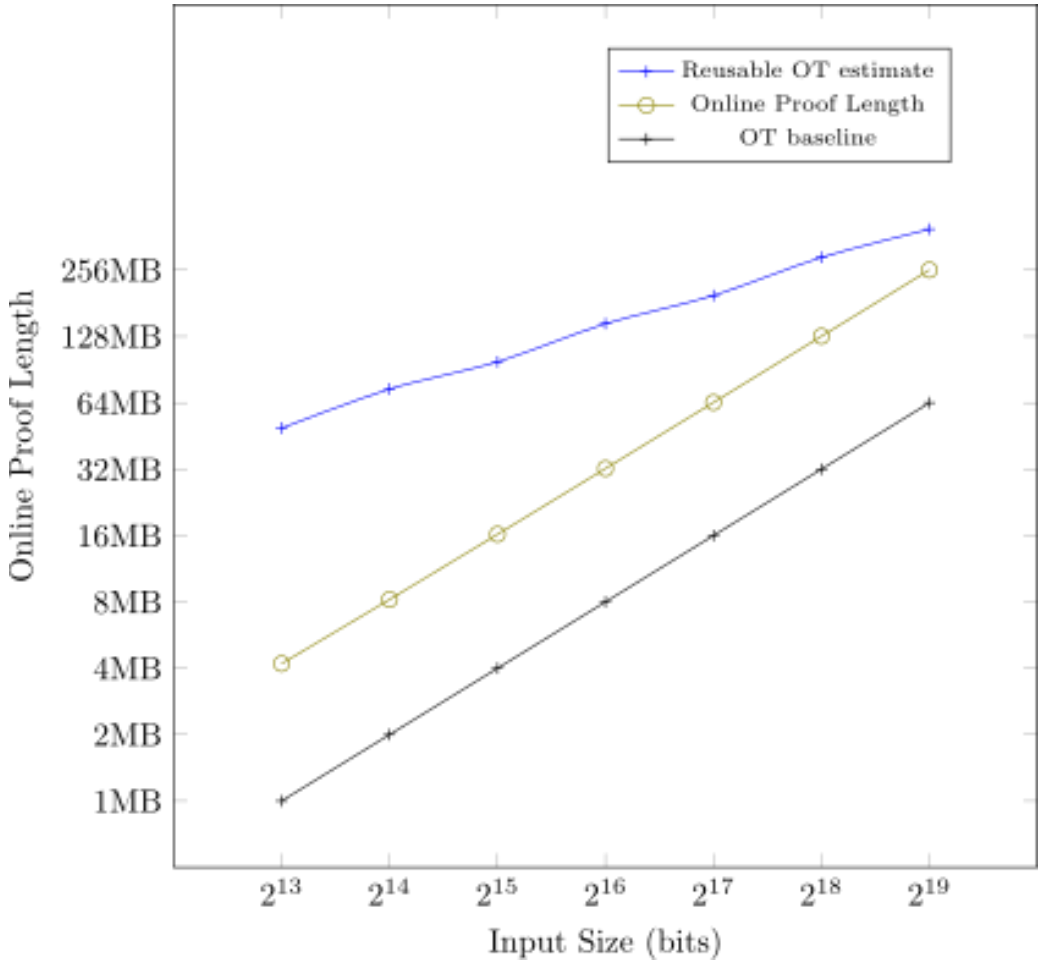
- Variant 1: Certified OT
  - Implementation!
  - Most communication efficient 2PC to date
  - Competitive computational complexity
- Variant 2: OT (Non-black-box on OT and PRG)
  - Larger ZKP in the online phase. Competitive for large input sizes
  - Reusable (Non-Interactive Secure Computation) NISC!

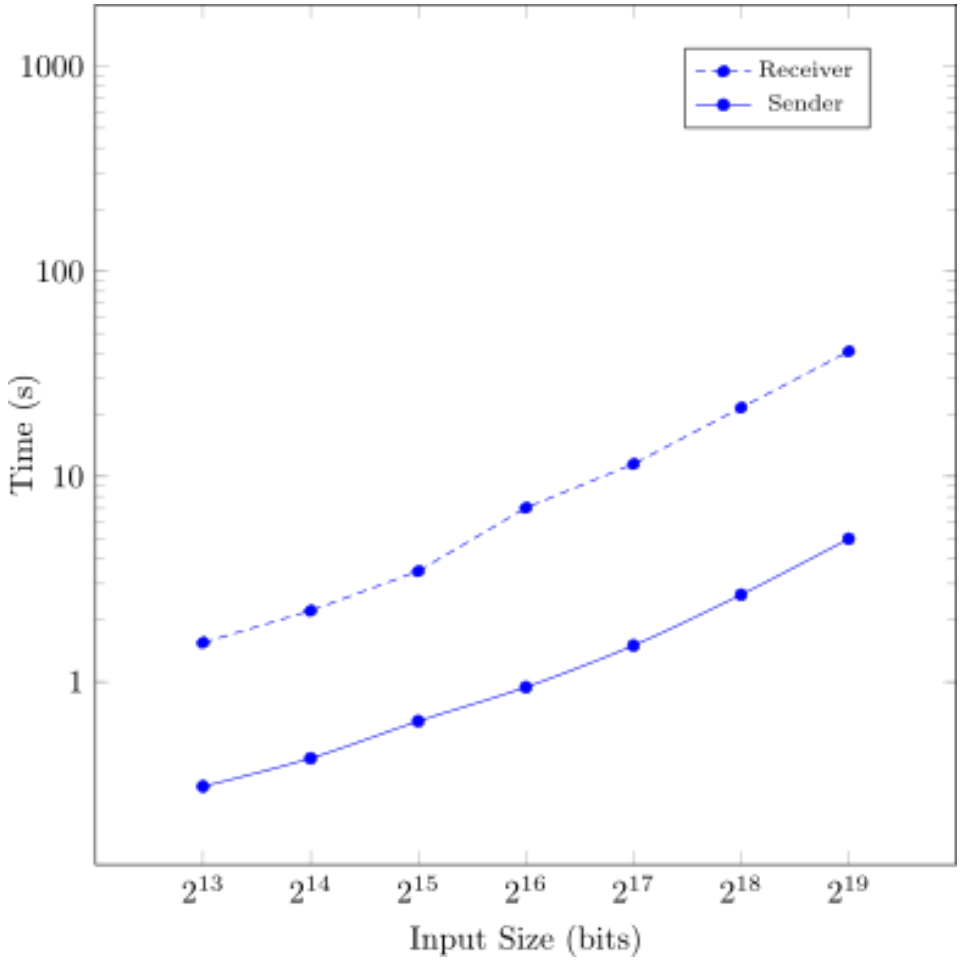
Results-Offline





Results-Online





Thank You