Zero Knowledge from MPC

CS 598 DH

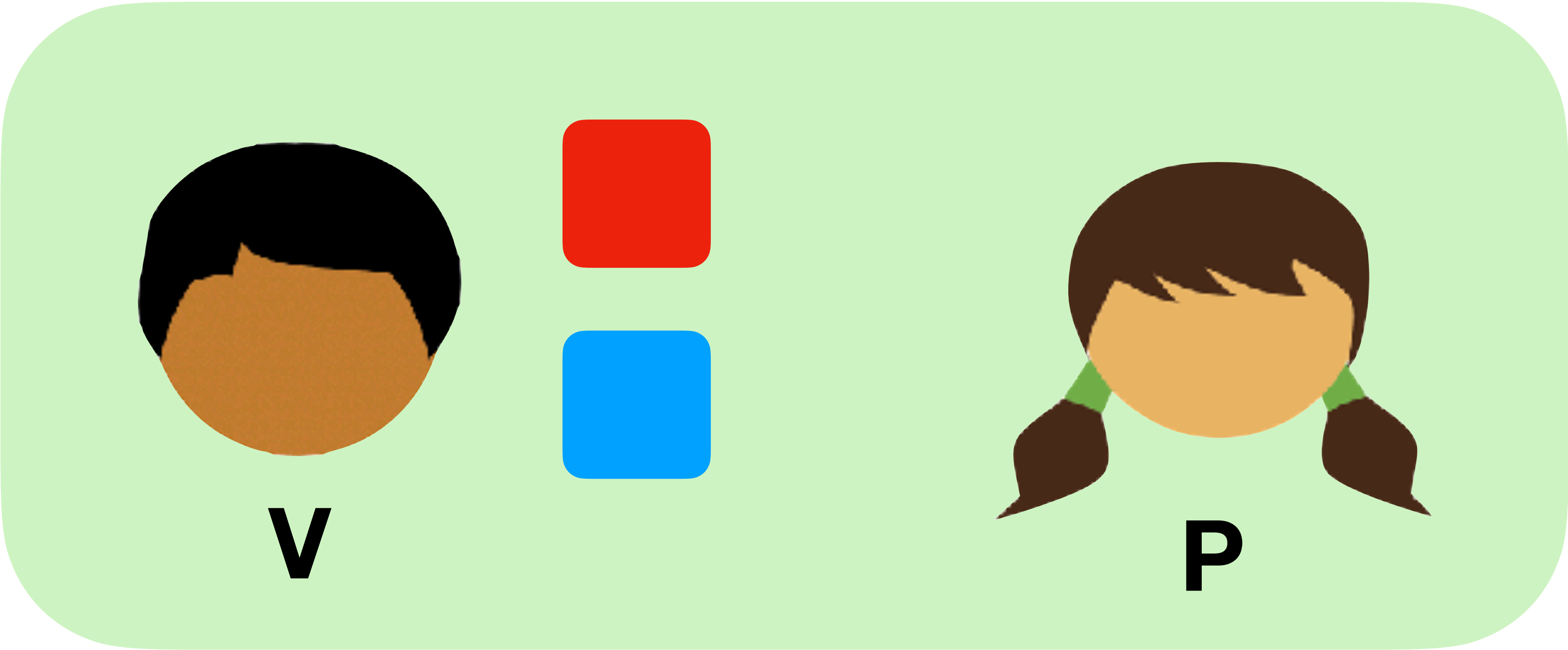
Today's objectives

Review Zero Knowledge Proofs

Construct ZK for circuit satisfiability problem

- From Garbled Circuits
- Via "MPC in the Head"

What is a *zero-knowledge* proof?



Completeness: If $x \in \mathscr{L}$ and if P and V are honest, then V accepts the proof (except with negligible probability)

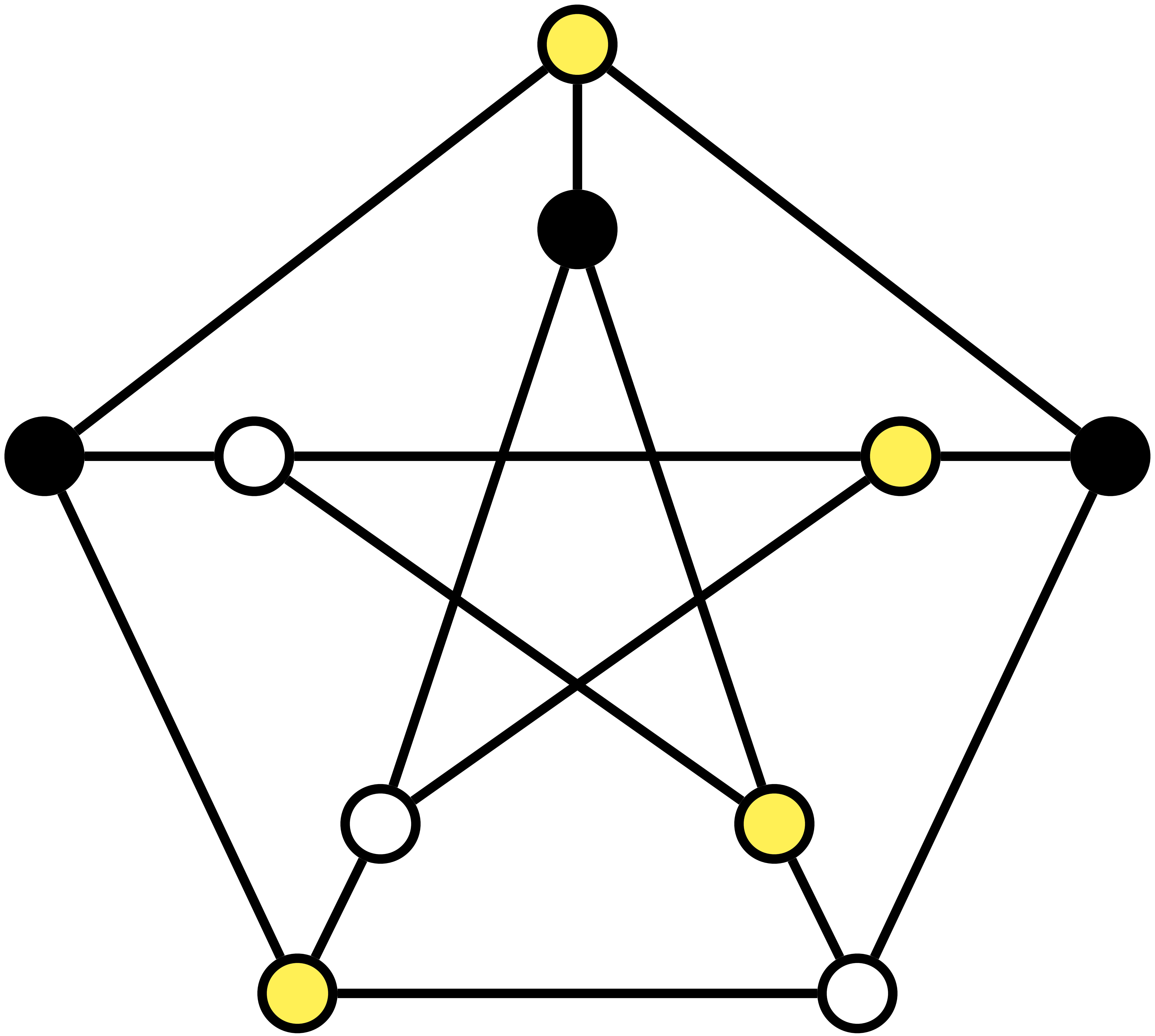
"P can prove true things"

Soundness: If $x \notin \mathscr{L}$, even malicious P cannot cause honest V to accept the proof

"P cannot prove false things"

Zero Knowledge: "V learns nothing except that the thing is true"

Graph 3-Coloring



ZK Proof system for 3-colorability

Statement: a graph
"this graph is 3-colorable"

Witness: a coloring

Basic cryptographic tool: Commitments

Setting

Semi-honest Security

GMW
Compiler

Malicious Security

Zero Knowledge

General-Purpose Tools

GMW Protocol

Multi-party

Multi-round

Garbled Circuit

Constant Round

Two Party

Primitives

Oblivious Transfer

Pseudorandom functions/encryption

Commitments

Setting

Semi-honest Security

GMW Compiler

Malicious Security

Zero Knowledge

General-Purpose Tools

GMW Protocol

Multi-party

Multi-round

Garbled Circuit

Constant Round

Two Party

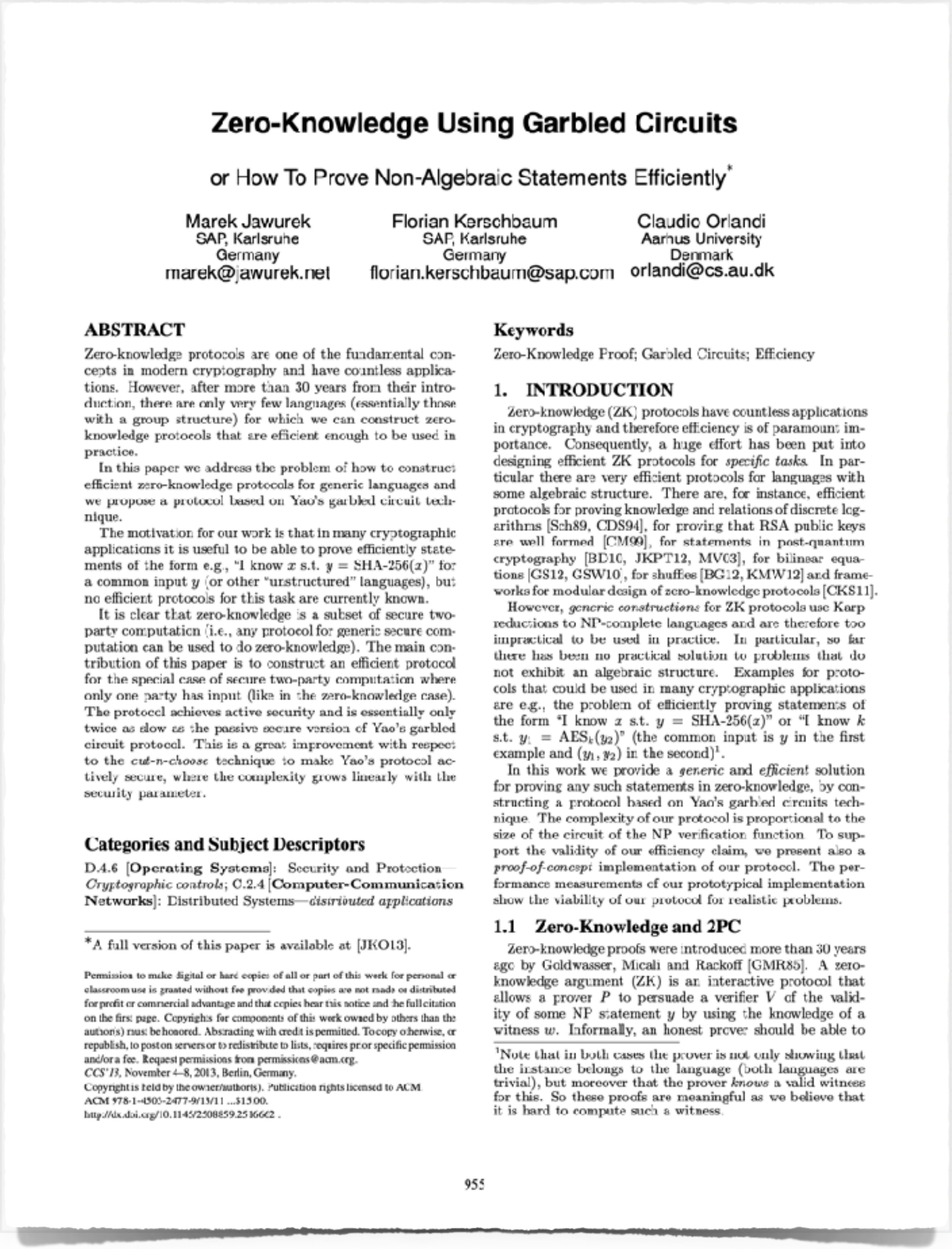
Primitives

Oblivious Transfer

Pseudorandom functions/encryption

Commitments

Zero Knowledge Proofs of Circuit Satisfiability



Zero-Knowledge Using Garbled Circuits

or How To Prove Non-Algebraic Statements Efficiently*

Marek Jawurek
SAP, Karlsruhe
Germany
marek@jawurek.net

Florian Kerschbaum
SAP, Karlsruhe
Germany
florian.kerschbaum@sap.com

Claudio Orlandi
Aarhus University
Denmark
orlandi@cs.au.dk

ABSTRACT

Zero-knowledge protocols are one of the fundamental concepts in modern cryptography and have countless applications. However, after more than 30 years from their introduction, there are only very few languages (essentially those with a group structure) for which we can construct zero-knowledge protocols that are efficient enough to be used in practice.

In this paper we address the problem of how to construct efficient zero-knowledge protocols for generic languages and we propose a protocol based on Yao's garbled circuit technique.

The motivation for our work is that in many cryptographic applications it is useful to be able to prove efficiently statements of the form e.g., "I know x s.t. $y = \text{SHA-256}(x)$" for a common input y (or other "unstructured" languages), but no efficient protocols for this task are currently known.

It is clear that zero-knowledge is a subset of secure two-party computation (i.e., any protocol for generic secure computation can be used to do zero-knowledge). The main contribution of this paper is to construct an efficient protocol for the special case of secure two-party computation where only one party has input (like in the zero-knowledge case). The protocol achieves active security and is essentially only twice as slow as the passive secure version of Yao's garbled circuit protocol. This is a great improvement with respect to the *cut-n-choose* technique to make Yao's protocol actively secure, where the complexity grows linearly with the security parameter.

Categories and Subject Descriptors

D.4.6 [**Operating Systems**]: Security and Protection—*Cryptographic controls*; C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*distributed applications*

*A full version of this paper is available at [JKO13].

CCS'13, November 4–8, 2013, Berlin, Germany.

ACM 978-1-4503-2477-9/13/11 ...$15.00.
http://dx.doi.org/10.1145/2508859.2516662 .

Keywords

Zero-Knowledge Proof; Garbled Circuits; Efficiency

1. INTRODUCTION

Zero-knowledge (ZK) protocols have countless applications in cryptography and therefore efficiency is of paramount importance. Consequently, a huge effort has been put into designing efficient ZK protocols for *specific tasks*. In particular there are very efficient protocols for languages with some algebraic structure. There are, for instance, efficient protocols for proving knowledge and relations of discrete logarithms [Sch89, CDS94], for proving that RSA public keys are well formed [CM99], for statements in post-quantum cryptography [BD10, JKPT12, MV03], for bilinear equations [GS12, GSW10], for shuffles [BG12, KMW12] and frameworks for modular design of zero-knowledge protocols [CKS11].

However, *generic constructions* for ZK protocols use Karp reductions to NP-complete languages and are therefore too impractical to be used in practice. In particular, so far there has been no practical solution to problems that do not exhibit an algebraic structure. Examples for protocols that could be used in many cryptographic applications are e.g., the problem of efficiently proving statements of the form "I know x s.t. $y = \text{SHA-256}(x)$" or "I know k s.t. $y_1 = \text{AES}_k(y_2)$" (the common input is y in the first example and (y_1, y_2) in the second)[1].

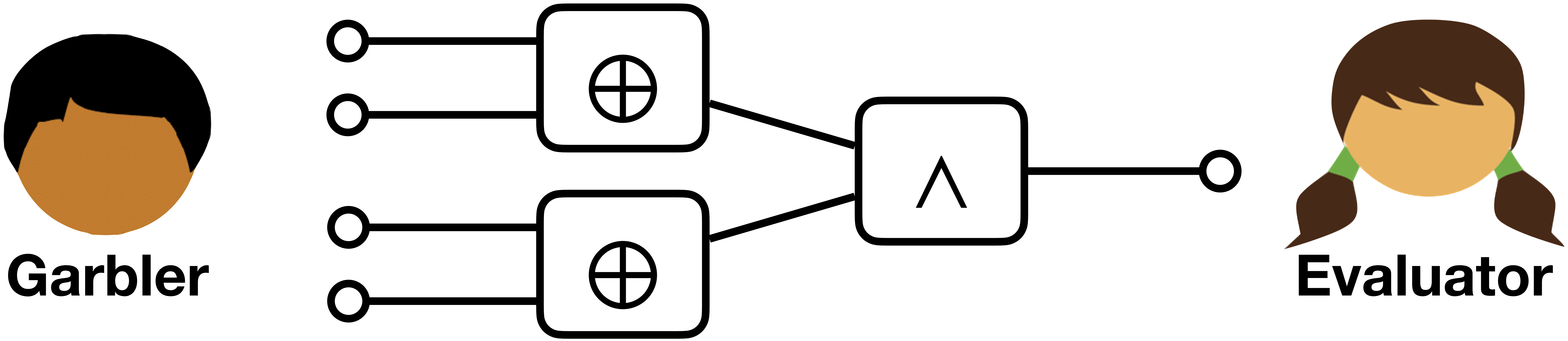
In this work we provide a *generic* and *efficient* solution for proving any such statements in zero-knowledge, by constructing a protocol based on Yao's garbled circuits technique. The complexity of our protocol is proportional to the size of the circuit of the NP verification function. To support the validity of our efficiency claim, we present also a *proof-of-concept* implementation of our protocol. The performance measurements of our prototypical implementation show the viability of our protocol for realistic problems.

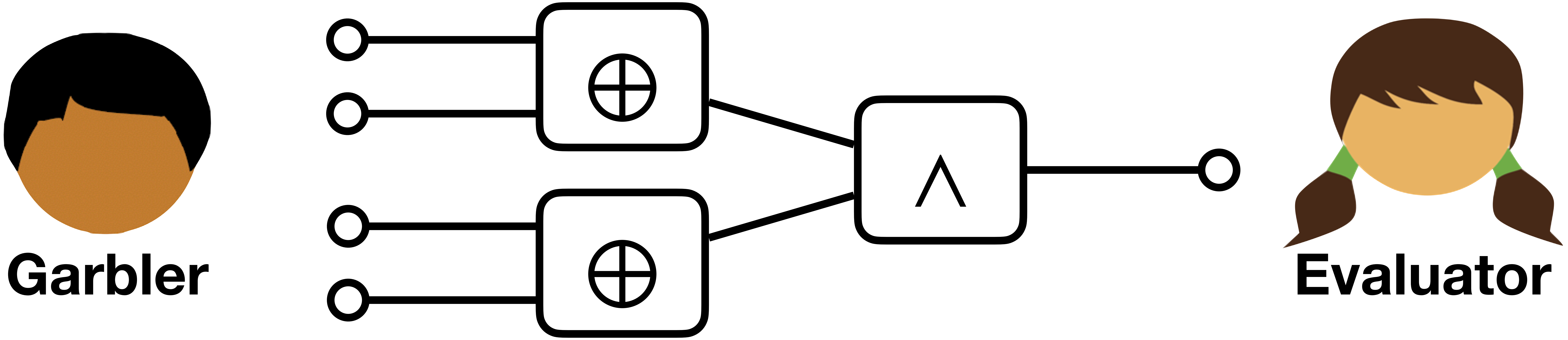
1.1 Zero-Knowledge and 2PC

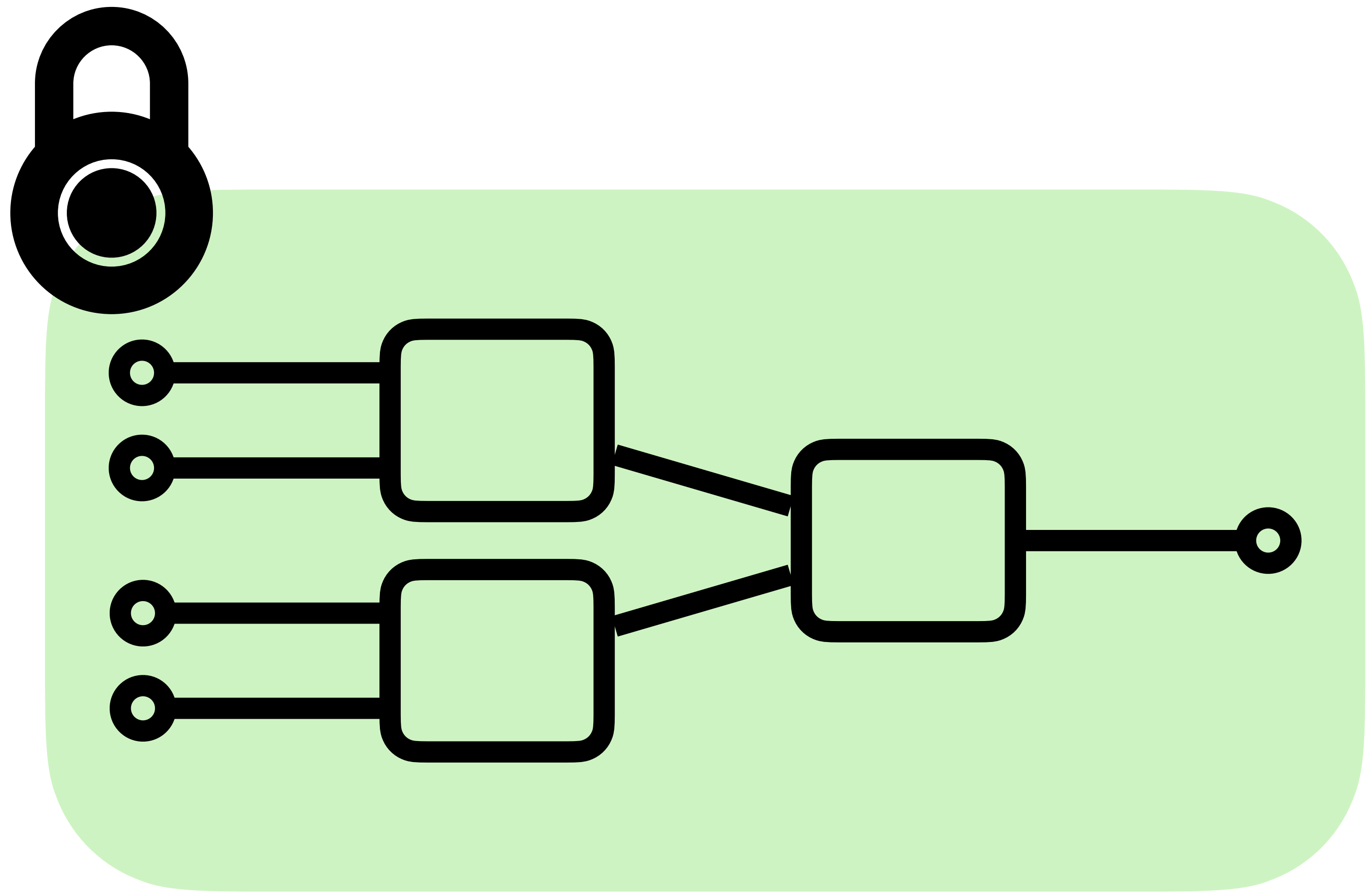
Zero-knowledge proofs were introduced more than 30 years ago by Goldwasser, Micali and Rackoff [GMR85]. A zero-knowledge argument (ZK) is an interactive protocol that allows a prover P to persuade a verifier V of the validity of some NP statement y by using the knowledge of a witness w. Informally, an honest prover should be able to

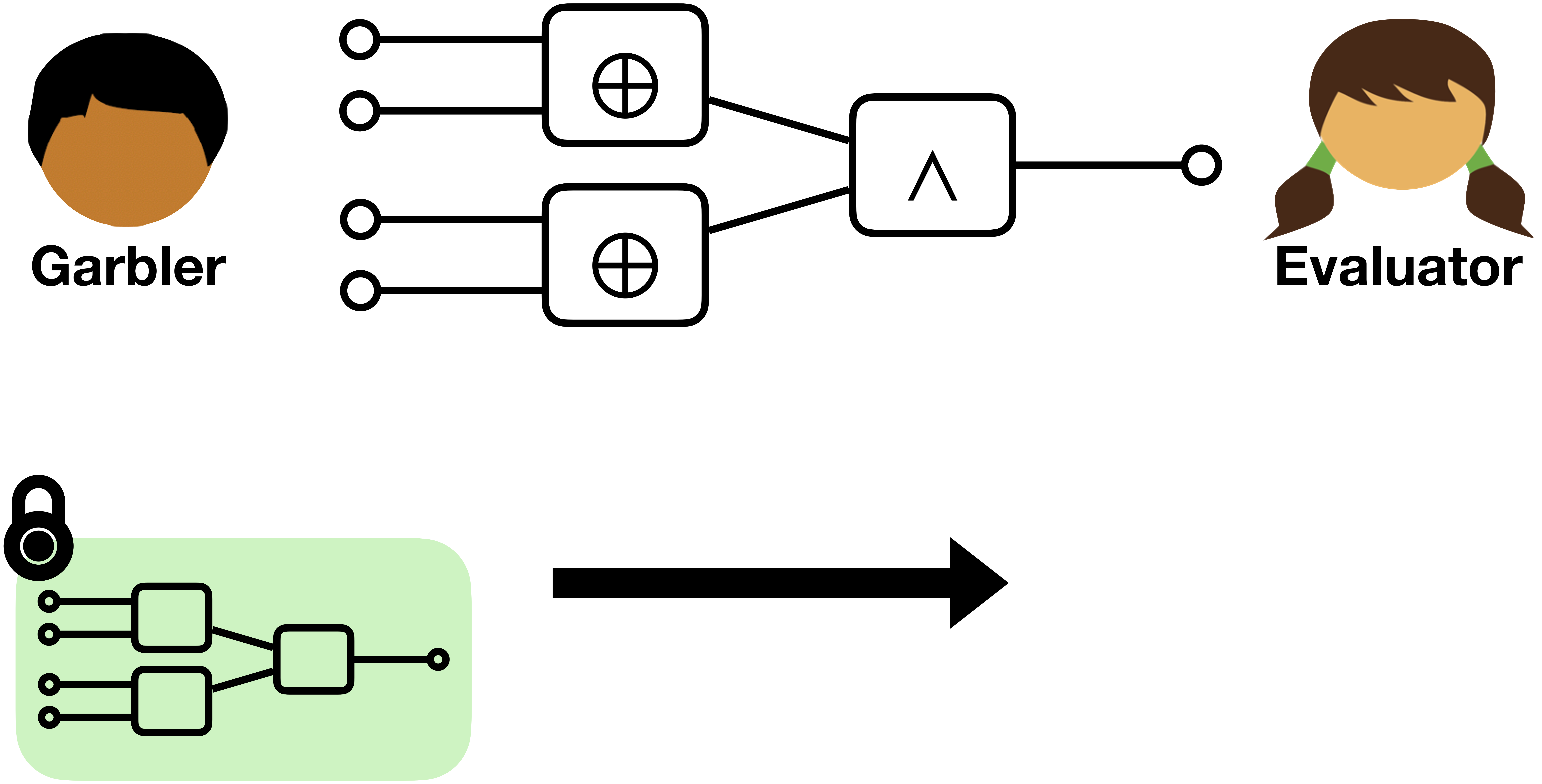
[1]Note that in both cases the prover is not only showing that the instance belongs to the language (both languages are trivial), but moreover that the prover *knows* a valid witness for this. So these proofs are meaningful as we believe that it is hard to compute such a witness.

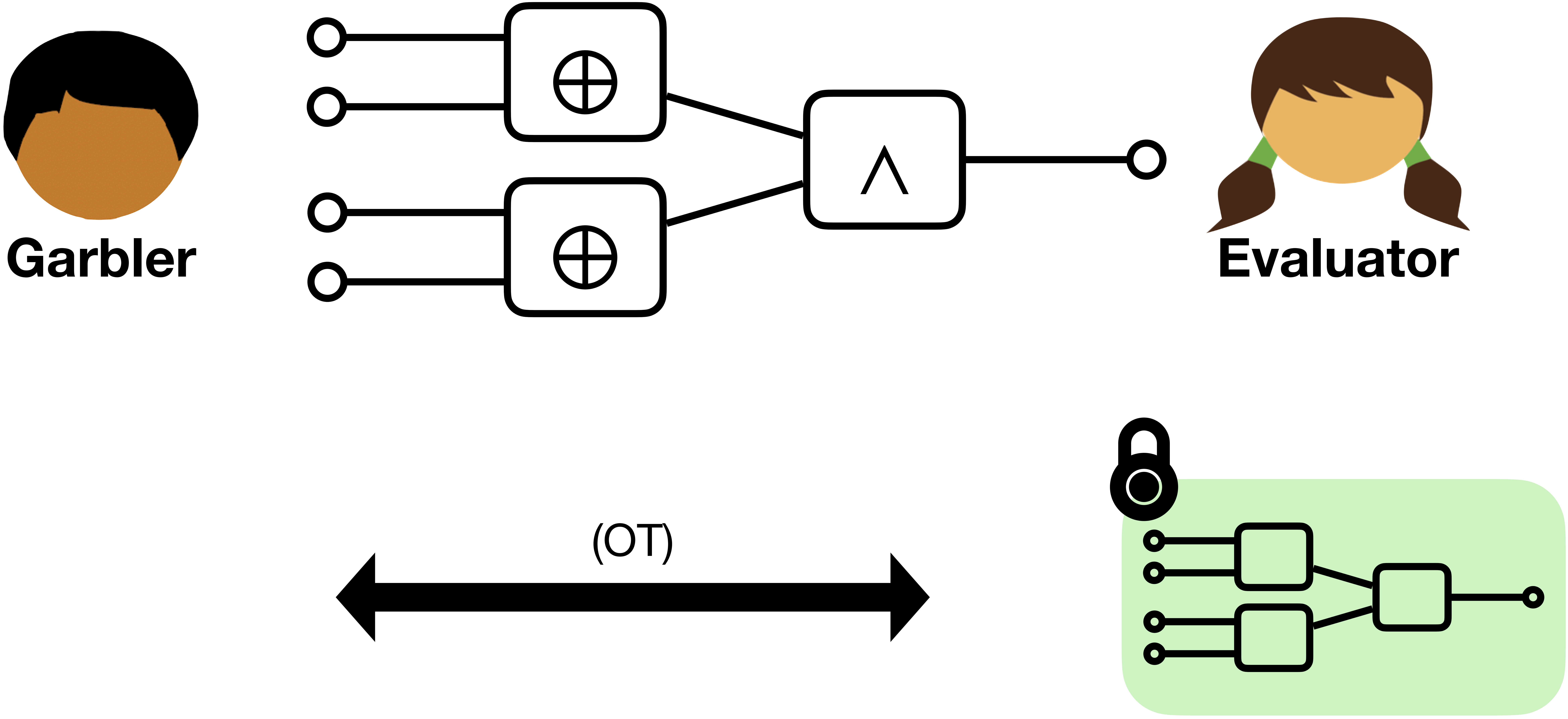
Refresher: Garbled Circuits

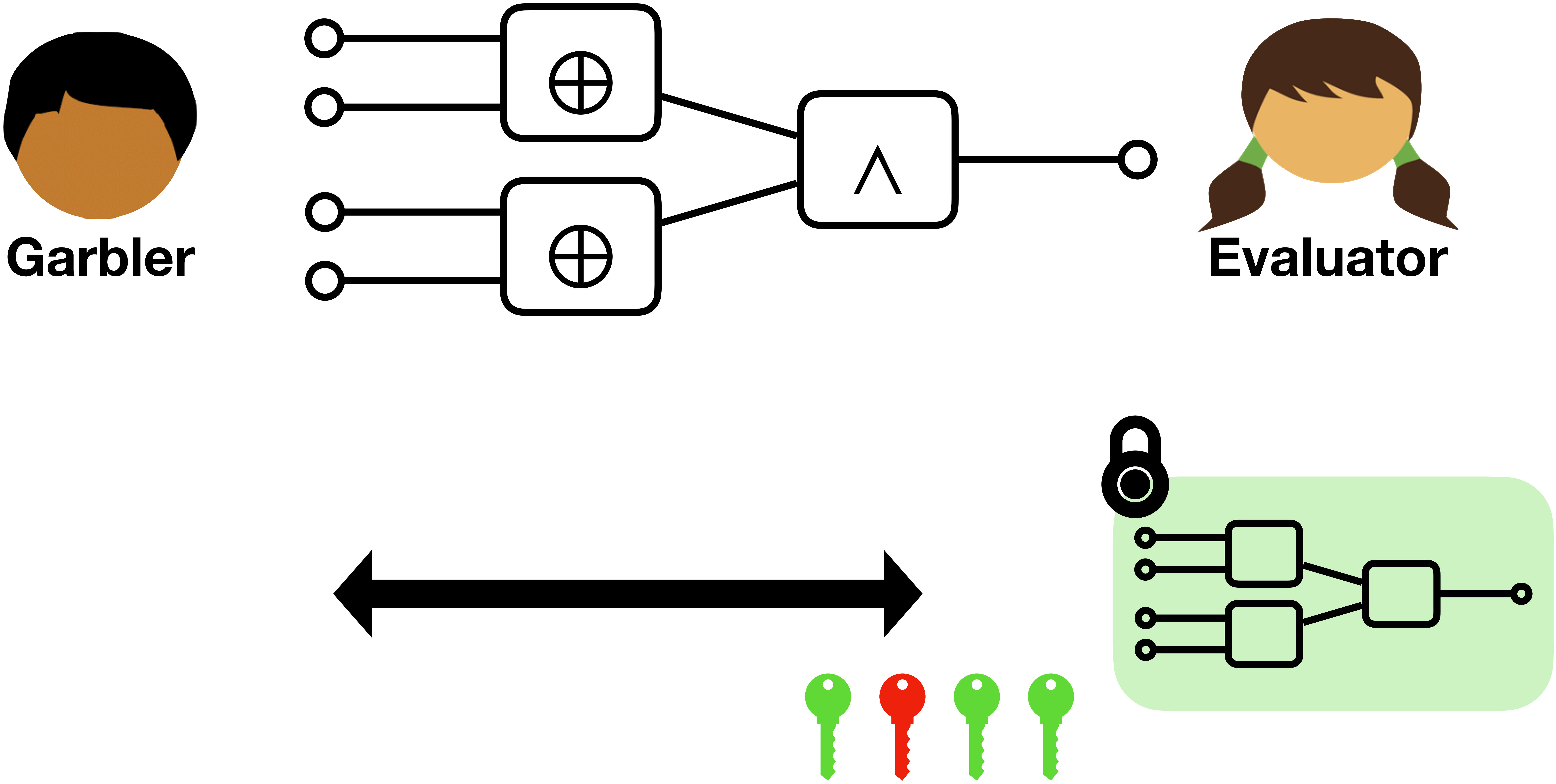

Garbler
⊕
⊕
∧
Evaluator

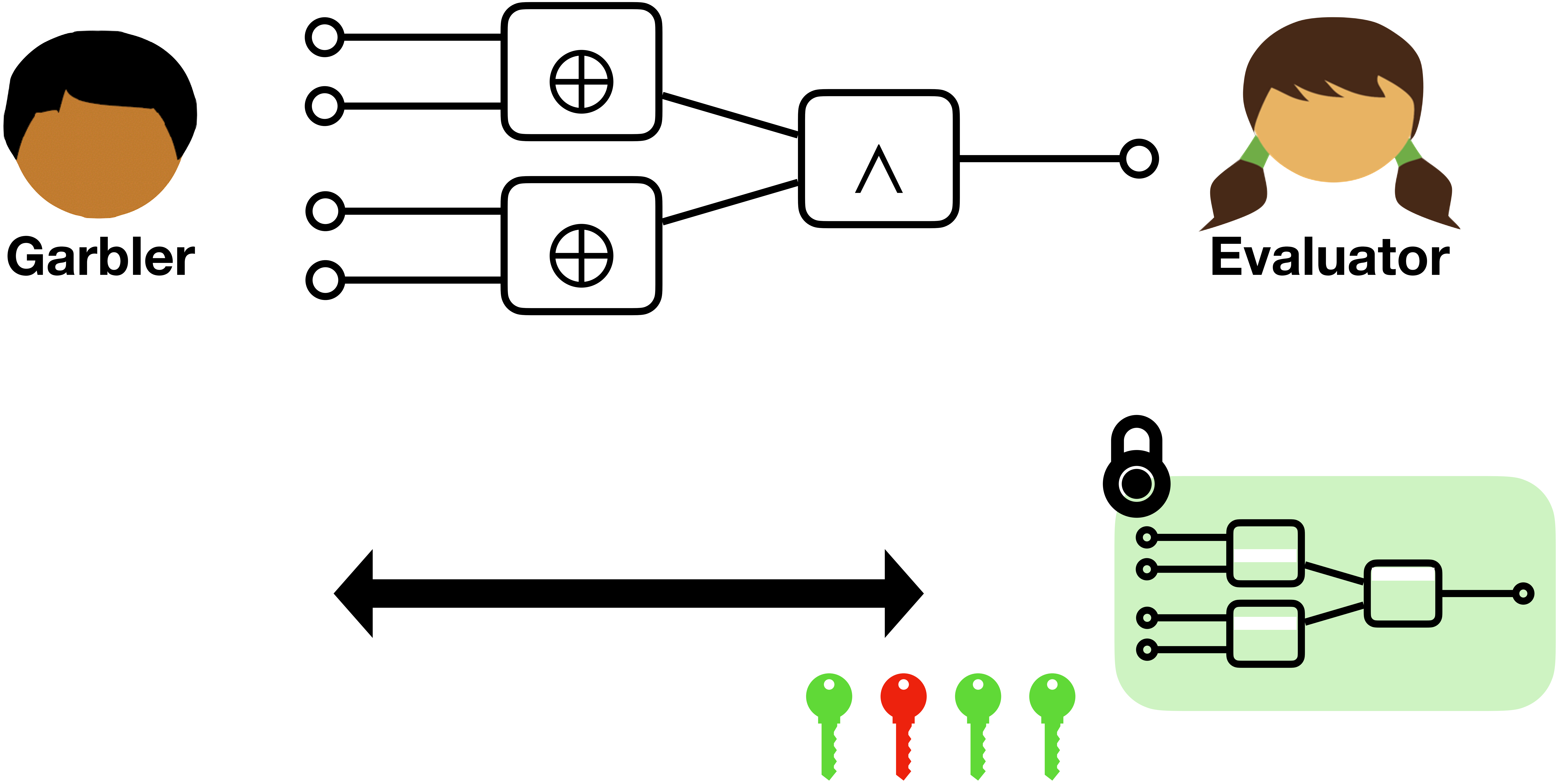

Garbler
⊕
⊕
∧
Evaluator



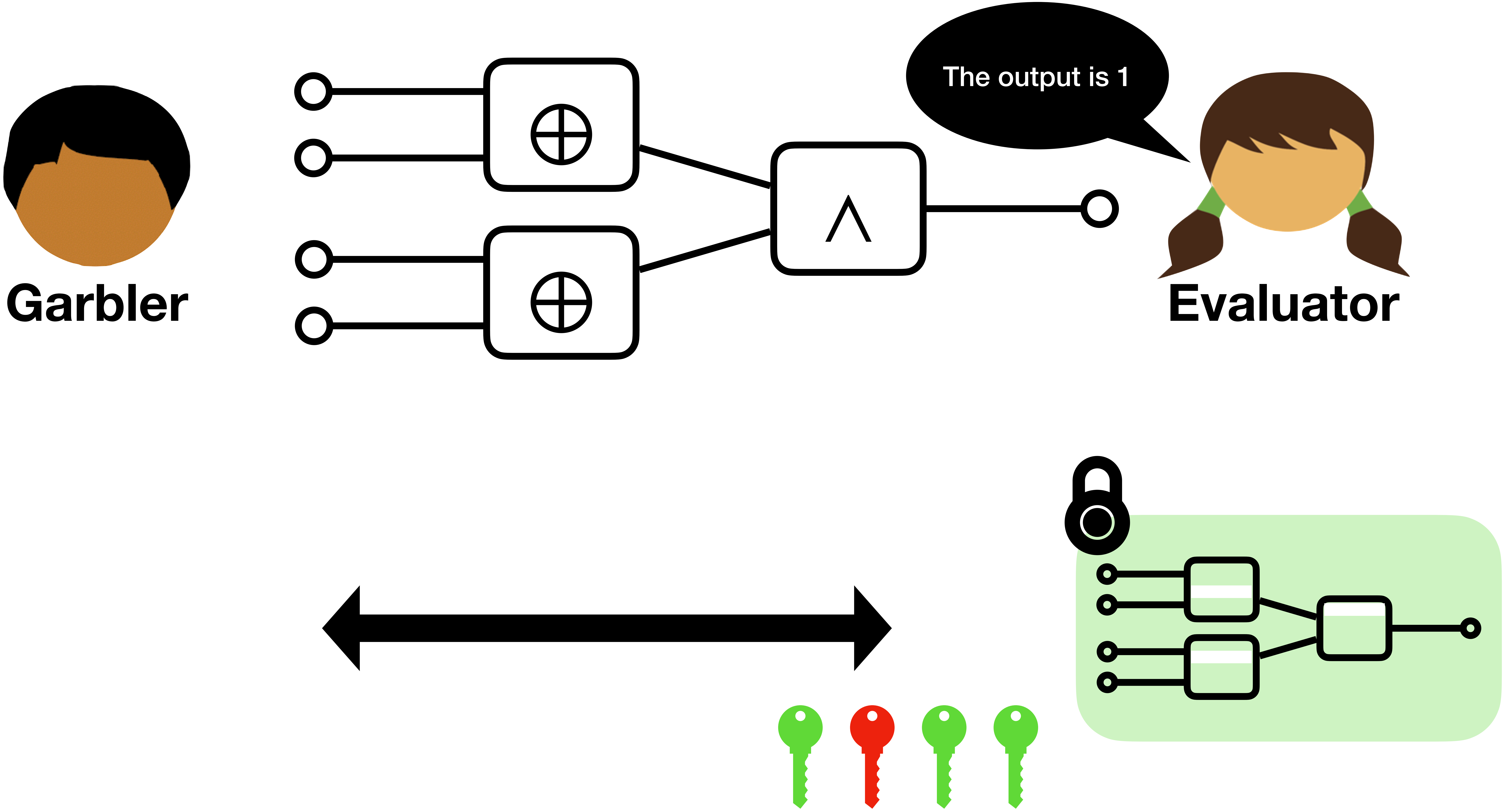

Garbler
⊕
⊕
∧
Evaluator

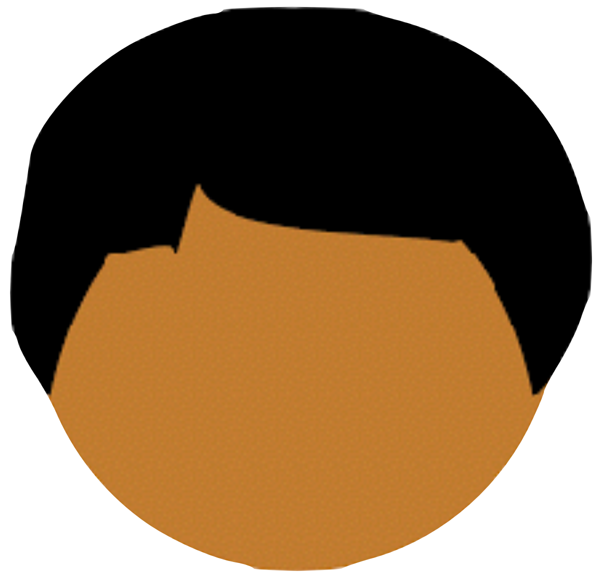

Garbler
⊕
⊕
∧
Evaluator
(OT)


Garbler
Evaluator

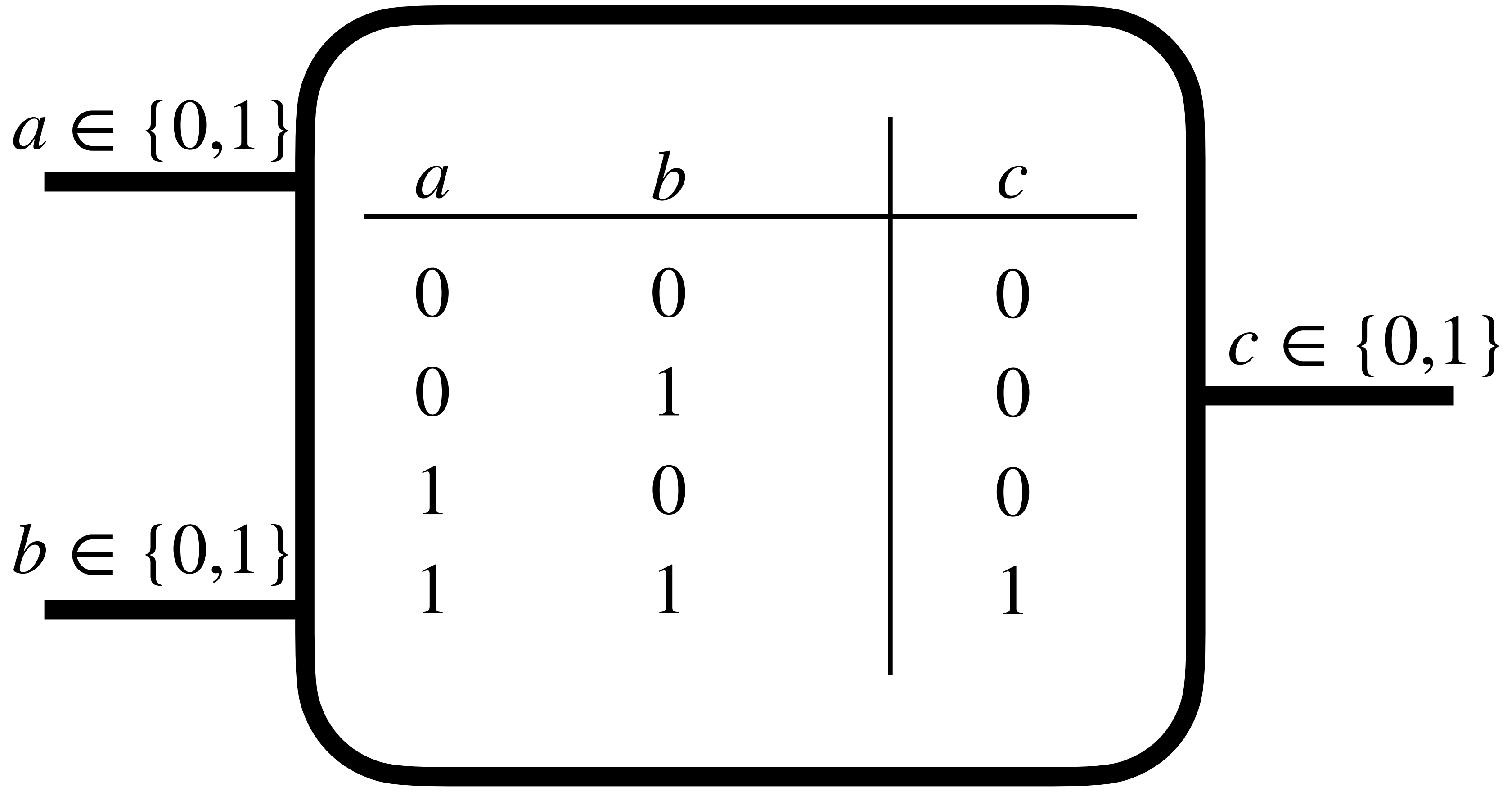


Garbler
Evaluator

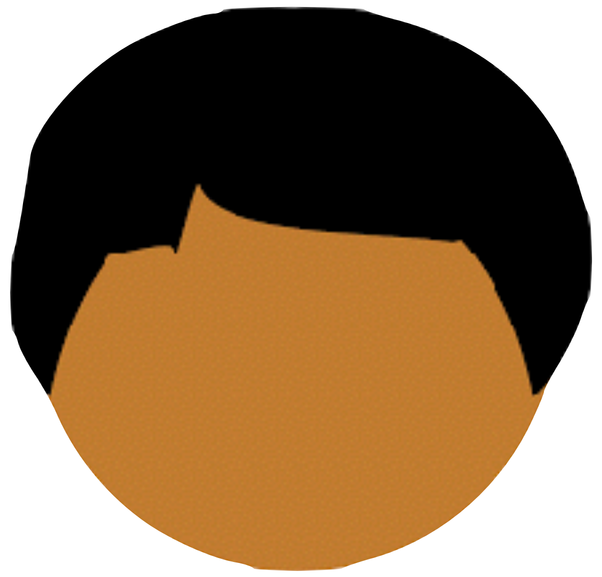

The output is 1
Garbler
⊕
⊕
∧
Evaluator



G



a	b	c
0	0	0
0	1	0
1	0	0
1	1	1



G

$a \in \{0,1\}$

$b \in \{0,1\}$

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

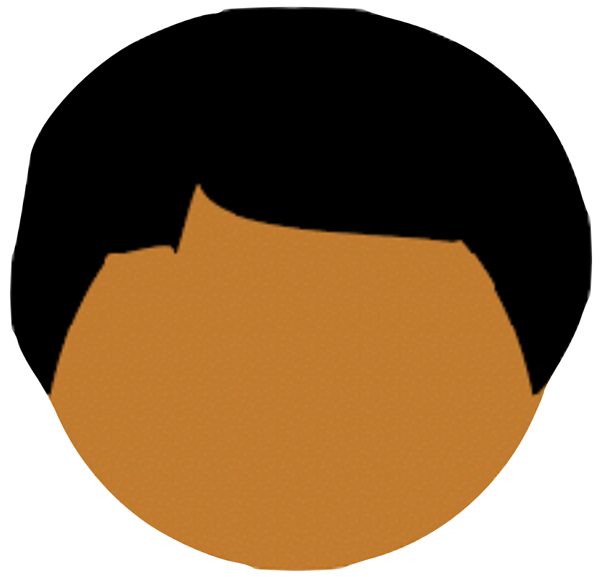
$c \in \{0,1\}$

$a \in \{K_a^0, K_a^1\}$

$b \in \{K_b^0, K_b^1\}$

a	b	c
K_a^0	K_b^0	K_c^0
K_a^0	K_b^1	K_c^0
K_a^1	K_b^0	K_c^0
K_a^1	K_b^1	K_c^1

$c \in \{K_c^0, K_c^1\}$



G

$a \in \{0,1\}$

$b \in \{0,1\}$

a	b	c
0	0	0
0	1	0
1	0	0
1	1	1

$c \in \{0,1\}$

$a \in \{K_a^0, K_a^1\}$

$b \in \{K_b^0, K_b^1\}$

a	b	c
K_a^0	K_b^0	K_c^0
K_a^0	K_b^1	K_c^0
K_a^1	K_b^0	K_c^0
K_a^1	K_b^1	K_c^1

$c \in \{K_c^0, K_c^1\}$

$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^0, K_c^0))$$
$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^1, K_c^0))$$
$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^0, K_c^0))$$
$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^1, K_c^1))$$

Zero Knowledge proof of circuit satisfiability

Proof system allows proofs of the form *"this circuit is satisfiable"*

There exists an input
s.t. the circuit outputs 1

Statement:

Witness:

Zero Knowledge proof of circuit satisfiability

Proof system allows proofs of the form *"this circuit is satisfiable"*

There exists an input
s.t. the circuit outputs 1

Statement: Boolean circuit C

Witness: A string x **s.t.** $C(x) = 1$

Zero Knowledge proof of circuit satisfiability

Proof system allows proofs of the form *"this circuit is satisfiable"*

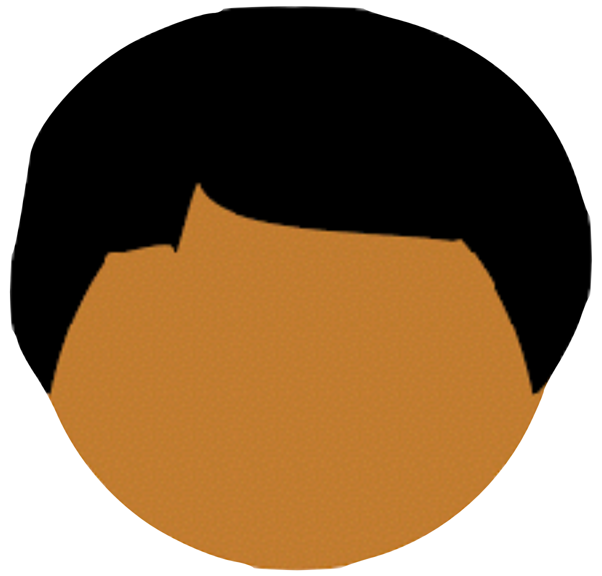
There exists an input
s.t. the circuit outputs 1

Statement: Boolean circuit C

Witness: A string x **s.t.** $C(x) = 1$

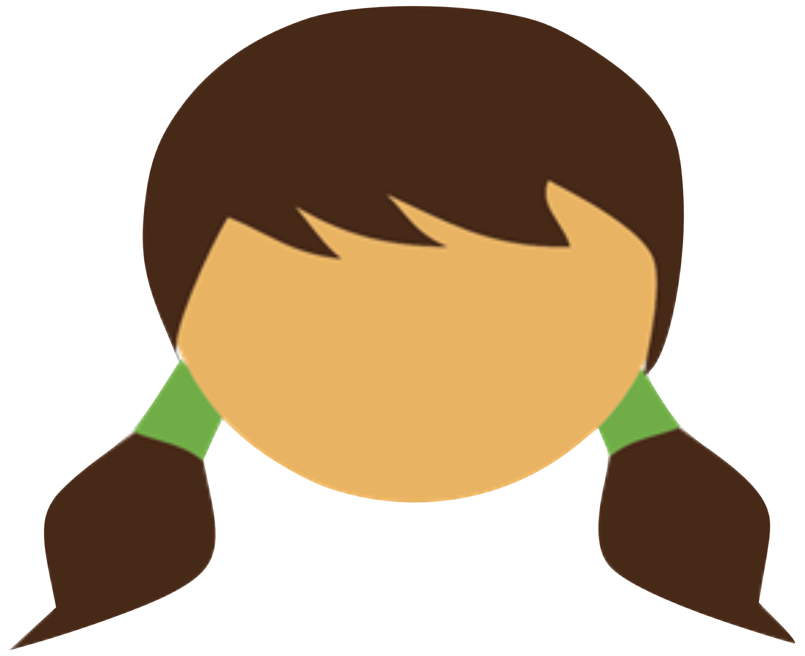
It is relatively easy to compile arbitrary provable (NP) statements to circuits

Zero Knowledge from Garbled Circuits



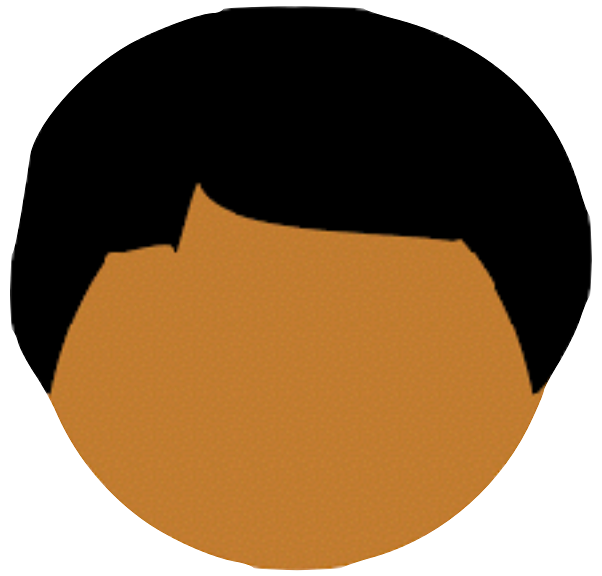
Verifier
Garbler

x

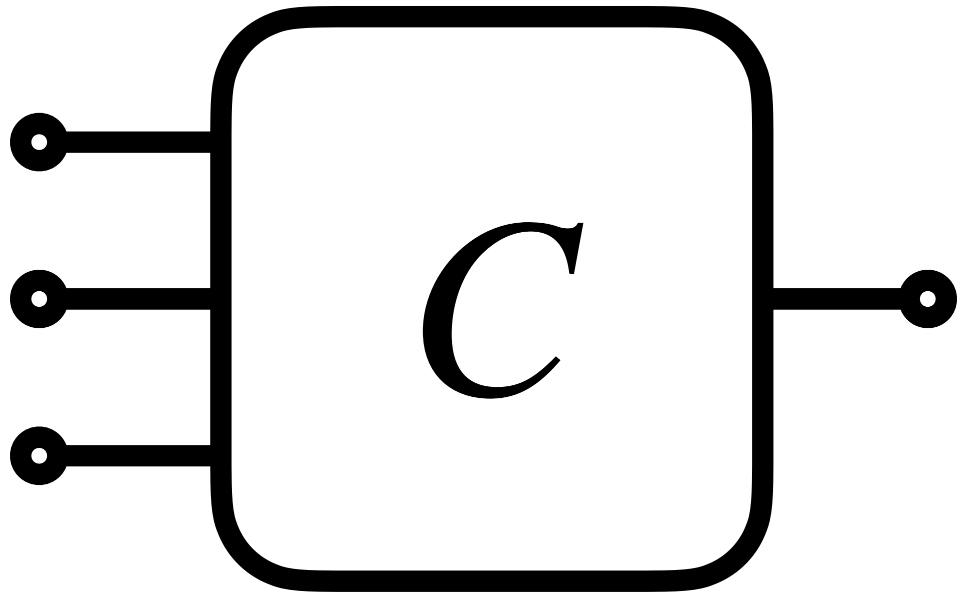


Prover
Evaluator

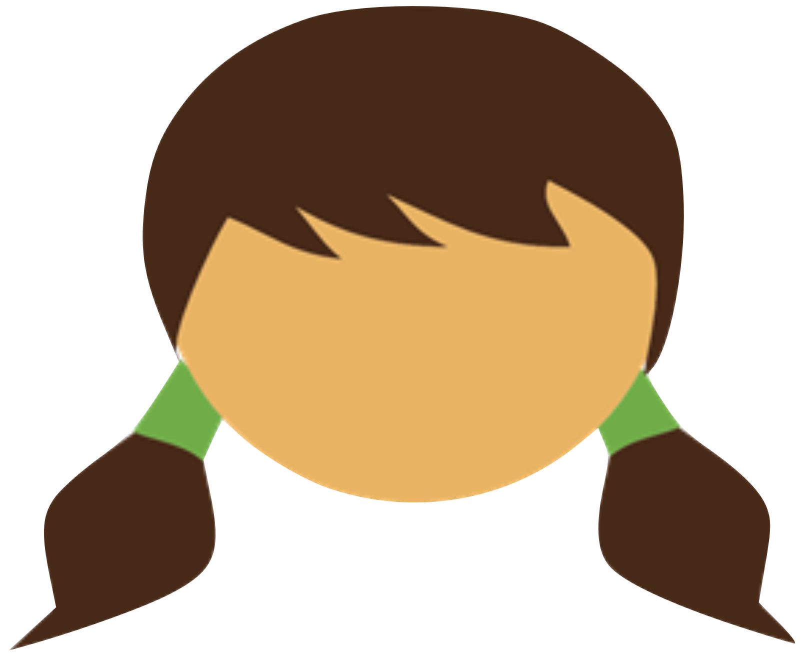
Zero Knowledge from Garbled Circuits



Verifier
Garbler

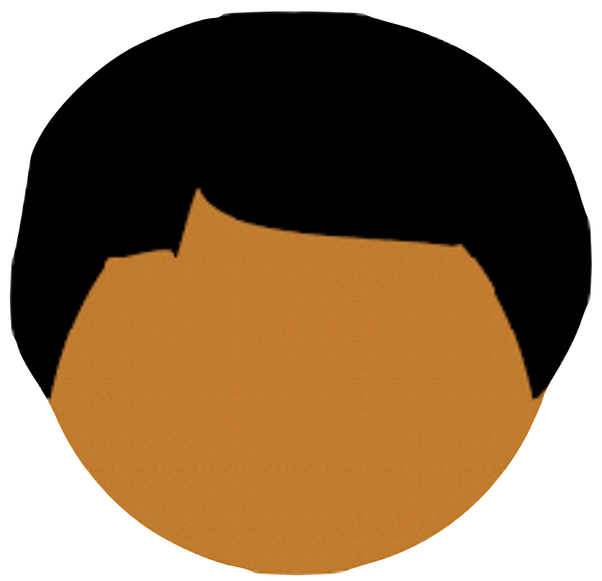


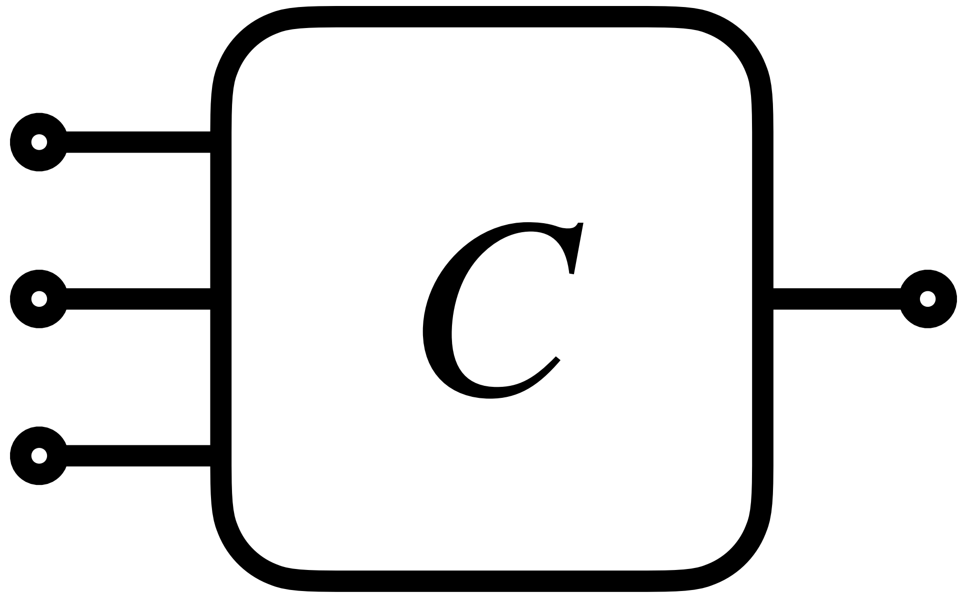
x



Prover
Evaluator

Zero Knowledge from Garbled Circuits



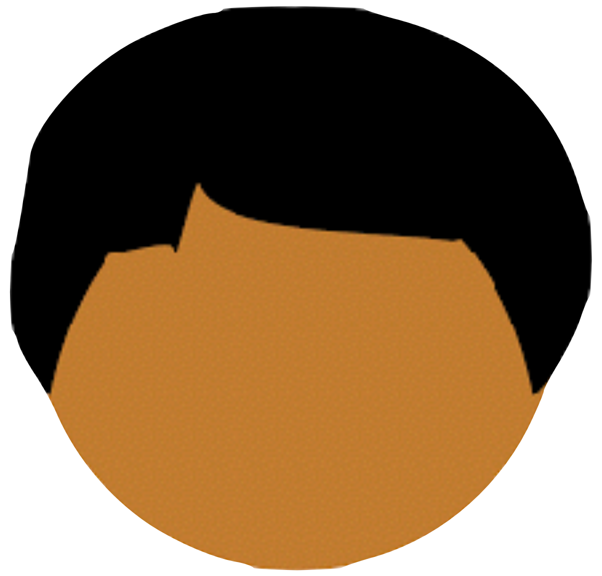


Verifier
Garbler

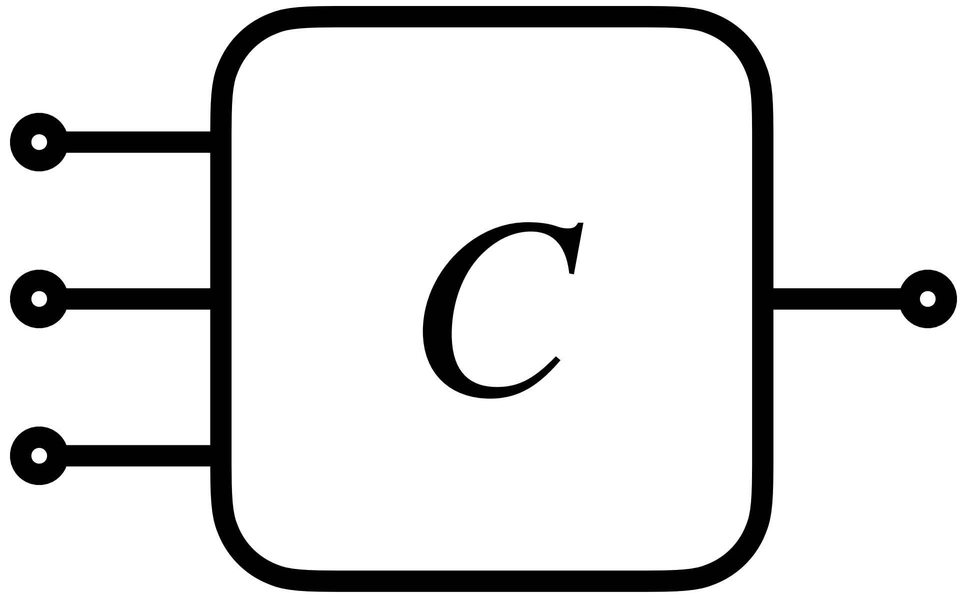
x **Prover**
Evaluator

Intuition: Garbled Circuit provides natural protection against cheating evaluator

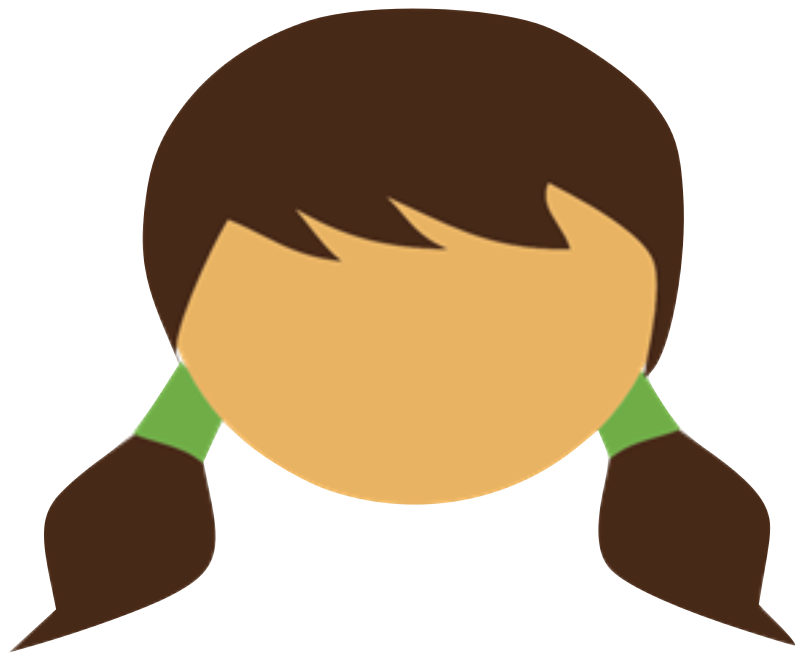
Zero Knowledge from Garbled Circuits



Verifier
Garbler



x

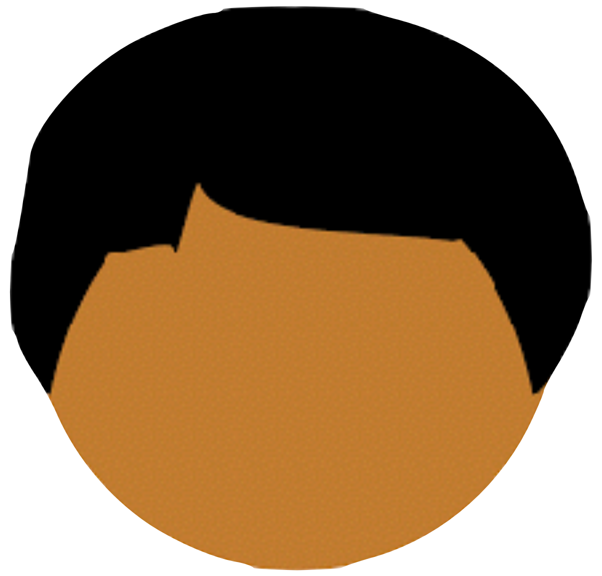


Prover
Evaluator

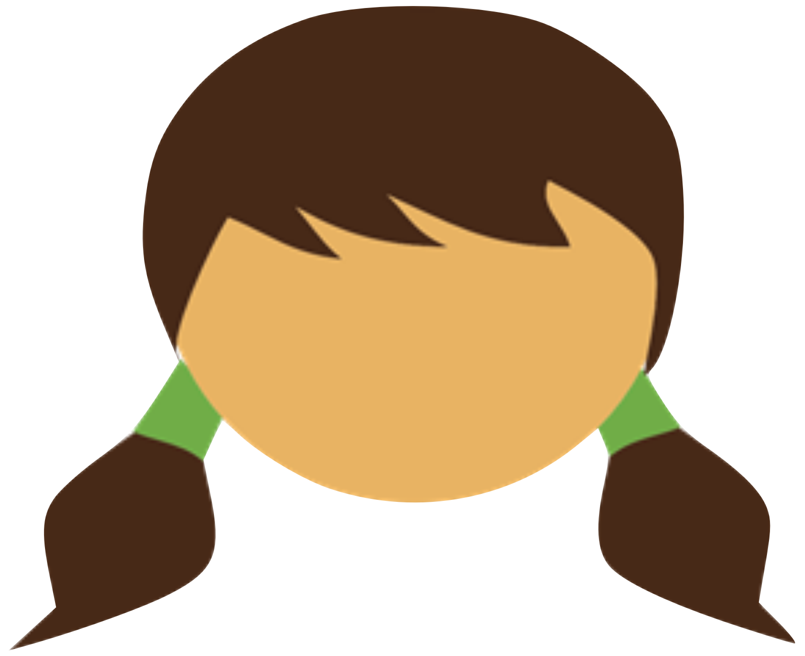
Intuition: Garbled Circuit provides natural protection against cheating evaluator

Just force evaluator to evaluate a garbling of C; the fact that she can come up with an output key that encodes 1 is convincing evidence she has a witness

Zero Knowledge from Garbled Circuits

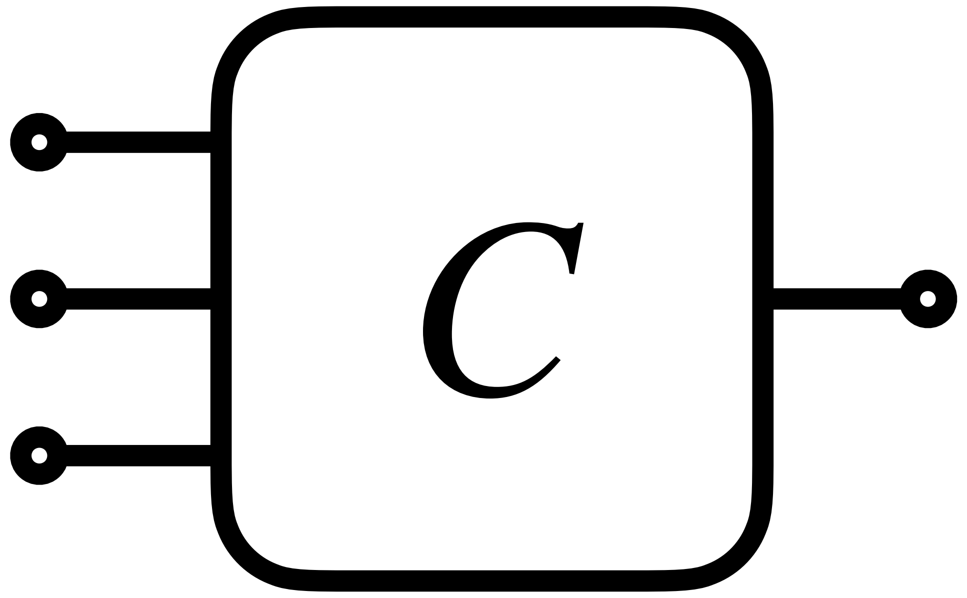


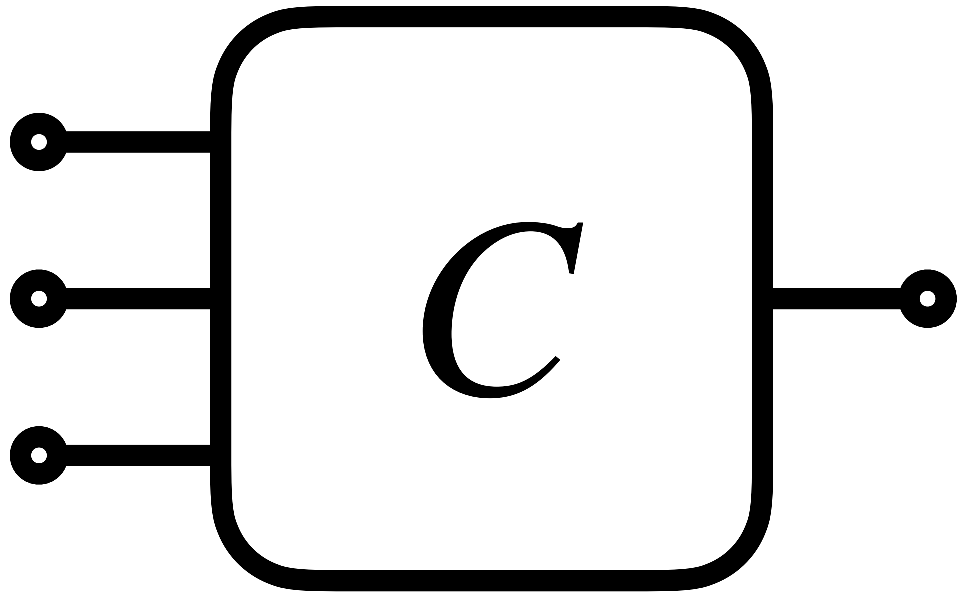
Verifier
Garbler



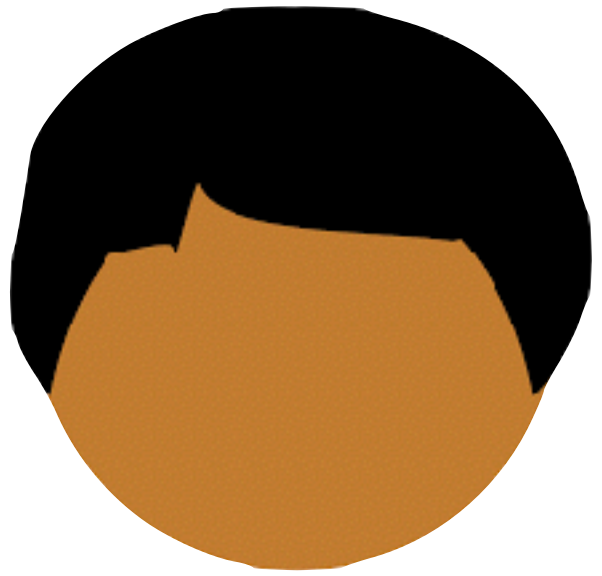
x

Prover
Evaluator

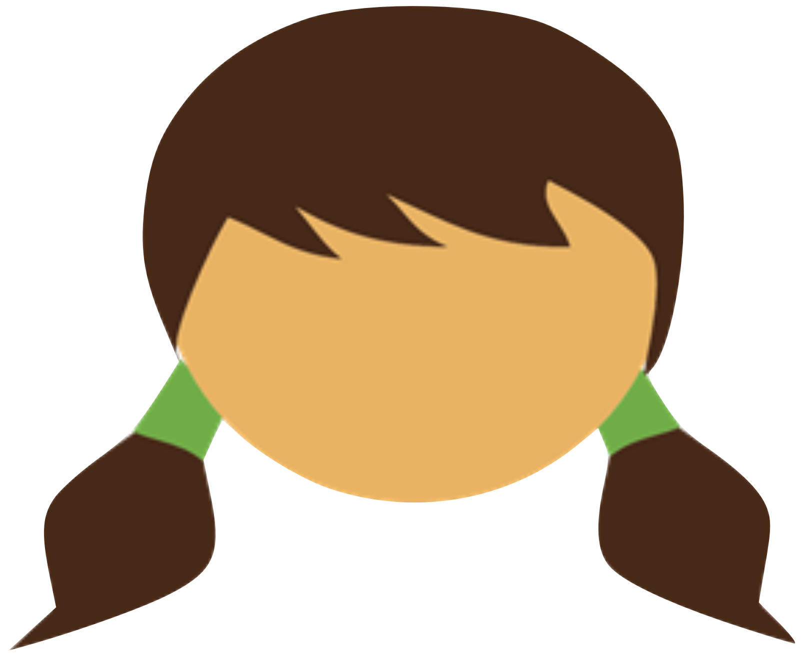




Zero Knowledge from Garbled Circuits

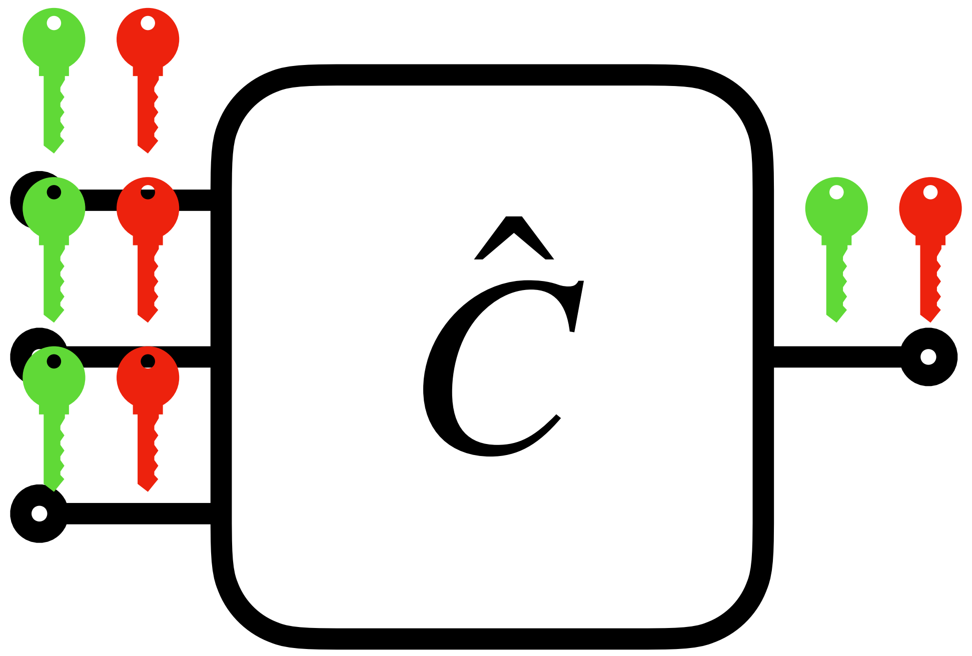


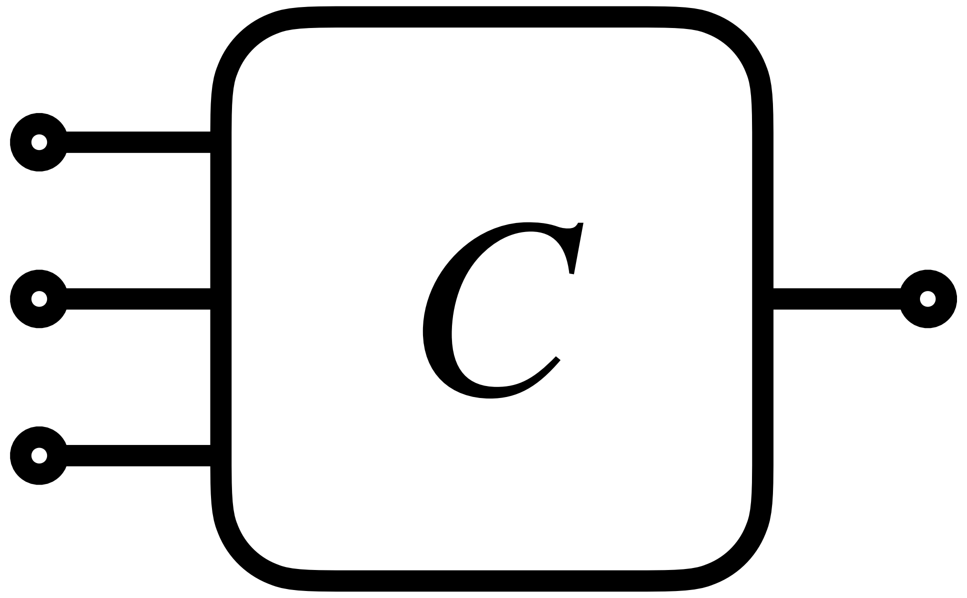
Verifier
Garbler



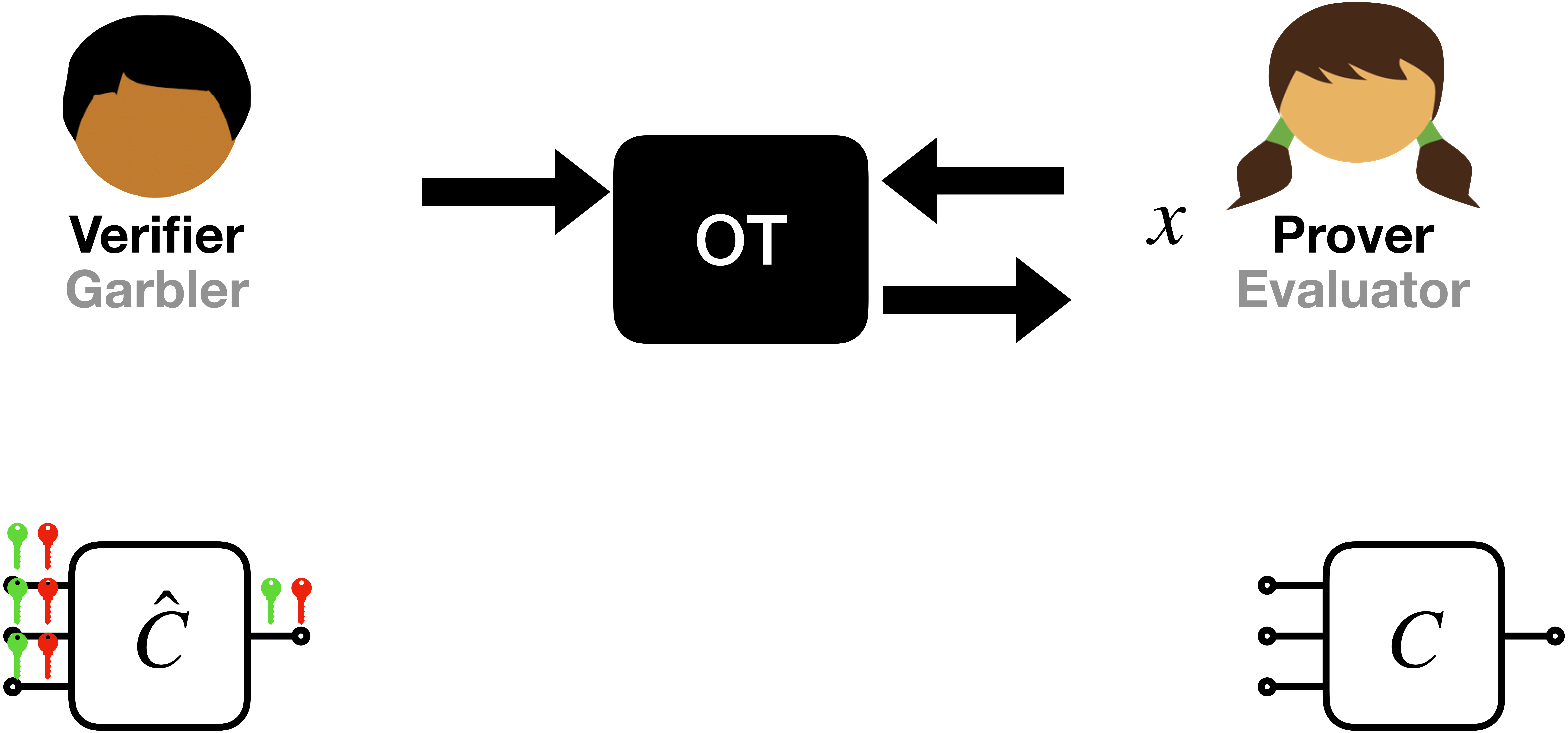
x

Prover
Evaluator

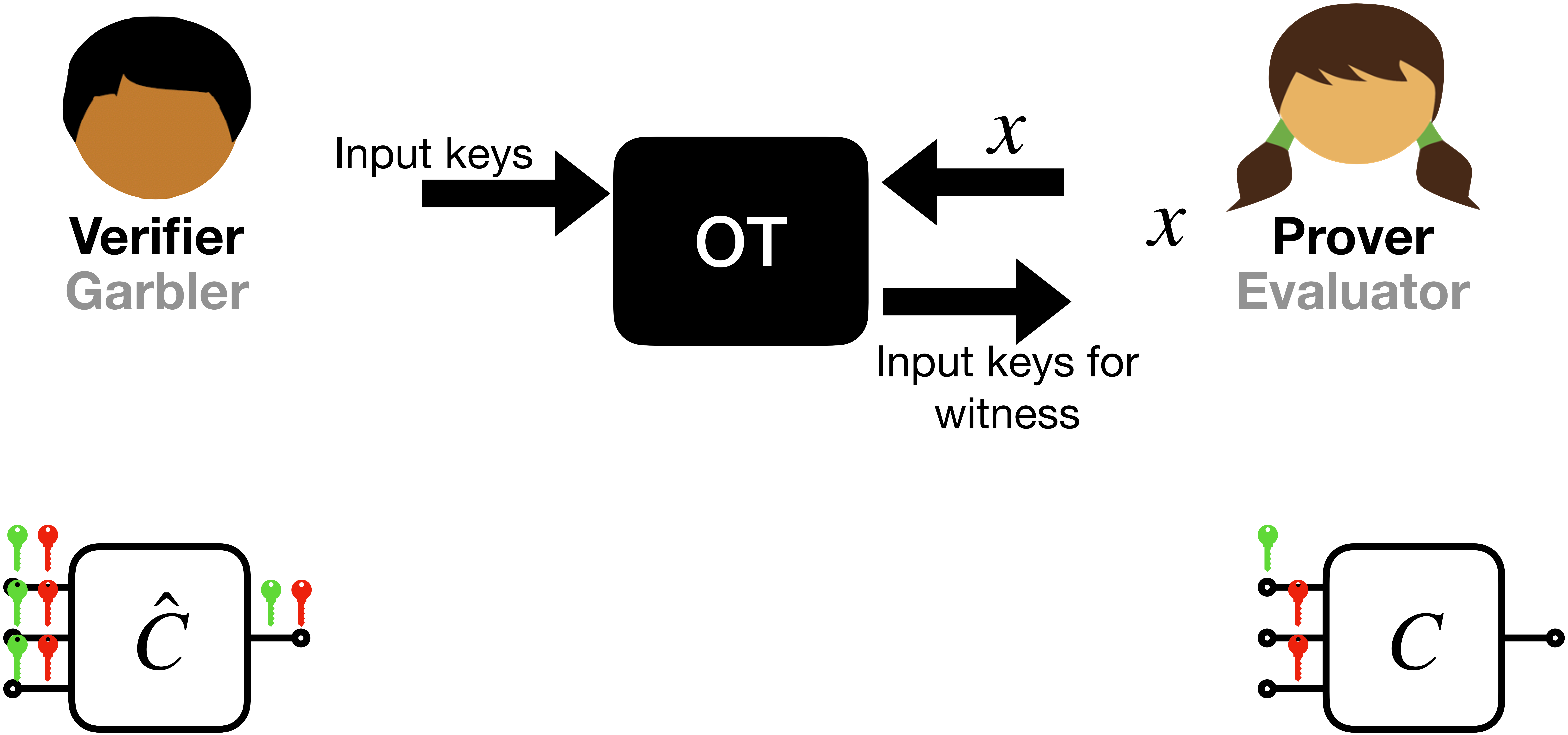




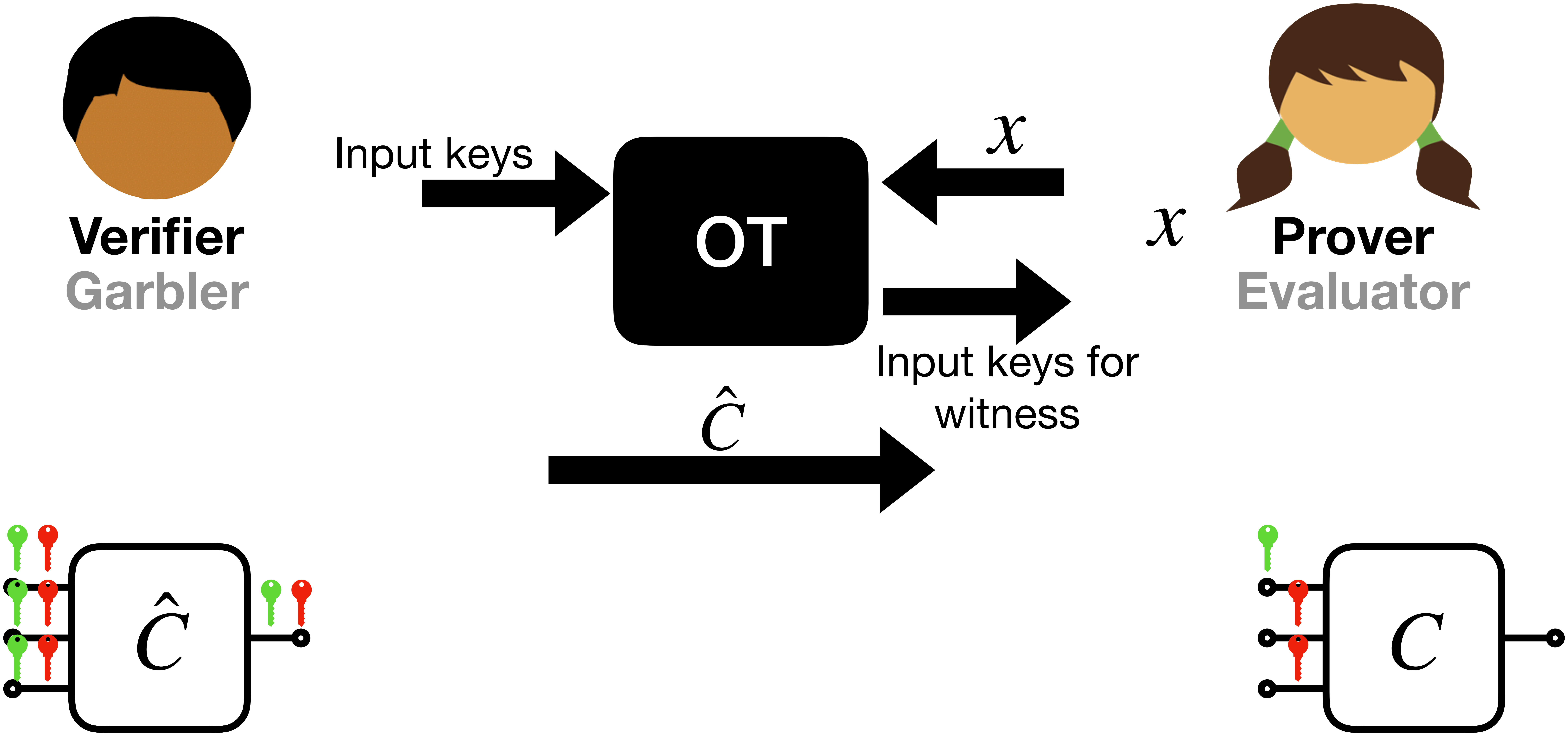
Zero Knowledge from Garbled Circuits



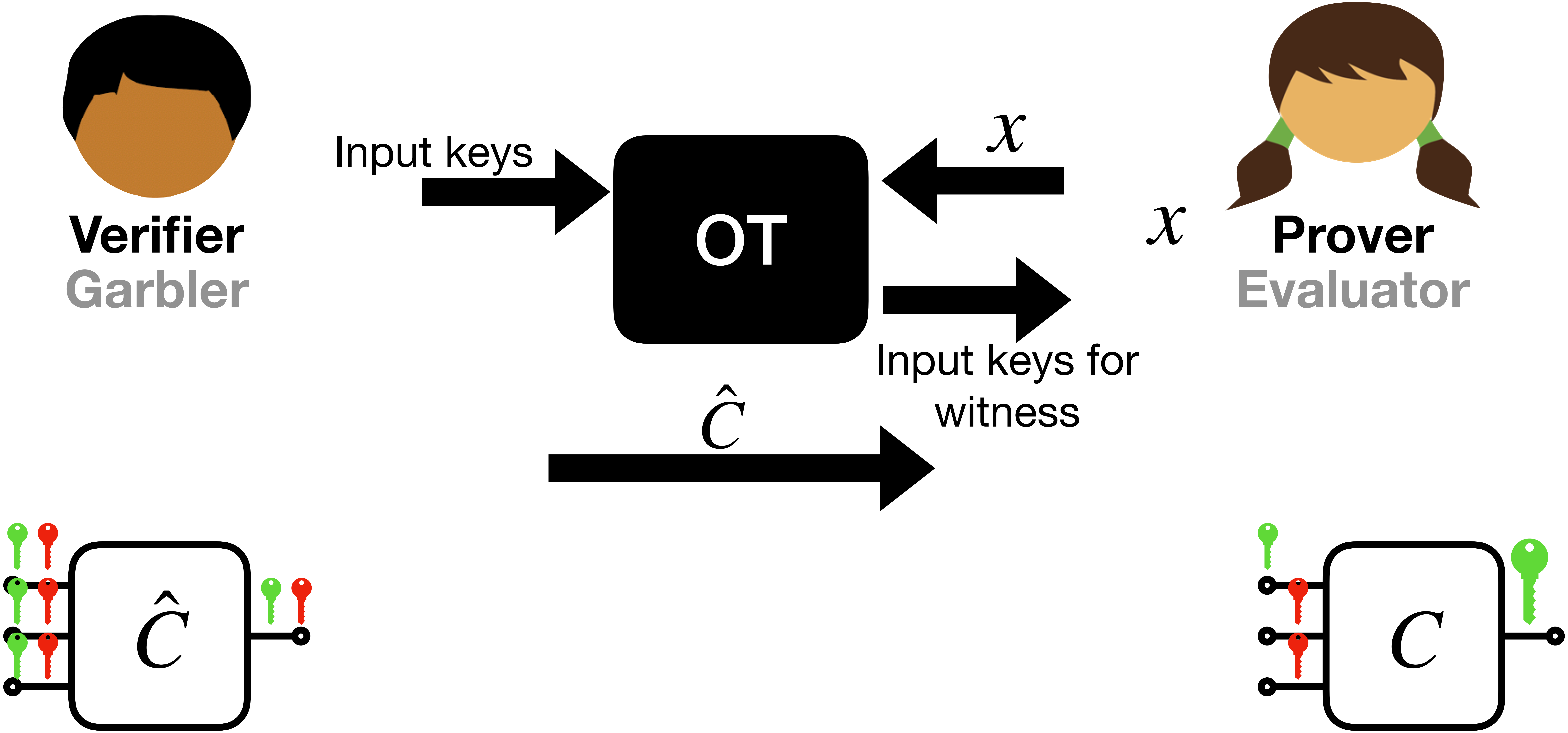
Zero Knowledge from Garbled Circuits



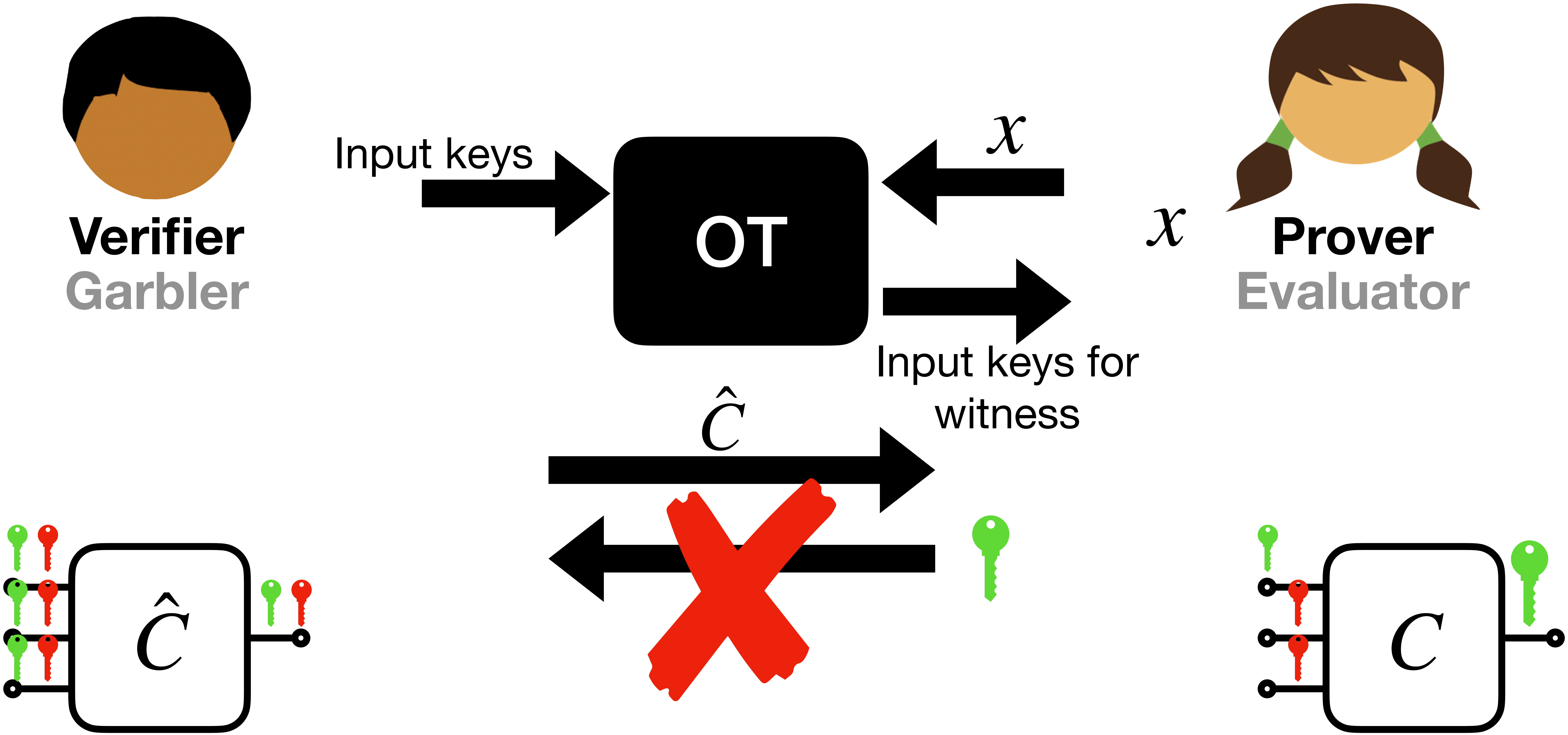
Zero Knowledge from Garbled Circuits

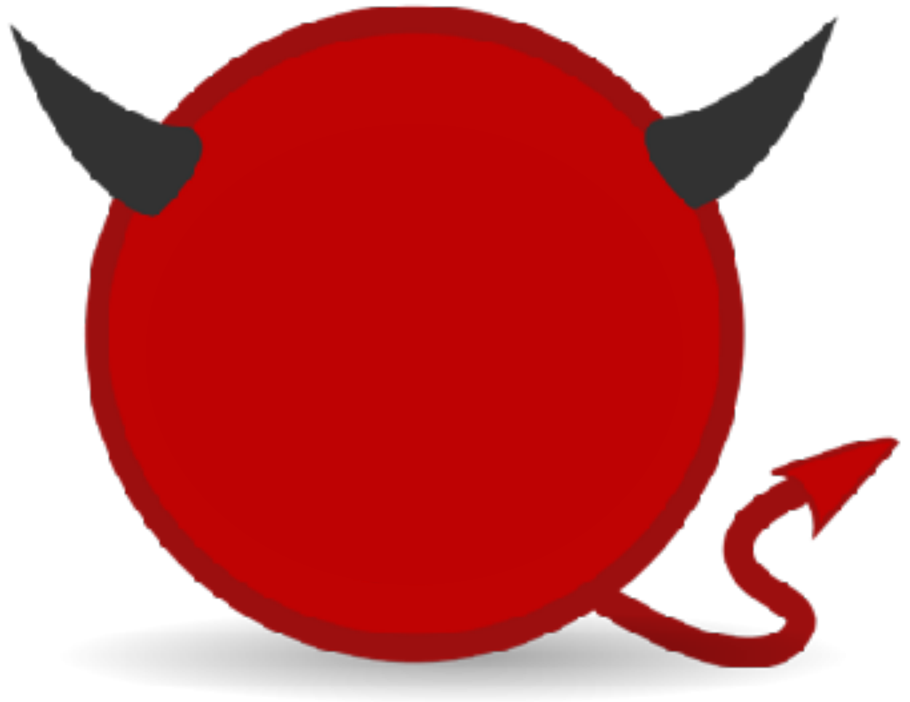


Zero Knowledge from Garbled Circuits



Zero Knowledge from Garbled Circuits





Verifier

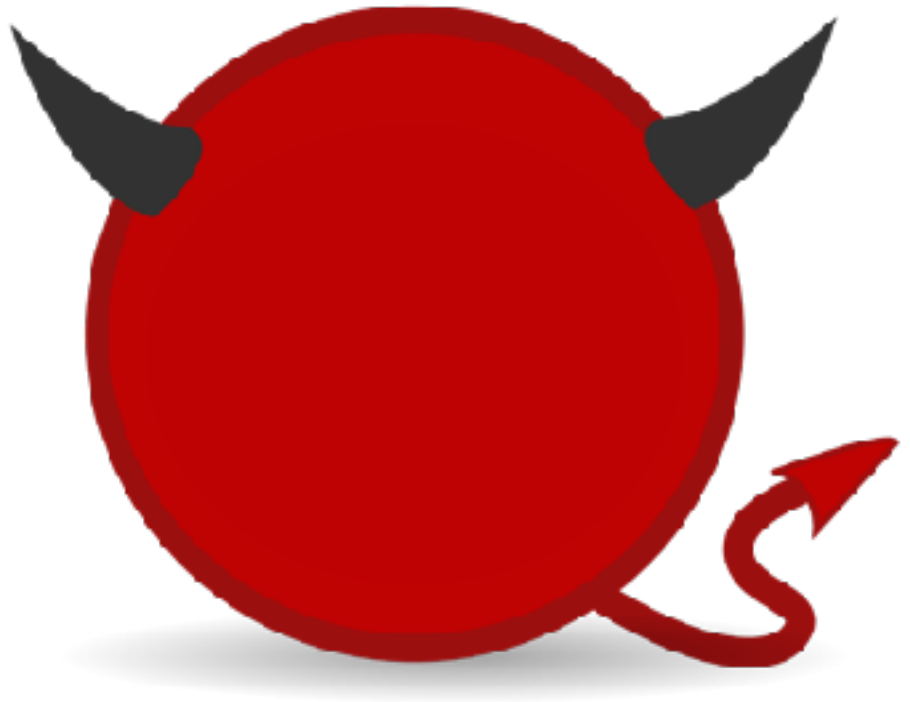
Garbler

$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^0, K_c^0))$$

$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^1, K_c^0))$$

$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^0, K_c^0))$$

$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^1, K_c^1))$$



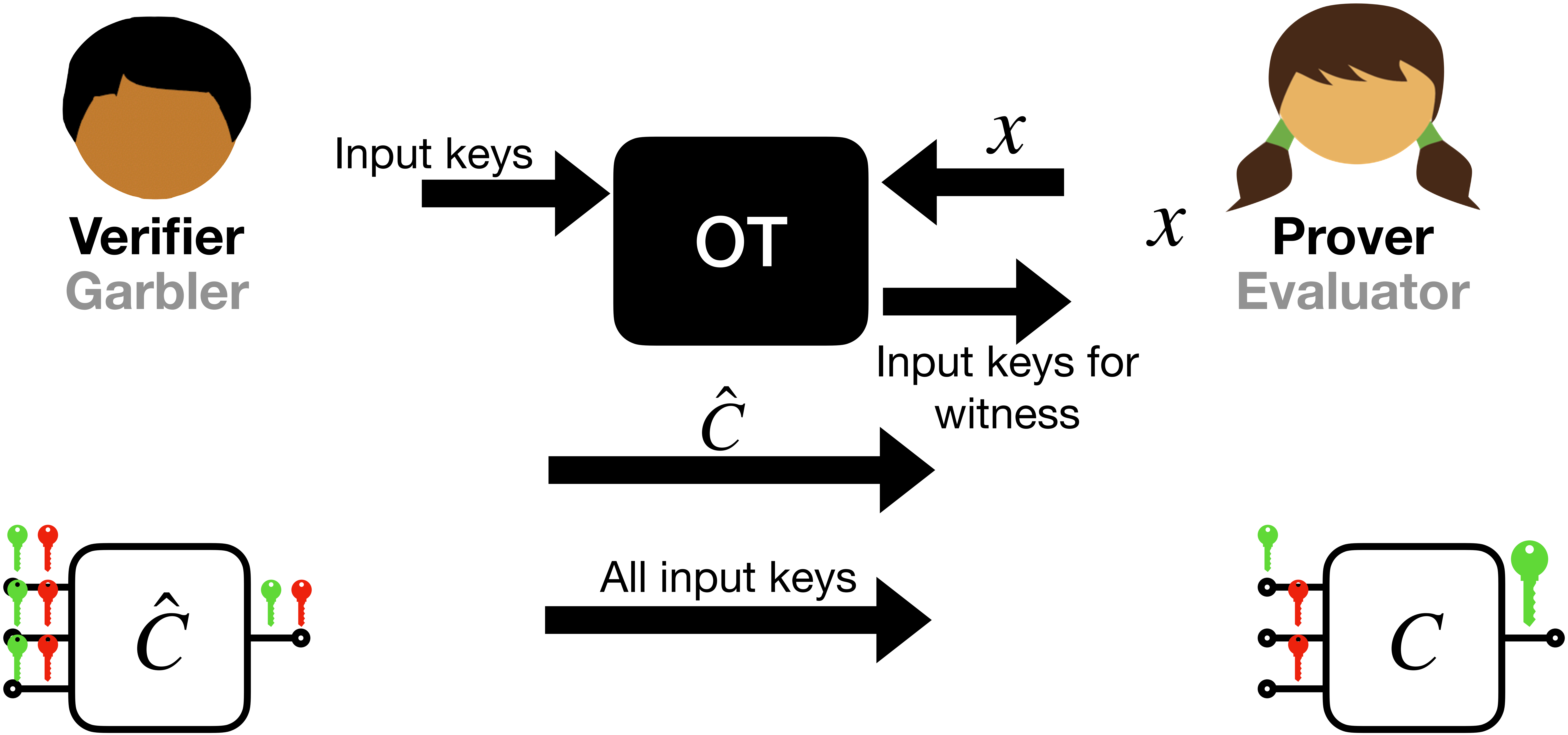
Verifier

Garbler

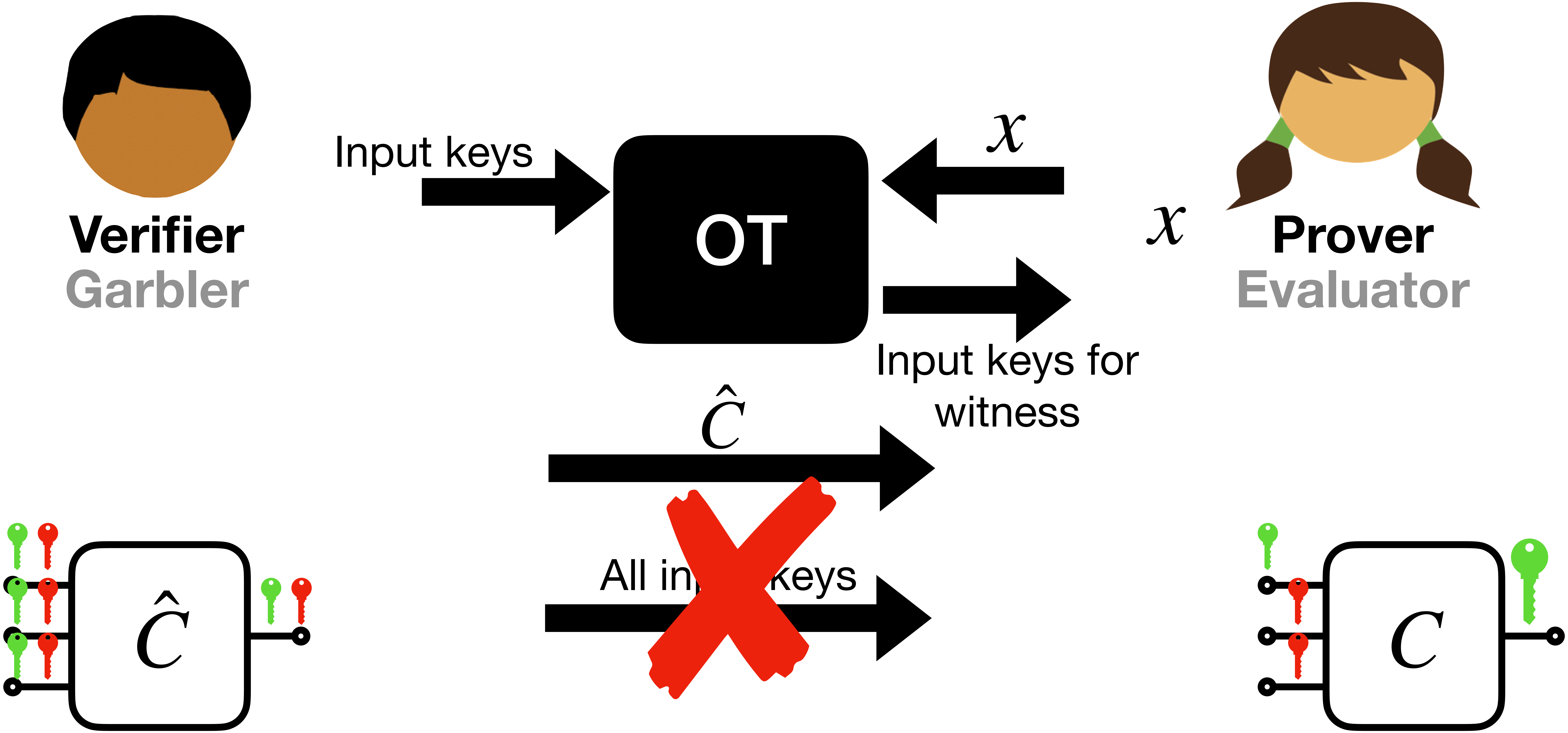
$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^0, K_c^0))$$
$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^1, K_c^0))$$
$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^0, K_c^0))$$
$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^1, K_c^1))$$

$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^0, K_c^1))$$
$$\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^1, K_c^1))$$
$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^0, K_c^1))$$
$$\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^1, K_c^1))$$

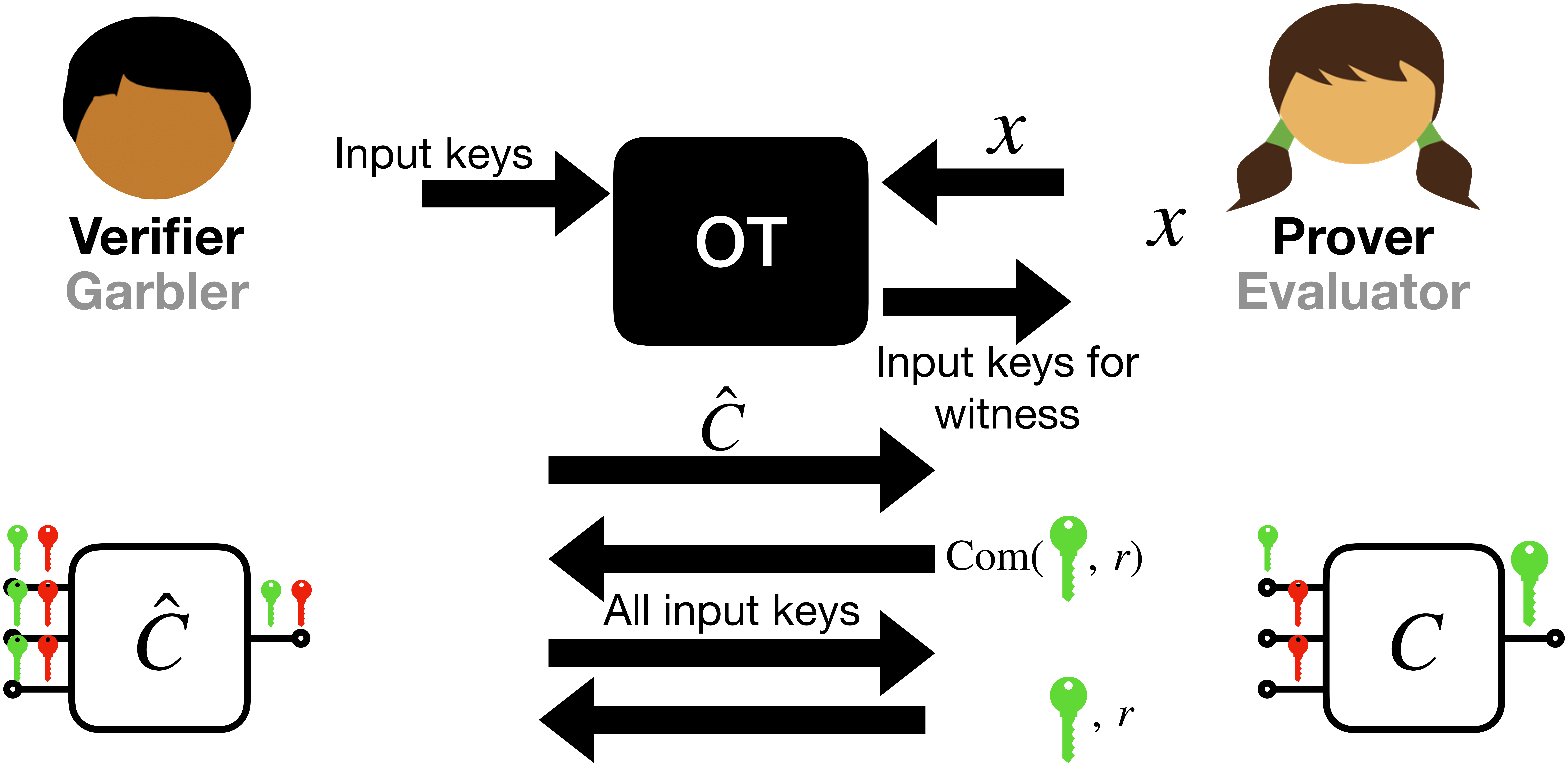
Zero Knowledge from Garbled Circuits



Zero Knowledge from Garbled Circuits



Zero Knowledge from Garbled Circuits



Zero Knowledge from Garbled Circuits

Completeness? (Honest prover can prove true things)

Zero Knowledge from Garbled Circuits

Completeness? (Honest prover can prove true things)

From **correctness** of GC

Zero Knowledge from Garbled Circuits

Completeness? (Honest prover can prove true things)

From **correctness** of GC

Soundness? (Malicious prover cannot prove false things)

From **authenticity** of GC

Zero Knowledge from Garbled Circuits

Completeness? (Honest prover can prove true things)

From **correctness** of GC

Soundness? (Malicious prover cannot prove false things)

From **authenticity** of GC

Zero Knowledge? (Malicious verifier learns nothing)

Zero Knowledge from Garbled Circuits

Completeness? (Honest prover can prove true things)

From **correctness** of GC

Soundness? (Malicious prover cannot prove false things)

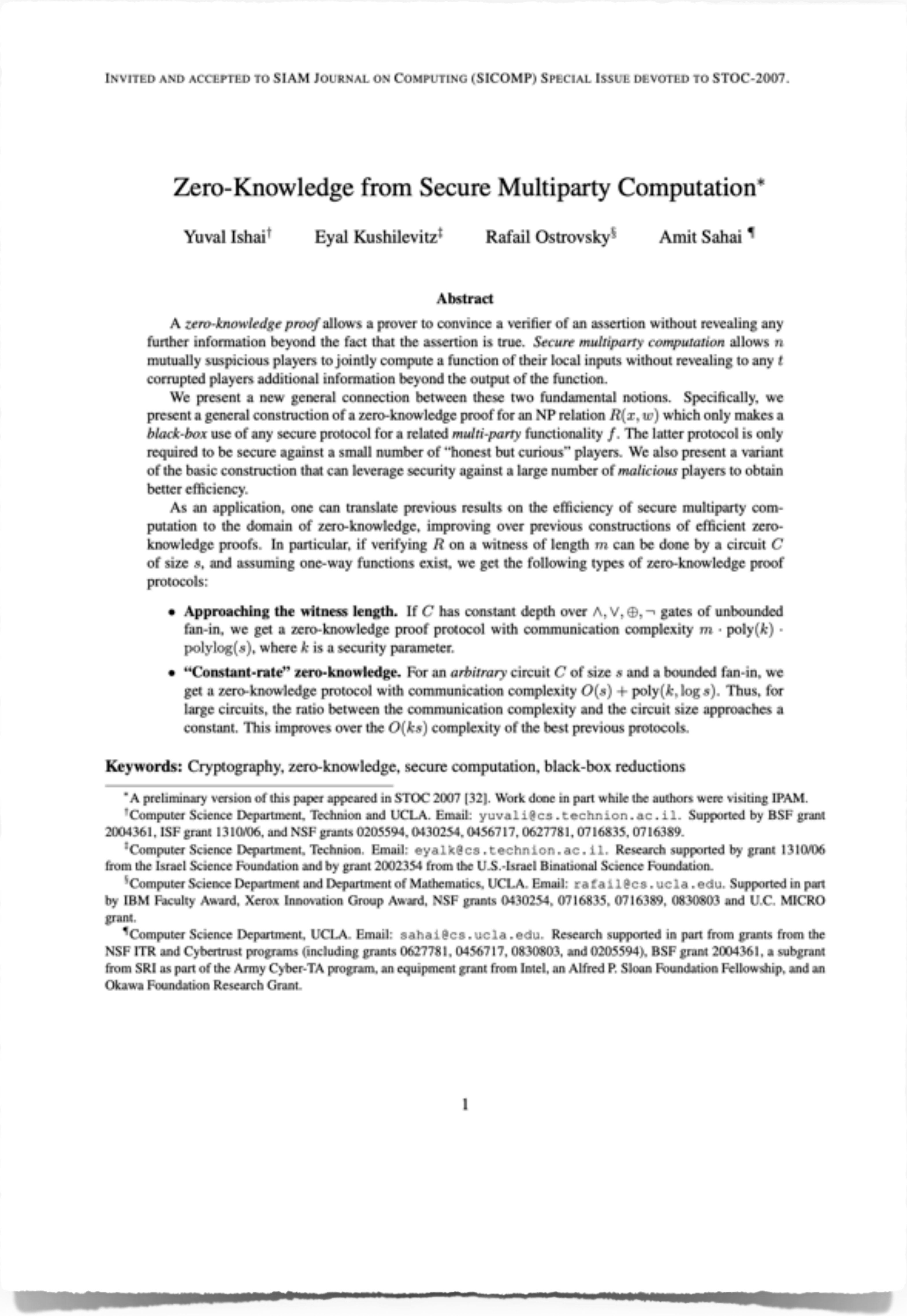
From **authenticity** of GC

Zero Knowledge? (Malicious verifier learns nothing)

Verifier is forced to produce a valid GC, and sees only the encoding of a one output key

Simulator can get all input keys from verifier, then send the one key to the verifier

MPC in the Head Paradigm



INVITED AND ACCEPTED TO SIAM JOURNAL ON COMPUTING (SICOMP) SPECIAL ISSUE DEVOTED TO STOC-2007.

Zero-Knowledge from Secure Multiparty Computation*

Yuval Ishai[†] Eyal Kushilevitz[‡] Rafail Ostrovsky[§] Amit Sahai[¶]

Abstract

A *zero-knowledge proof* allows a prover to convince a verifier of an assertion without revealing any further information beyond the fact that the assertion is true. *Secure multiparty computation* allows n mutually suspicious players to jointly compute a function of their local inputs without revealing to any t corrupted players additional information beyond the output of the function.

We present a new general connection between these two fundamental notions. Specifically, we present a general construction of a zero-knowledge proof for an NP relation $R(x, w)$ which only makes a *black-box* use of any secure protocol for a related *multi-party* functionality f. The latter protocol is only required to be secure against a small number of "honest but curious" players. We also present a variant of the basic construction that can leverage security against a large number of *malicious* players to obtain better efficiency.

As an application, one can translate previous results on the efficiency of secure multiparty computation to the domain of zero-knowledge, improving over previous constructions of efficient zero-knowledge proofs. In particular, if verifying R on a witness of length m can be done by a circuit C of size s, and assuming one-way functions exist, we get the following types of zero-knowledge proof protocols:

- **Approaching the witness length.** If C has constant depth over $\wedge, \vee, \oplus, \neg$ gates of unbounded fan-in, we get a zero-knowledge proof protocol with communication complexity $m \cdot \text{poly}(k) \cdot \text{polylog}(s)$, where k is a security parameter.
- **"Constant-rate" zero-knowledge.** For an *arbitrary* circuit C of size s and a bounded fan-in, we get a zero-knowledge protocol with communication complexity $O(s) + \text{poly}(k, \log s)$. Thus, for large circuits, the ratio between the communication complexity and the circuit size approaches a constant. This improves over the $O(ks)$ complexity of the best previous protocols.

Keywords: Cryptography, zero-knowledge, secure computation, black-box reductions

*A preliminary version of this paper appeared in STOC 2007 [32]. Work done in part while the authors were visiting IPAM.

†Computer Science Department, Technion and UCLA. Email: yuvali@cs.technion.ac.il. Supported by BSF grant 2004361, ISF grant 1310/06, and NSF grants 0205594, 0430254, 0456717, 0627781, 0716835, 0716389.

‡Computer Science Department, Technion. Email: eyalk@cs.technion.ac.il. Research supported by grant 1310/06 from the Israel Science Foundation and by grant 2002354 from the U.S.-Israel Binational Science Foundation.

§Computer Science Department and Department of Mathematics, UCLA. Email: rafail@cs.ucla.edu. Supported in part by IBM Faculty Award, Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389, 0830803 and U.C. MICRO grant.

¶Computer Science Department, UCLA. Email: sahai@cs.ucla.edu. Research supported in part from grants from the NSF ITR and Cybertrust programs (including grants 0627781, 0456717, 0830803, and 0205594), BSF grant 2004361, a subgrant from SRI as part of the Army Cyber-TA program, an equipment grant from Intel, an Alfred P. Sloan Foundation Fellowship, and an Okawa Foundation Research Grant.

1

MPC in the Head Paradigm

INVITED AND ACCEPTED TO SIAM JOURNAL ON COMPUTING (SICOMP) SPECIAL ISSUE DEVOTED TO STOC-2007.

Zero-Knowledge from Secure Multiparty Computation*

Yuval Ishai[†] Eyal Kushilevitz[‡] Rafail Ostrovsky[§] Amit Sahai[¶]

Abstract

A *zero-knowledge proof* allows a prover to convince a verifier of an assertion without revealing any further information beyond the fact that the assertion is true. *Secure multiparty computation* allows n mutually suspicious players to jointly compute a function of their local inputs without revealing to any t corrupted players additional information beyond the output of the function.

We present a new general connection between these two fundamental notions. Specifically, we present a general construction of a zero-knowledge proof for an NP relation $R(x, w)$ which only makes a *black-box* use of any secure protocol for a related *multi-party* functionality f. The latter protocol is only required to be secure against a small number of "honest but curious" players. We also present a variant of the basic construction that can leverage security against a large number of *malicious* players to obtain better efficiency.

As an application, one can translate previous results on the efficiency of secure multiparty computation to the domain of zero-knowledge, improving over previous constructions of efficient zero-knowledge proofs. In particular, if verifying R on a witness of length m can be done by a circuit C of size s, and assuming one-way functions exist, we get the following types of zero-knowledge proof protocols:

- **Approaching the witness length.** If C has constant depth over $\wedge, \vee, \oplus, \neg$ gates of unbounded fan-in, we get a zero-knowledge proof protocol with communication complexity $m \cdot \mathrm{poly}(k) \cdot \mathrm{polylog}(s)$, where k is a security parameter.
- **"Constant-rate" zero-knowledge.** For an *arbitrary* circuit C of size s and a bounded fan-in, we get a zero-knowledge protocol with communication complexity $O(s) + \mathrm{poly}(k, \log s)$. Thus, for large circuits, the ratio between the communication complexity and the circuit size approaches a constant. This improves over the $O(ks)$ complexity of the best previous protocols.

Keywords: Cryptography, zero-knowledge, secure computation, black-box reductions

*A preliminary version of this paper appeared in STOC 2007 [32]. Work done in part while the authors were visiting IPAM.
†Computer Science Department, Technion and UCLA. Email: yuvali@cs.technion.ac.il. Supported by BSF grant 2004361, ISF grant 1310/06, and NSF grants 0205594, 0430254, 0456717, 0627781, 0716835, 0716389.
‡Computer Science Department, Technion. Email: eyalk@cs.technion.ac.il. Research supported by grant 1310/06 from the Israel Science Foundation and by grant 2002354 from the U.S.-Israel Binational Science Foundation.
§Computer Science Department and Department of Mathematics, UCLA. Email: rafail@cs.ucla.edu. Supported in part by IBM Faculty Award, Xerox Innovation Group Award, NSF grants 0430254, 0716835, 0716389, 0830803 and U.C. MICRO grant.
¶Computer Science Department, UCLA. Email: sahai@cs.ucla.edu. Research supported in part from grants from the NSF ITR and Cybertrust programs (including grants 0627781, 0456717, 0830803, and 0205594), BSF grant 2004361, a subgrant from SRI as part of the Army Cyber-TA program, an equipment grant from Intel, an Alfred P. Sloan Foundation Fellowship, and an Okawa Foundation Research Grant.

1

ZKBoo: Faster Zero-Knowledge for Boolean Circuits

Irene Giacomelli Jesper Madsen Claudio Orlandi
Computer Science Department, Aarhus University,
{giacomelli,jesbuus,orlandi}@cs.au.dk

Post-Quantum Zero-Knowledge and Signatures from Symmetric-Key Primitives*

Melissa Chase, Microsoft Research, melissac@microsoft.com
David Derler, Graz University of Technology, david.derler@tugraz.at
Steven Goldfeder, Princeton University, stevenag@cs.princeton.edu
Claudio Orlandi, Aarhus University, orlandi@cs.au.dk
Sebastian Ramacher, Graz University of Technology, sebastian.ramacher@tugraz.at
Christian Rechberger, Graz University of Technology & Denmark Technical University, christian.rechberger@tugraz.at
Daniel Slamanig, AIT Austrian Institute of Technology, daniel.slamanig@ait.ac.at
Greg Zaverucha, Microsoft Research, gregz@microsoft.com

Improved Non-Interactive Zero Knowledge with Applications to Post-Quantum Signatures

Jonathan Katz, University of Maryland, jkatz@cs.umd.edu
Vladimir Kolesnikov, Georgia Tech, kolesnikov@gatech.edu
Xiao Wang, University of Maryland, wangxiao@cs.umd.edu

Session J1: Outsourcing CCS'17, October 30-November 3, 2017, Dallas, TX, USA

Ligero: Lightweight Sublinear Arguments Without a Trusted Setup

Scott Ames, University of Rochester, sames@cs.rochester.edu
Carmit Hazay, Bar-Ilan University, carmit.hazay@cs.biu.ac.il
Yuval Ishai, Technion and UCLA, yuvali@cs.technion.ac.il
Muthuramakrishnan Venkitasubramaniam, University of Rochester, muthuv@cs.rochester.edu

ABSTRACT

We design and implement a simple zero-knowledge argument protocol for NP whose communication complexity is proportional to the square-root of the verification circuit size. The protocol can be based on any collision-resistant hash function. Alternatively, it can be made non-interactive in the random oracle model, yielding concretely efficient zk-SNARKs that do not require a trusted setup or public-key cryptography.

Our protocol is attractive not only for very large verification circuits but also for moderately large circuits that arise in appli-

a concretely efficient argument protocol for NP whose communication complexity is proportional to the square root of the size of a circuit verifying the NP witness. Our argument system is in fact a zero-knowledge argument of knowledge, and it only requires the verifier to send public coins to the prover. The latter feature implies that it can be made non-interactive via the Fiat-Shamir transform [19], yielding an efficient implementation of zero-knowledge succinct non-interactive arguments of knowledge (zk-SNARKs [11]) without a trusted setup.

To put our work in the proper context, we give some relevant

Non-interactive Zero Knowledge

Can give succinct proofs

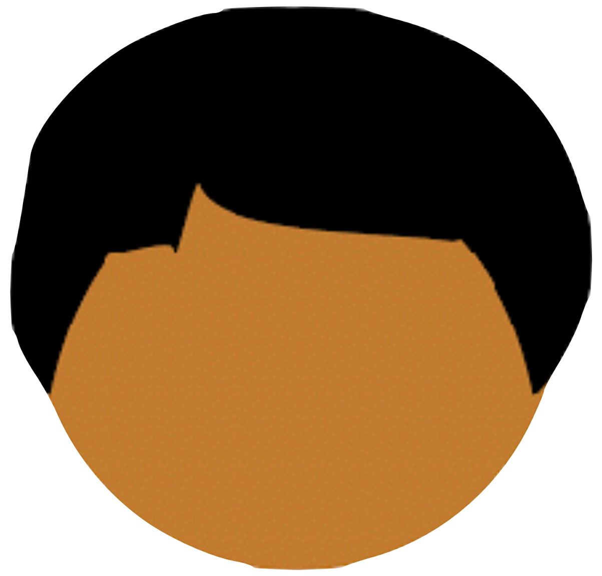
Plausibly post-quantum secure

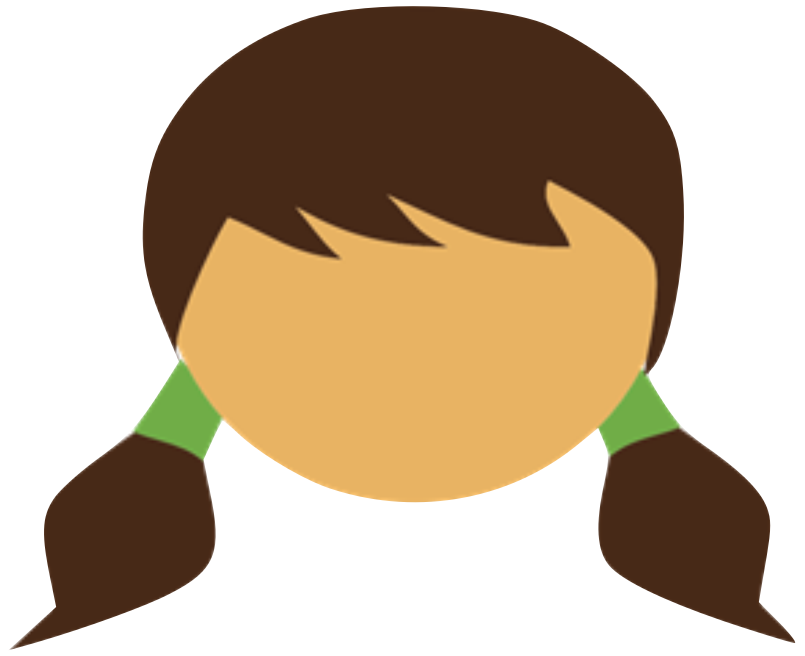
“Meta”-quality of some kinds of cryptography

ZK from MPC in the Head

Statements: Boolean circuit satisfiability

Witness: circuit input causing circuit to output 1



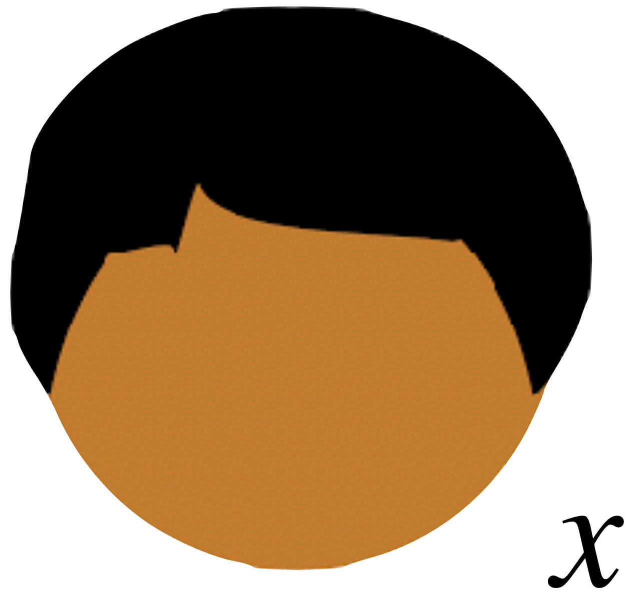


ZK from MPC in the Head

Statements: Boolean circuit satisfiability

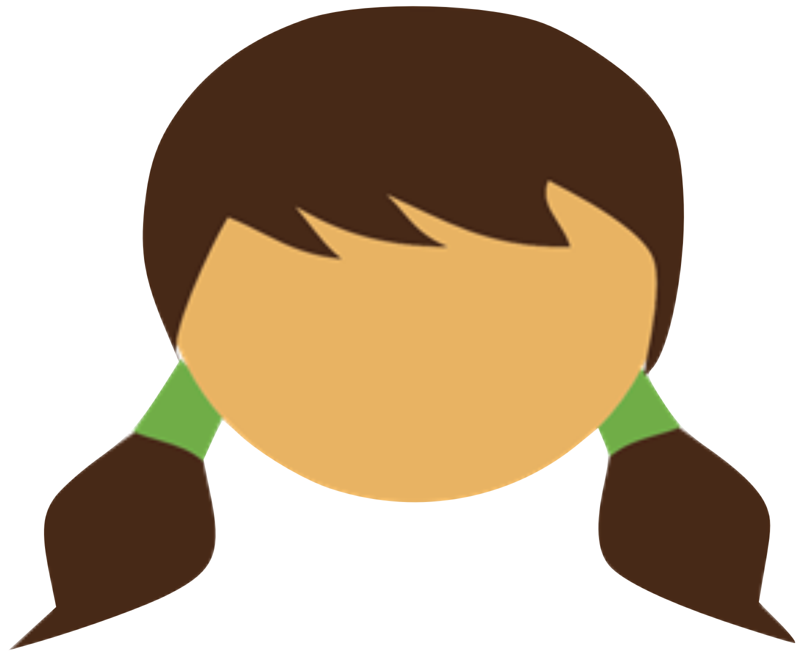
Witness: circuit input causing circuit to output 1

C



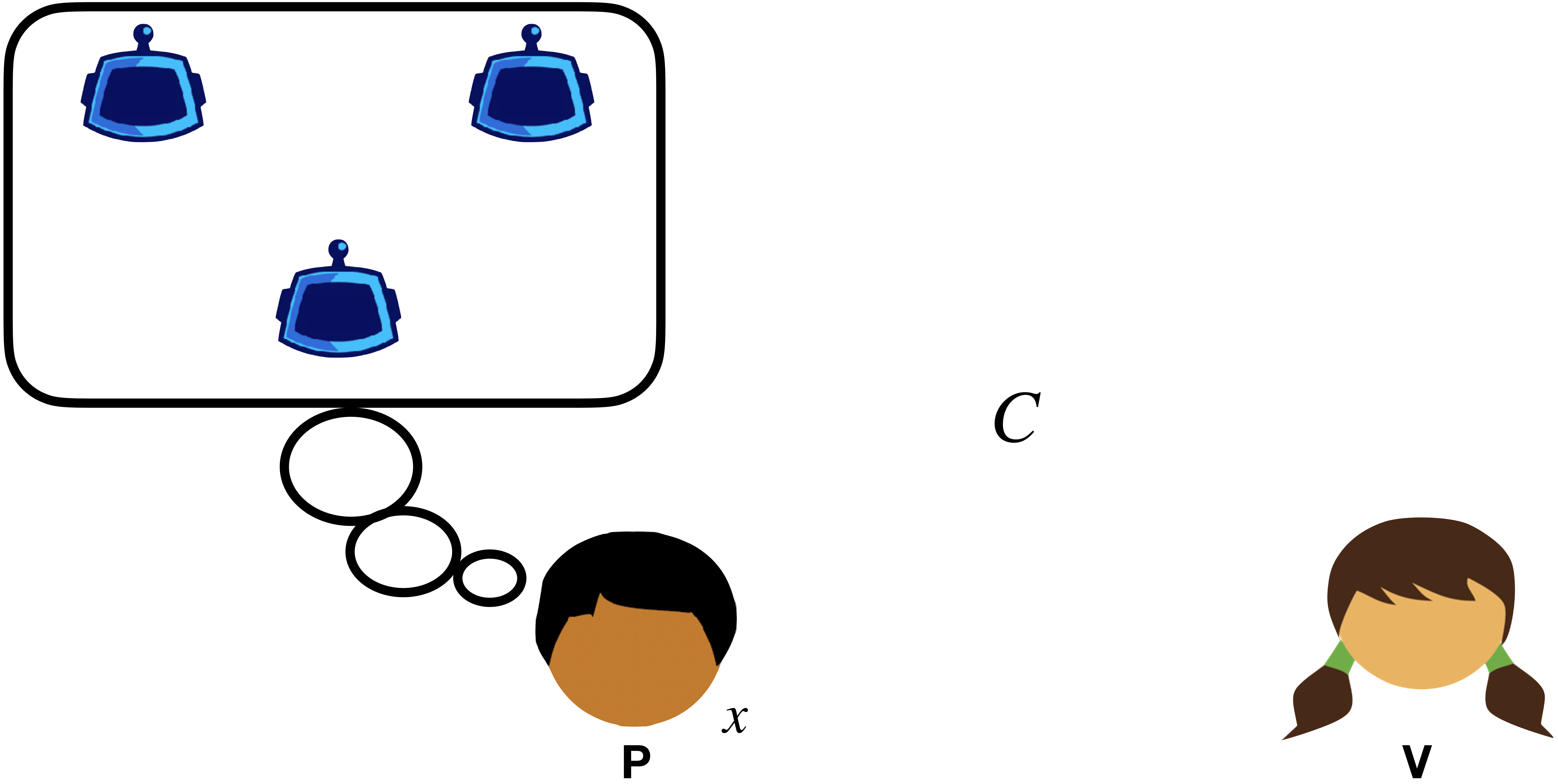
x

P

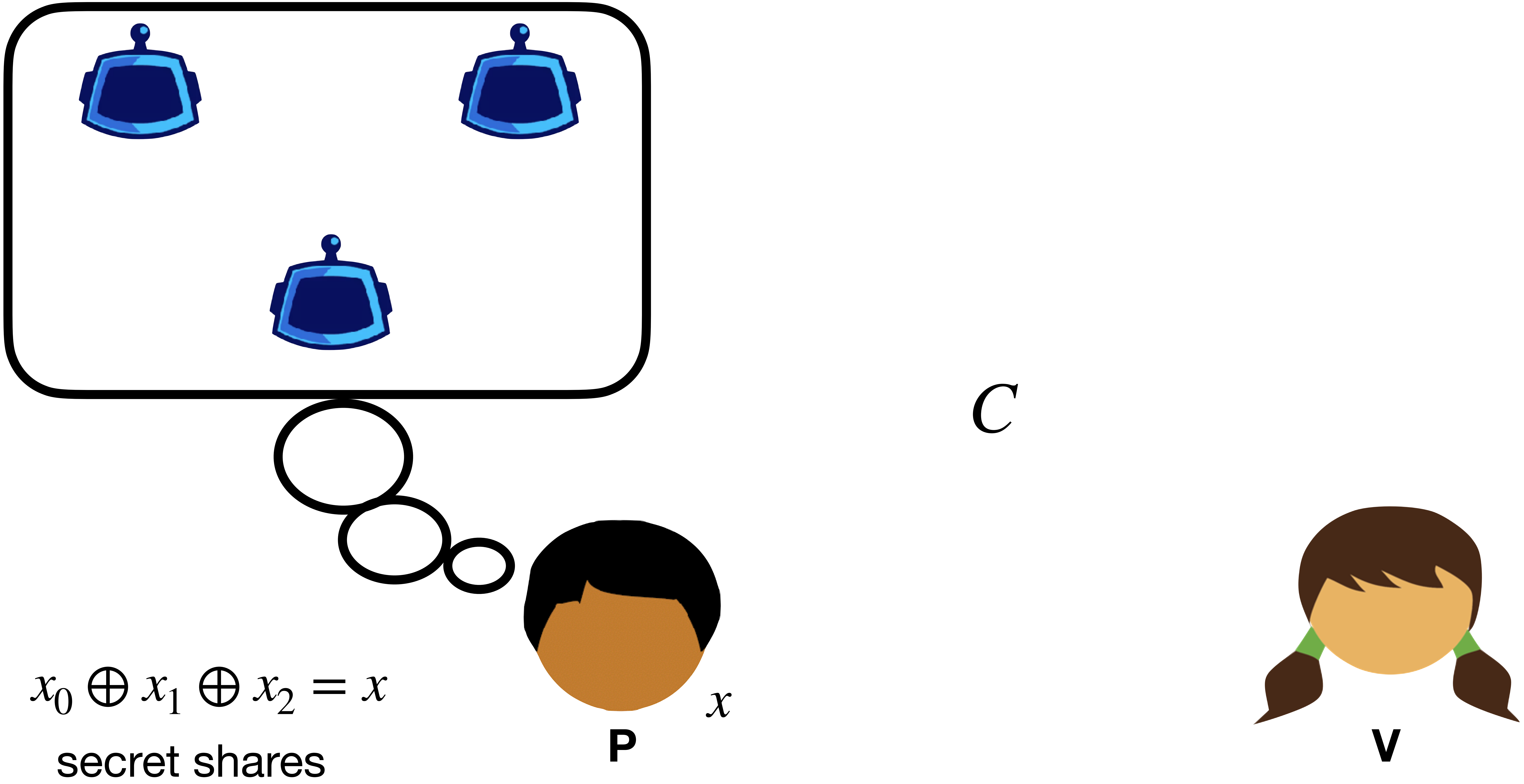


V

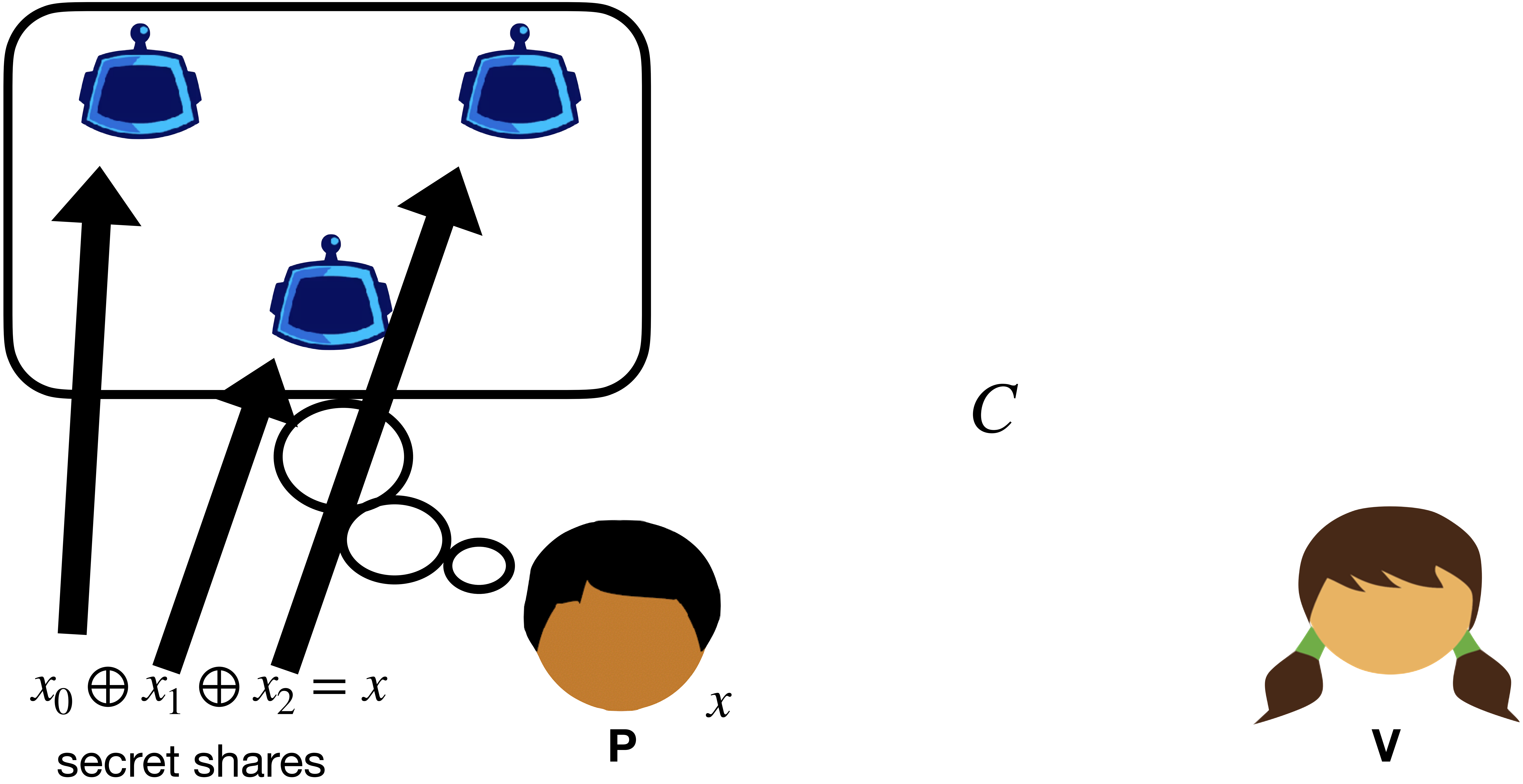
ZK from MPC in the Head



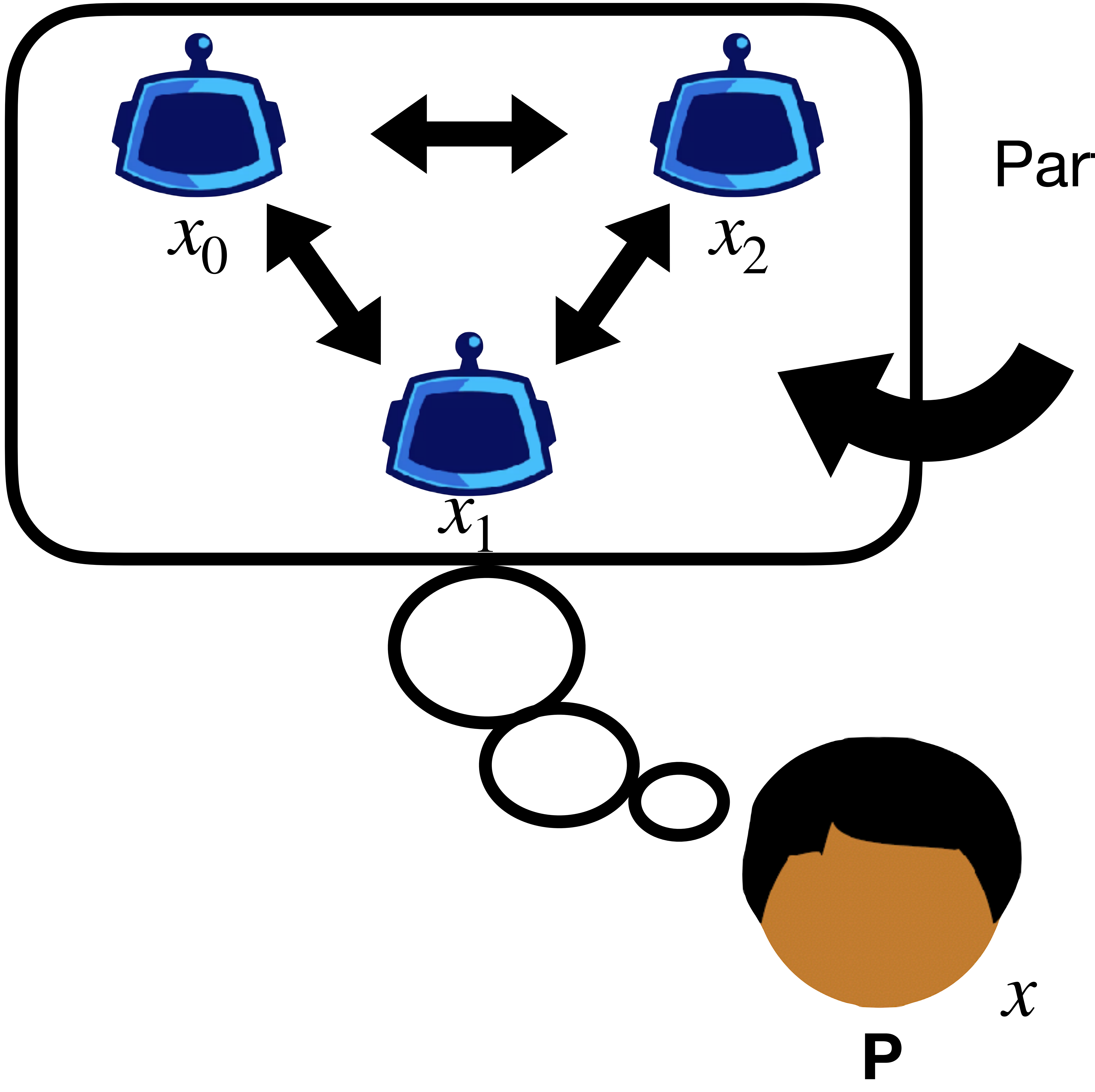
ZK from MPC in the Head



ZK from MPC in the Head

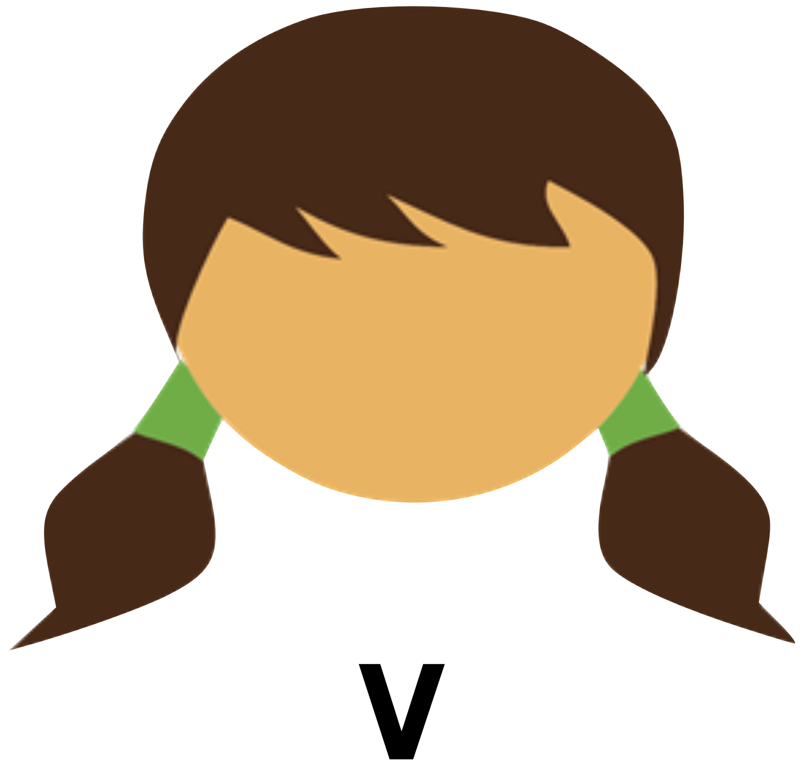


ZK from MPC in the Head

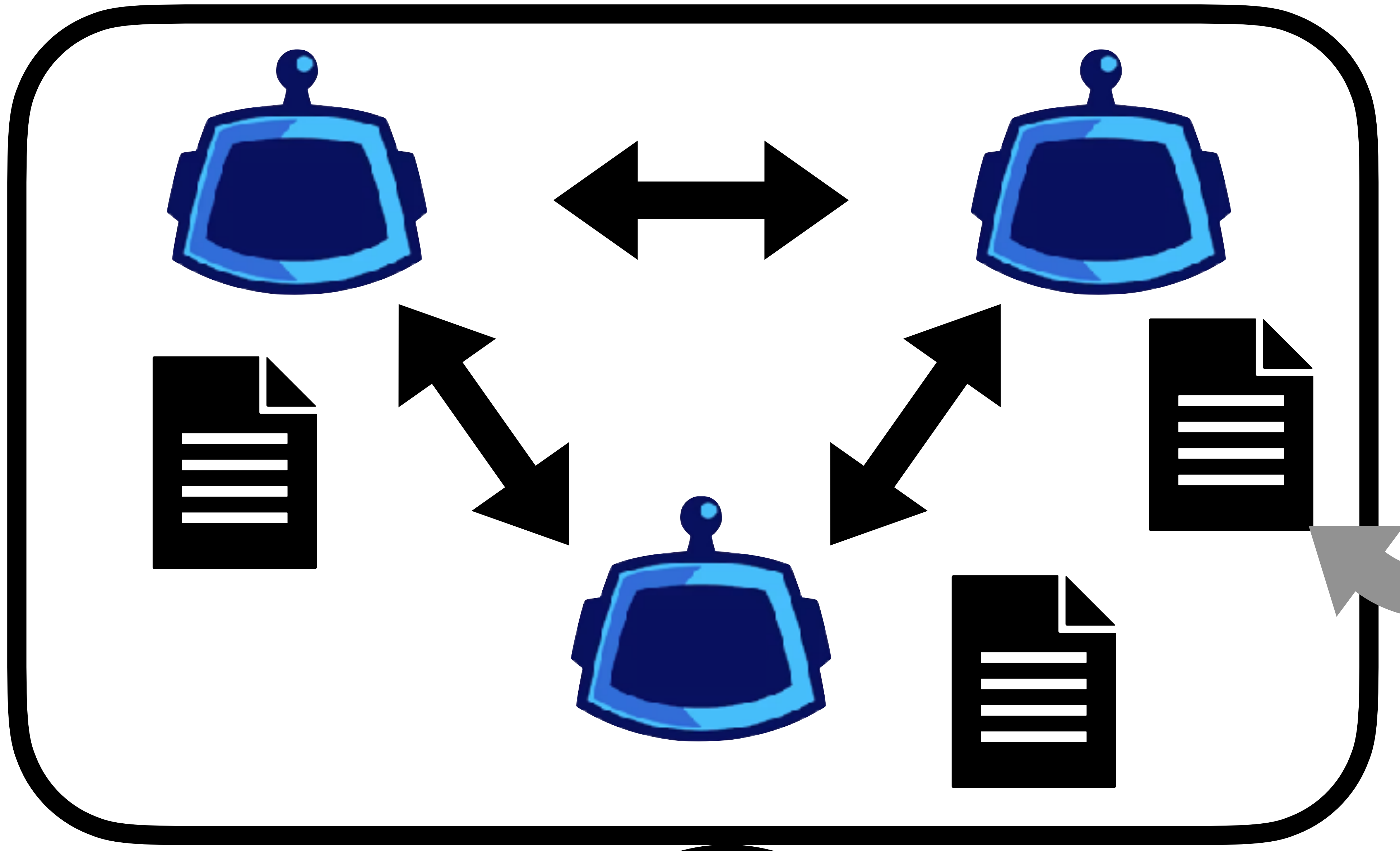


Parties compute $C(x_0 \oplus x_1 \oplus x_2)$
using the GMW protocol

C

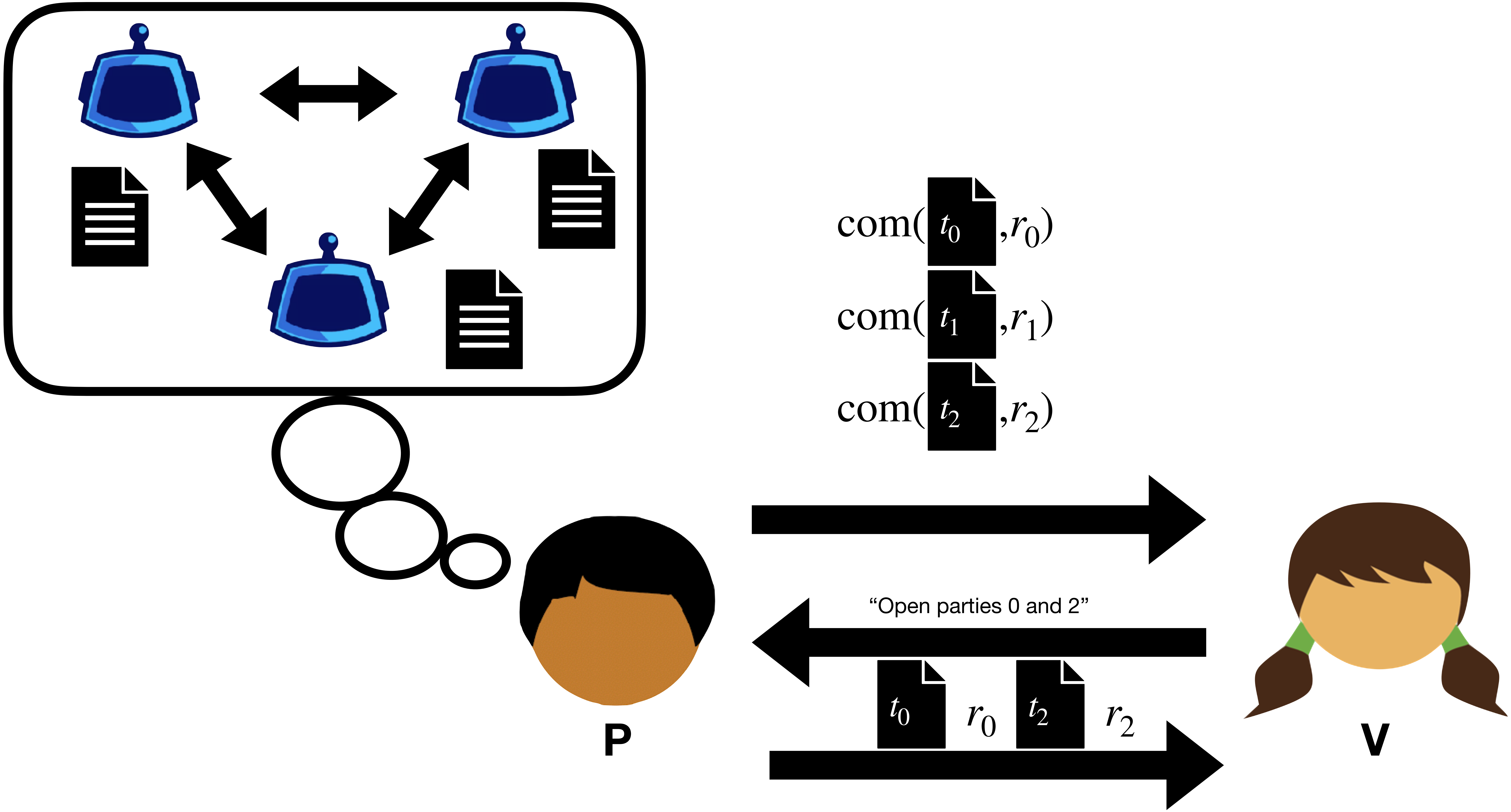


ZK from MPC in the Head



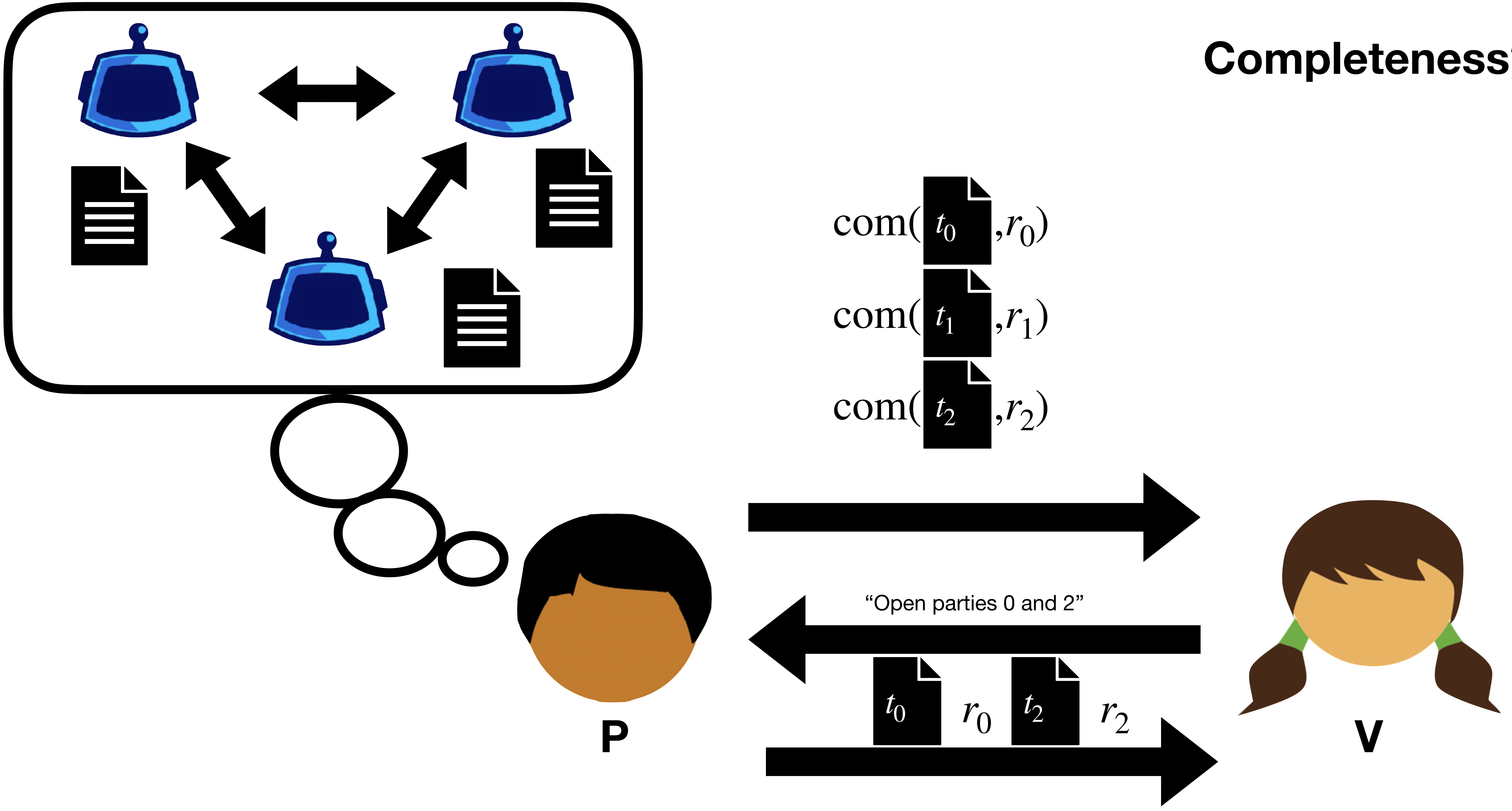
Protocol transcript from the perspective of virtual party P_i

ZK from MPC in the Head

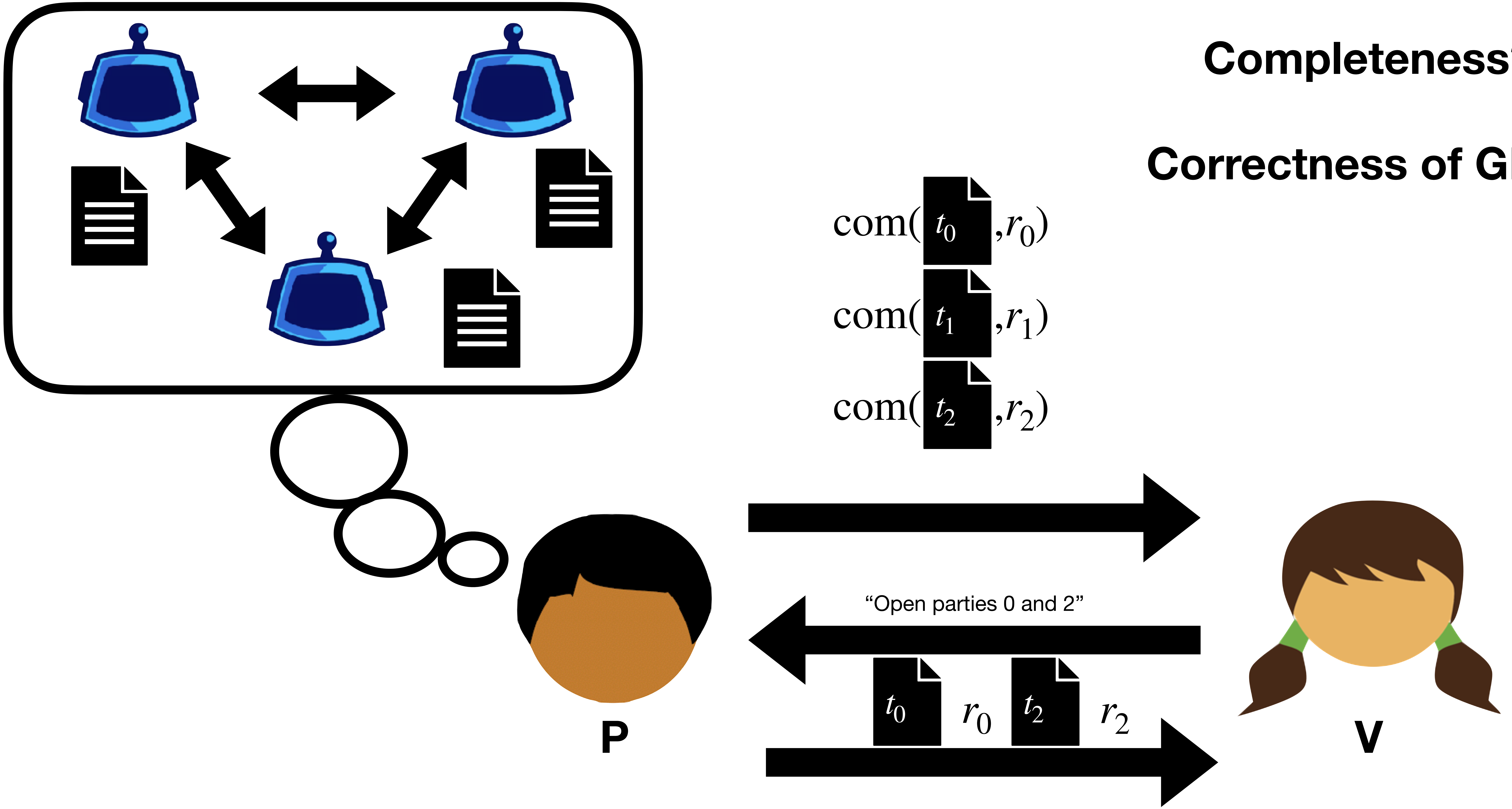


ZK from MPC in the Head

Completeness?



ZK from MPC in the Head

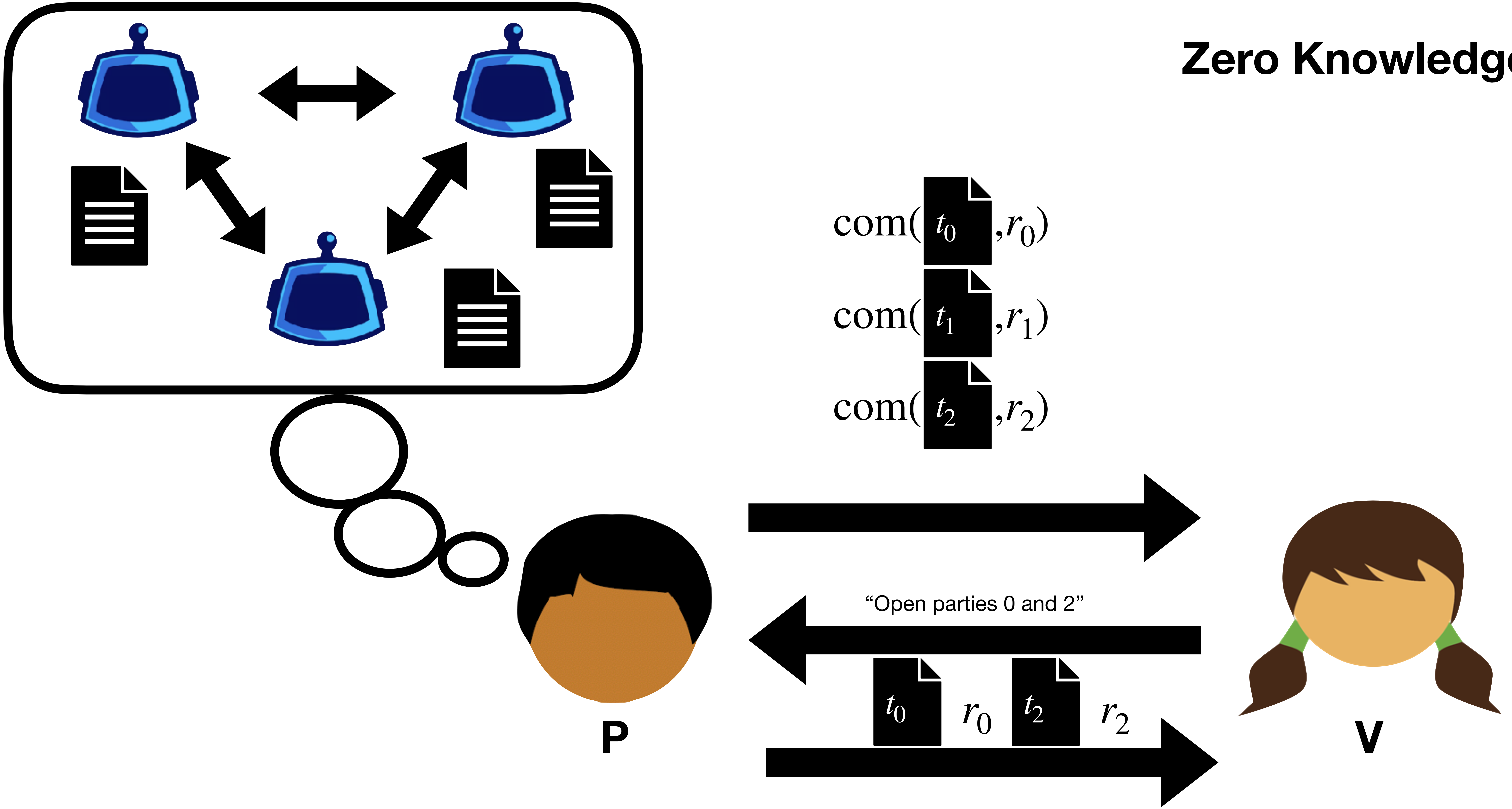


Completeness?

Correctness of GMW

ZK from MPC in the Head

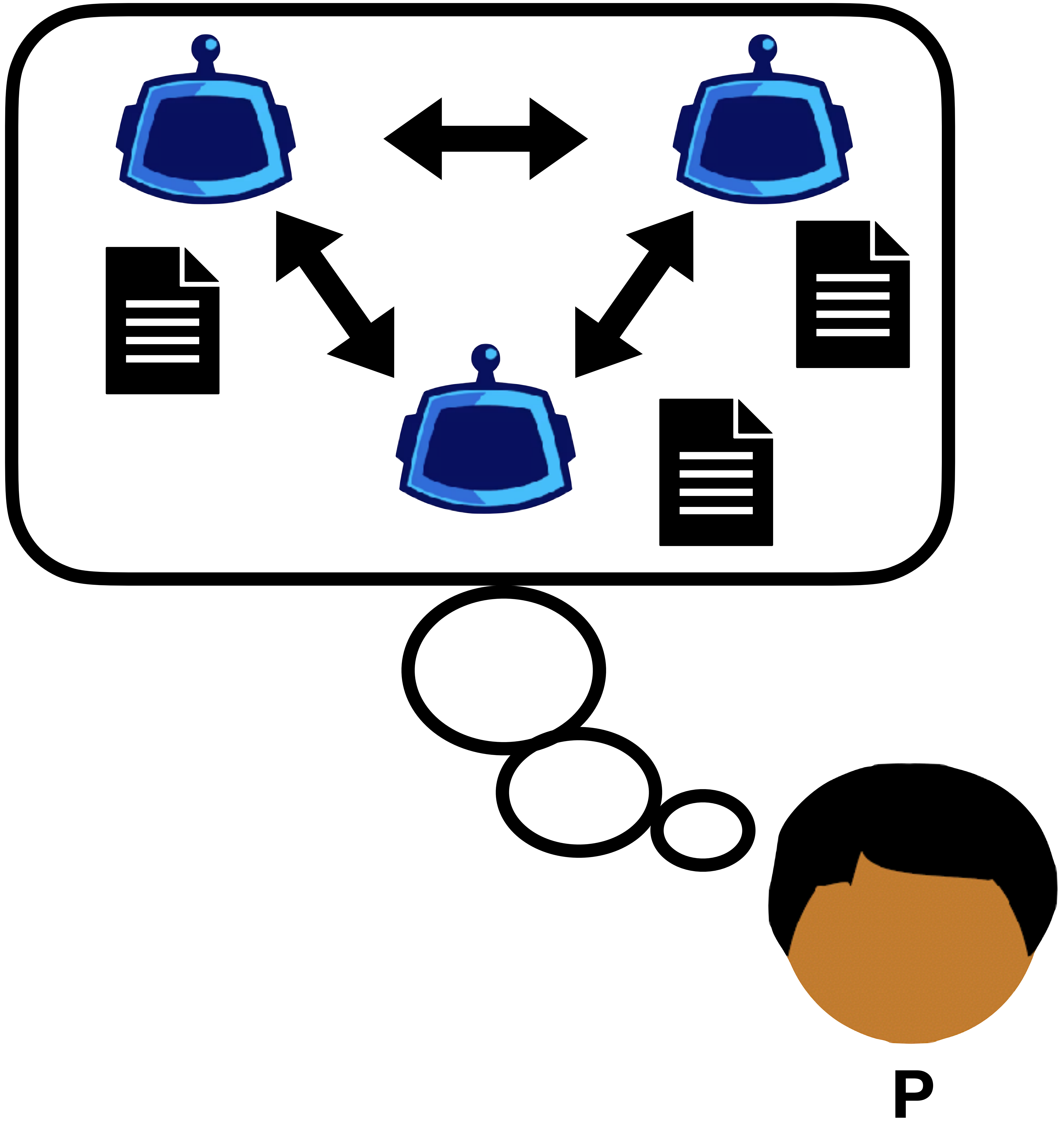
Zero Knowledge?



ZK from MPC in the Head

Zero Knowledge?

Security of GMW



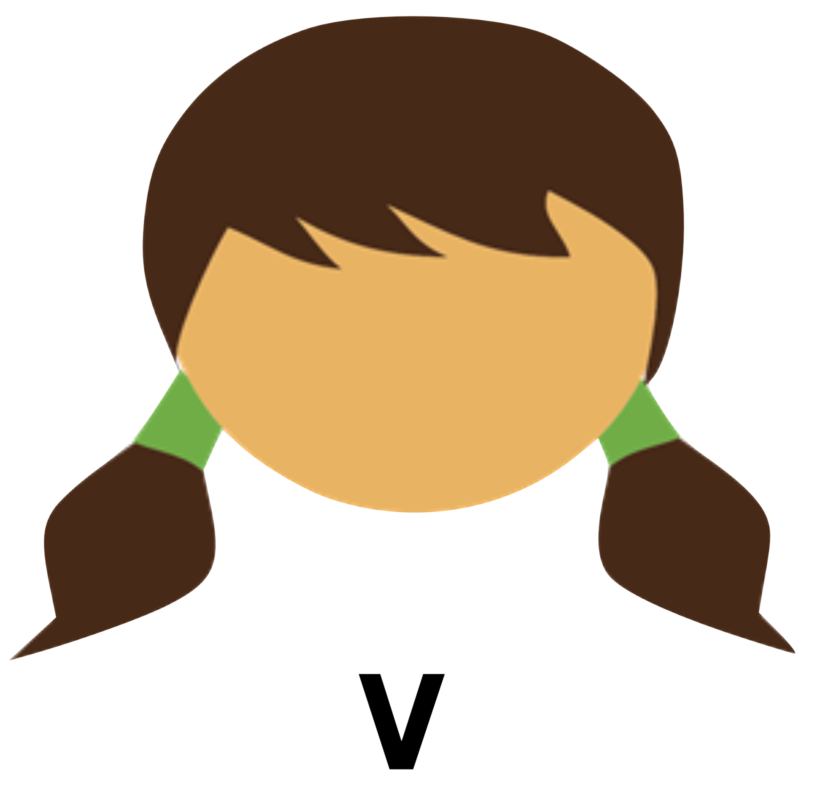
$\mathrm{com}(t_0, r_0)$

$\mathrm{com}(t_1, r_1)$

$\mathrm{com}(t_2, r_2)$

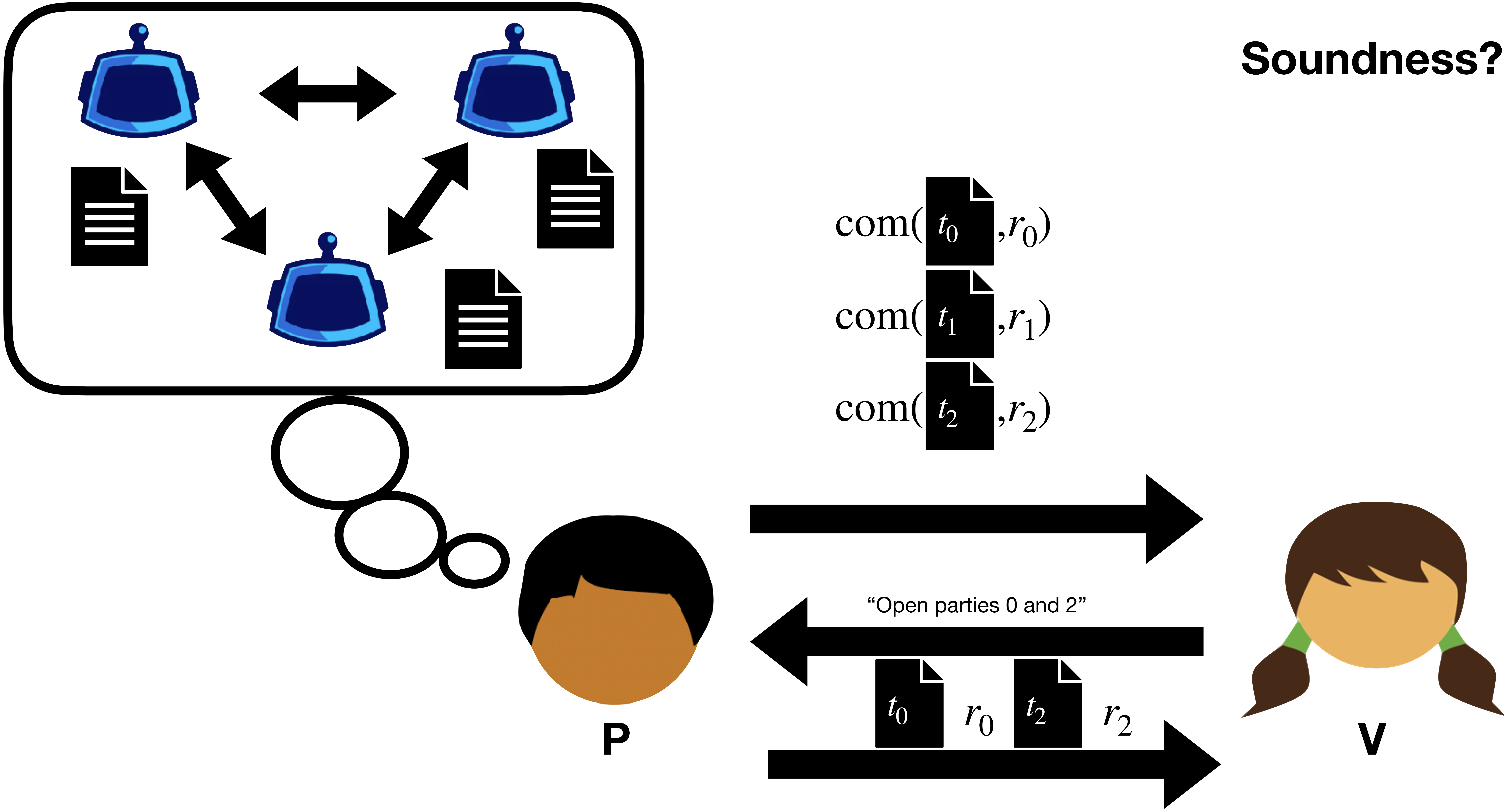
"Open parties 0 and 2"

$t_0 \quad r_0 \quad t_2 \quad r_2$

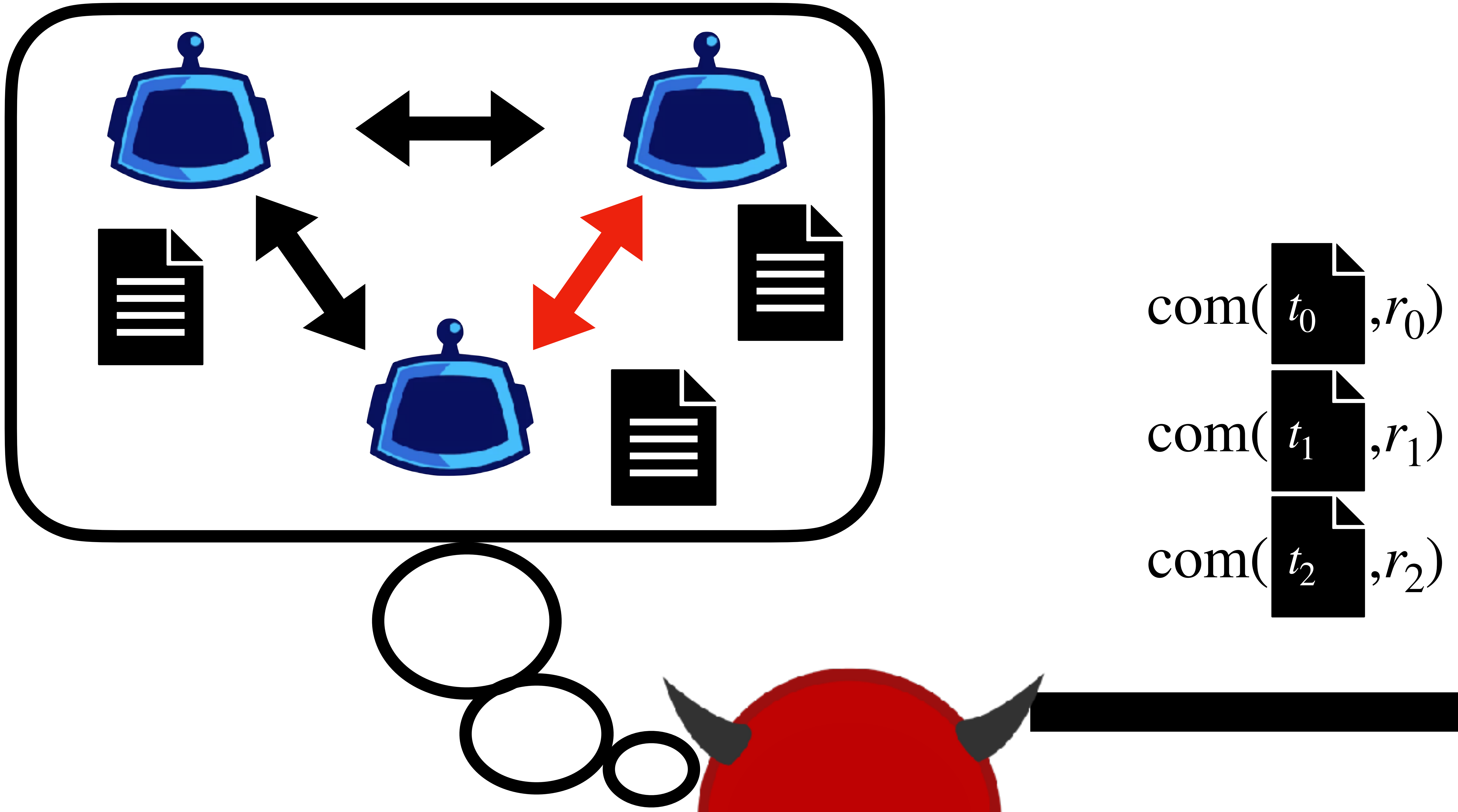


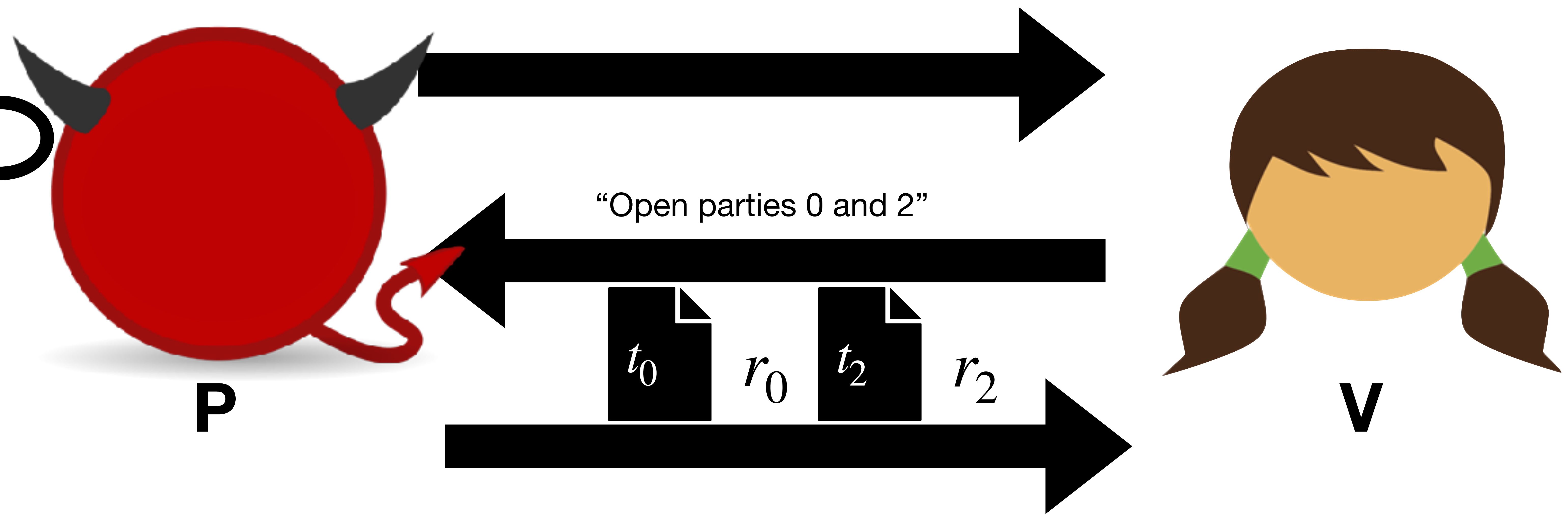
ZK from MPC in the Head

Soundness?



ZK from MPC in the Head

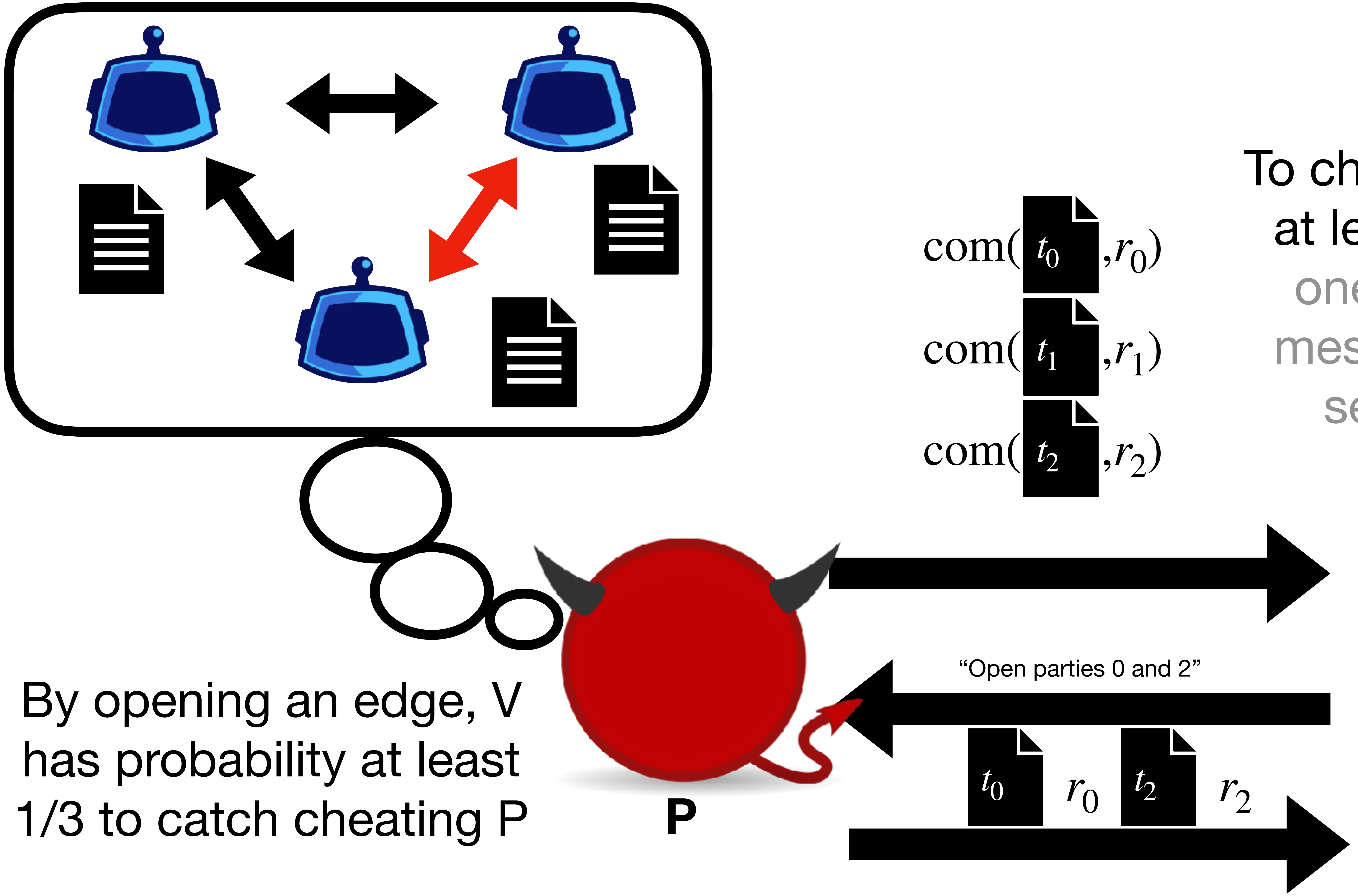


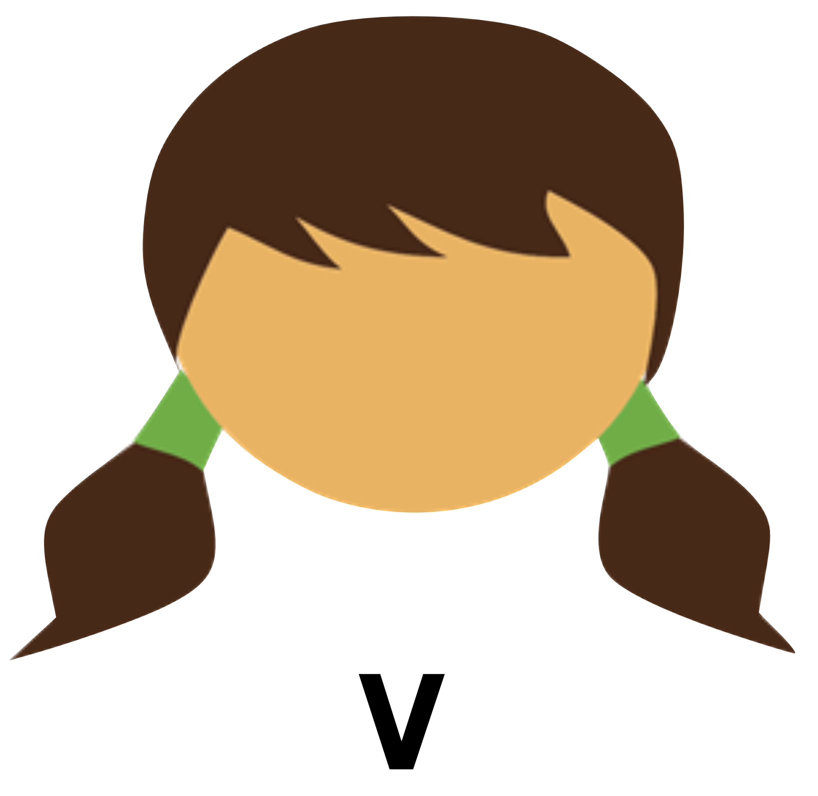


Soundness?

To cheat, P must corrupt at least one edge (i.e., one party receives a message that was not sent by the other)

ZK from MPC in the Head



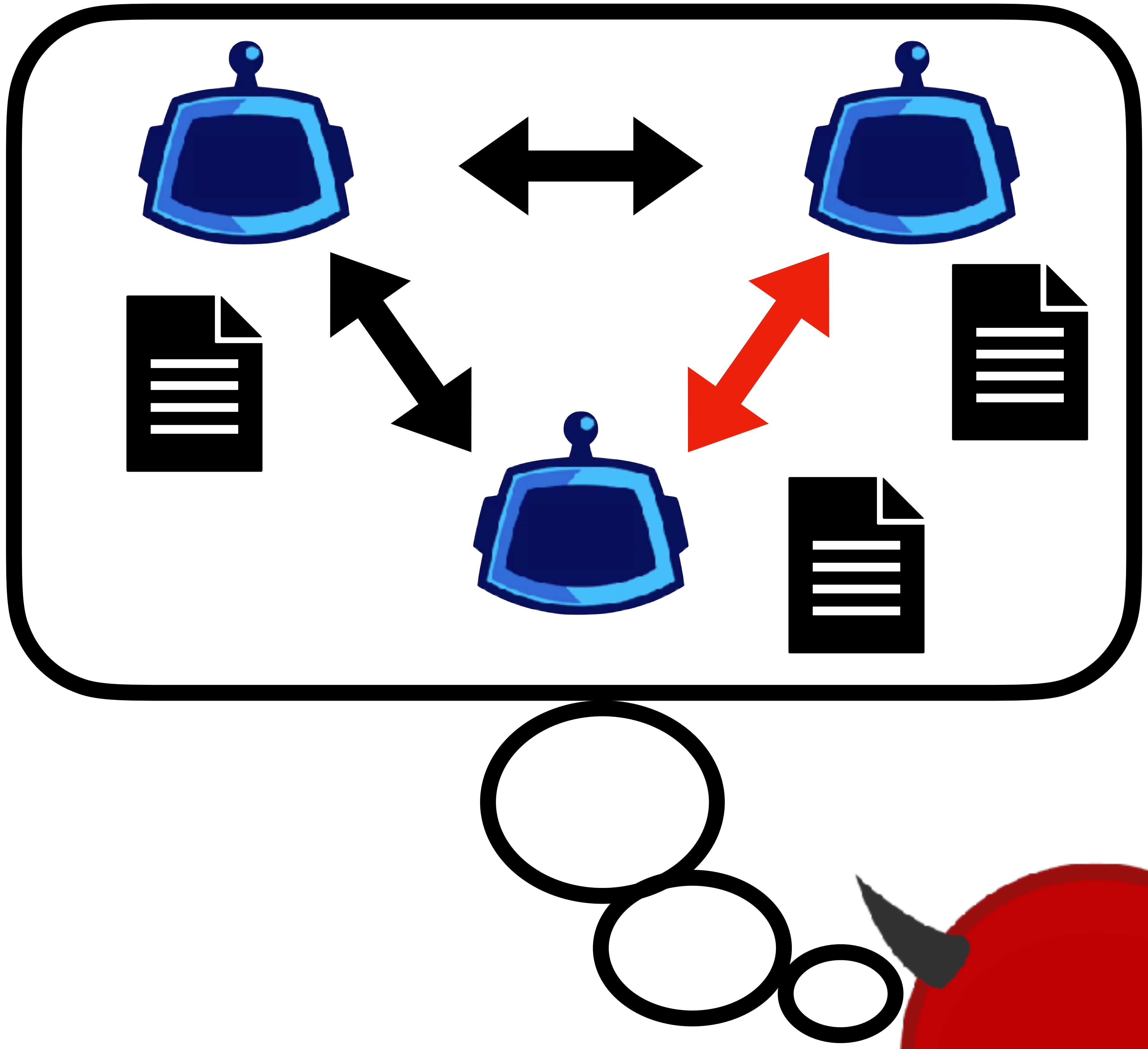


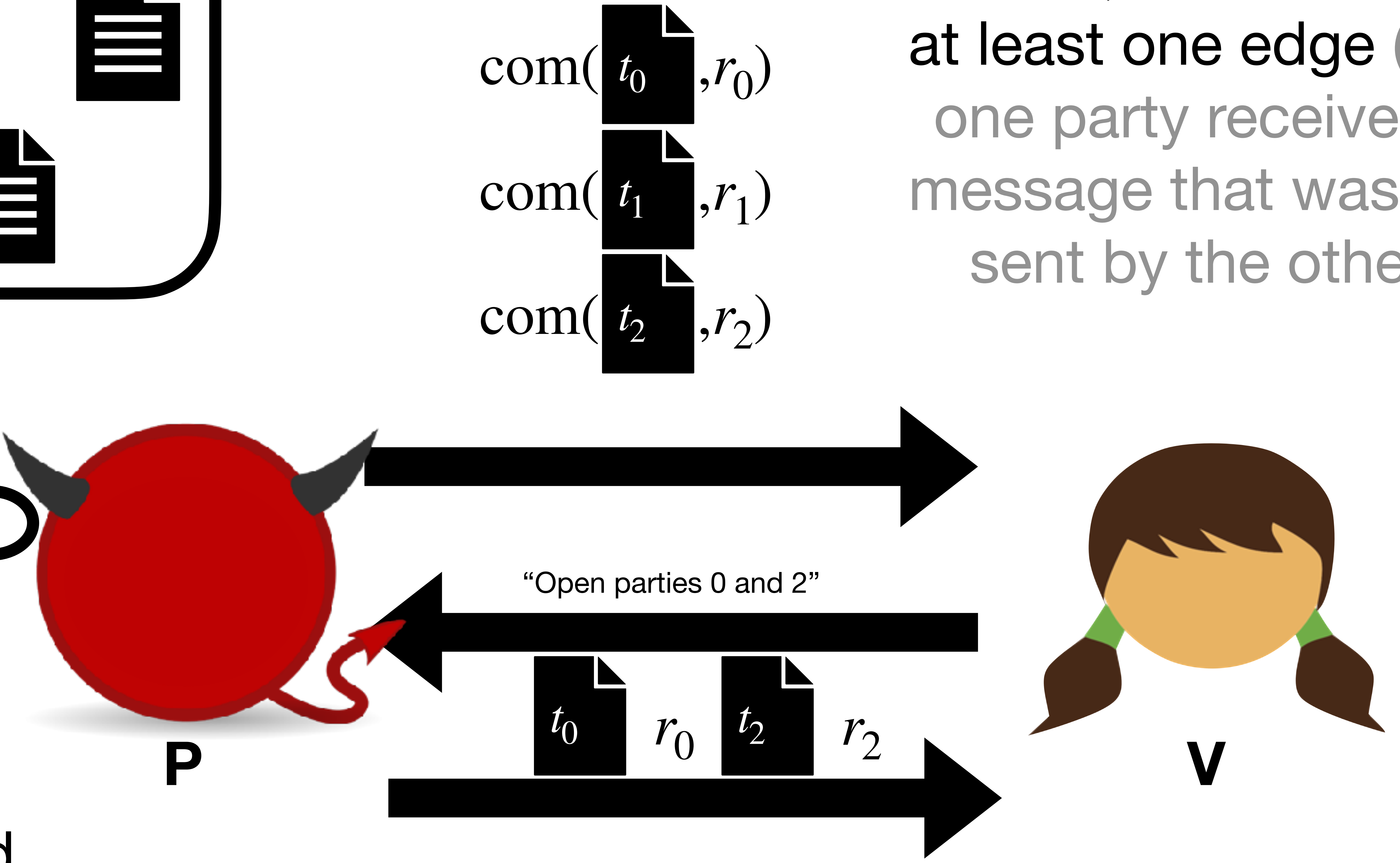
Soundness?

To cheat, P must corrupt at least one edge (i.e., one party receives a message that was not sent by the other)

By opening an edge, V has probability at least 1/3 to catch cheating P

ZK from MPC in the Head





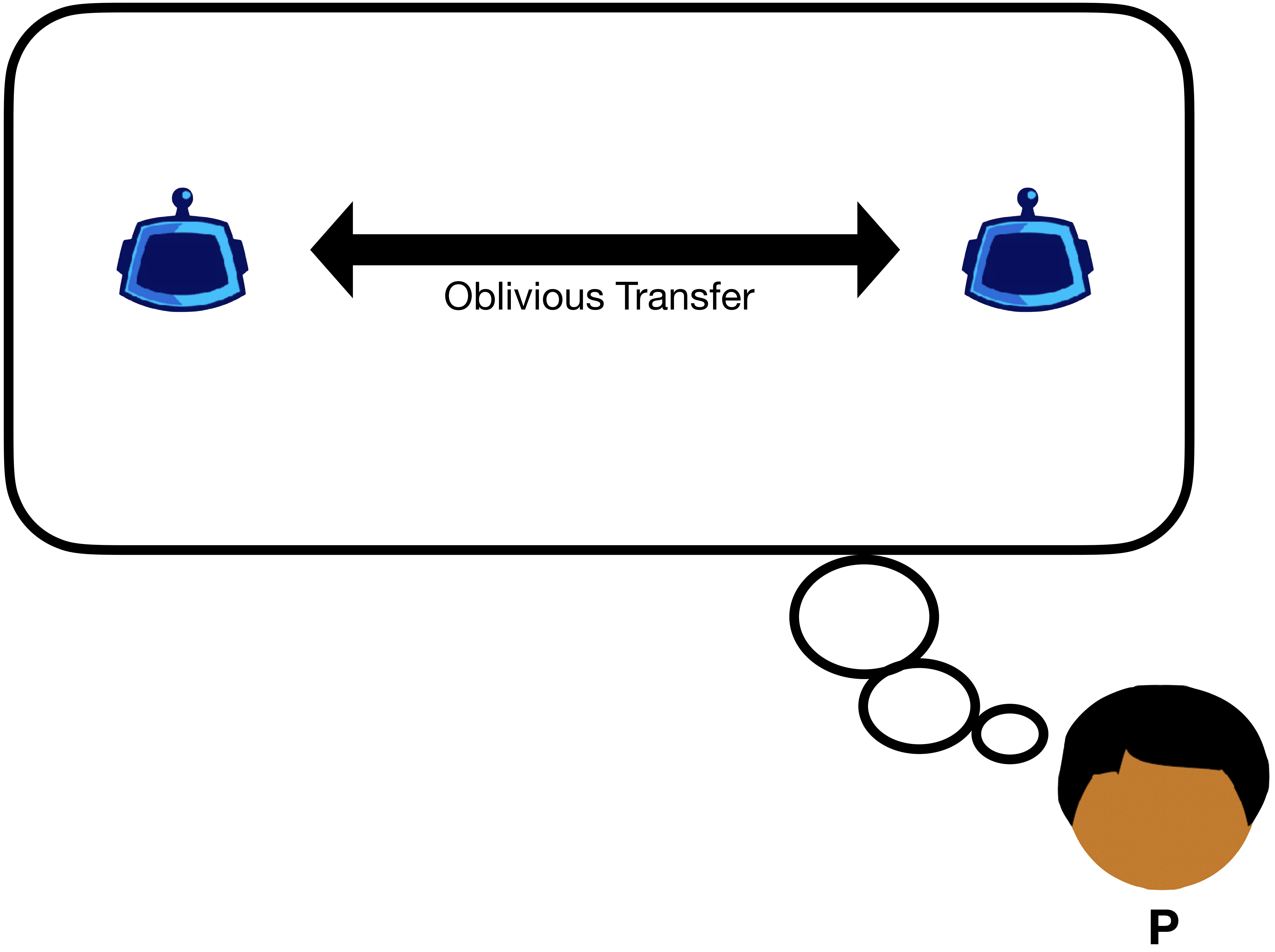
Soundness?

To cheat, P must corrupt at least one edge (i.e., one party receives a message that was not sent by the other)

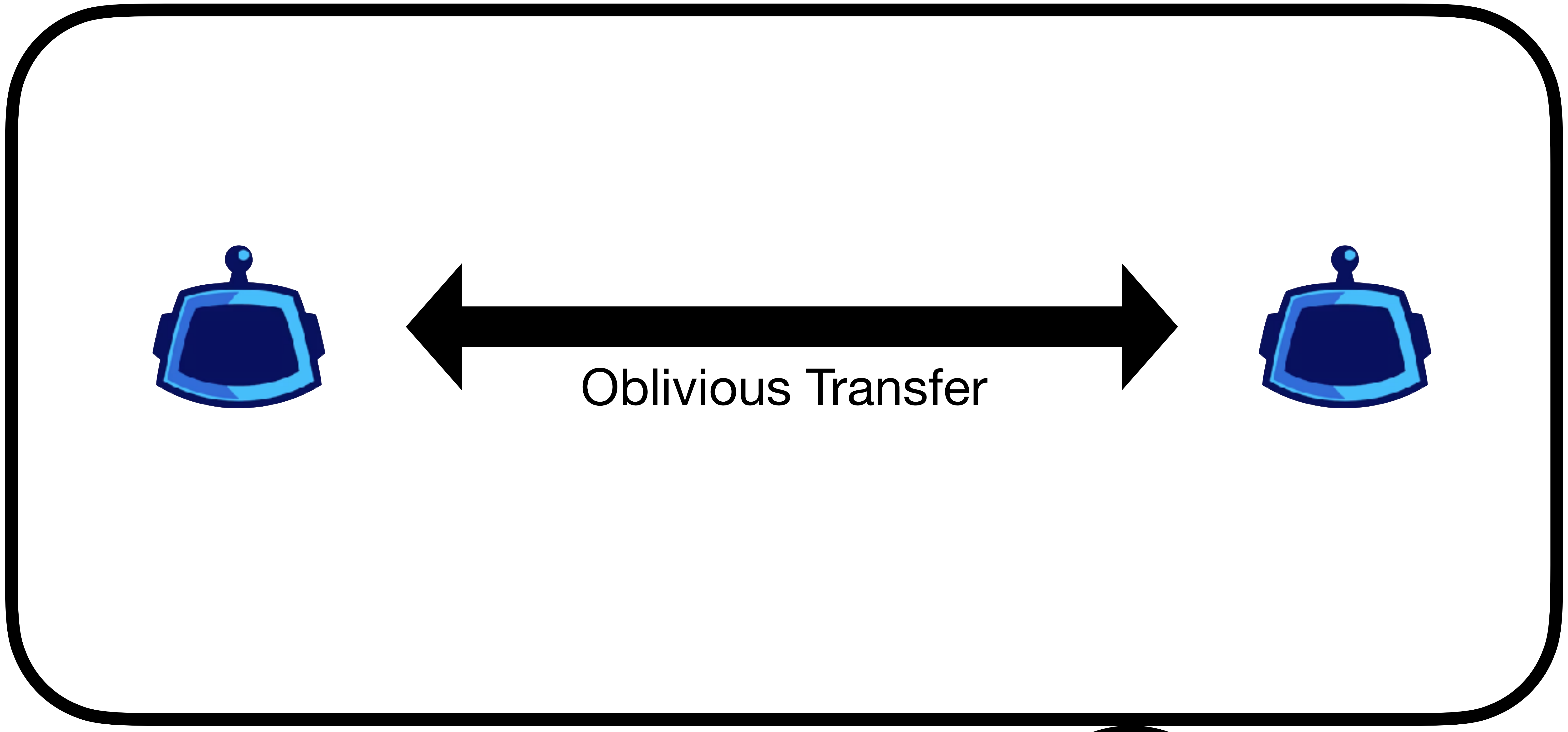
By opening an edge, V has probability at least 1/3 to catch cheating P

Repeat to obtain desired soundness

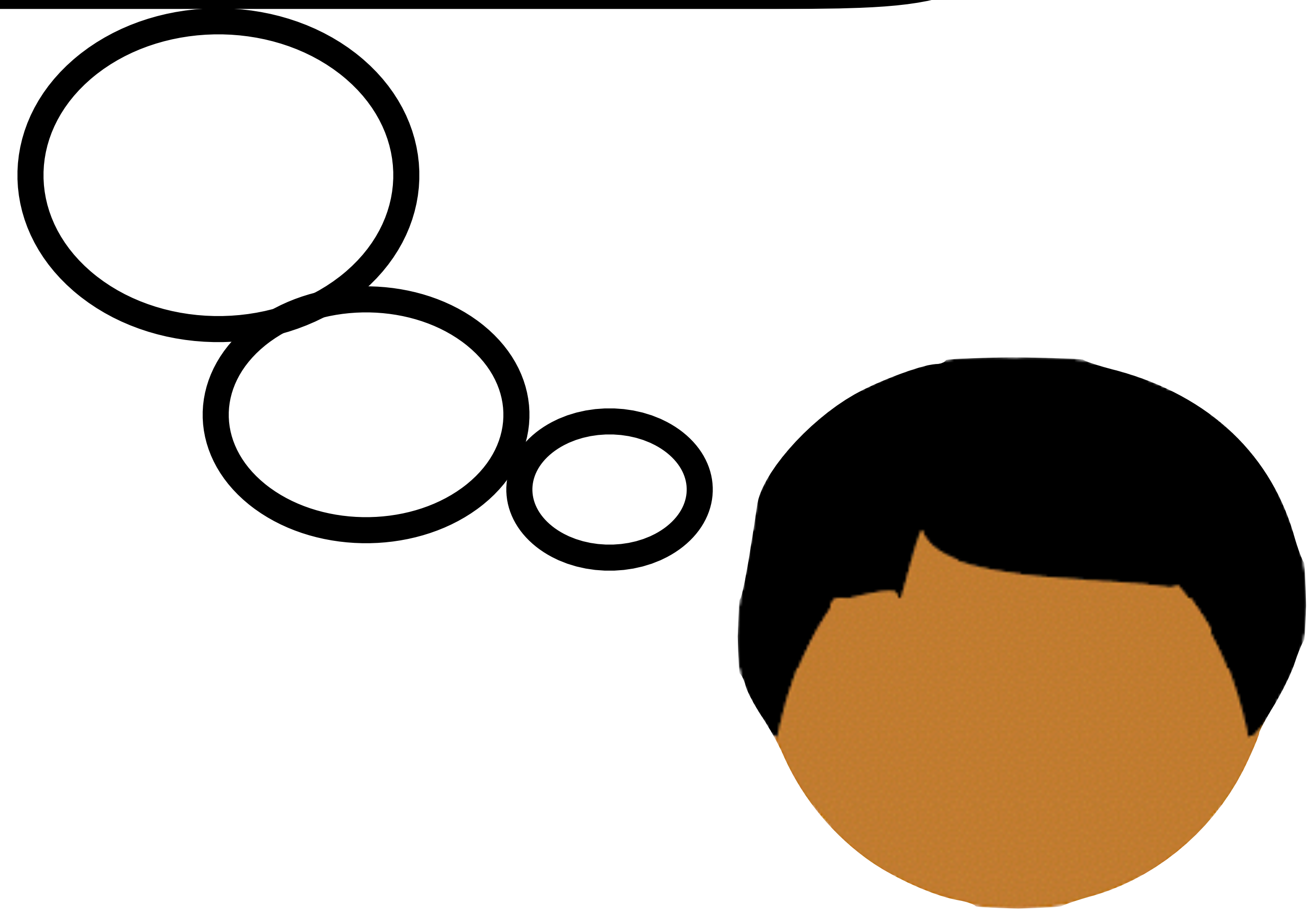
ZK from MPC in the Head



ZK from MPC in the Head

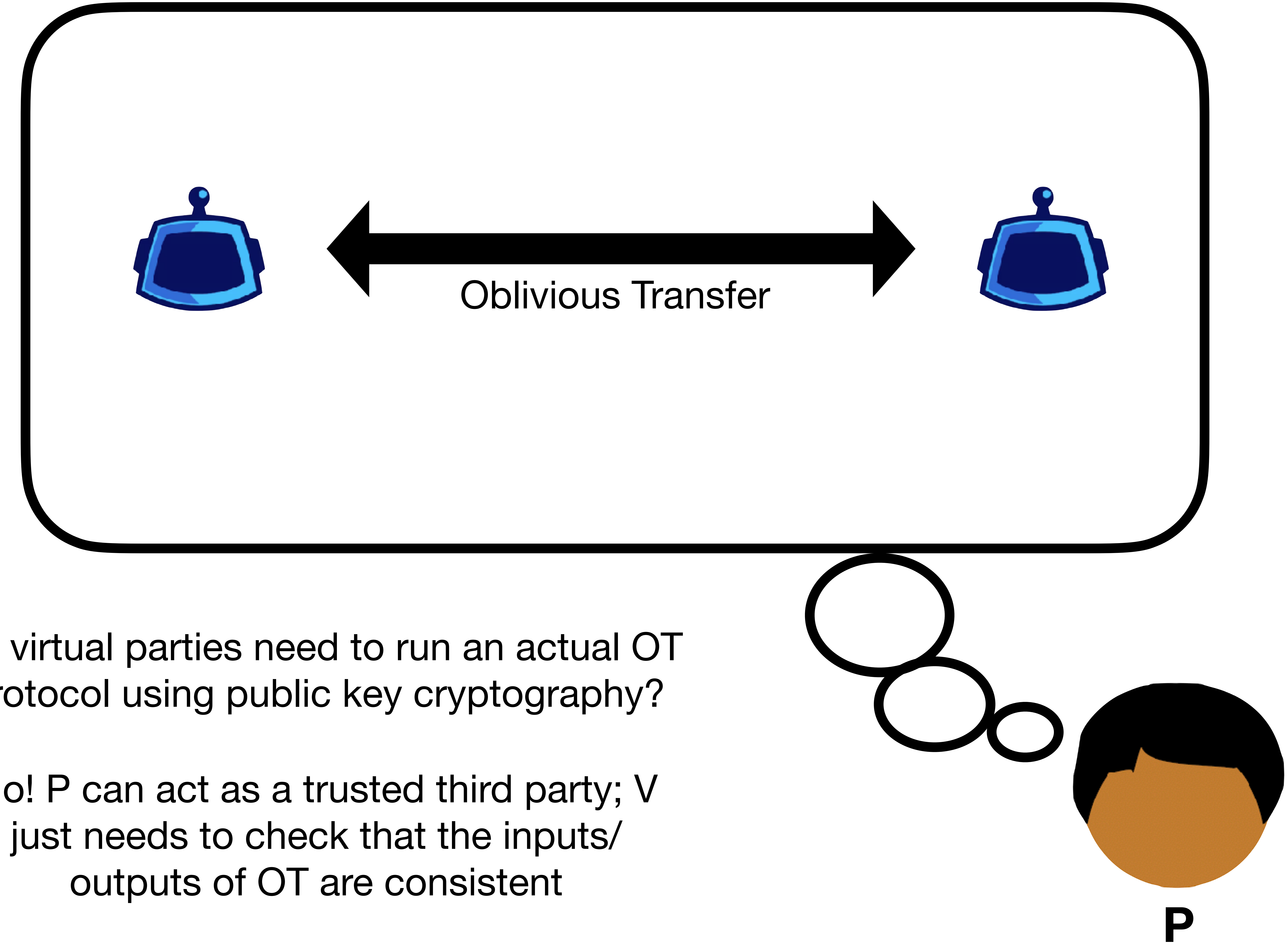


Do virtual parties need to run an actual OT protocol using public key cryptography?



P

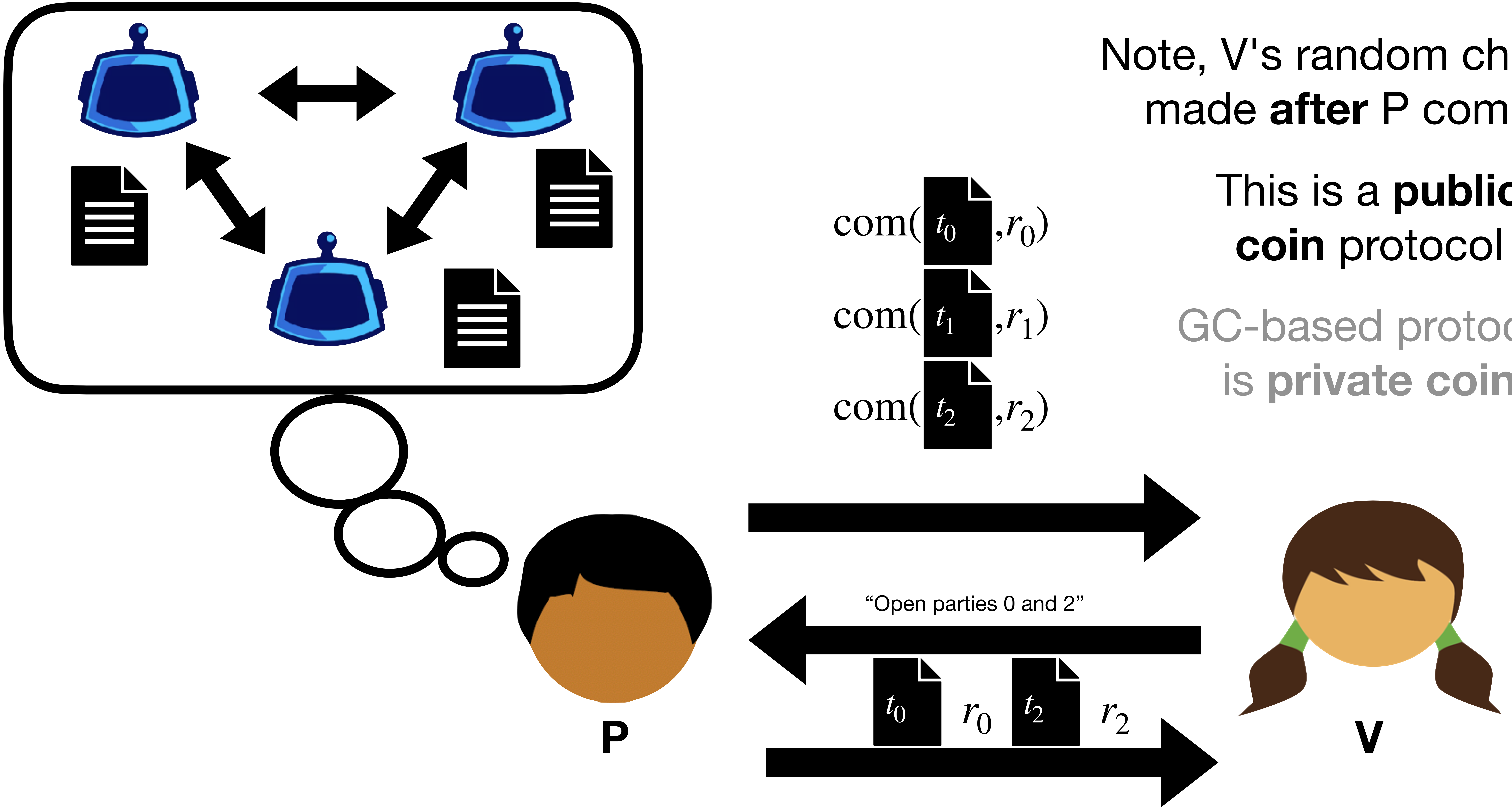
ZK from MPC in the Head



Do virtual parties need to run an actual OT protocol using public key cryptography?

No! P can act as a trusted third party; V just needs to check that the inputs/outputs of OT are consistent

ZK from MPC in the Head



Note, V's random choice is made **after** P commits

This is a **public coin** protocol

GC-based protocol is **private coin**

Fiat Shamir Heuristic


How To Prove Yourself:
Practical Solutions to Identification
and Signature Problems

Amos Fiat and Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract.

In this paper we describe simple identification and signature schemes which enable any user to prove his identity and the authenticity of his messages to any other user without shared or public keys. The schemes are provably secure against any known or chosen message attack if factoring is difficult, and typical implementations require only 1% to 4% of the number of modular multiplications required by the RSA scheme. Due to their simplicity, security and speed, these schemes are ideally suited for microprocessor-based devices such as smart cards, personal computers, and remote control systems.

1. Introduction

Creating unforgeable ID cards based on the emerging technology of smart cards is an important problem with numerous commercial and military applications. The problem becomes particularly challenging when the two parties (the prover A and the verifier B) are adversaries, and we want to make it impossible for B to misrepresent himself as A even after he witnesses and verifies arbitrarily many proofs of identity generated by A. Typical applications include passports (which are often inspected and photocopied by hostile governments), credit cards (whose numbers can be copied to blank cards or used over the phone), computer passwords (which are vulnerable to hackers and wire tappers) and military command and control systems (whose terminals may fall into enemy hands). We distinguish between three levels of protection:

1) Authentication schemes: A can prove to B that he is A, but someone else cannot prove to B that he is A.

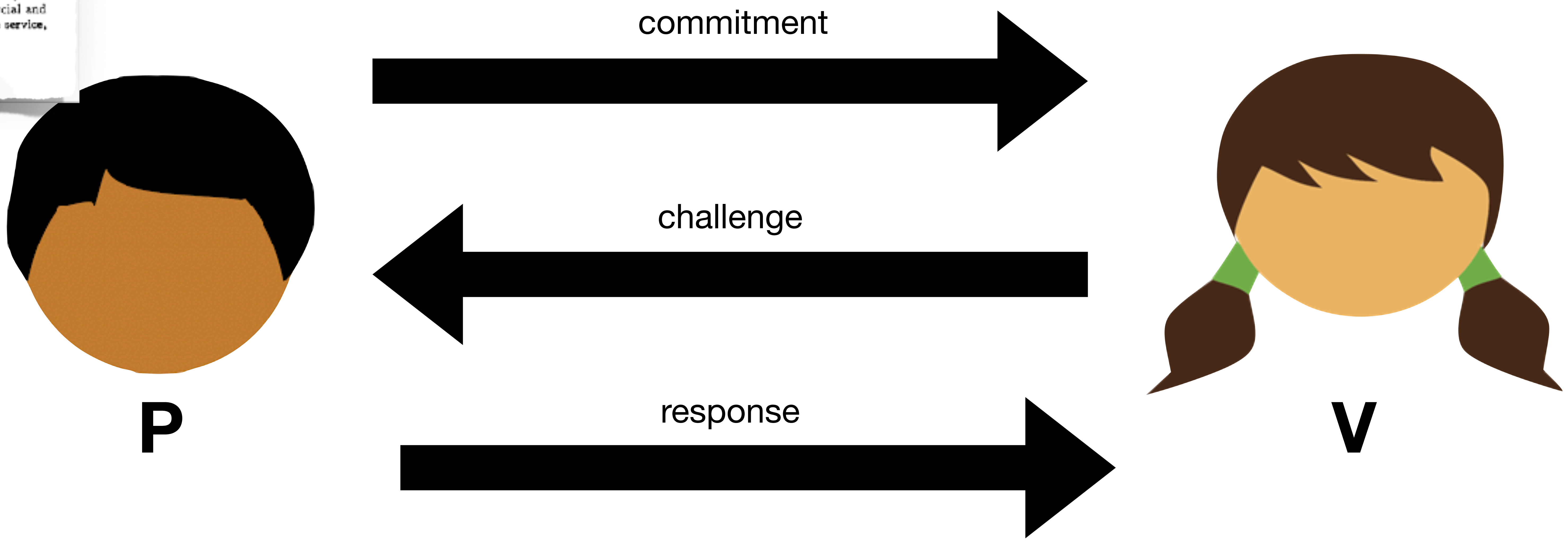
2) Identification schemes: A can prove to B that he is A, but B cannot prove to someone else that he is A.

3) Signature schemes: A can prove to B that he is A, but B cannot prove even to himself that he is A.

Authentication schemes are useful only against external threats when A and B cooperate. The distinction between identification and signature schemes is subtle, and manifests itself mainly when the proof is interactive and the verifier later wants to prove its existence to a judge: In identification schemes B can create a credible transcript of an imaginary communication by carefully choosing both the questions and the answers in the dialog, while in signature schemes only real communication with A could generate a credible transcript. However, in many commercial and military applications the main problem is to detect forgeries in real time and to deny the service,

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 186-194, 1987.
© Springer-Verlag Berlin Heidelberg 1987

Public coin ZK can be made **non-interactive**




How To Prove Yourself:
Practical Solutions to Identification
and Signature Problems

Amos Fiat and Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract.

In this paper we describe simple identification and signature schemes which enable any user to prove his identity and the authenticity of his messages to any other user without shared or public keys. The schemes are provably secure against any known or chosen message attack if factoring is difficult, and typical implementations require only 1% to 4% of the number of modular multiplications required by the RSA scheme. Due to their simplicity, security and speed, these schemes are ideally suited for microprocessor-based devices such as smart cards, personal computers, and remote control systems.

1. Introduction

Creating unforgeable ID cards based on the emerging technology of smart cards is an important problem with numerous commercial and military applications. The problem becomes particularly challenging when the two parties (the prover A and the verifier B) are adversaries, and we want to make it impossible for B to misrepresent himself as A even after he witnesses and verifies arbitrarily many proofs of identity generated by A. Typical applications include passports (which are often inspected and photocopied by hostile governments), credit cards (whose numbers can be copied to blank cards or used over the phone), computer passwords (which are vulnerable to hackers and wire tappers) and military command and control systems (whose terminals may fall into enemy hands). We distinguish between three levels of protection:

1) Authentication schemes: A can prove to B that he is A, but someone else cannot prove to B that he is A.

2) Identification schemes: A can prove to B that he is A, but B cannot prove to someone else that he is A.

3) Signature schemes: A can prove to B that he is A, but B cannot prove even to himself that he is A.

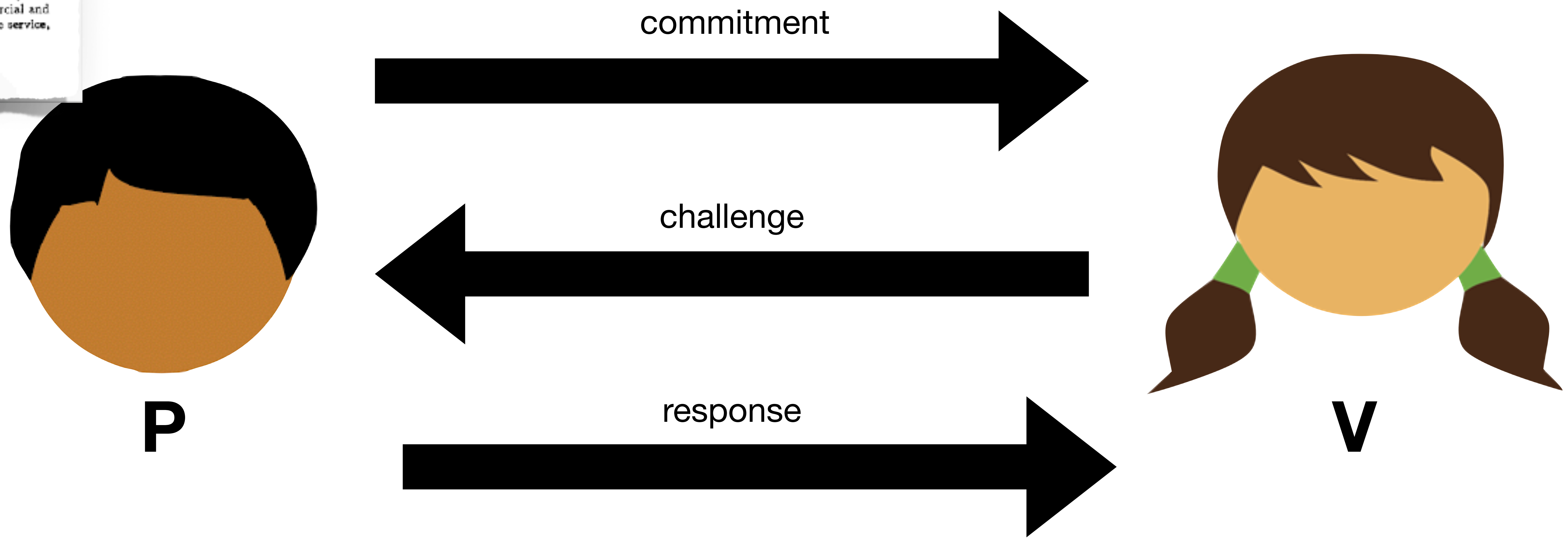
Authentication schemes are useful only against external threats when A and B cooperate. The distinction between identification and signature schemes is subtle, and manifests itself mainly when the proof is interactive and the verifier later wants to prove its existence to a judge: In identification schemes B can create a credible transcript of an imaginary communication by carefully choosing both the questions and the answers in the dialog, while in signature schemes only real communication with A could generate a credible transcript. However, in many commercial and military applications the main problem is to detect forgeries in real time and to deny the service,

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 186-194, 1987.
© Springer-Verlag Berlin Heidelberg 1987

Fiat Shamir Heuristic

Public coin ZK can be made **non-interactive**

Simple idea: P can choose the challenge itself



Fiat Shamir Heuristic

How To Prove Yourself:
Practical Solutions to Identification
and Signature Problems

Amos Fiat and Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract.

In this paper we describe simple identification and signature schemes which enable any user to prove his identity and the authenticity of his messages to any other user without shared or public keys. The schemes are provably secure against any known or chosen message attack if factoring is difficult, and typical implementations require only 1% to 4% of the number of modular multiplications required by the RSA scheme. Due to their simplicity, security and speed, these schemes are ideally suited for microprocessor-based devices such as smart cards, personal computers, and remote control systems.

1. Introduction

Creating unforgeable ID cards based on the emerging technology of smart cards is an important problem with numerous commercial and military applications. The problem becomes particularly challenging when the two parties (the prover A and the verifier B) are adversaries, and we want to make it impossible for B to misrepresent himself as A even after he witnesses and verifies arbitrarily many proofs of identity generated by A. Typical applications include passports (which are often inspected and photocopied by hostile governments), credit cards (whose numbers can be copied to blank cards or used over the phone), computer passwords (which are vulnerable to hackers and wire tappers) and military command and control systems (whose terminals may fall into enemy hands). We distinguish between three levels of protection:

1) Authentication schemes: A can prove to B that he is A, but someone else cannot prove to B that he is A.

2) Identification schemes: A can prove to B that he is A, but B cannot prove to someone else that he is A.

3) Signature schemes: A can prove to B that he is A, but B cannot prove even to himself that he is A.

Authentication schemes are useful only against external threats when A and B cooperate. The distinction between identification and signature schemes is subtle, and manifests itself mainly when the proof is interactive and the verifier later wants to prove its existence to a judge: In identification schemes B can create a credible transcript of an imaginary communication by carefully choosing both the questions and the answers in the dialog, while in signature schemes only real communication with A could generate a credible transcript. However, in many commercial and military applications the main problem is to detect forgeries in real time and to deny the service,

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 186-194, 1987.
© Springer-Verlag Berlin Heidelberg 1987

Public coin ZK can be made **non-interactive**

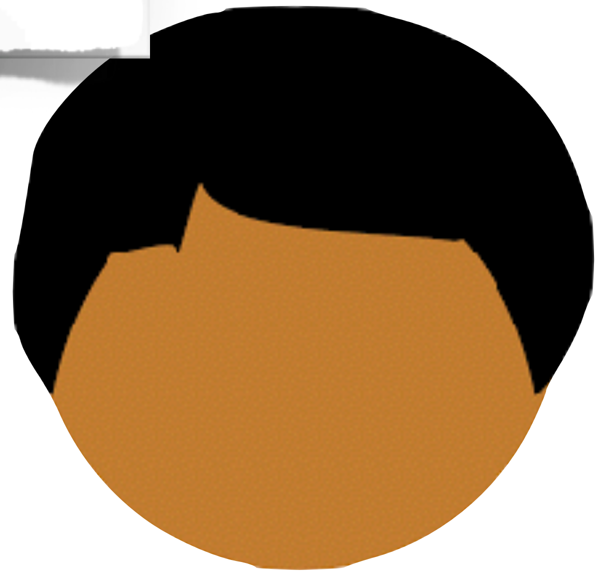
Simple idea: P can choose the challenge itself

Cryptographic hash function (e.g. SHA 256)
Formally, a random oracle

commitment

challenge = H(commitment)

response



P

How To Prove Yourself:
Practical Solutions to Identification
and Signature Problems

Amos Fiat and Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract.

In this paper we describe simple identification and signature schemes which enable any user to prove his identity and the authenticity of his messages to any other user without shared or public keys. The schemes are provably secure against any known or chosen message attack if factoring is difficult, and typical implementations require only 1% to 4% of the number of modular multiplications required by the RSA scheme. Due to their simplicity, security and speed, these schemes are ideally suited for microprocessor-based devices such as smart cards, personal computers, and remote control systems.

1. Introduction

Creating unforgeable ID cards based on the emerging technology of smart cards is an important problem with numerous commercial and military applications. The problem becomes particularly challenging when the two parties (the prover A and the verifier B) are adversaries, and we want to make it impossible for B to misrepresent himself as A even after he witnesses and verifies arbitrarily many proofs of identity generated by A. Typical applications include passports (which are often inspected and photocopied by hostile governments), credit cards (whose numbers can be copied to blank cards or used over the phone), computer passwords (which are vulnerable to hackers and wire tappers) and military command and control systems (whose terminals may fall into enemy hands). We distinguish between three levels of protection:

1) Authentication schemes: A can prove to B that he is A, but someone else cannot prove to B that he is A.

2) Identification schemes: A can prove to B that he is A, but B cannot prove to someone else that he is A.

3) Signature schemes: A can prove to B that he is A, but B cannot prove even to himself that he is A.

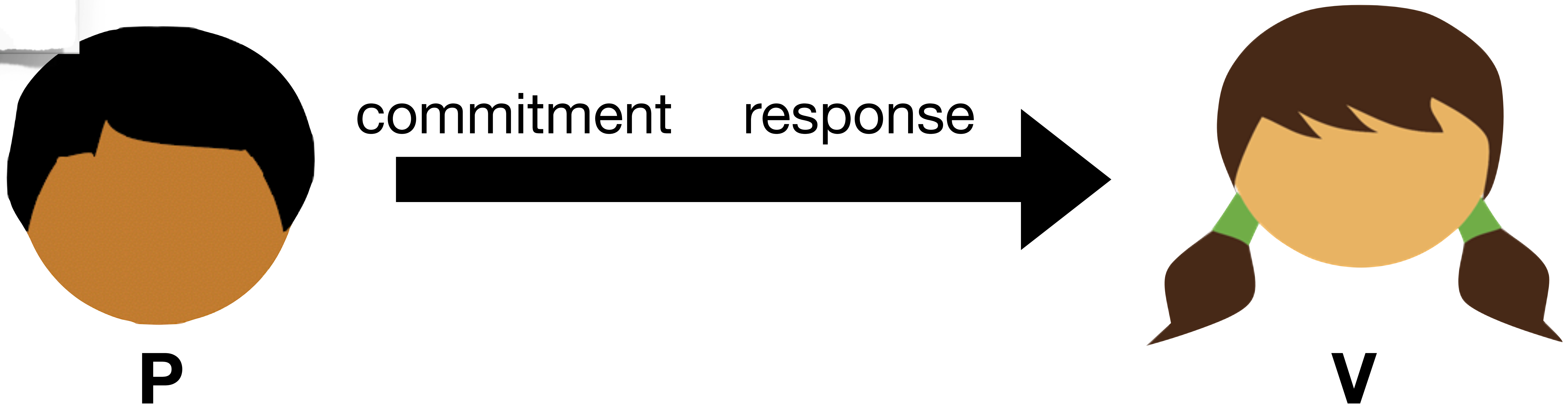
Authentication schemes are useful only against external threats when A and B cooperate. The distinction between identification and signature schemes is subtle, and manifests itself mainly when the proof is interactive and the verifier later wants to prove its existence to a judge: In identification schemes B can create a credible transcript of an imaginary communication by carefully choosing both the questions and the answers in the dialog, while in signature schemes only real communication with A could generate a credible transcript. However, in many commercial and military applications the main problem is to detect forgeries in real time and to deny the service,

A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 186-194, 1987.
© Springer-Verlag Berlin Heidelberg 1987

Fiat Shamir Heuristic

Public coin ZK can be made **non-interactive**

Simple idea: P can choose the challenge itself



Fiat Shamir Heuristic

How To Prove Yourself:
Practical Solutions to Identification
and Signature Problems

Amos Fiat and Adi Shamir
Department of Applied Mathematics
The Weizmann Institute of Science
Rehovot 76100, Israel

Abstract.

In this paper we describe simple identification and signature schemes which enable any user to prove his identity and the authenticity of his messages to any other user without shared or public keys. The schemes are provably secure against any known or chosen message attack if factoring is difficult, and typical implementations require only 1% to 4% of the number of modular multiplications required by the RSA scheme. Due to their simplicity, security and speed, these schemes are ideally suited for microprocessor-based devices such as smart cards, personal computers, and remote control systems.

1. Introduction

Creating unforgeable ID cards based on the emerging technology of smart cards is an important problem with numerous commercial and military applications. The problem becomes particularly challenging when the two parties (the prover A and the verifier B) are adversaries, and we want to make it impossible for B to misrepresent himself as A even after he witnesses and verifies arbitrarily many proofs of identity generated by A. Typical applications include passports (which are often inspected and photocopied by hostile governments), credit cards (whose numbers can be copied to blank cards or used over the phone), computer passwords (which are vulnerable to hackers and wire tappers) and military command and control systems (whose terminals may fall into enemy hands). We distinguish between three levels of protection:

1) Authentication schemes: A can prove to B that he is A, but someone else cannot prove to B that he is A.

2) Identification schemes: A can prove to B that he is A, but B cannot prove to someone else that he is A.

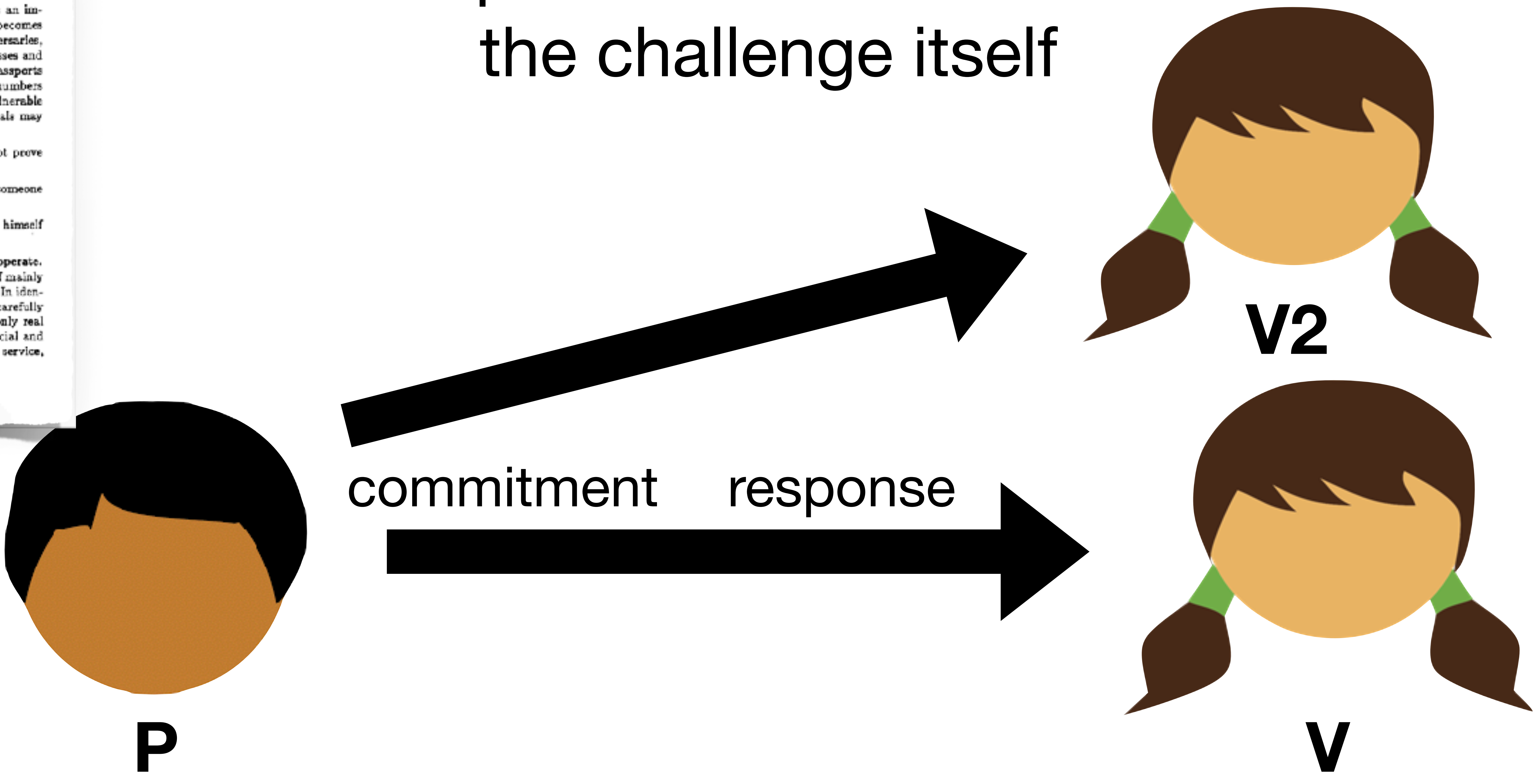
3) Signature schemes: A can prove to B that he is A, but B cannot prove even to himself that he is A.

Authentication schemes are useful only against external threats when A and B cooperate. The distinction between identification and signature schemes is subtle, and manifests itself mainly when the proof is interactive and the verifier later wants to prove its existence to a judge: In identification schemes B can create a credible transcript of an imaginary communication by carefully choosing both the questions and the answers in the dialog, while in signature schemes only real communication with A could generate a credible transcript. However, in many commercial and military applications the main problem is to detect forgeries in real time and to deny the service,

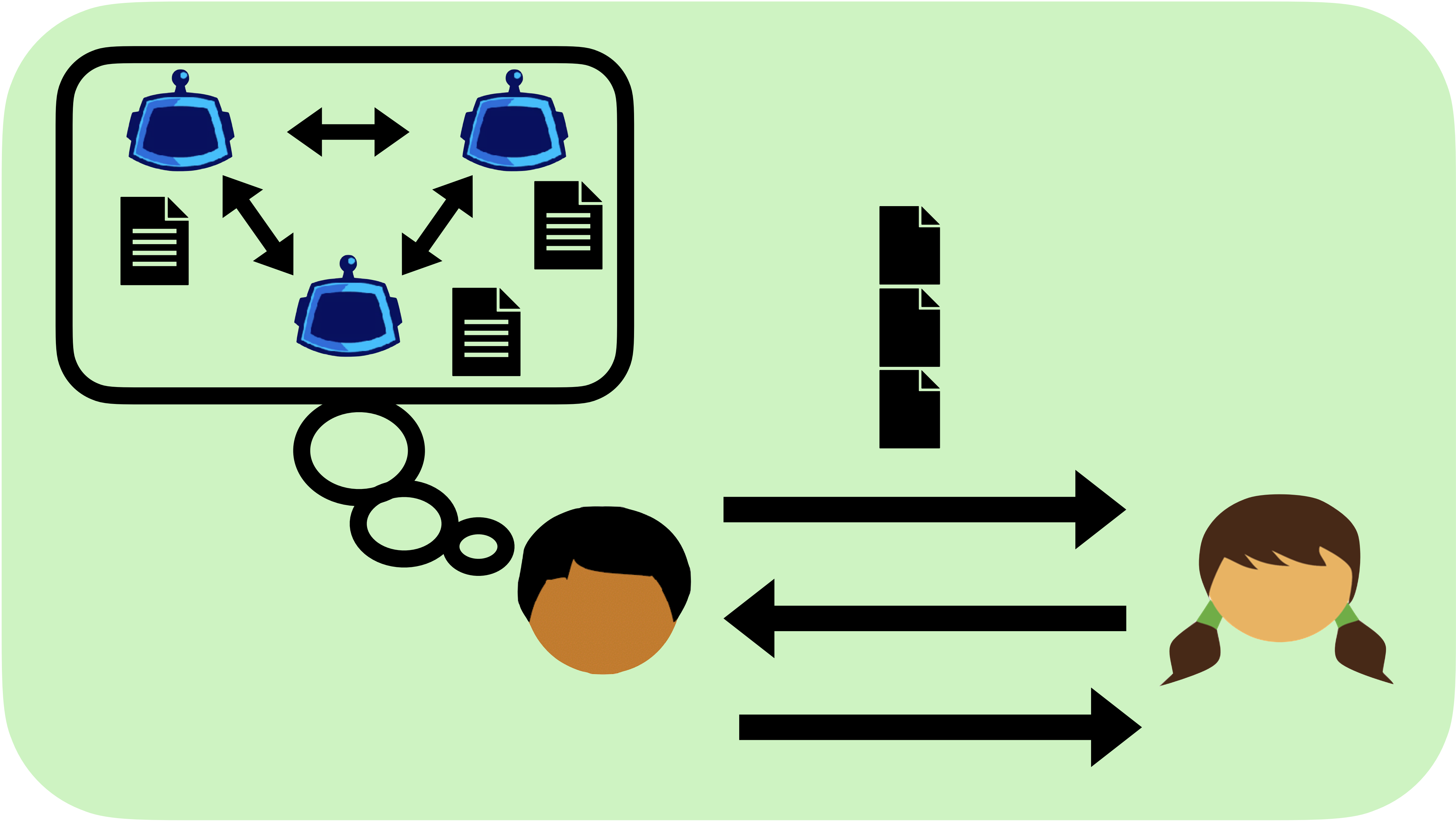
A.M. Odlyzko (Ed.): Advances in Cryptology - CRYPTO '86, LNCS 263, pp. 186-194, 1987.

Public coin ZK can be made **non-interactive**

Simple idea: P can choose the challenge itself



ZK from MPC in the Head



P runs an MPC protocol in its head to calculate transcripts

V challenges a subset of the transcripts

P opens, and therefore is caught with some probability if cheating

P can commit to multiple repetitions of the protocol to amplify soundness

P can calculate its own challenges using a hash function (Fiat Shamir)

Today's objectives

Review Zero Knowledge Proofs

Construct ZK for circuit satisfiability problem

- From Garbled Circuits
- Via "MPC in the Head"