Malicious Garbled Circuits

CS 598 DH

Today’s objectives

Understand Cut and Choose

Understand Information-Theoretic MACs

Sketch authenticated garbling protocol

Setting

Semi-honest Security

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

ORAM

Setting

Semi-honest Security

Malicious Security

Zero Knowledge

Previously

General-Purpose Tools

GMW Protocol

- Multi-party
- Multi-round

Garbled Circuit

- Constant Round
- Two Party

Primitives

Oblivious Transfer

Pseudorandom functions/encryption

Commitments

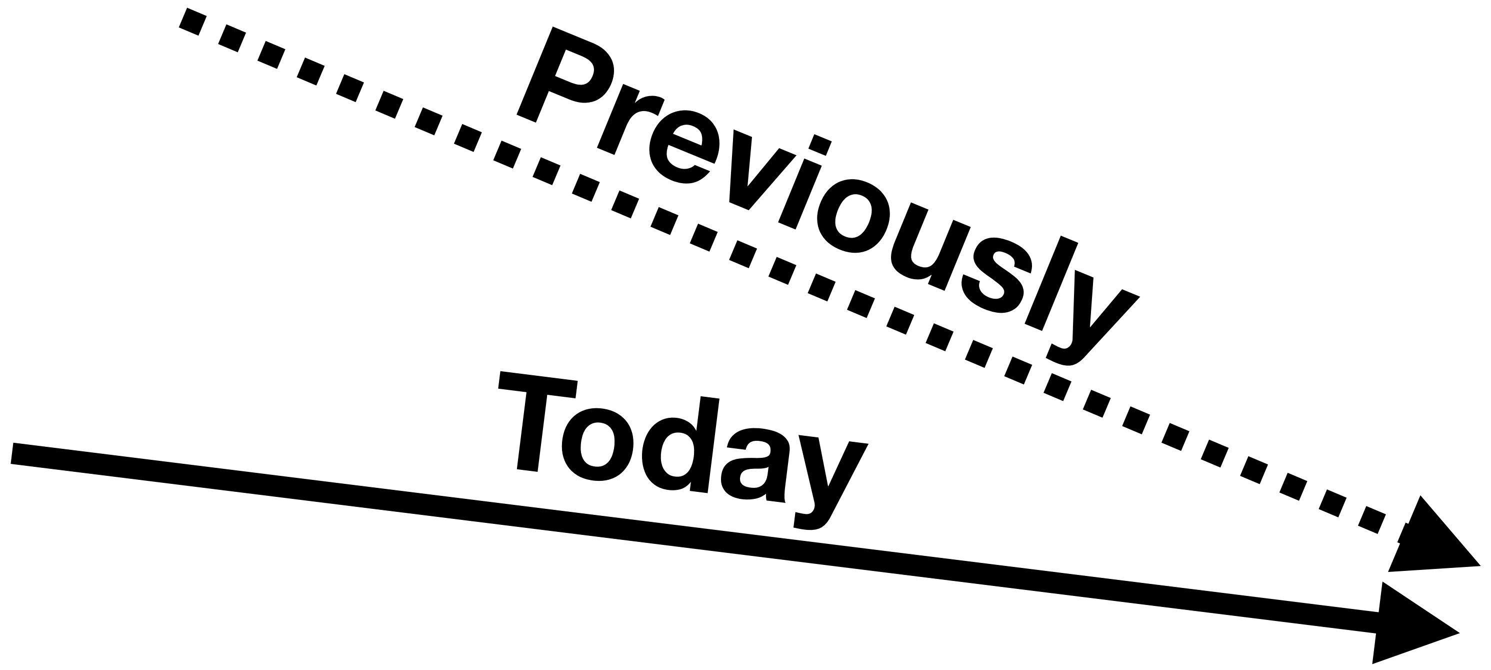
ORAM

Setting

Semi-honest Security

Malicious Security

Zero Knowledge



General-Purpose Tools

GMW Protocol

- Multi-party
- Multi-round

Garbled Circuit

- Constant Round
- Two Party

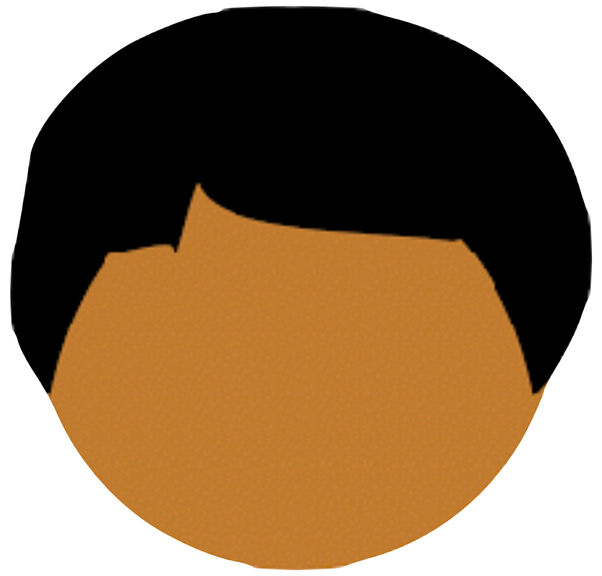
Primitives

Oblivious Transfer

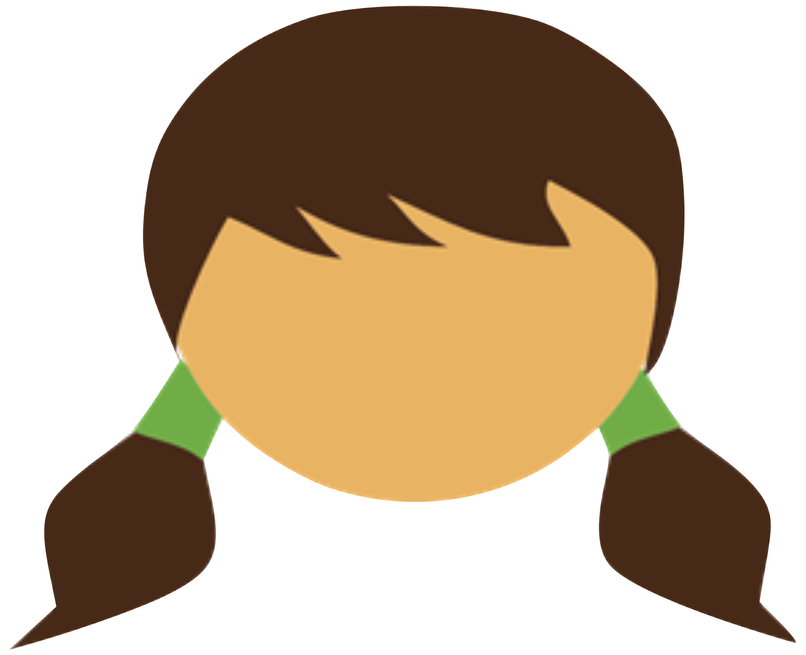
Pseudorandom functions/encryption

Commitments

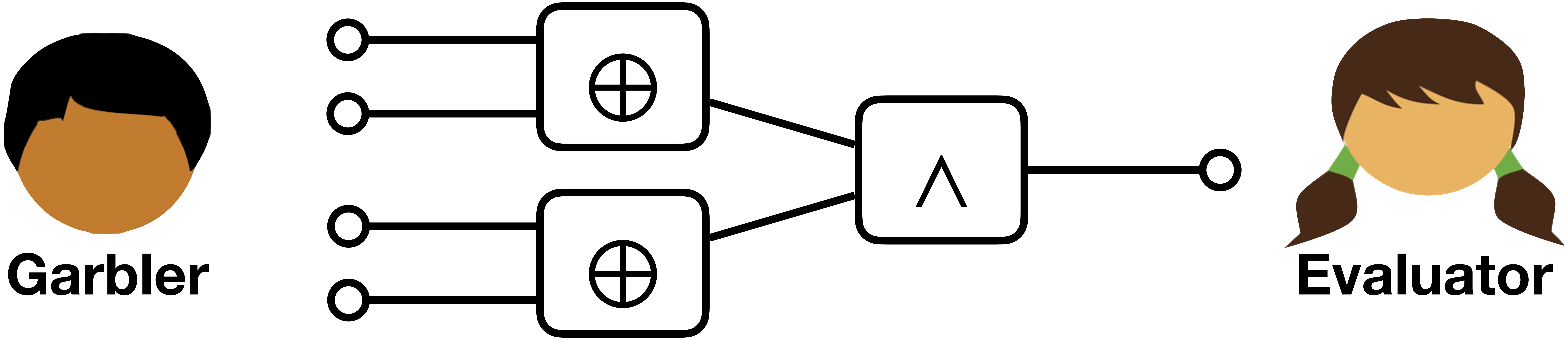
ORAM

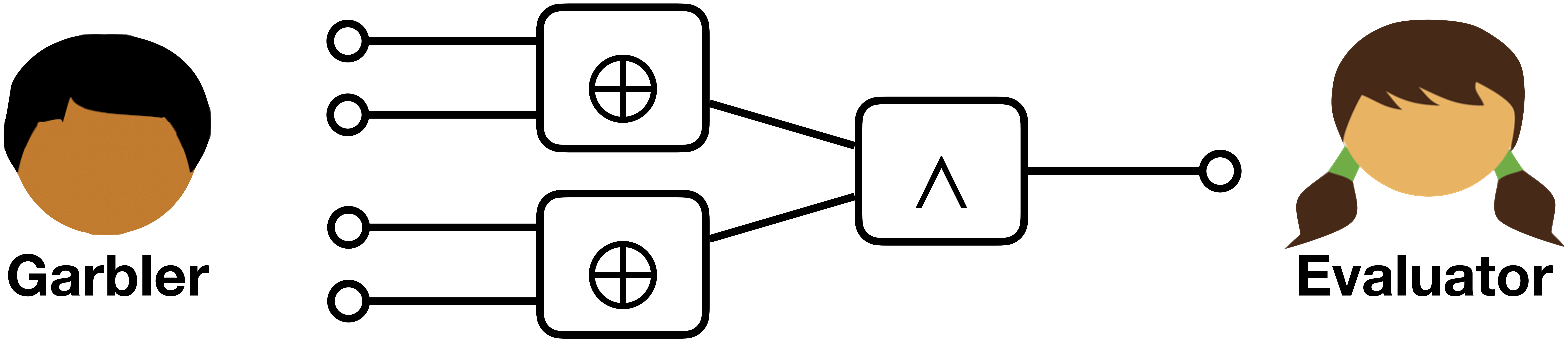


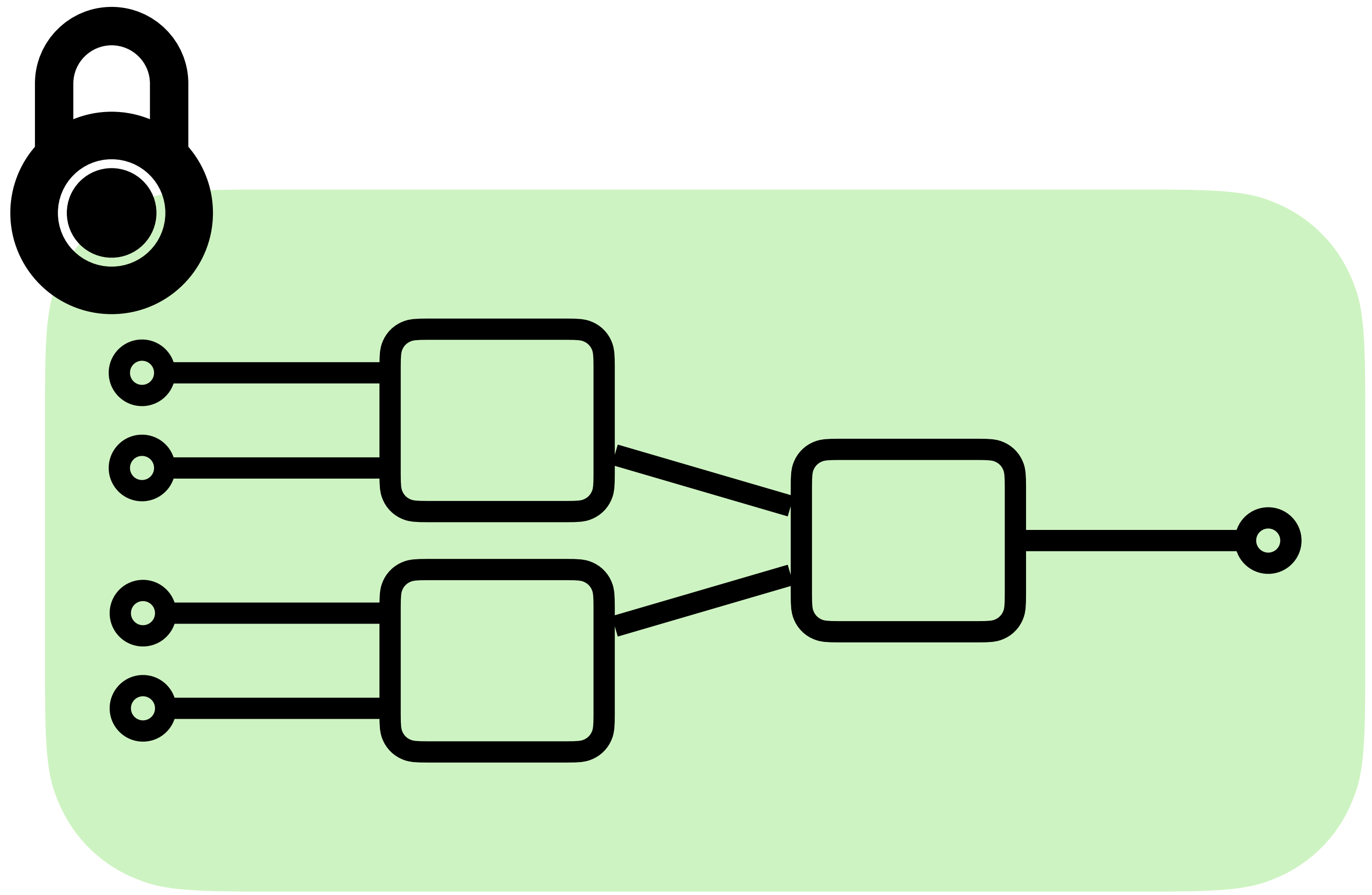
Garbler

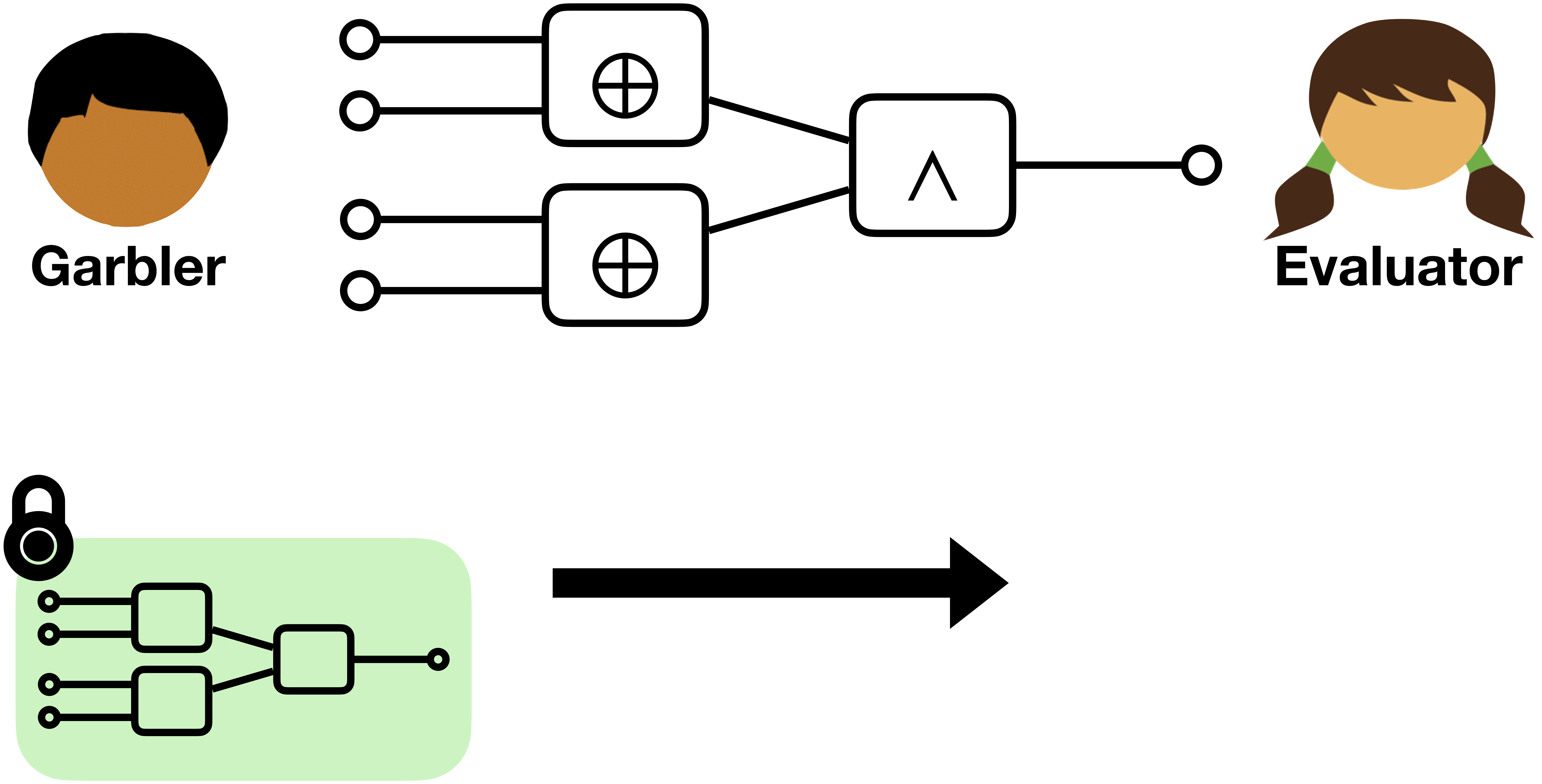


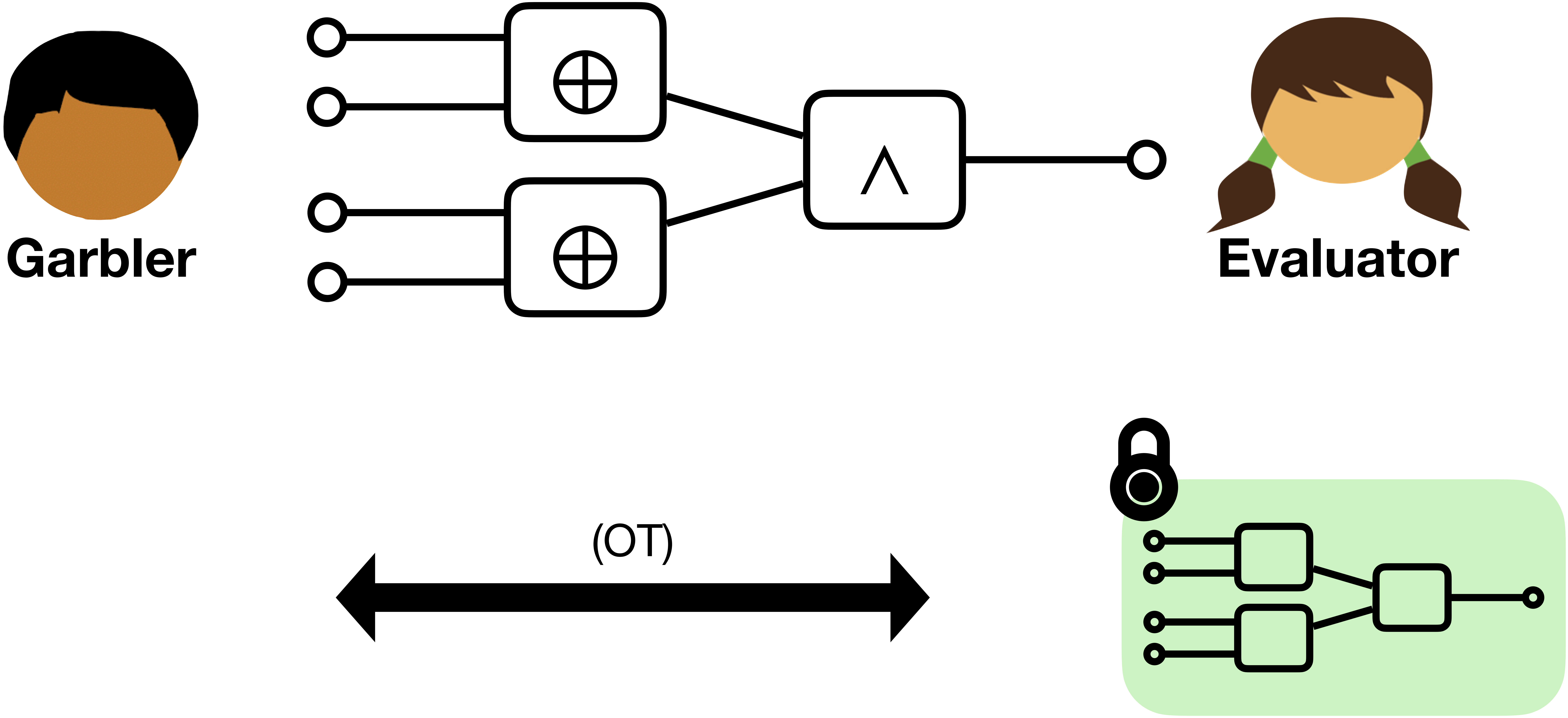
Evaluator

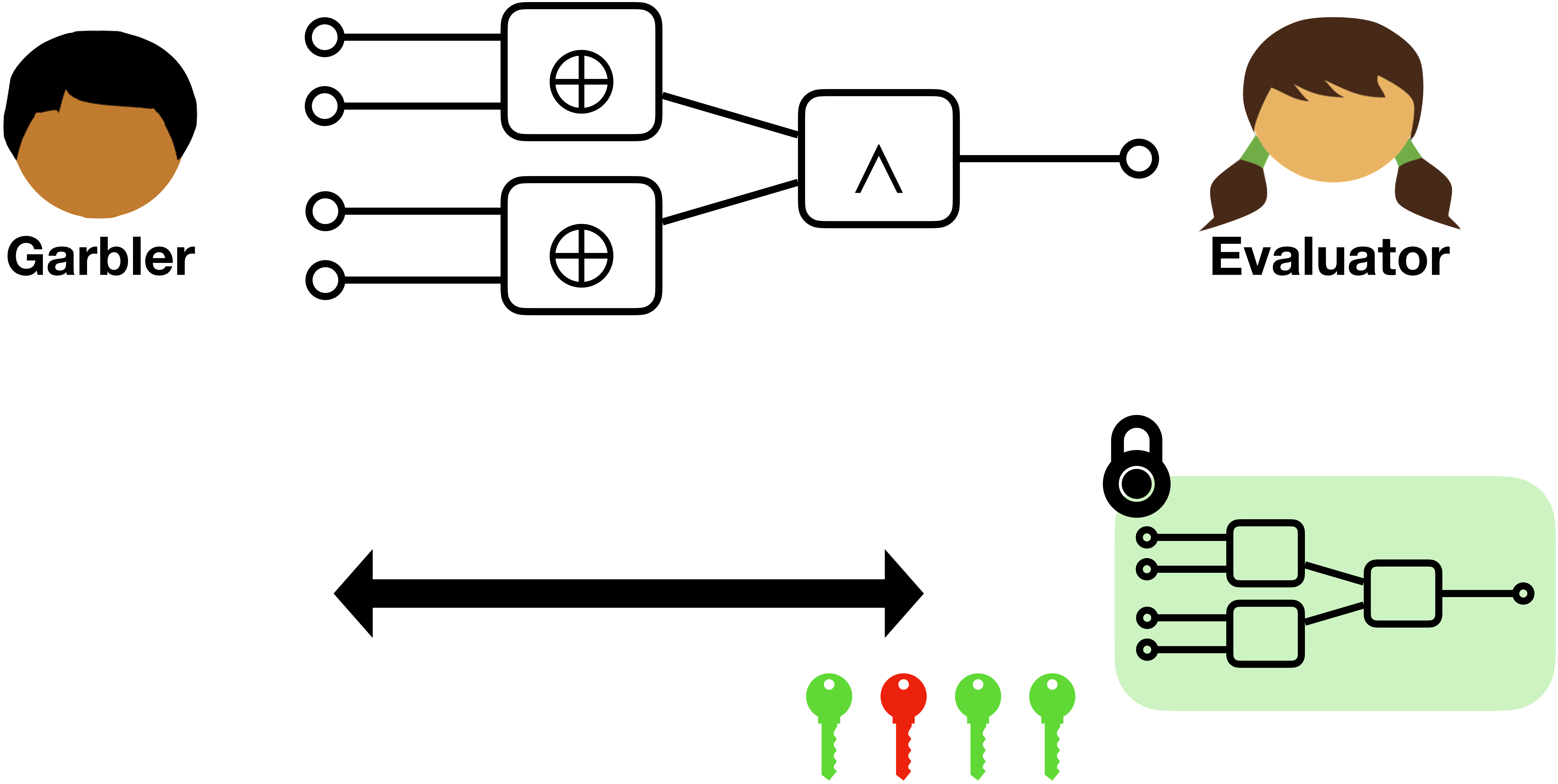

Garbler
⊕
⊕
∧
Evaluator

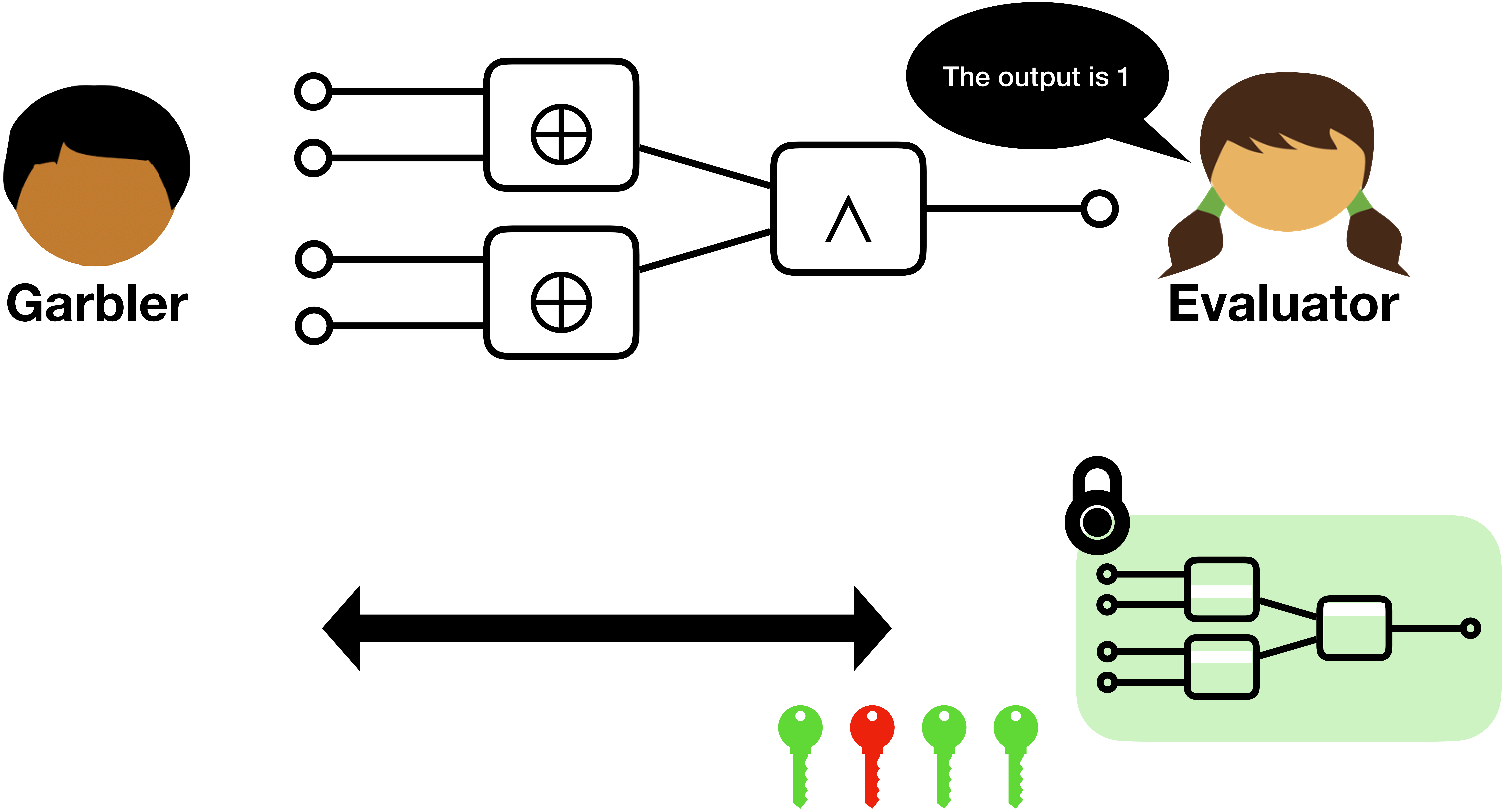

Garbler
⊕
⊕
∧
Evaluator



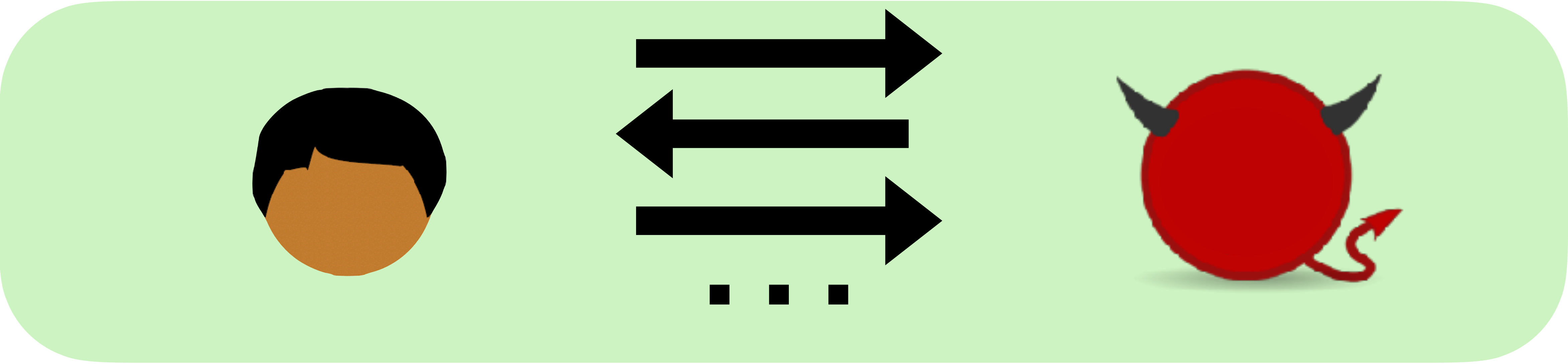

Garbler
⊕
⊕
∧
Evaluator


Garbler
Evaluator
(OT)


Garbler
Evaluator


The output is 1
Garbler
Evaluator

Malicious Security (with abort)



*A protocol Π securely realizes a functionality f in the presence of a malicious (with abort) adversary if for **every** real-world adversary $\mathscr{A}$ corrupting party i, **there exists** an ideal-world adversary $\mathscr{S}_i$ (a simulator) such that for all inputs x, y the following holds:*

$$\mathrm{Real}^{\Pi}_{\mathscr{A}}(x, y) \approx \mathrm{Ideal}^{f}_{\mathscr{S}_i}(x, y)$$

Ensemble of outputs of ***each*** party

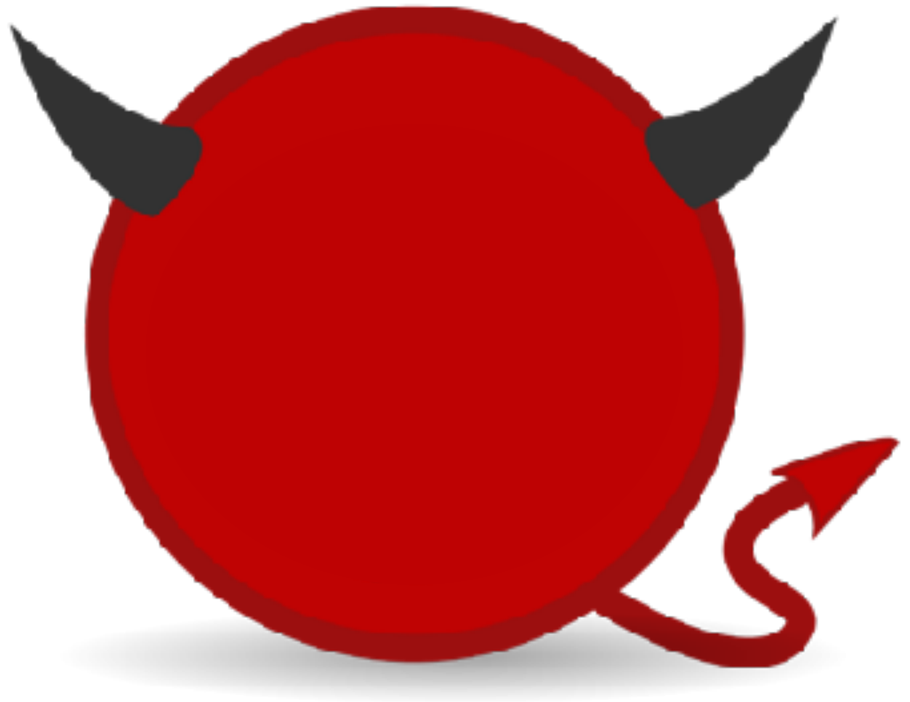
Why can't we simulate G?

Garbler

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

G can encrypt each gate *freely*

E has no way to tell if gate it correctly garbled



Garbler

Why can't we simulate G?

G can encrypt each gate *freely*

E has no way to tell if gate it correctly garbled

$$\begin{aligned}
&\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^0, K_c^0))\\
&\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^1, K_c^0))\\
&\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^0, K_c^0))\\
&\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^1, K_c^1))
\end{aligned}$$

$\Rightarrow$

$$\begin{aligned}
&\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^0, K_c^0))\\
&\mathrm{Enc}(K_a^0, \mathrm{Enc}(K_b^1, K_c^0))\\
&\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^0, K_c^1))\\
&\mathrm{Enc}(K_a^1, \mathrm{Enc}(K_b^1, K_c^1))
\end{aligned}$$

The Cut-and-Choose Game and its Application to Cryptographic Protocols

Ruiyu Zhu, *Indiana University*
Yan Huang, *Indiana University*
Jonathan Katz, *University of Maryland*
abhi shelat, *Northeastern University*

A New Approach to Practical
Active-Secure Two-Party Computation

Jesper Buus Nielsen[1], Peter Sebastian Nordholt[1], Claudio Orlandi[2], Sai Sheshank Burra[3]

[1] Aarhus University
[2] Bar-Ilan University
[3] Indian Institute of Technology Guwahati

Abstract. We propose a new approach to practical two-party computation secure against an active adversary. All prior practical protocols were based on Yao's garbled circuits. We use an OT-based approach and get efficiency via OT extension in the random oracle model. To get a practical protocol we introduce a number of novel techniques for relating the outputs and inputs of OTs in a larger construction.
We also report on an implementation of this approach, that shows that our protocol is more efficient than any previous one: For big enough circuits, we can evaluate more than 20000 Boolean gates per second. As an example, evaluating one oblivious AES encryption (~ 34000 gates) takes 64 seconds, but when repeating the task 27 times it only takes less than 3 seconds per instance.

Problem with GC

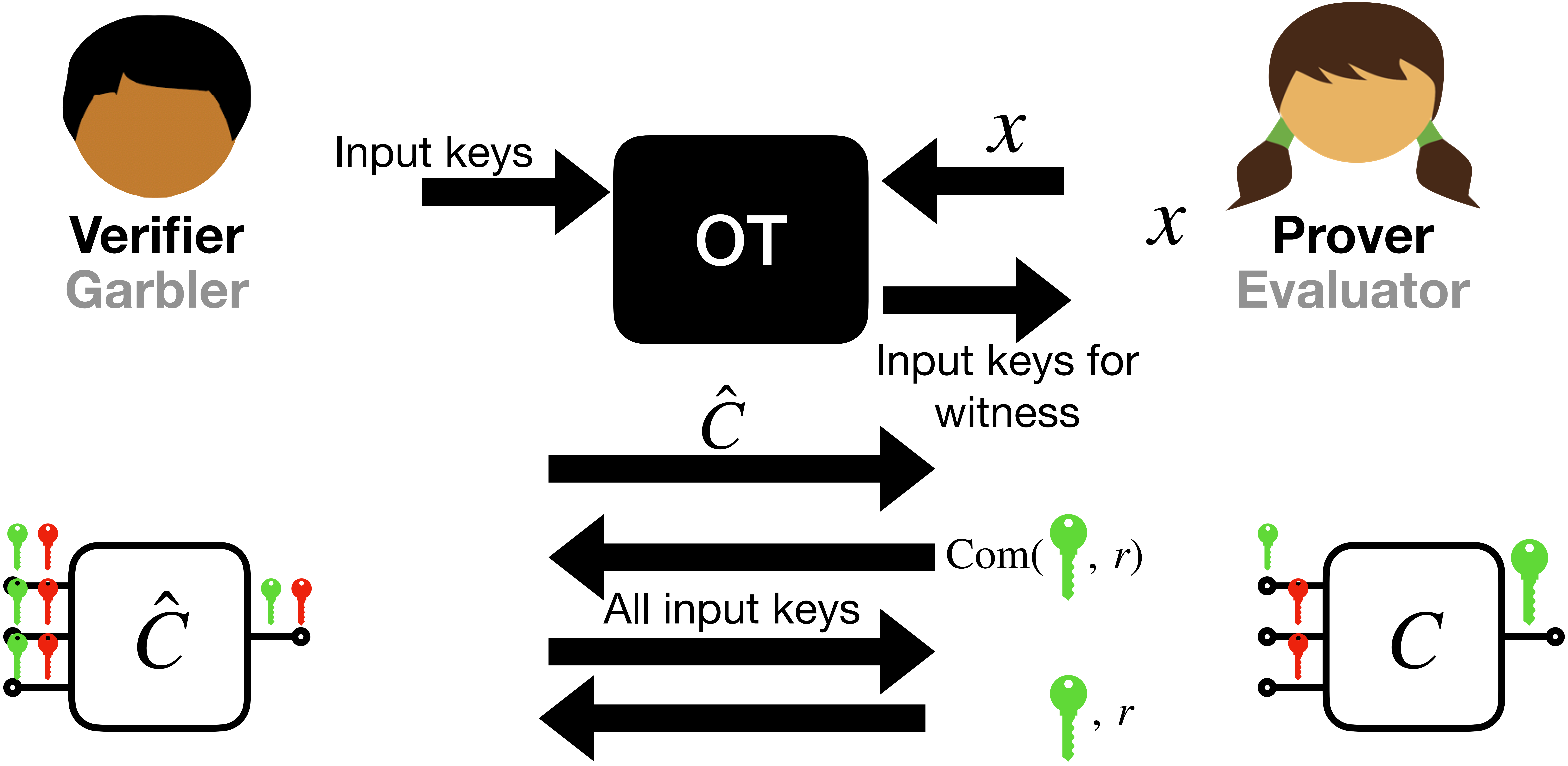
Adversary can arbitrarily
corrupt the GC encryptions

Solution Space

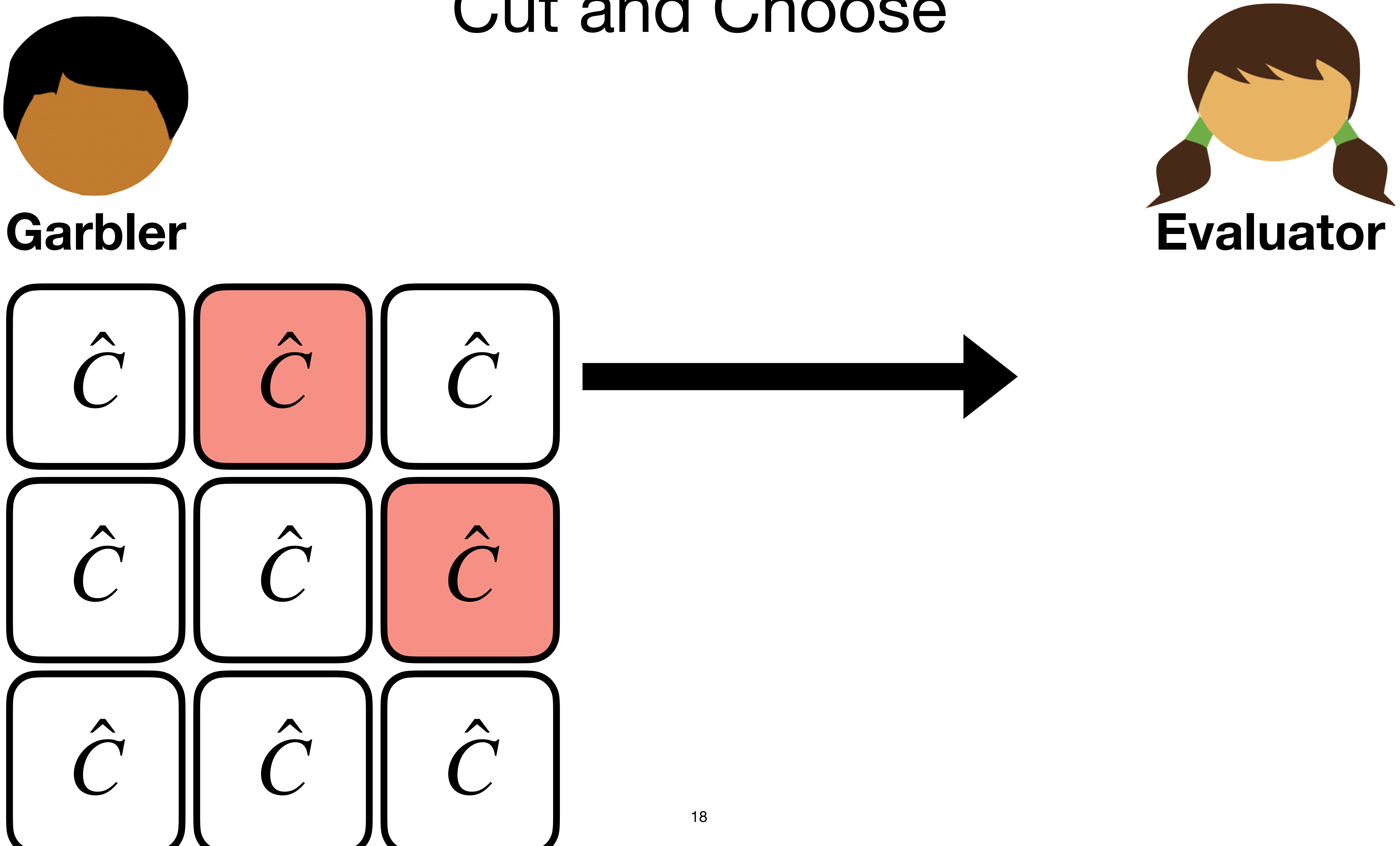
Evaluator can check a
garbled circuit is well-formed

We can repeat multiple times

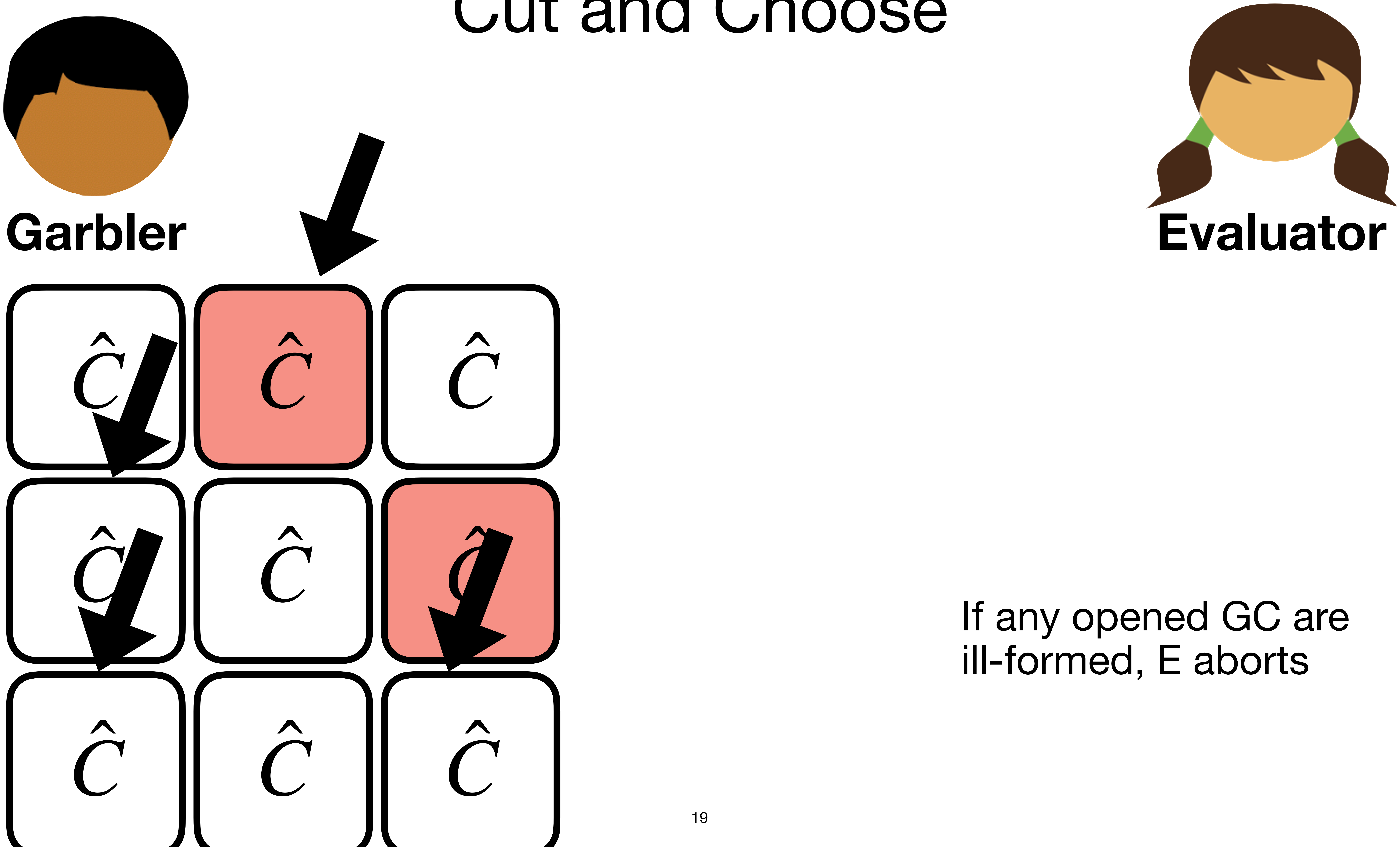
Zero Knowledge from Garbled Circuits



Cut and Choose

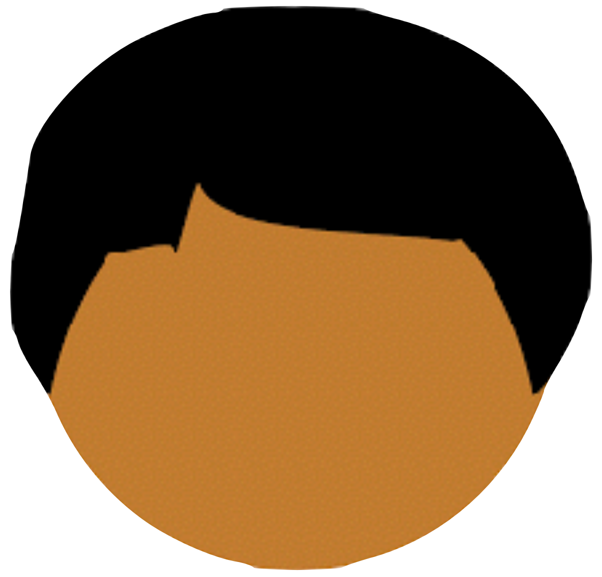


Cut and Choose

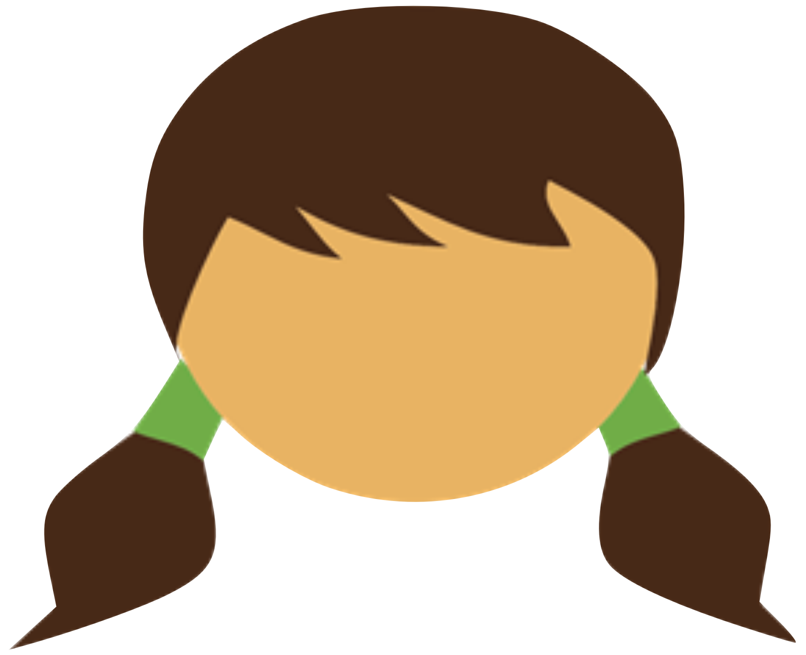


If any opened GC are ill-formed, E aborts

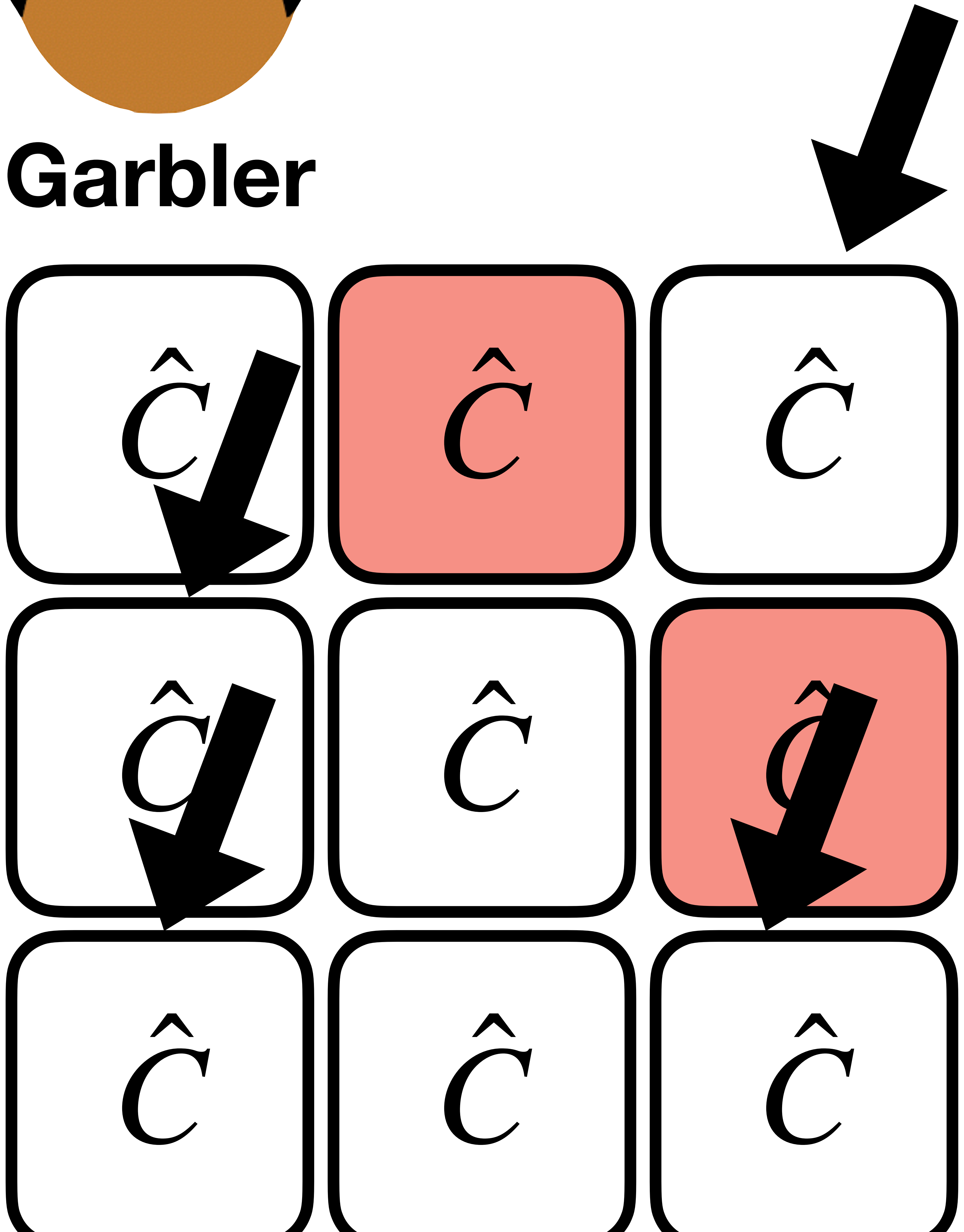
Cut and Choose



Garbler

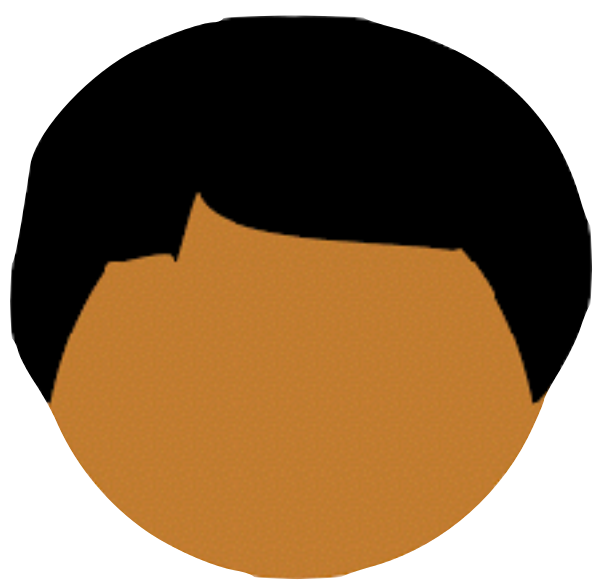


Evaluator

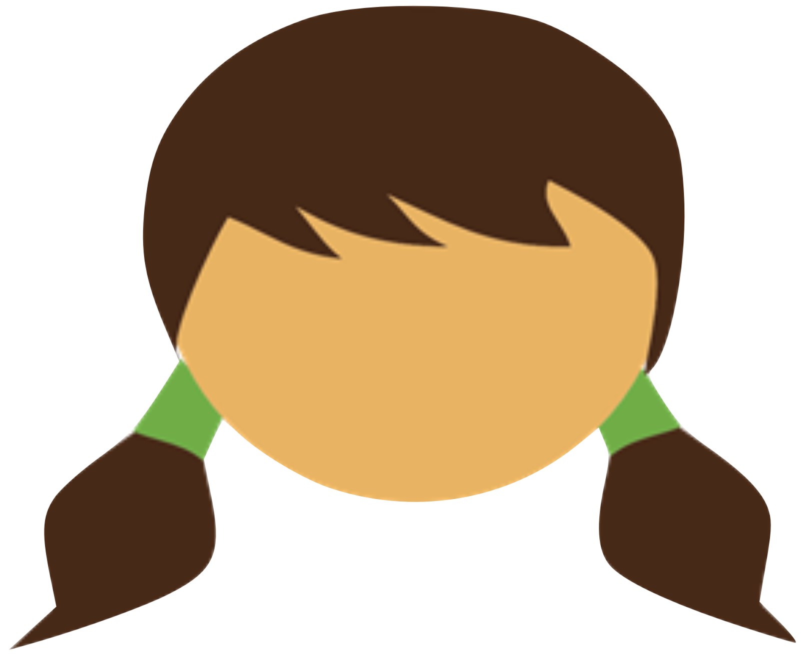


If all opened GC are well-formed, parties continue

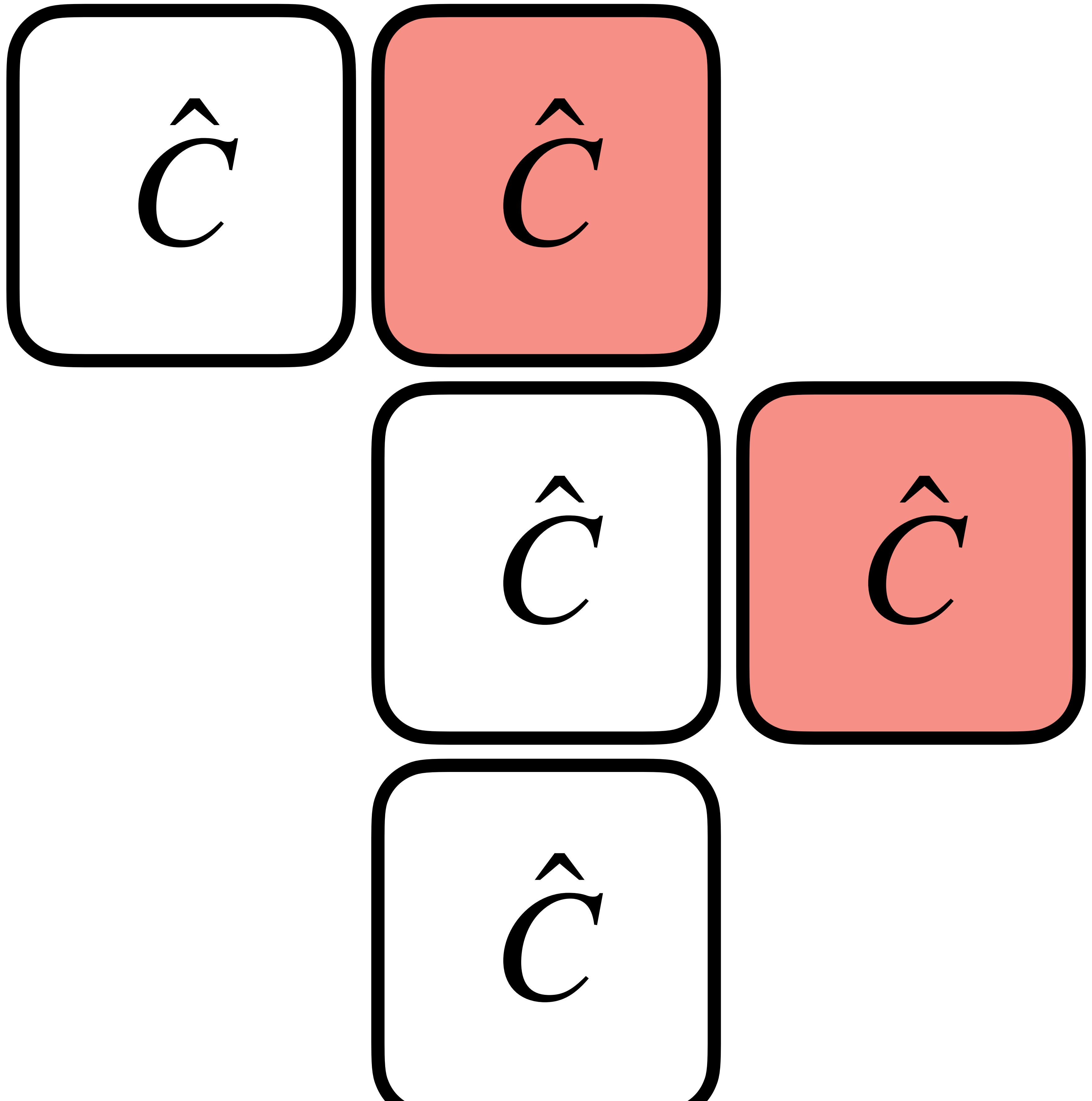
Cut and Choose



Garbler

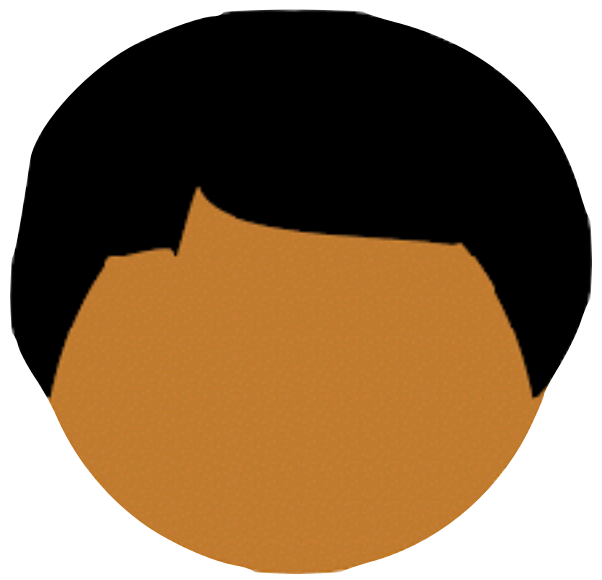


Evaluator

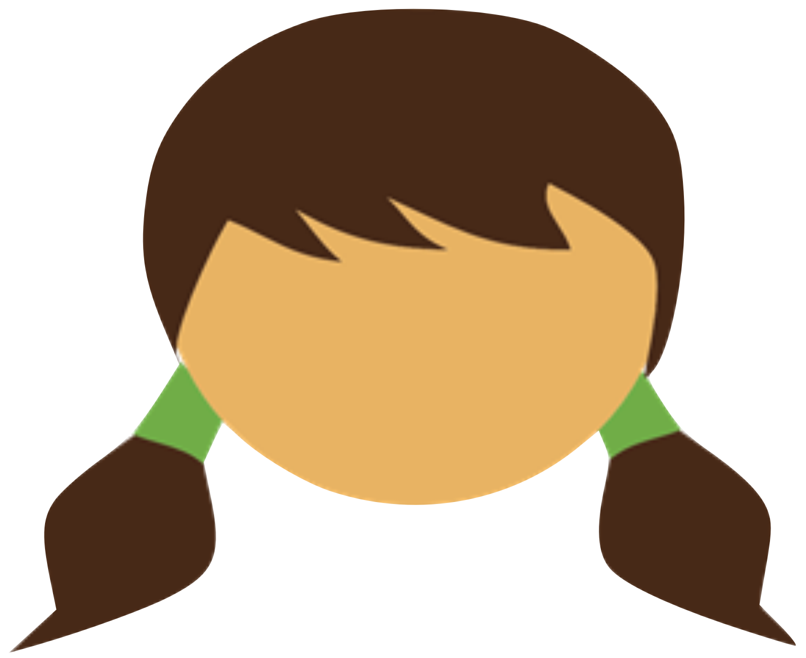


Parties evaluate remaining GCs, and E obtains outputs from each GC

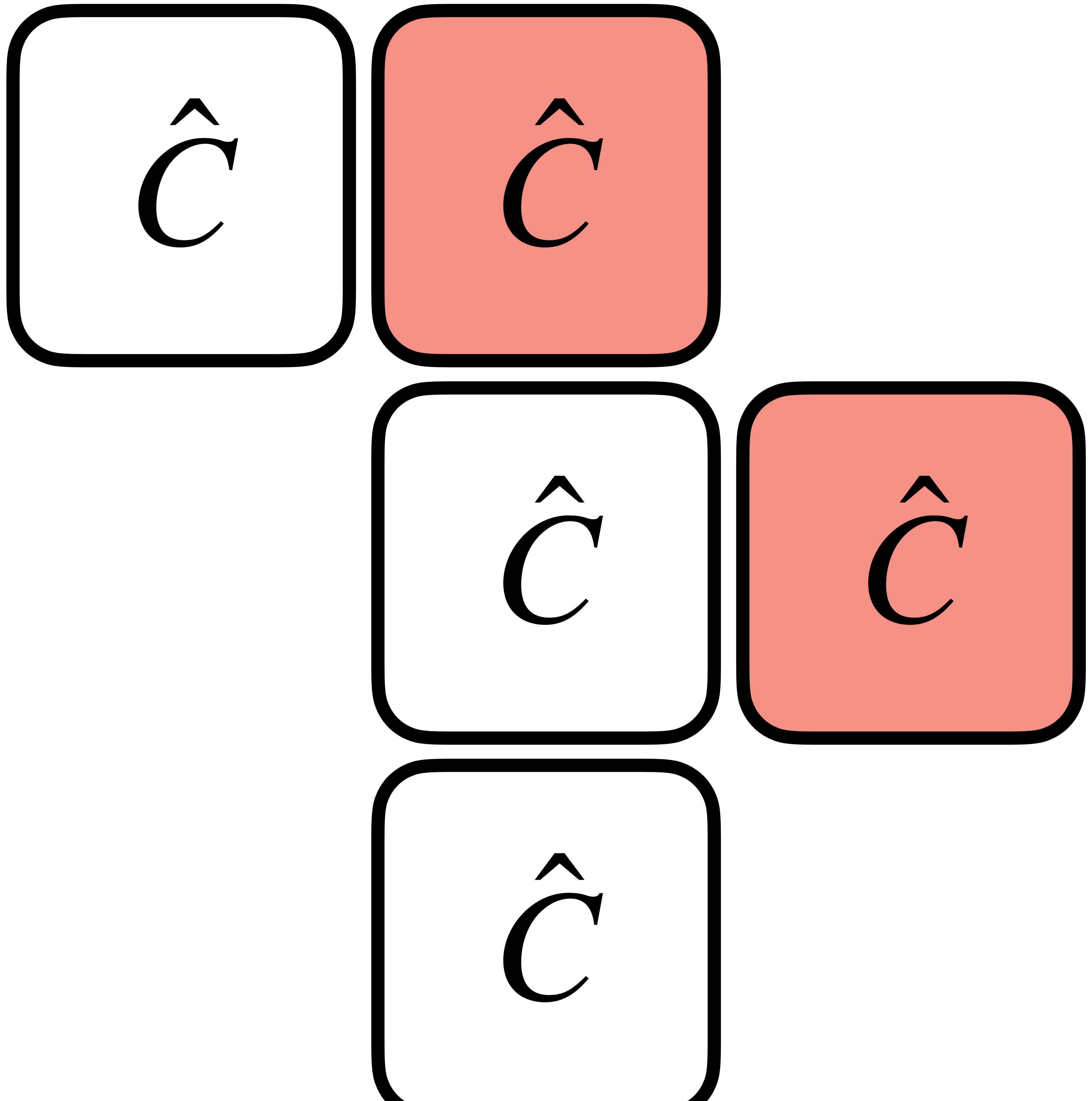
Cut and Choose



Garbler



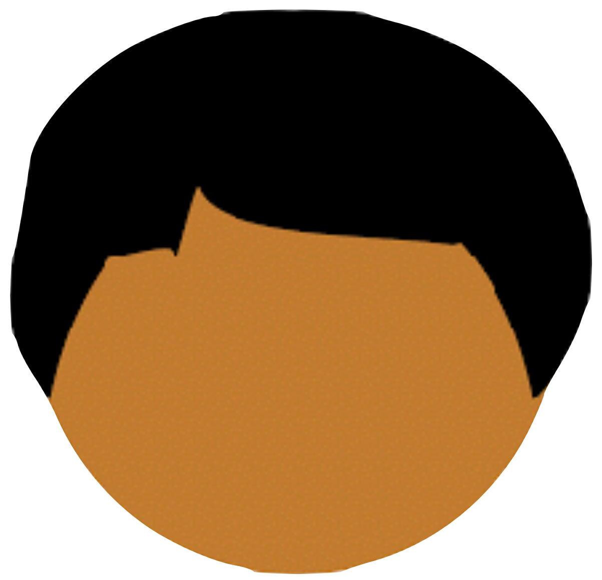
Evaluator



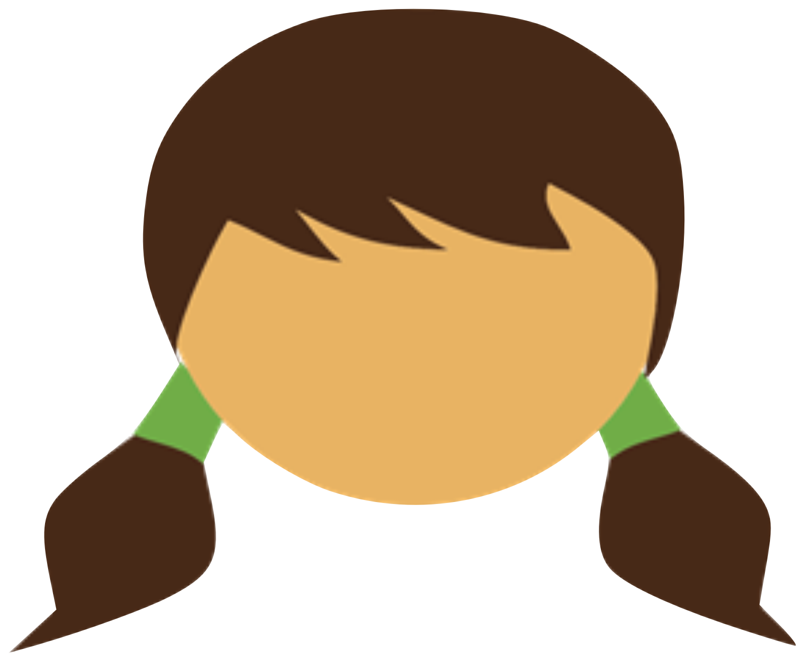
Parties evaluate remaining GCs, and E obtains outputs from each GC

Now what?

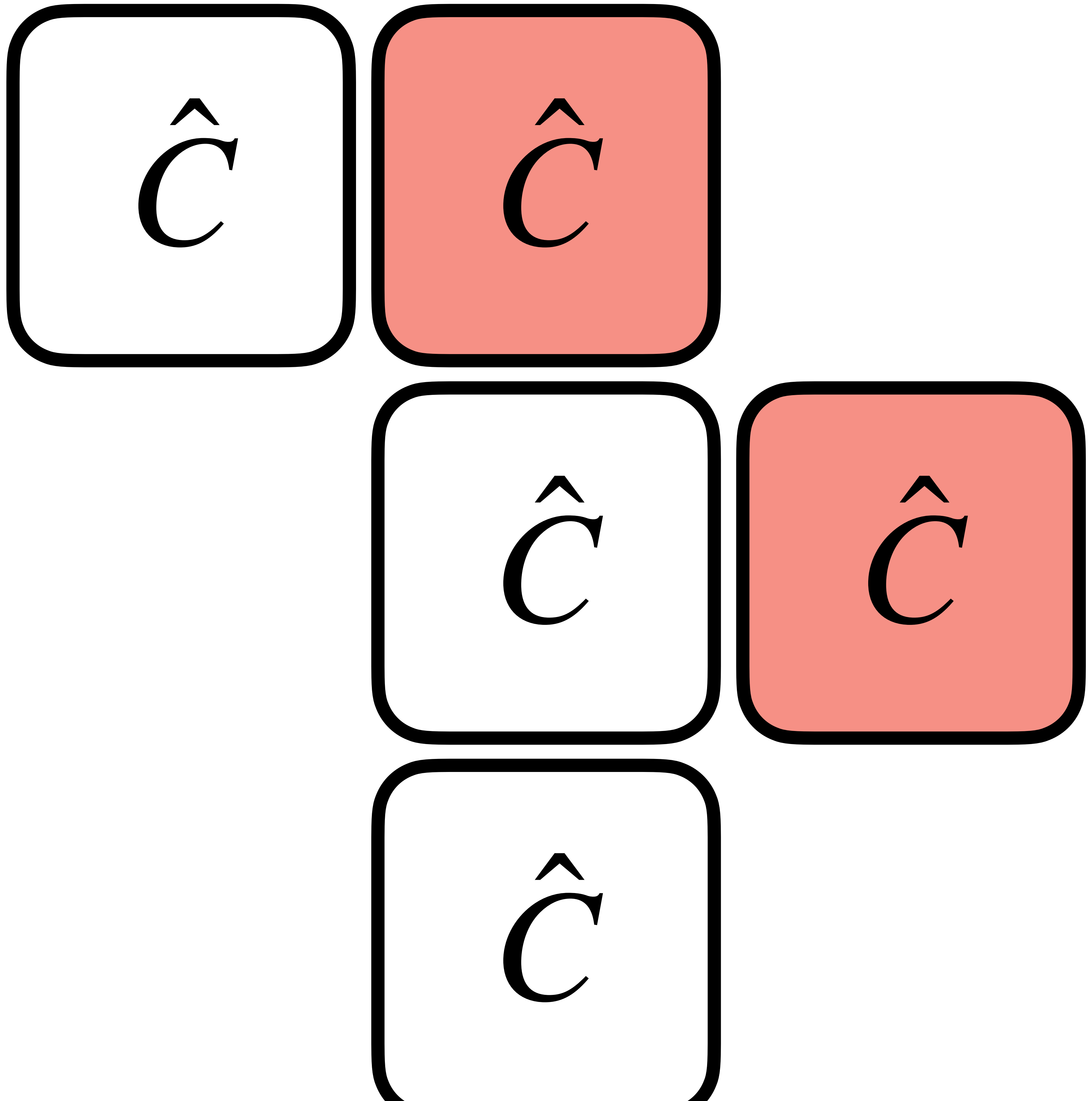
Cut and Choose



Garbler



Evaluator



Parties evaluate remaining GCs, and E obtains outputs from each GC

Now what?

Evaluator takes majority output

Authenticated Garbling and Efficient Maliciously Secure Two-Party Computation

Xiao Wang
University of Maryland
wangxiao@cs.umd.edu

Samuel Ranellucci
University of Maryland
George Mason University
samuel@umd.edu

Jonathan Katz
University of Maryland
jkatz@cs.umd.edu

Abstract

We propose a simple and efficient framework for obtaining efficient constant-round protocols for maliciously secure two-party computation. Our framework uses a function-independent preprocessing phase to generate authenticated information for the two parties; this information is then used to construct a *single* "authenticated" garbled circuit which is transmitted and evaluated.

We also show how to efficiently instantiate the preprocessing phase by designing a highly optimized version of the TinyOT protocol by Nielsen et al. Our overall protocol outperforms existing work in both the single-execution and amortized settings, with or without preprocessing:

- In the single-execution setting, our protocol evaluates an AES circuit with malicious security in 37 ms with an online time of just 1 ms. Previous work with the best online time (also 1 ms) requires 124 ms in total; previous work with the best total time requires 62 ms (with 14 ms online time).
- If we amortize the computation over 1024 executions, each AES computation requires just 6.7 ms with roughly the same online time as above. The best previous work in the amortized setting has roughly the same total time but does not support function-independent preprocessing.

Our work shows that the performance penalty for maliciously secure two-party computation (as compared to semi-honest security) is much smaller than previously believed.

1 Introduction

Protocols for secure two-party computation (2PC) allow two parties to compute an agreed-upon function of their inputs without revealing anything additional to each other. Although originally viewed as impractical, protocols for generic 2PC in the semi-honest setting based on Yao's garbled-circuit protocol [Yao86] have seen tremendous efficiency improvements over the past several years [MNPS04, HEKM11, ZRE15, KS08, KMR14, ALSZ13, BHKR13, PSSW09].

While these results are impressive, semi-honest security—which assumes that both parties follow the protocol honestly yet may try to learn additional information from the execution—is clearly not sufficient for all applications. This has motivated researchers to construct protocols achieving the stronger notion of *malicious* security. One popular approach for designing constant-round maliciously secure protocols is to apply the "cut-and-choose" technique [LP07, sS11, sS13, KSS12, LP11, HKE13, Lin13, Bra13, FJN14, AMPR14] to Yao's garbled-circuit protocol. For statistical security $2^{-\rho}$, the best approaches using this paradigm require ρ garbled circuits (which is optimal); the most efficient instantiation of this approach, by Wang et al. [WMK17], securely evaluates an AES circuit in 62 ms.

The cut-and-choose approach incurs significant overhead when large circuits are evaluated precisely because ρ garbled circuits need to be transmitted (typically, $\rho \geq 40$). In order to mitigate this, recent works have explored secure computation in an *amortized* setting where the same function is evaluated multiple times

Optimizing Authenticated Garbling for Faster Secure Two-Party Computation

Jonathan Katz
University of Maryland
jkatz@cs.umd.edu

Samuel Ranellucci
University of Maryland
George Mason University
samuel@umd.edu

Mike Rosulek
Oregon State University
rosulekm@eecs.oregonstate.edu

Xiao Wang
University of Maryland
wangxiao@cs.umd.edu

October 10, 2018

Authenticated Garbling from Simple Correlations

Samuel Dittmer[1][0000-0003-0018-6354], Yuval Ishai[2], Steve Lu[1][0000-0003-1837-8864], and Rafail Ostrovsky[1,3][0000-0002-1501-1330]

[1] Stealth Software Technologies, Inc.
[2] Technion - Israel Institute of Technology
[3] University of California, Los Angeles

Abstract. We revisit the problem of constant-round malicious secure two-party computation by considering the use of *simple correlations*, namely sources of correlated randomness that can be securely generated with sublinear communication complexity and good concrete efficiency. The current state-of-the-art protocol of Katz et al. (Crypto 2018) achieves malicious security by realizing a variant of the *authenticated garbling* functionality of Wang et al. (CCS 2017). Given oblivious transfer correlations, the communication cost of this protocol (with 40 bits of statistical security) is comparable to roughly 10 garbled circuits (GCs). This protocol inherently requires more than 2 rounds of interaction.
In this work, we use other kinds of simple correlations to realize the authenticated garbling functionality with better efficiency. Concretely, we get the following reduced costs in the random oracle model:
- Using variants of both vector oblivious linear evaluation (VOLE) and multiplication triples (MT), we reduce the cost to 1.31 GCs.
- Using only variants of VOLE, we reduce the cost to 2.25 GCs.
- Using only variants of MT, we obtain a *non-interactive* (i.e., 2-message) protocol with cost comparable to 8 GCs.

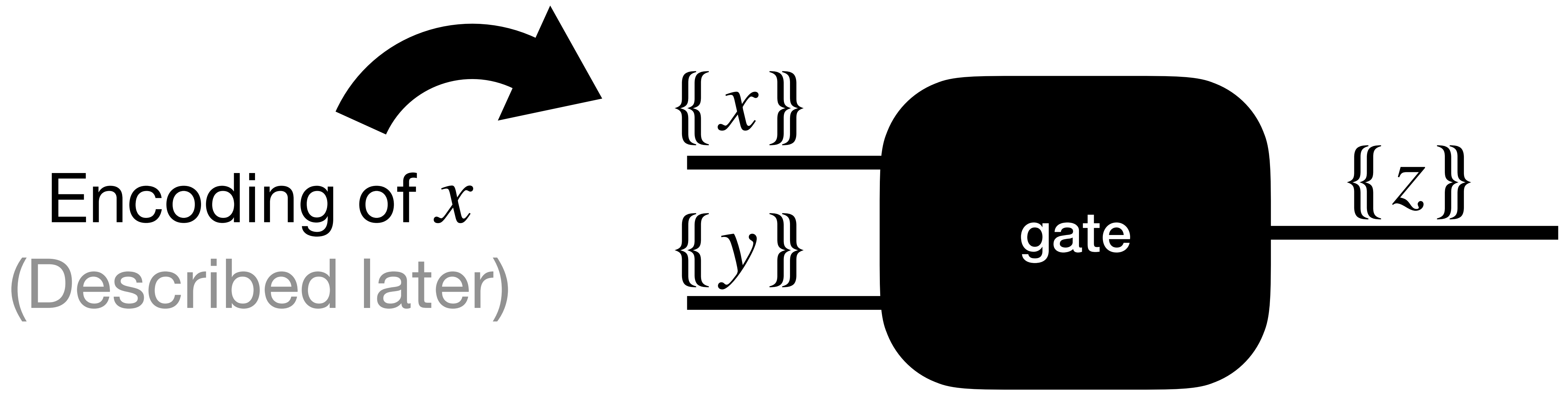
Finally, we show that by using recent constructions of pseudorandom correlation generators (Boyle et al., CCS 2018, Crypto 2019, 2020), the simple correlations consumed by our protocols can be securely realized without forming an efficiency bottleneck.

1 Introduction

Authenticated Garbling

Crucial Insight: use information-theoretic MACs on each wire so that GC can reveal internal values to E.

E can tell if a the revealed value is corrupted.



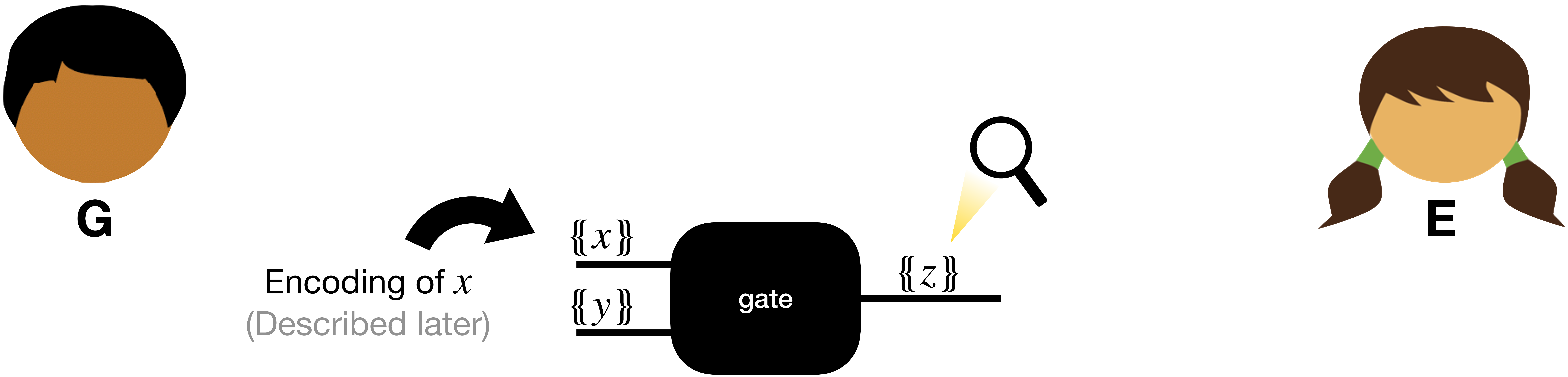
Authenticated Garbling

Just like classic GC, gate-by-gate evaluation in constant rounds

However, the technique prevents G from cheating

Authenticated Garbling

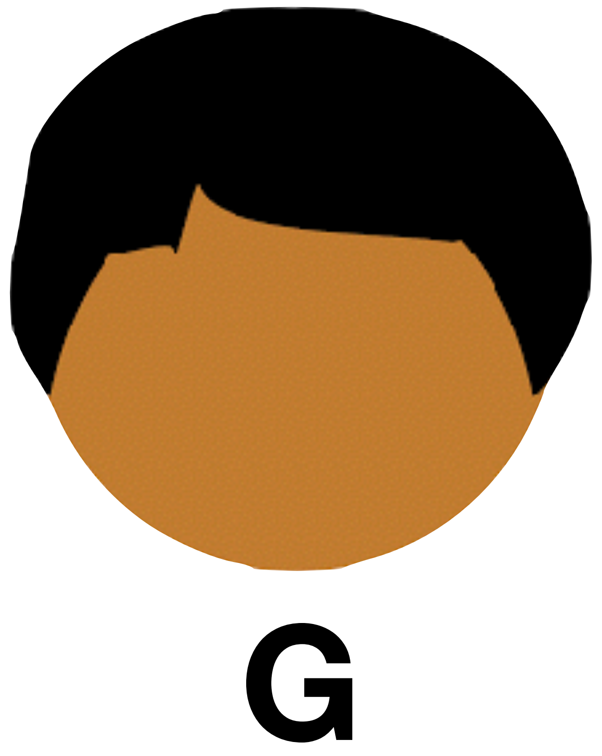
Crucial Insight: add a mechanism by GC can reveal internal values to E.
E can tell if a the revealed value is corrupted.

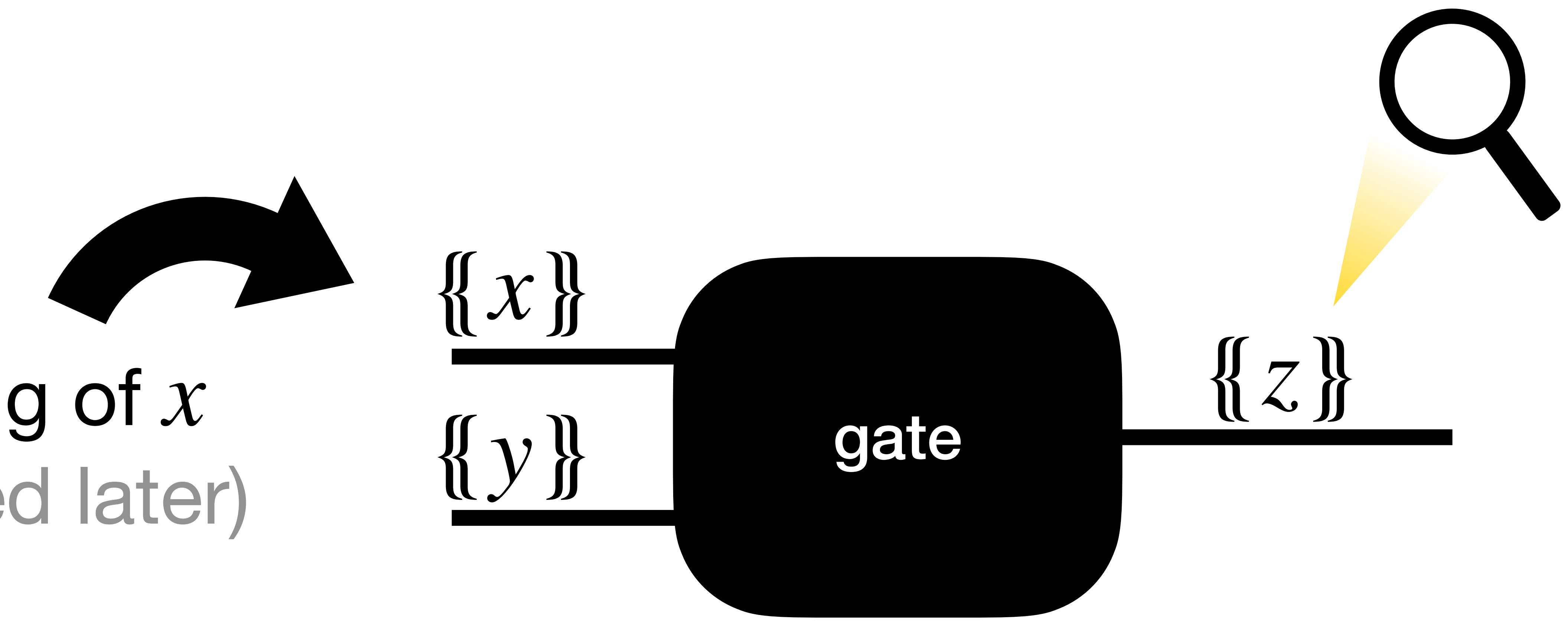


Authenticated Garbling

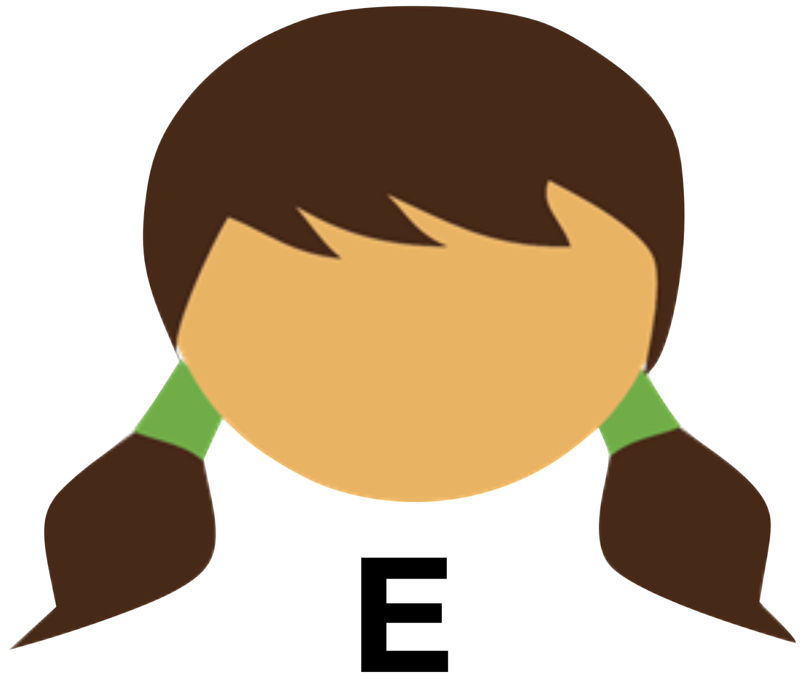
Crucial Insight: add a mechanism by GC can reveal internal values to E.

E can tell if a the revealed value is corrupted.





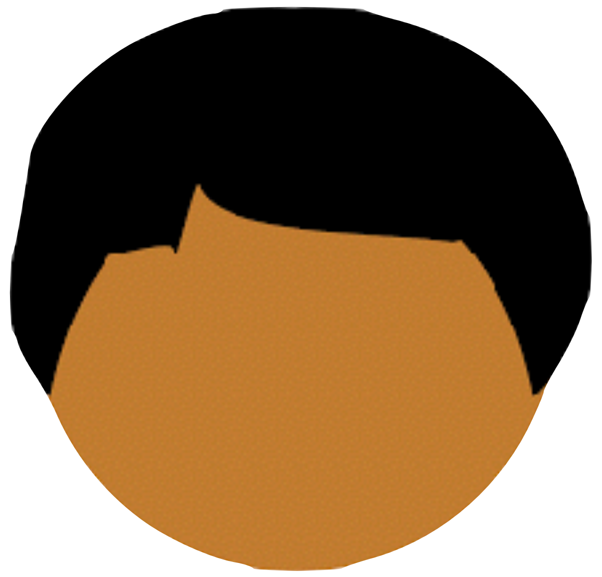
Encoding of x
(Described later)



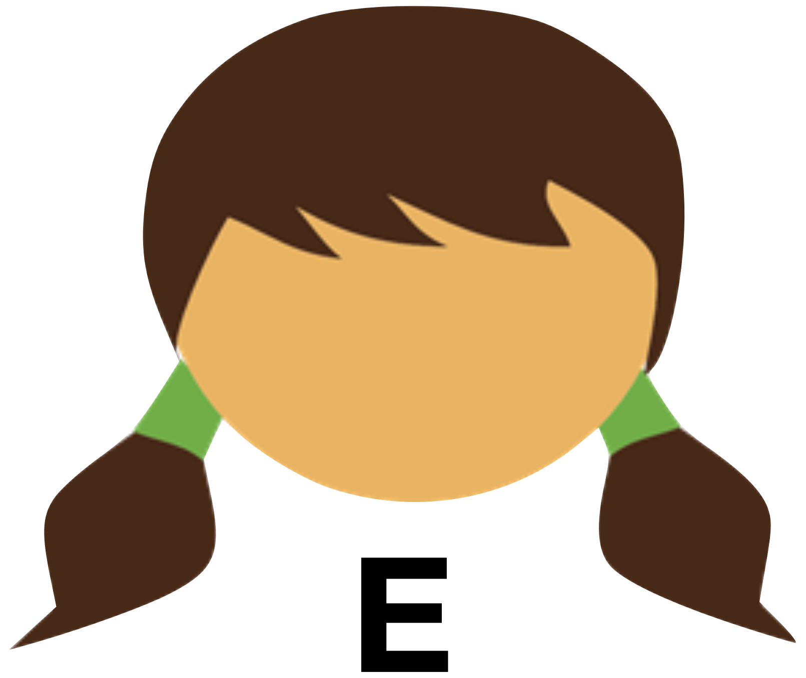
If G tries to corrupt the GC, then E will notice z is ill-formed with overwhelming probability

Authenticated Garbling

$\llbracket x \rrbracket$



G



E

Authenticated Garbling

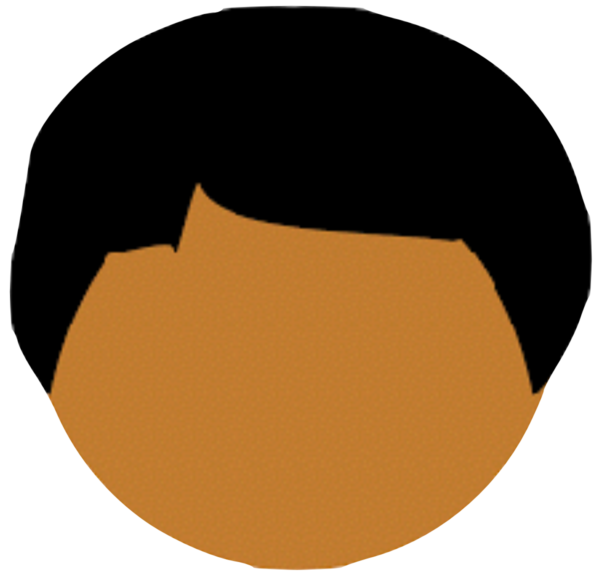
Key

$\llbracket x \rrbracket$

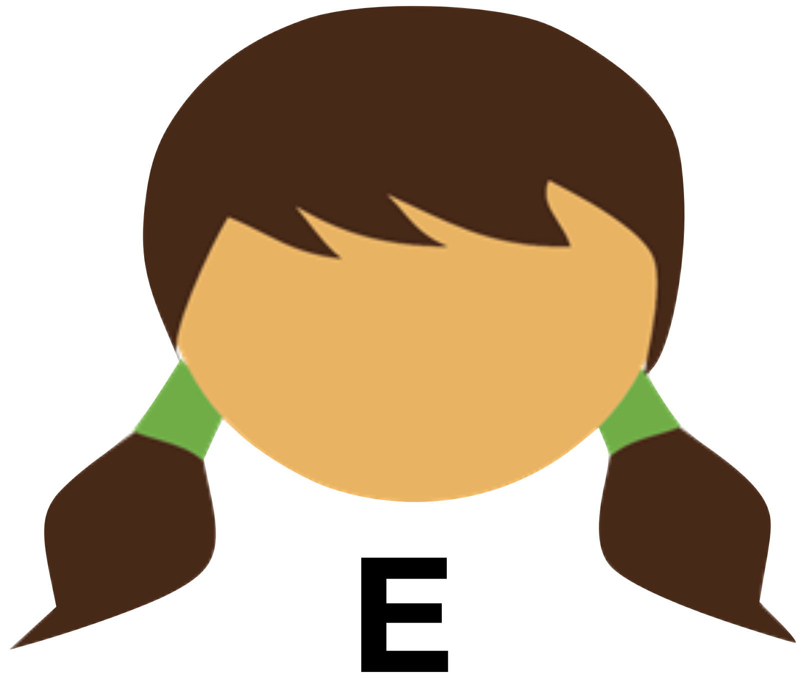
Authenticator

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$



G



E

Authenticated Garbling

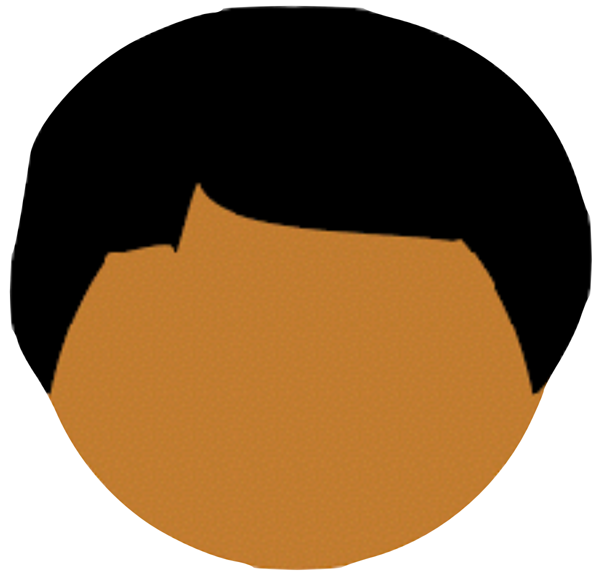
Key

$\llbracket x \rrbracket$

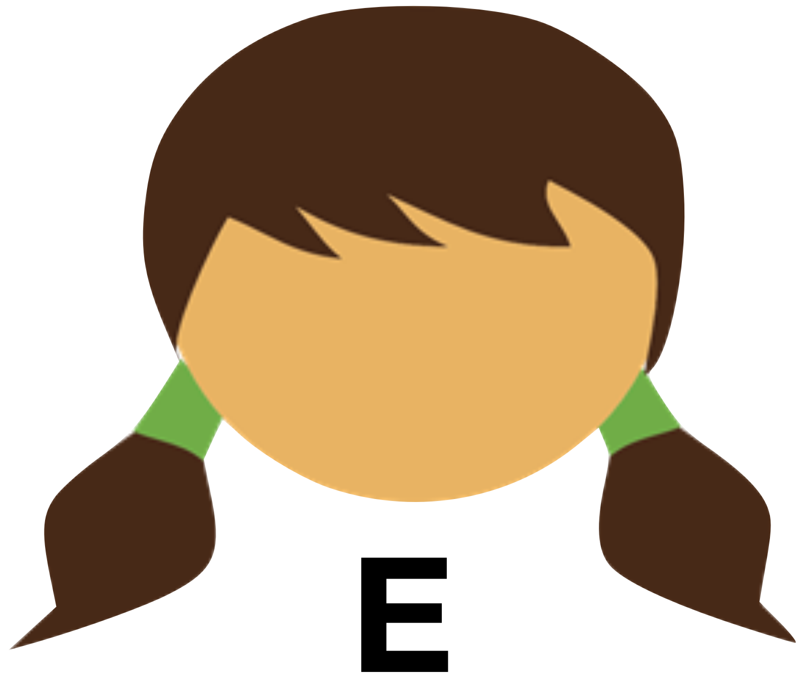
Authenticator

$$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$$

$$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$$



G



E

$$\llbracket x \rrbracket = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$

Authenticated Garbling

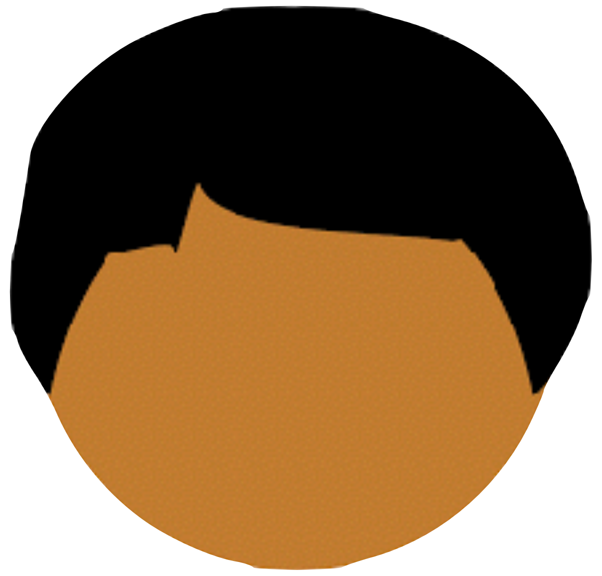
$\llbracket x \rrbracket$ ______

Key

$$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$$

Authenticator

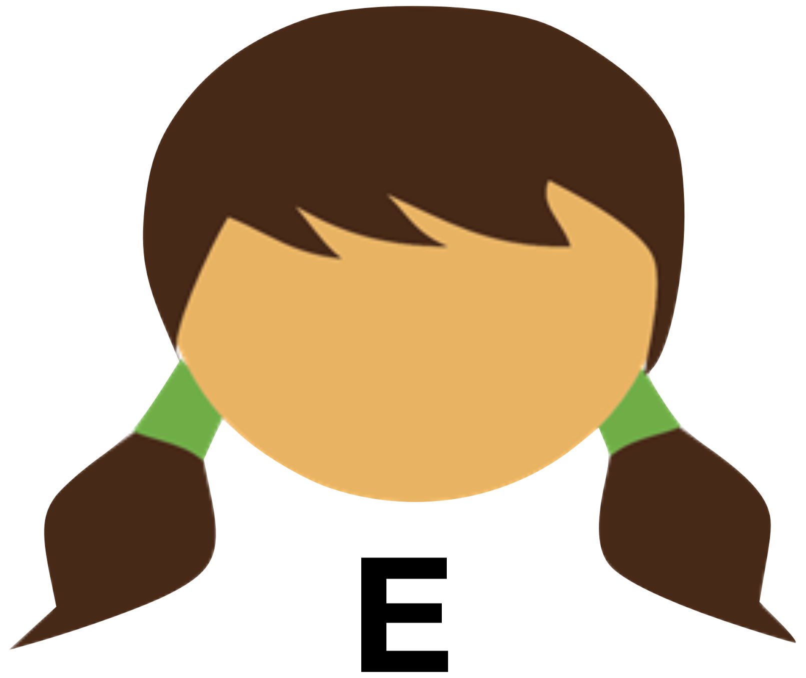
$$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$$



G

$$X \in \{0,1\}^{\lambda+\sigma+1}$$

$$\llbracket x \rrbracket = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$



E

Authenticated Garbling

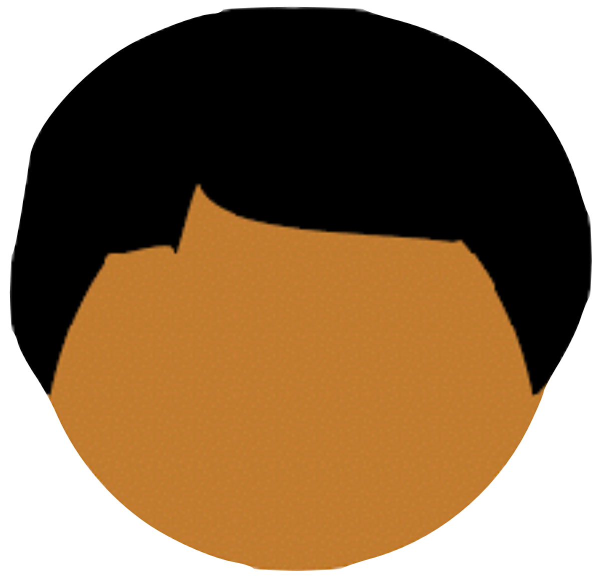
Key

$\llbracket x \rrbracket$ ______

Authenticator

$$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$$

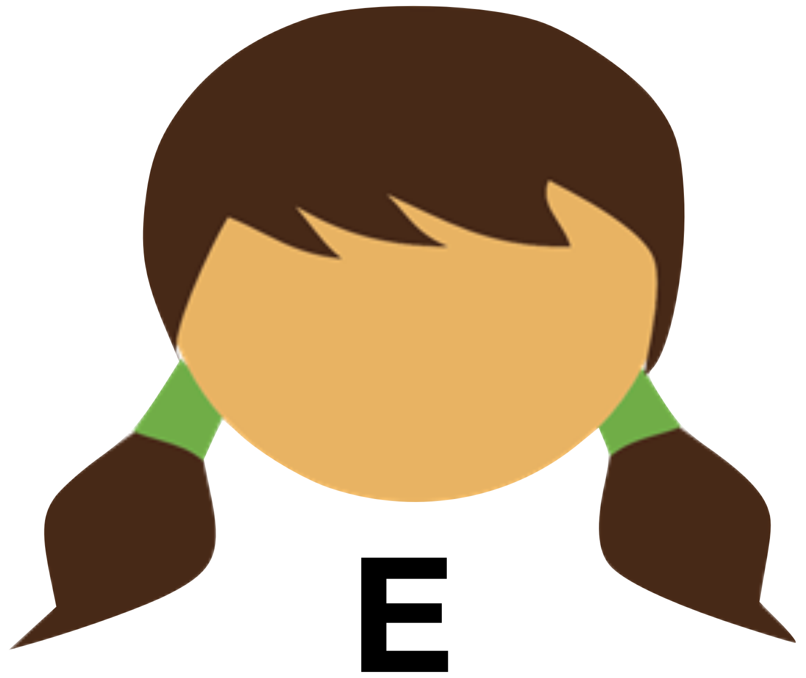
$$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$$



G

$$X \in \{0,1\}^{\lambda+\sigma+1}$$

$$\llbracket x \rrbracket = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$



E

Key part

Authenticator part

Value part

Authenticated Garbling

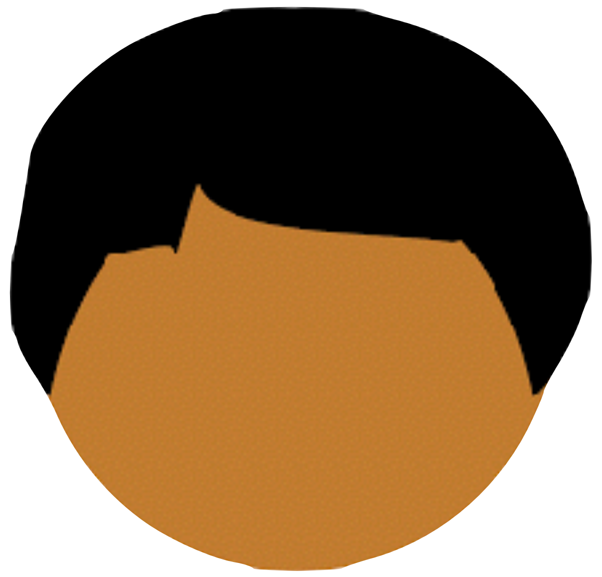
$\llbracket x \rrbracket$

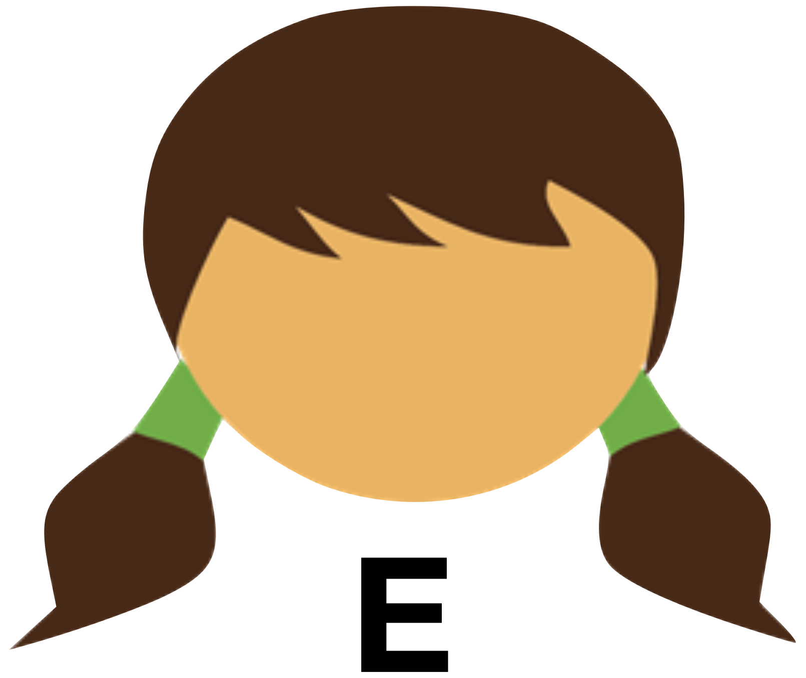
Key

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$

Authenticator

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$


G


E

$X \in \{0,1\}^{\lambda+\sigma+1}$

$$\llbracket x \rrbracket = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$

Secret-share of parts

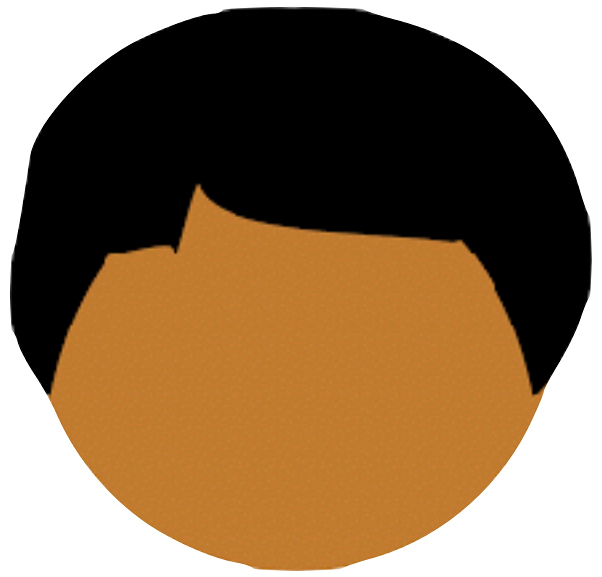
Key part

Authenticator part

Value part

Authenticated Garbling

$$\{\!\{x\}\!\} = [x \cdot \Delta, x \cdot \mu, x]$$



G

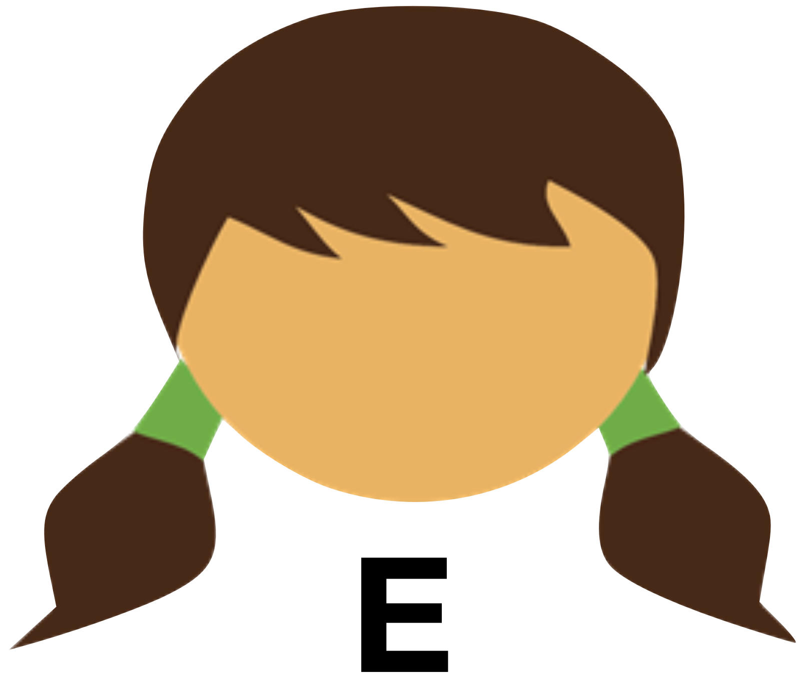
$$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$$

$$\{\!\{x\}\!\} = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$

Key part

Authenticator part

Value part



E

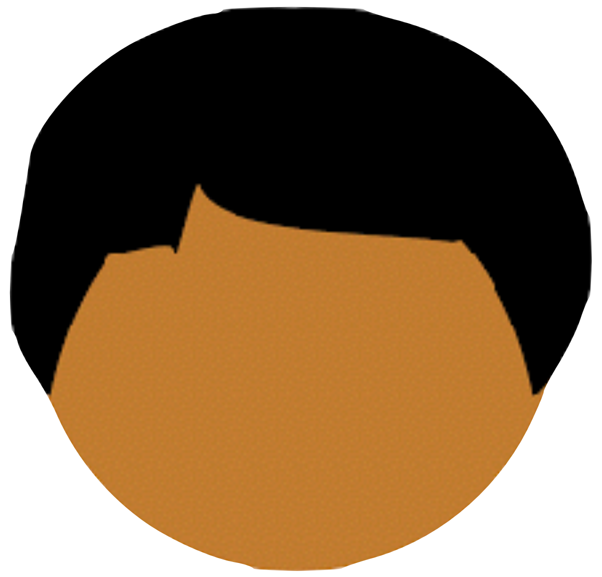
$$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$$

Authenticated Garbling

$$\{\!\{x\}\!\} = [x \cdot \Delta, x \cdot \mu, x]$$

open authenticator, value

$x \cdot \mu, x$



G

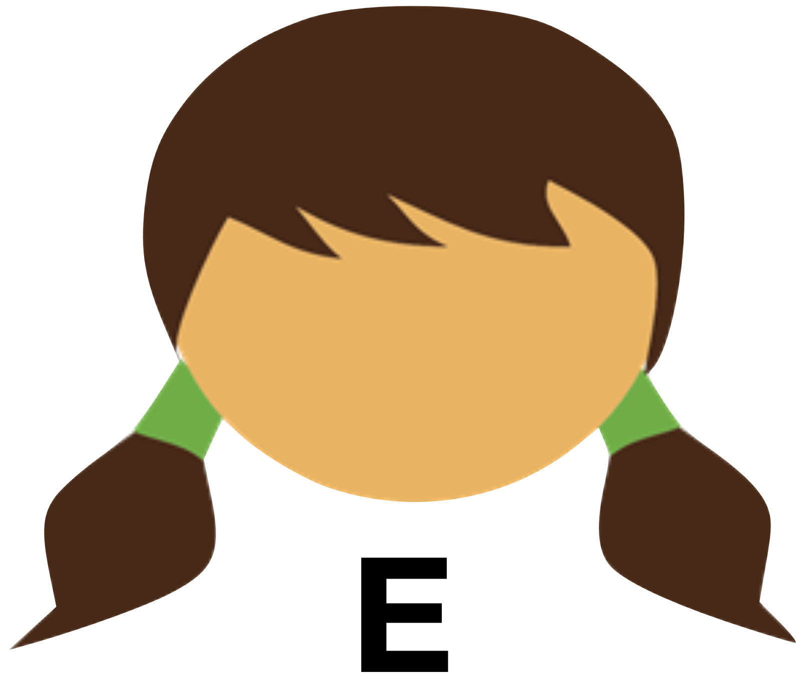
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$

$$\{\!\{x\}\!\} = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$

Key part

Authenticator part

Value part



E

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

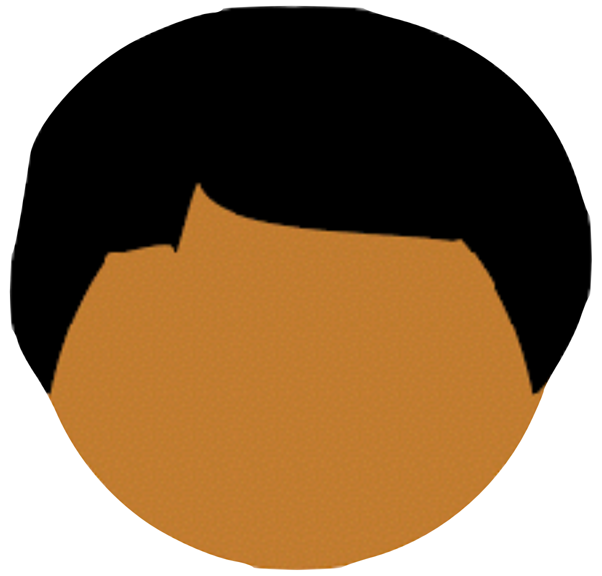
Authenticated Garbling

$\{\!\{x\}\!\} = [x \cdot \Delta, x \cdot \mu, x]$

open authenticator, value

$x \cdot \mu, x$

G cannot flip bit, because
G does not know μ



G

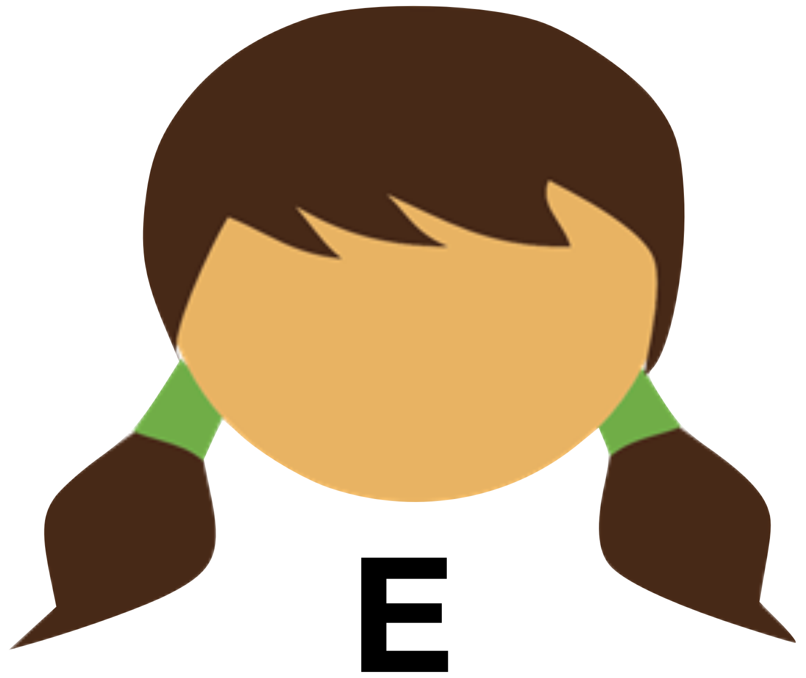
$\Delta \stackrel{\$}{\leftarrow} \{0,1\}^{\lambda}$

$$\{\!\{x\}\!\} = \left\langle X, \begin{cases} X & \text{if } x = 0 \\ X \oplus (\Delta, \mu, 1) & \text{if } x = 1 \end{cases} \right\rangle$$

Key part

Authenticator part

Value part

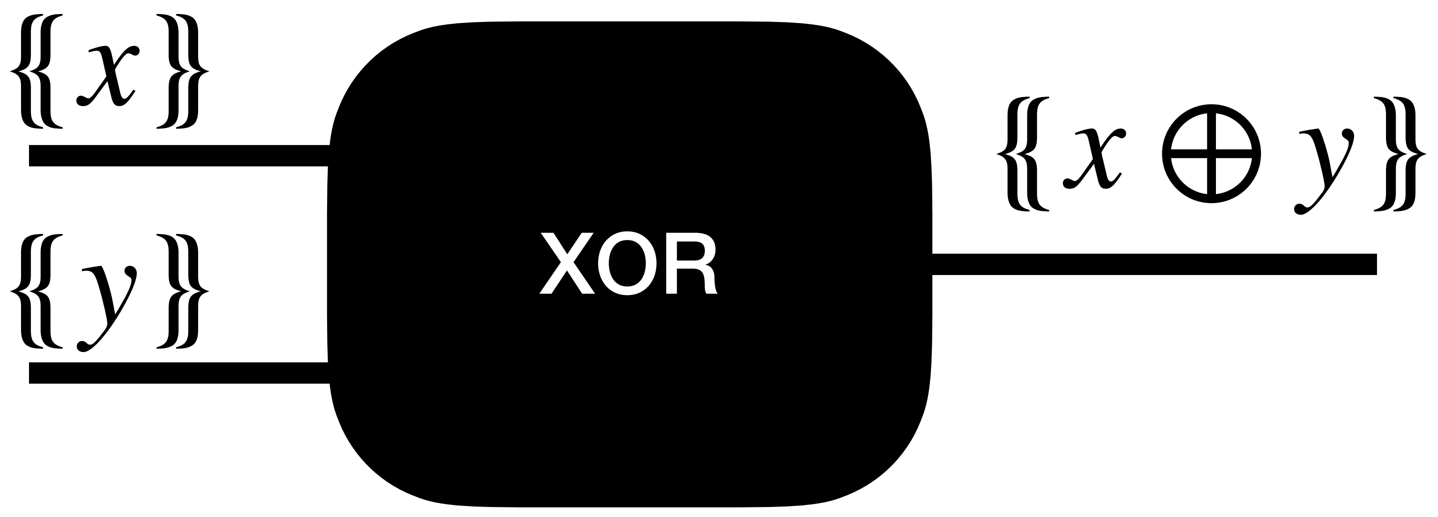


E

$\mu \stackrel{\$}{\leftarrow} \{0,1\}^{\sigma}$

Authenticated Garbling

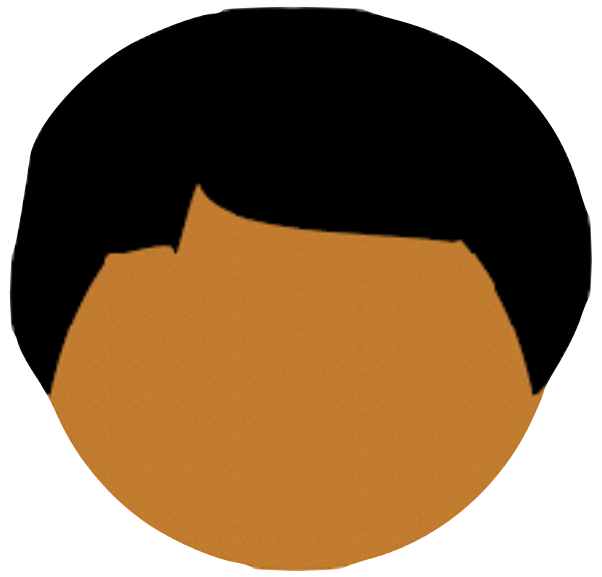
XOR gates are "free"



$$\{\!\{x\}\!\} = [x \cdot \Delta, x \cdot \mu, x]$$

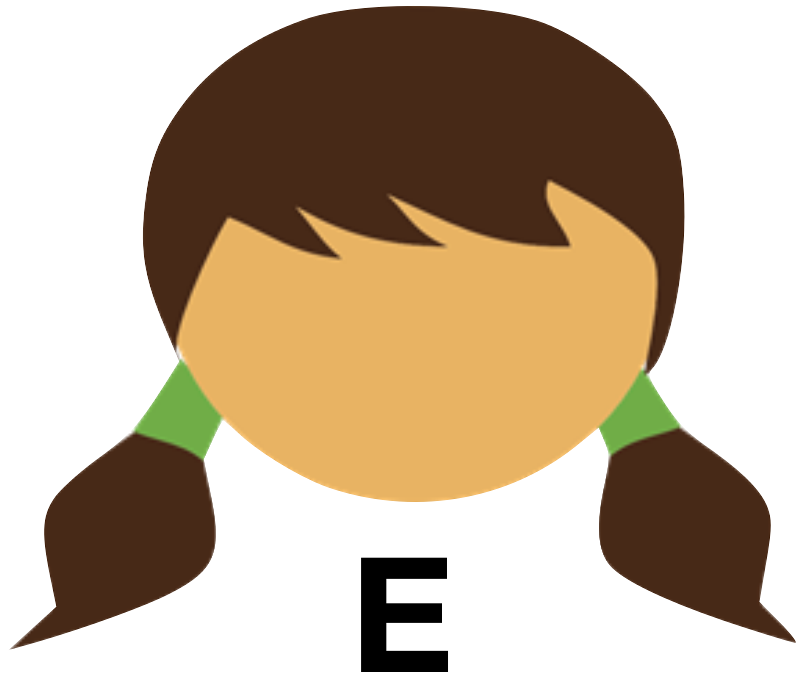
$$\{\!\{y\}\!\} = [y \cdot \Delta, y \cdot \mu, y]$$

$$\{\!\{x \oplus y\}\!\} = [(x \oplus y) \cdot \Delta, (x \oplus y) \cdot \mu, (x \oplus y)]$$



G

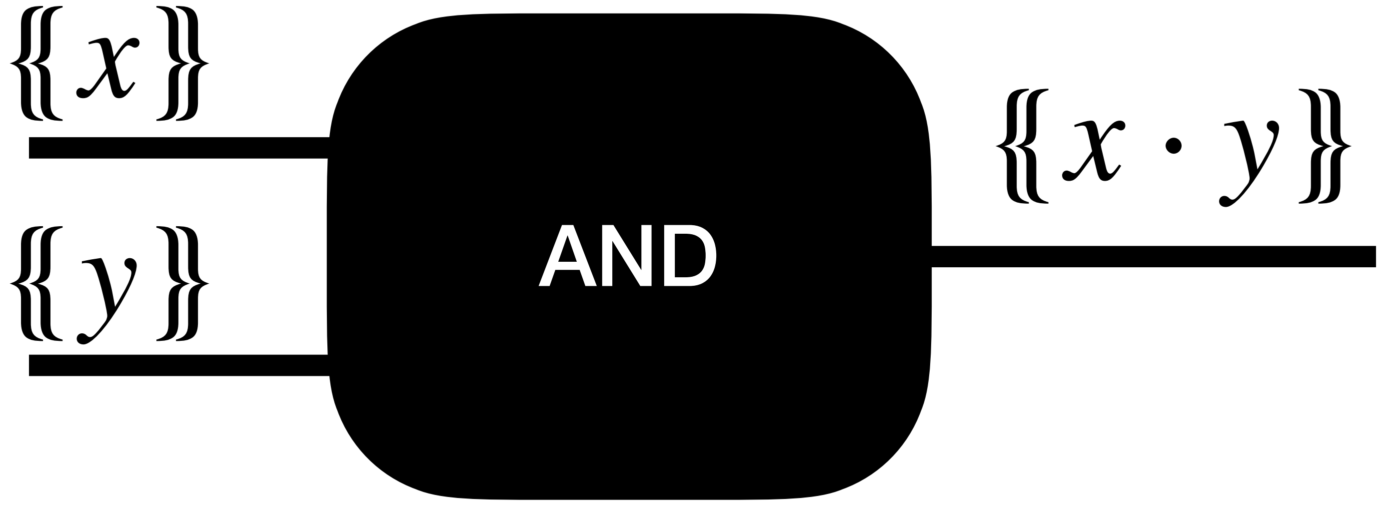
$\Delta \xleftarrow{\$} \{0,1\}^\lambda$

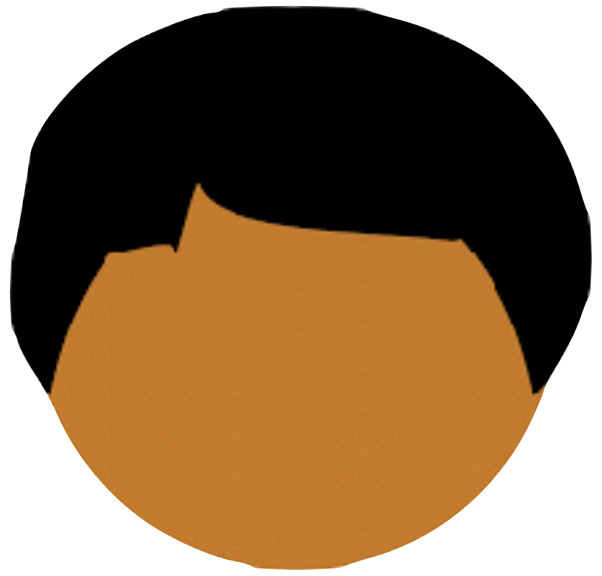


E

$\mu \xleftarrow{\$} \{0,1\}^\sigma$

Authenticated Garbling





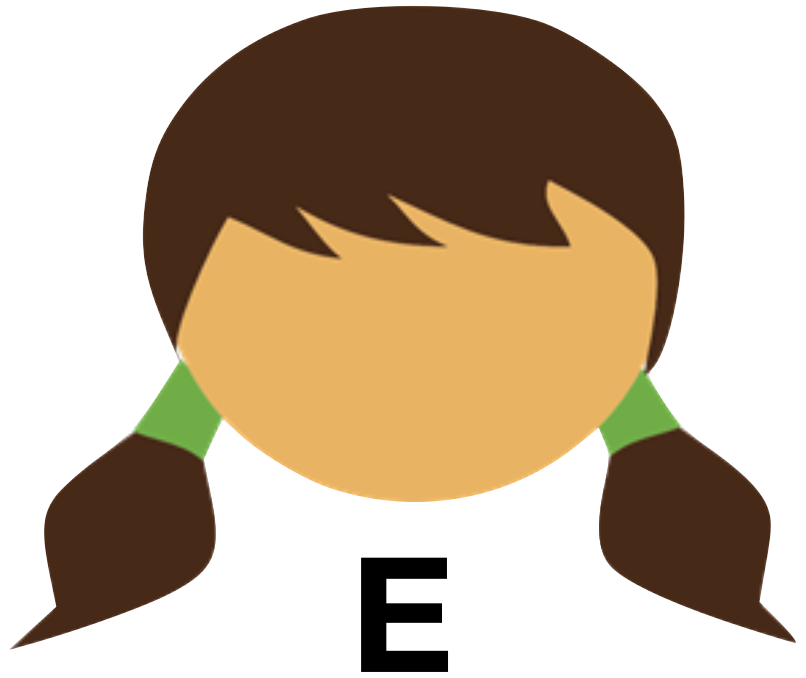
G

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$

Suppose G and E have access to a **doubly authenticated multiplication triple**

$$\llbracket \alpha \rrbracket, \llbracket \beta \rrbracket, \llbracket \alpha \cdot \beta \rrbracket$$

where $\alpha, \beta \xleftarrow{\$} \{0,1\}$



E

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

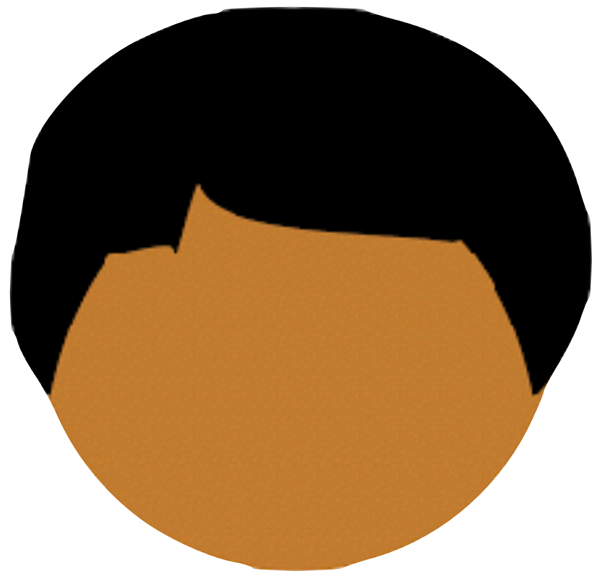
Authenticated Garbling


Two Halves Make a Whole
Reducing Data Transfer in Garbled Circuits using Half Gates
Samee Zahur, Mike Rosulek, and David Evans

$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

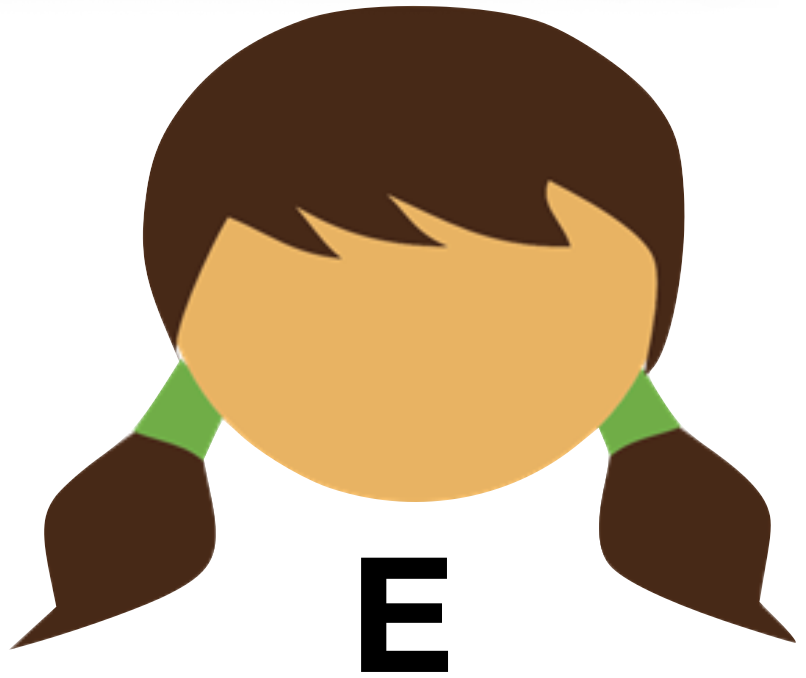
where $\alpha, \beta \overset{\$}{\leftarrow} \{0,1\}$

Observe: $(x \oplus \alpha) \cdot y \oplus (y \oplus \beta) \cdot \alpha \oplus \alpha \cdot \beta = x \cdot y$



G

$\Delta \overset{\$}{\leftarrow} \{0,1\}^{\lambda}$



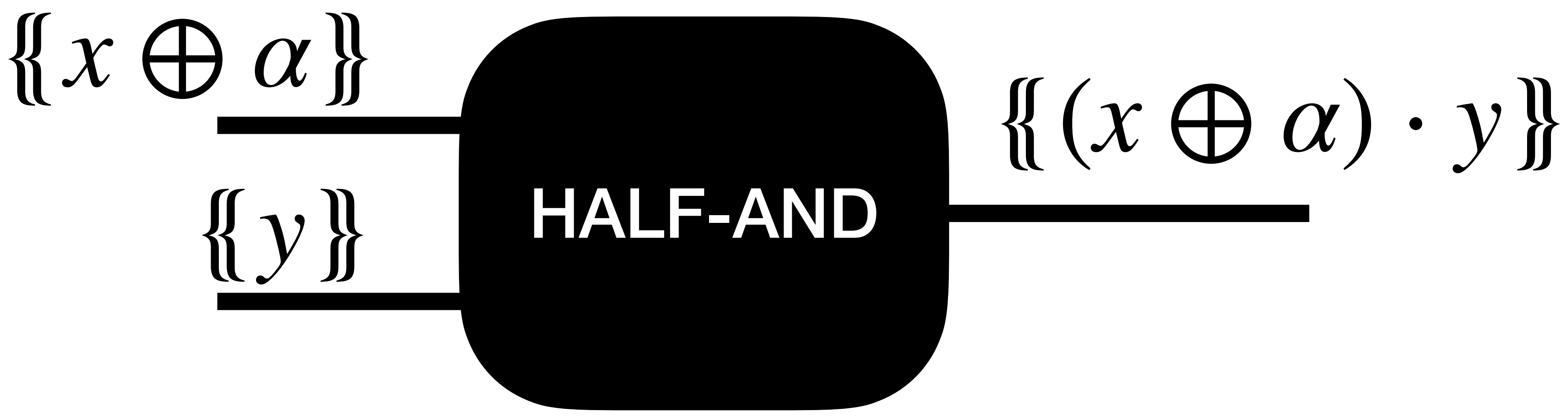
E

$\mu \overset{\$}{\leftarrow} \{0,1\}^{\sigma}$

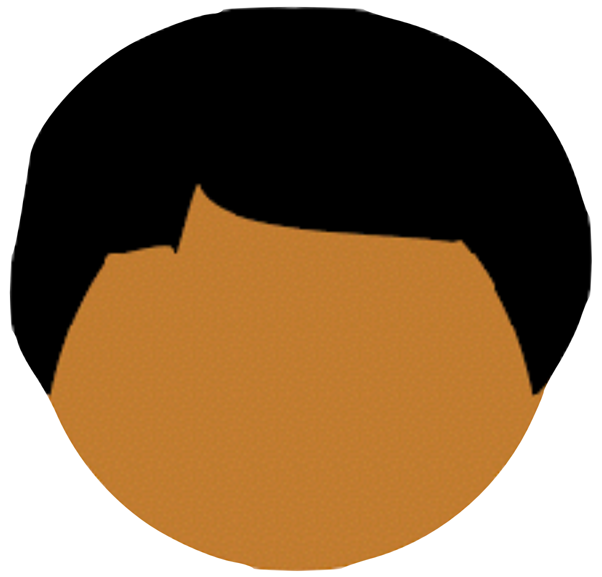
Authenticated Garbling

$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

where $\alpha, \beta \xleftarrow{\$} \{0,1\}$

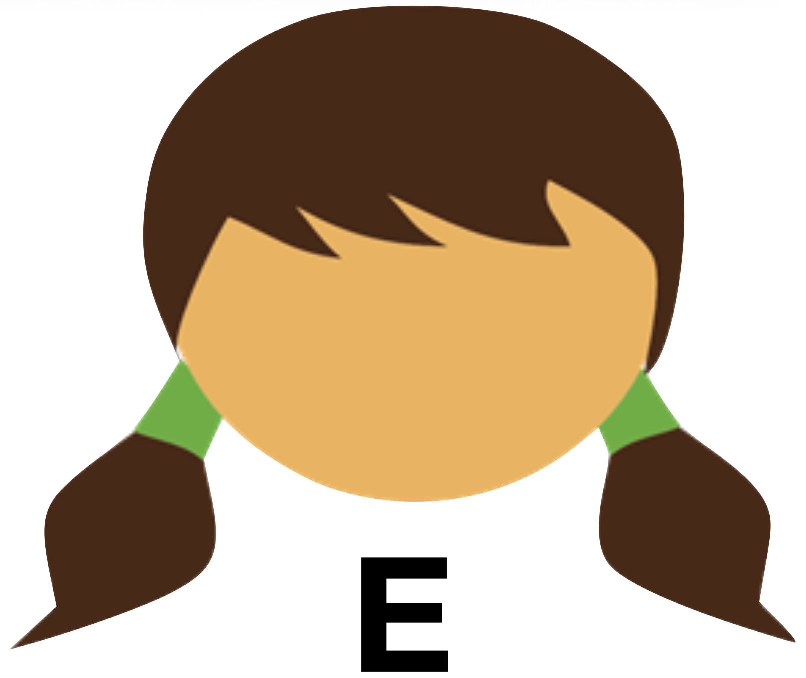



Two Halves Make a Whole
Reducing Data Transfer in Garbled Circuits using Half Gates
Samee Zahur, Mike Rosulek, and David Evans



G

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$



E

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

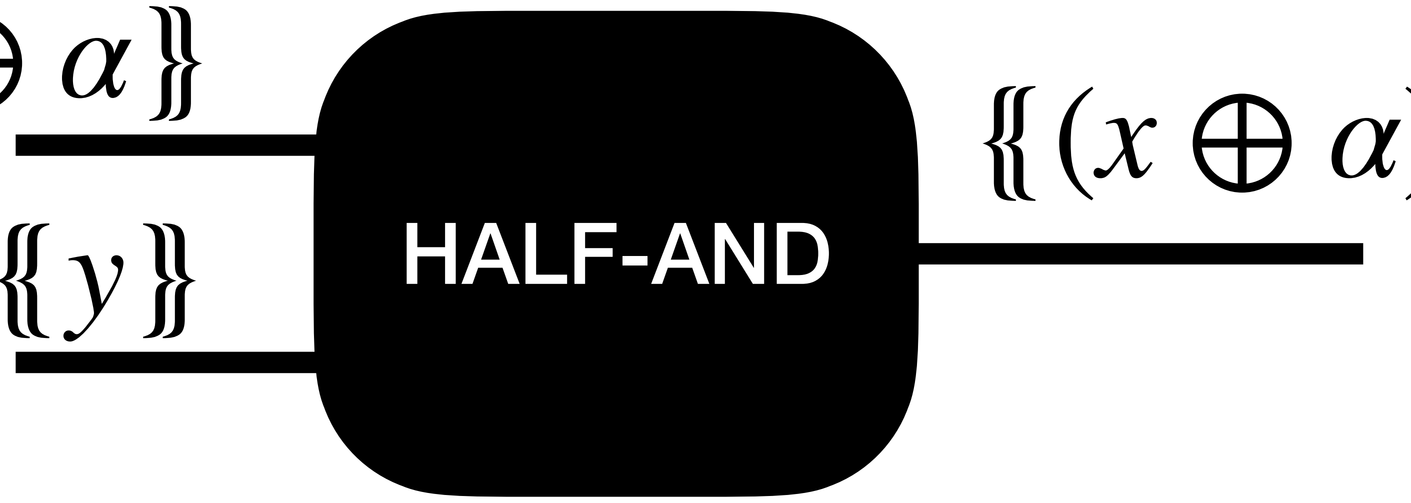
Authenticated Garbling

$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

where $\alpha, \beta \xleftarrow{\$} \{0,1\}$

$\{\!\{x \oplus \alpha\}\!\}$

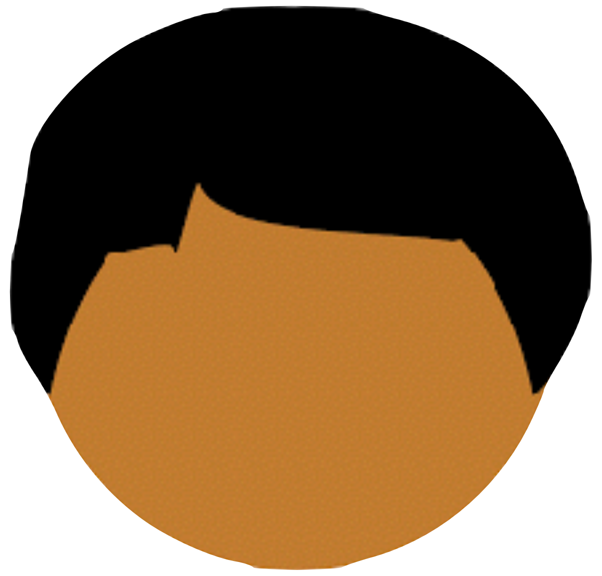
$\{\!\{y\}\!\}$



$\{\!\{(x \oplus \alpha) \cdot y\}\!\}$

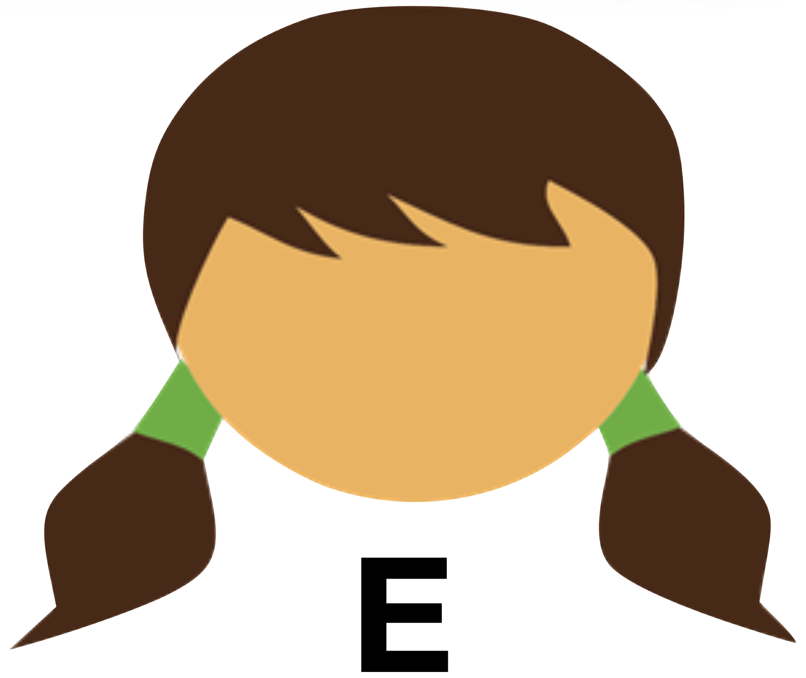
$$\langle X, X \oplus x \cdot \Delta \rangle = \text{keyPart}(\{\!\{x \oplus \alpha\}\!\})$$

$$\langle Y, Y \oplus y \cdot (\Delta, \mu, 1) \rangle = \{\!\{y\}\!\}$$



G

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$



E

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

$x \oplus \alpha$

Authenticated Garbling


In 34th Eurocrypt, Sofia, Bulgaria, April 2015

Two Halves Make a Whole

Reducing Data Transfer in Garbled Circuits using Half Gates

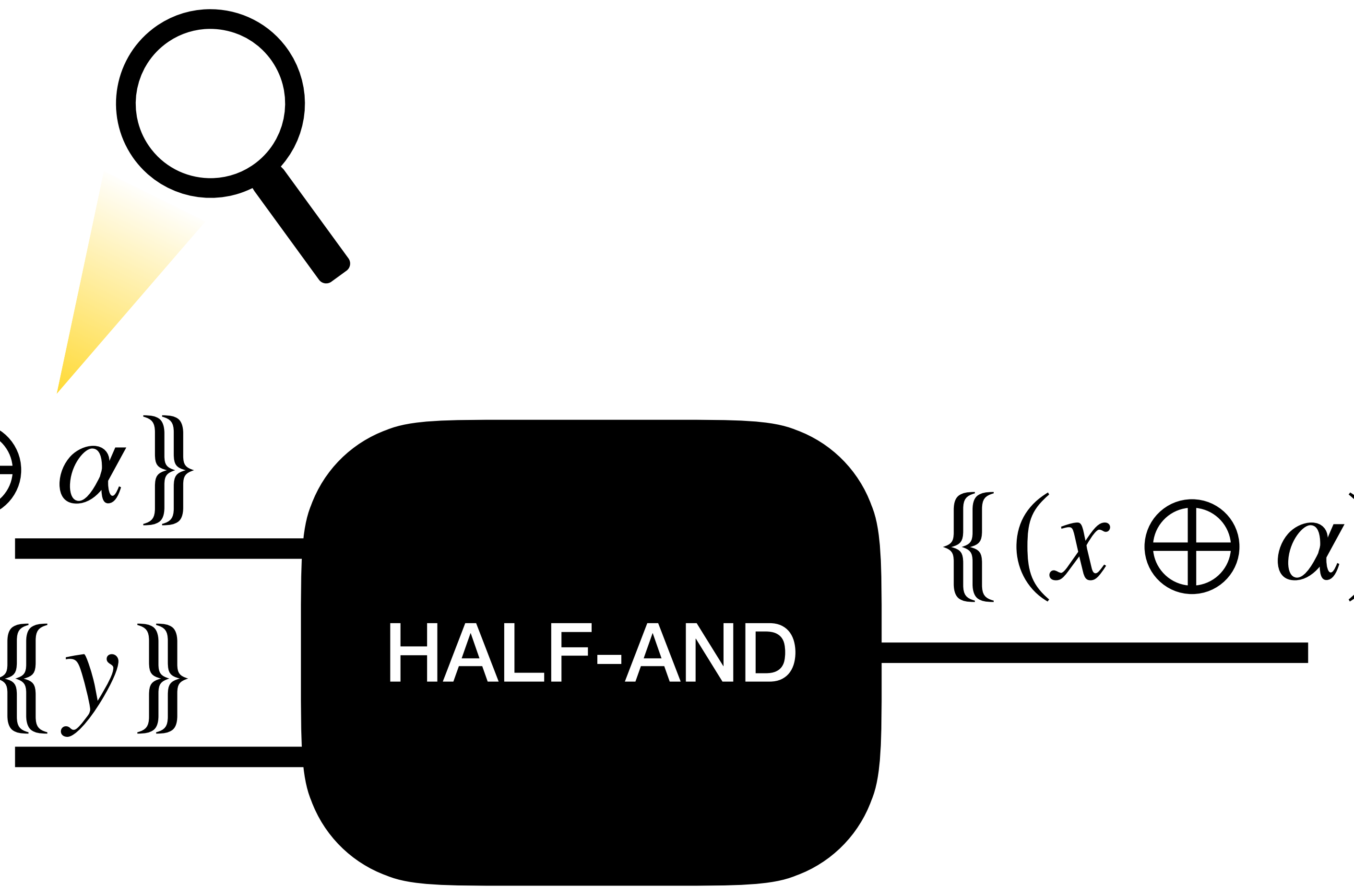
Samee Zahur, Mike Rosulek, and David Evans

University of Virginia

Oregon State University

$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

where $\alpha, \beta \stackrel{\$}{\leftarrow} \{0,1\}$



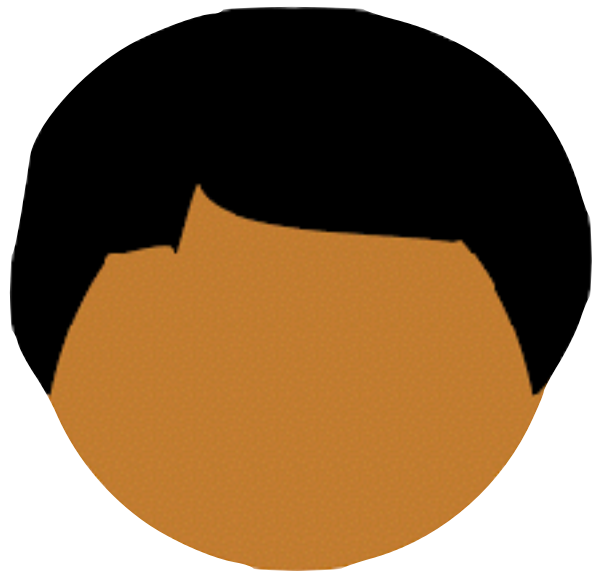
$\{\!\{x \oplus \alpha\}\!\}$

$\{\!\{y\}\!\}$

$\{\!\{(x \oplus \alpha) \cdot y\}\!\}$

$$\langle X, X \oplus x \cdot \Delta \rangle = \text{keyPart}(\{\!\{x \oplus \alpha\}\!\})$$

$$\langle Y, Y \oplus y \cdot (\Delta, \mu, 1) \rangle = \{\!\{y\}\!\}$$

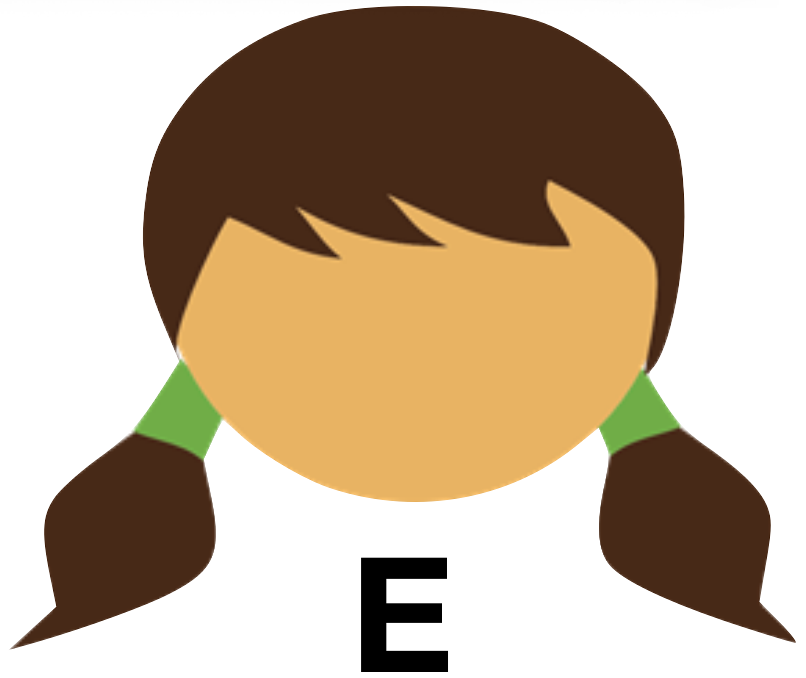


G

$\Delta \stackrel{\$}{\leftarrow} \{0,1\}^{\lambda}$

$\text{Enc}(X, Z)$

$\text{Enc}(X \oplus \Delta, Y \oplus Z)$



E

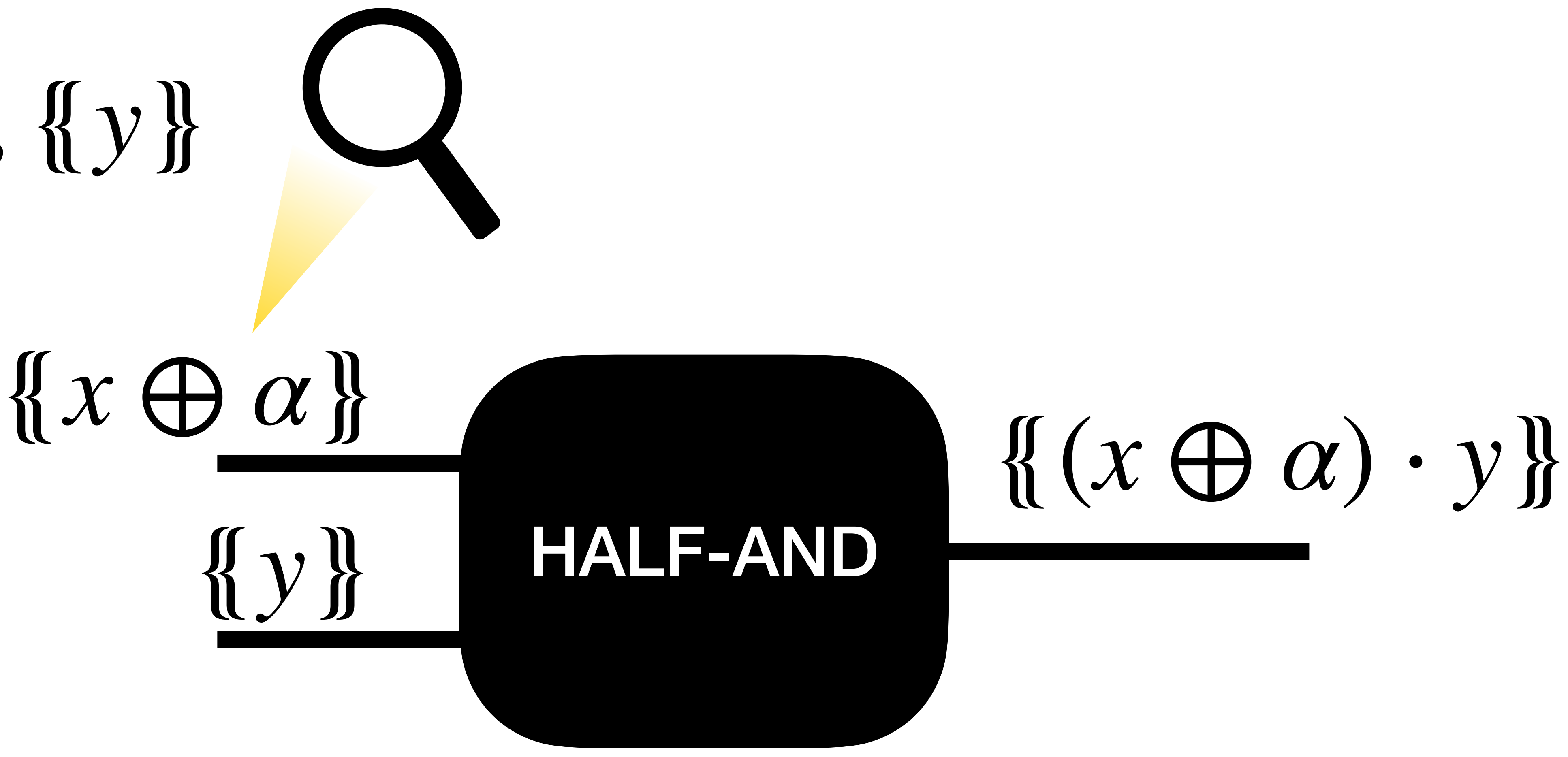
$\mu \stackrel{\$}{\leftarrow} \{0,1\}^{\sigma}$

$x \oplus \alpha$

Authenticated Garbling

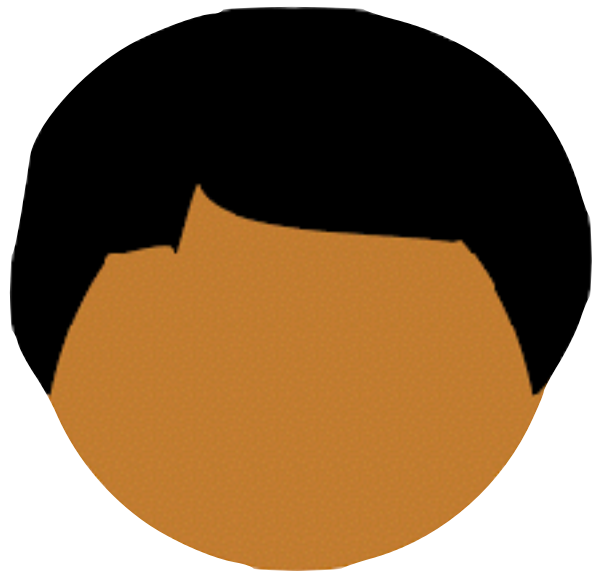
$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

where $\alpha, \beta \xleftarrow{\$} \{0,1\}$



$$\langle X, X \oplus x \cdot \Delta \rangle = \mathrm{keyPart}(\{\!\{x \oplus \alpha\}\!\})$$

$$\langle Y, Y \oplus y \cdot (\Delta, \mu, 1) \rangle = \{\!\{y\}\!\}$$



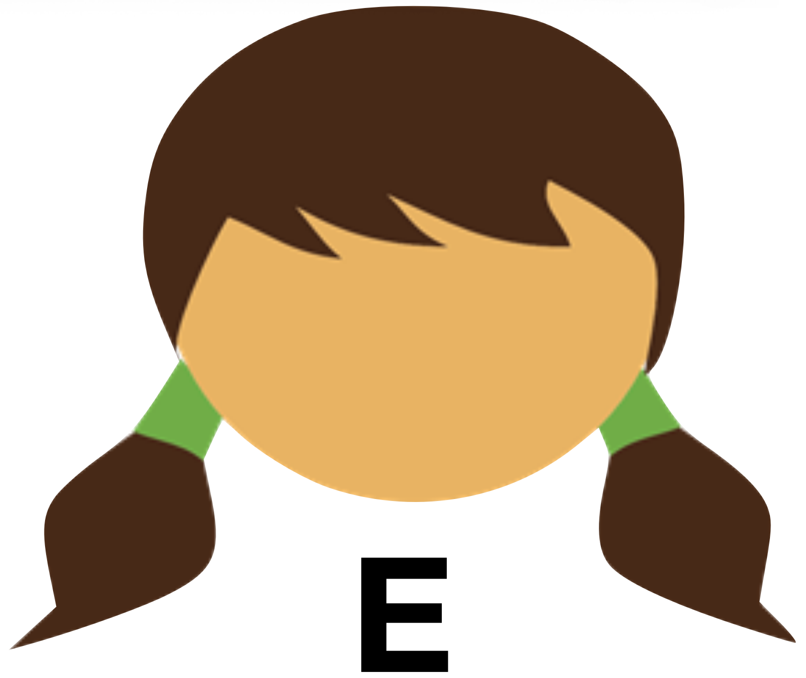
G

$\Delta \xleftarrow{\$} \{0,1\}^\lambda$

$\mathrm{Enc}(X, Z)$

$\mathrm{Enc}(X \oplus \Delta, Y \oplus Z)$

Garbled Circuit



E

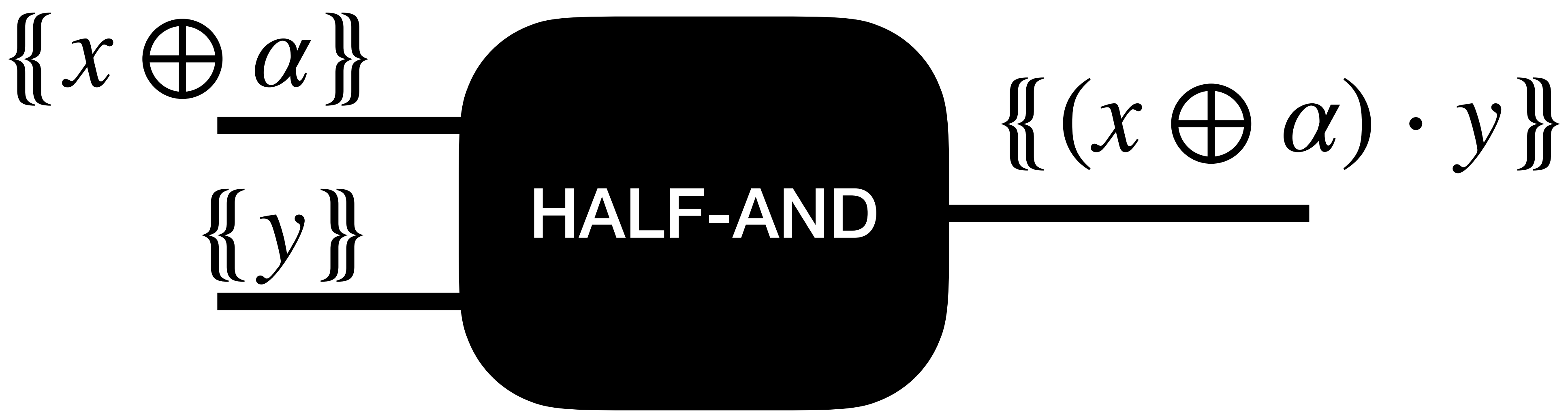
$\mu \xleftarrow{\$} \{0,1\}^\sigma$

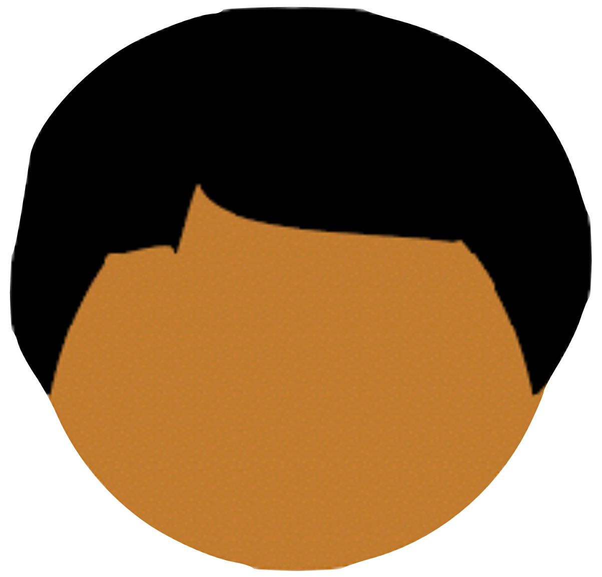
$x \oplus \alpha$

Authenticated Garbling

$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

where $\alpha, \beta \xleftarrow{\$} \{0,1\}$



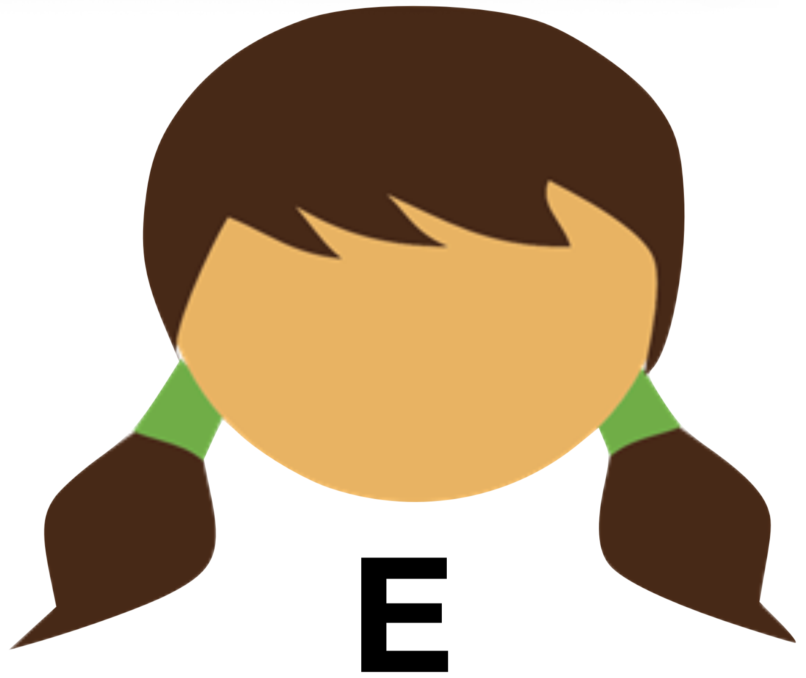


G

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$

$$\langle X, X \oplus (x \oplus \alpha) \cdot \Delta \rangle = \text{keyPart}(\{\!\{x \oplus \alpha\}\!\})$$

$$\langle Y, Y \oplus y \cdot (\Delta, \mu, 1) \rangle = \{\!\{y\}\!\}$$



E

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

$x \oplus \alpha$

$\text{Enc}(X, Z)$

$\text{Enc}(X \oplus \Delta, Y \oplus Z)$

$$\begin{cases} Z & \text{if } x \oplus \alpha = 0 \\ Z \oplus y \cdot (\Delta, \mu, 1) & \text{otherwise} \end{cases}$$

Authenticated Garbling

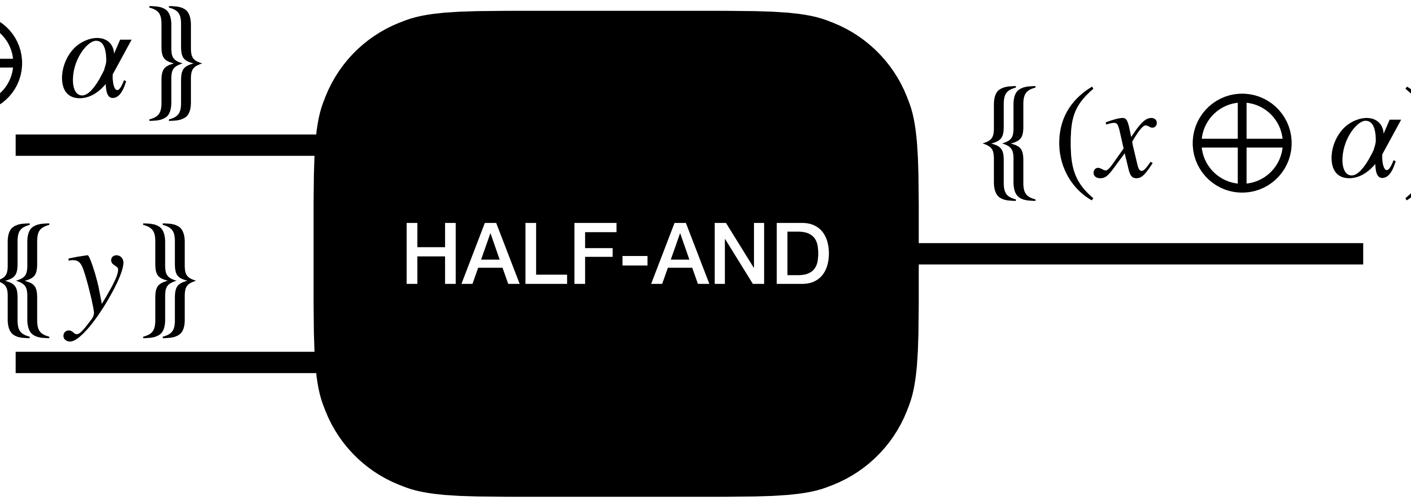
$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}$

where $\alpha, \beta \xleftarrow{\$} \{0,1\}$

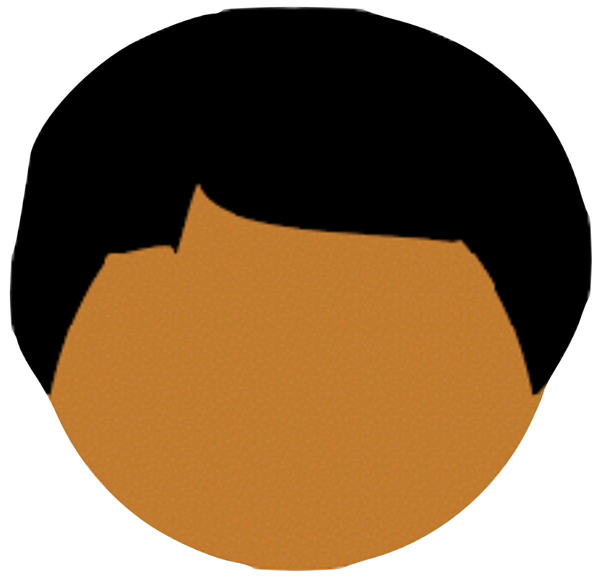
Two Halves Make a Whole
Reducing Data Transfer in Garbled Circuits using Half Gates
Samee Zahur, Mike Rosulek, and David Evans

$\{\!\{x \oplus \alpha\}\!\}$

$\{\!\{y\}\!\}$



$\{\!\{(x \oplus \alpha) \cdot y\}\!\}$

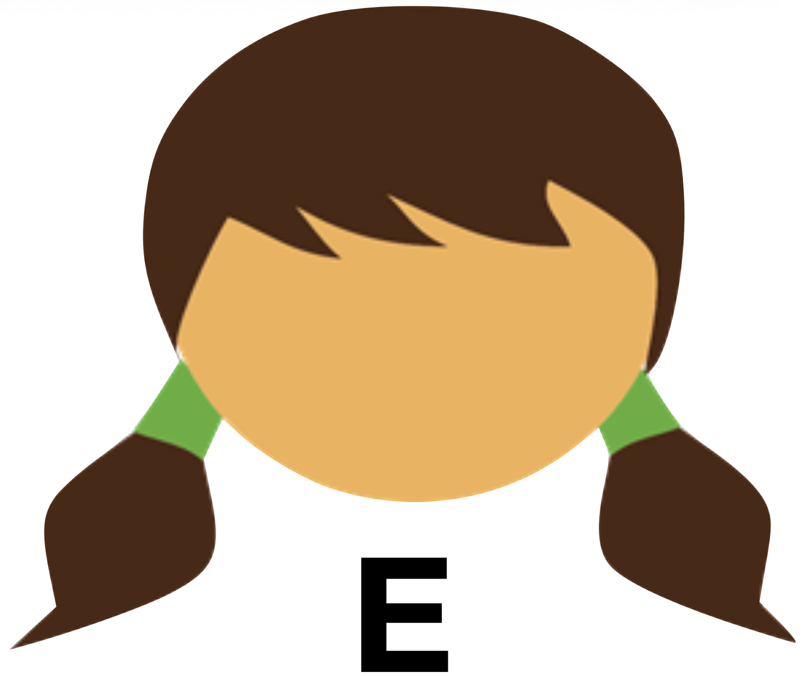


G

$\Delta \xleftarrow{\$} \{0,1\}^\lambda$

$$\langle X, X \oplus (x \oplus \alpha) \cdot \Delta \rangle = \text{keyPart}(\{\!\{x \oplus \alpha\}\!\})$$

$$\langle Y, Y \oplus y \cdot (\Delta, \mu, 1) \rangle = \{\!\{y\}\!\}$$



E

$\mu \xleftarrow{\$} \{0,1\}^\sigma$

$x \oplus \alpha$

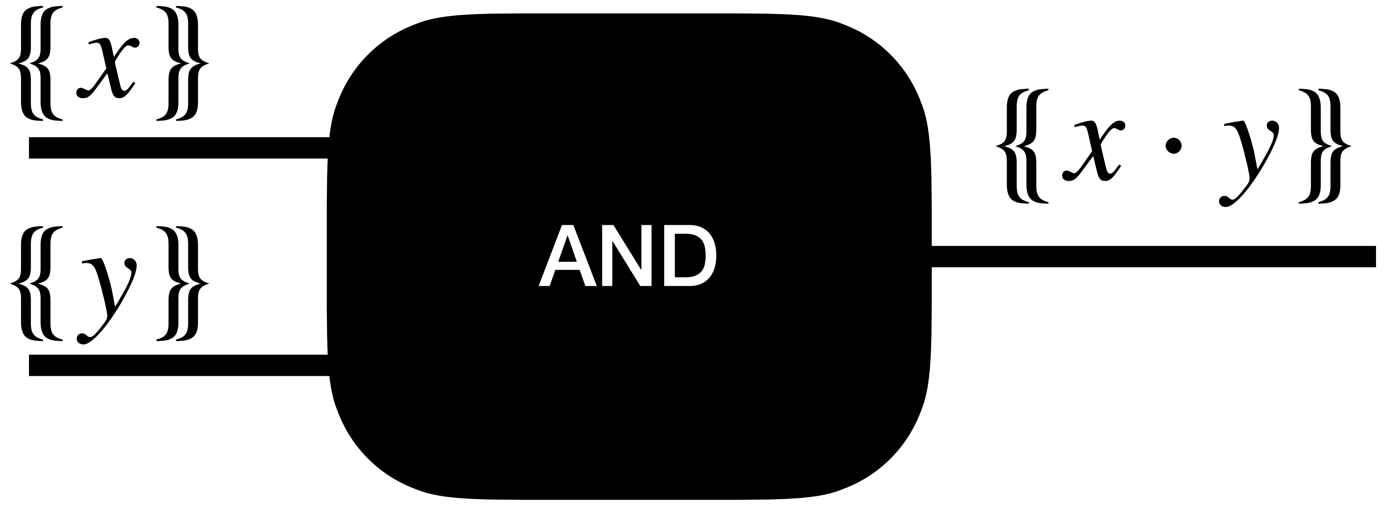
$\text{Enc}(X, Z)$

$\text{Enc}(X \oplus \Delta, Y \oplus Z)$

$$\begin{cases} Z & \text{if } x \oplus \alpha = 0 \\ Z \oplus y \cdot (\Delta, \mu, 1) & \text{otherwise} \end{cases}$$

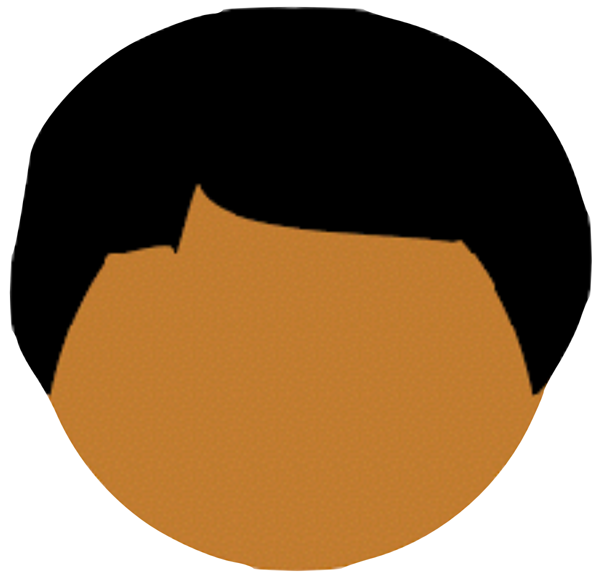
No opportunity for selective abort

Authenticated Garbling



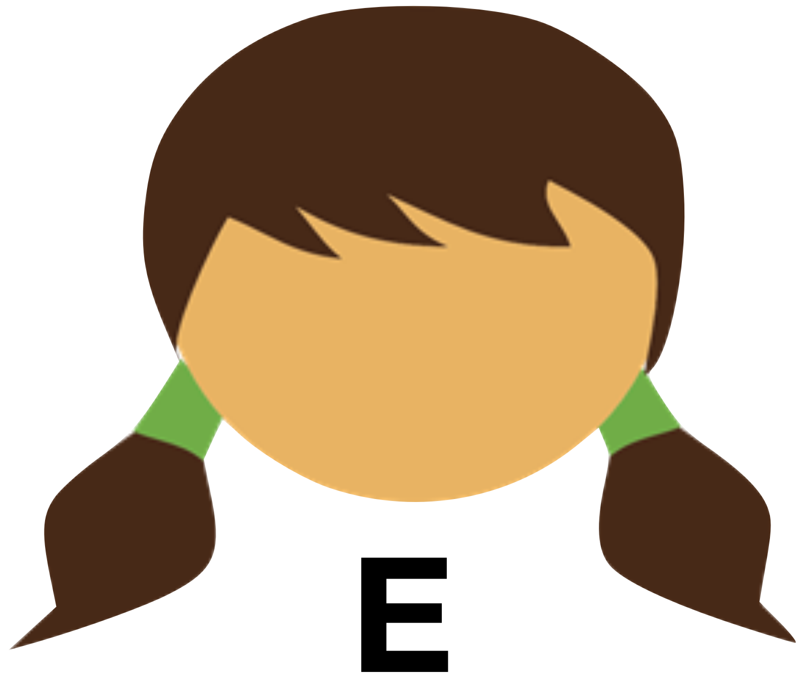
$$
\begin{array}{ll}
 & \llbracket (x \oplus \alpha) \cdot y \rrbracket \\
 & \llbracket (y \oplus \beta) \cdot \alpha \rrbracket \\
\oplus & \llbracket \alpha \cdot \beta \rrbracket \\
\hline
 & \llbracket x \cdot y \rrbracket
\end{array}
$$

} **half gates** (the first two lines)



G

$\Delta \xleftarrow{\$} \{0,1\}^\lambda$



E

$\mu \xleftarrow{\$} \{0,1\}^\sigma$

Authenticated Garbling

Authenticated Garbling

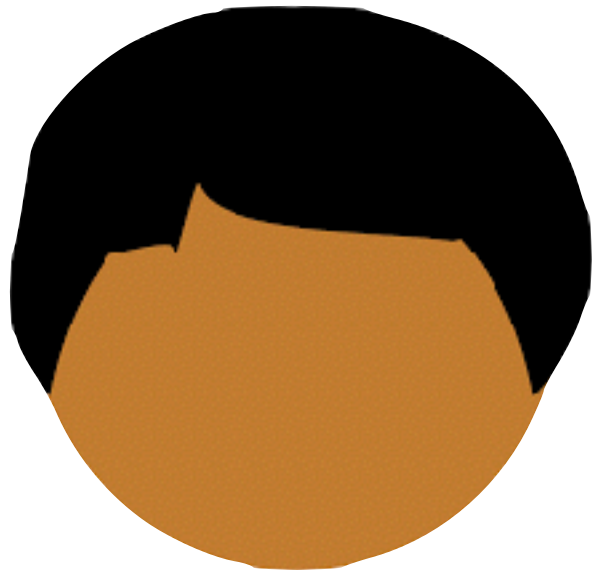
G input wire

$\{\!\{r\}\!\} = [r \cdot \Delta, r \cdot \mu, r]$

$r \cdot \Delta, r$

Open key part, value part ←

$r \oplus x$ →

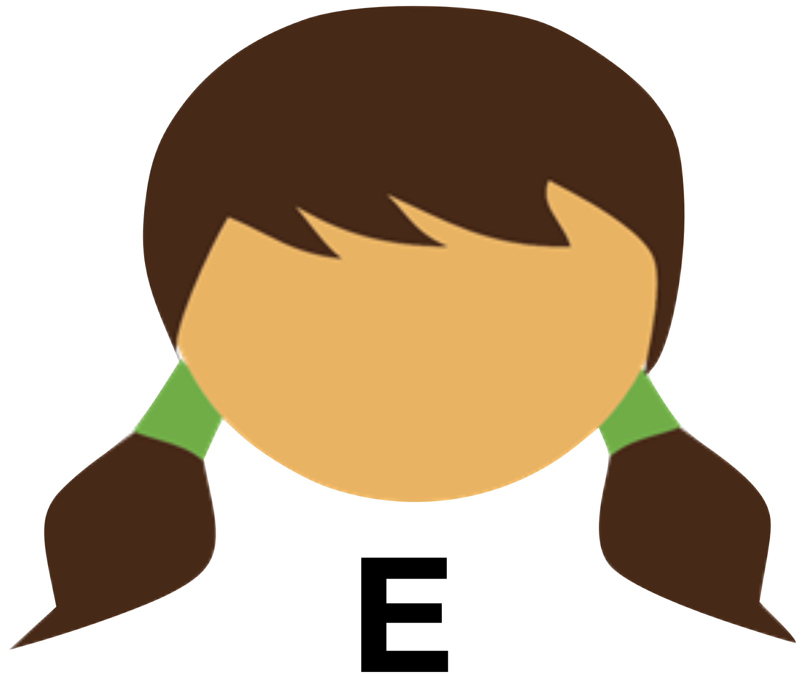


x

G

$\Delta \xleftarrow{\$} \{0,1\}^\lambda$

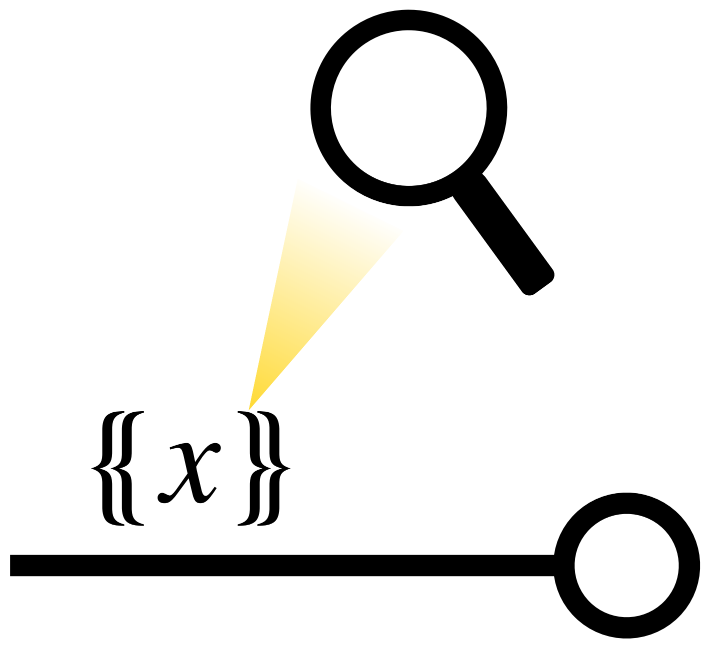
$\{\!\{x\}\!\} = [x \cdot \Delta, x \cdot \mu, x]$



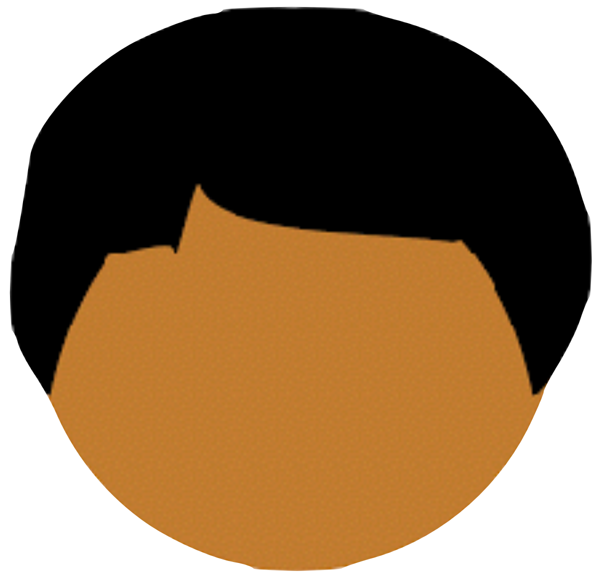
E

$\mu \xleftarrow{\$} \{0,1\}^\sigma$

Authenticated Garbling

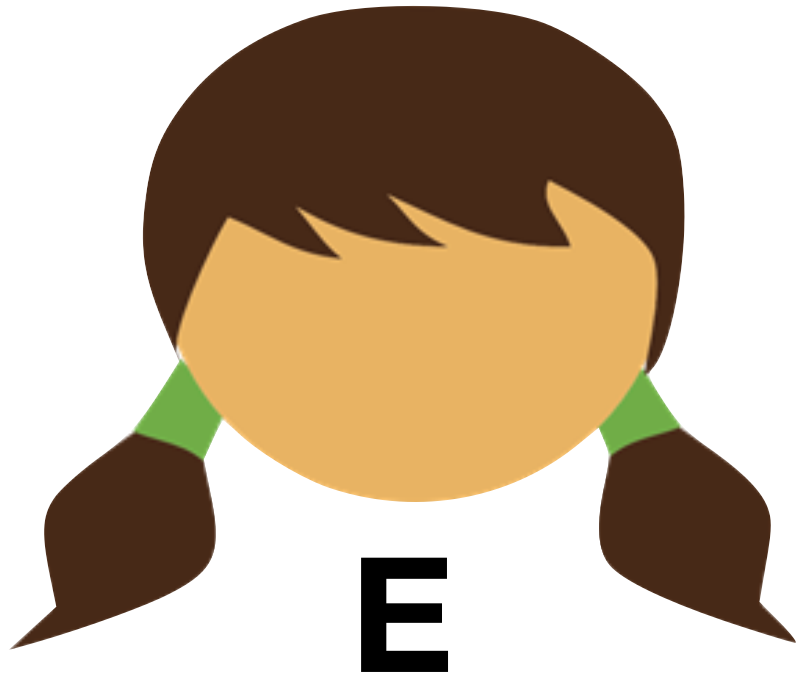


output wire



G

$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$



E

$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

Authenticated Garbling

Authenticated Garbling

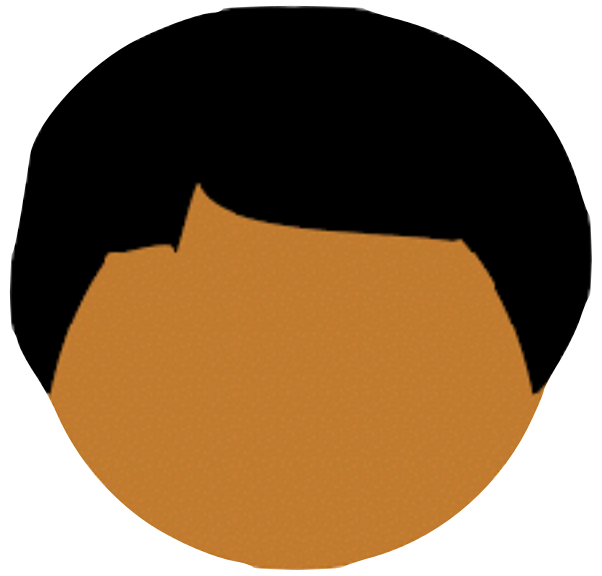
Preprocessing Functionality

Suppose G and E have access to a **doubly authenticated multiplication triple**

$$\llbracket \alpha \rrbracket, \llbracket \beta \rrbracket, \llbracket \alpha \cdot \beta \rrbracket$$

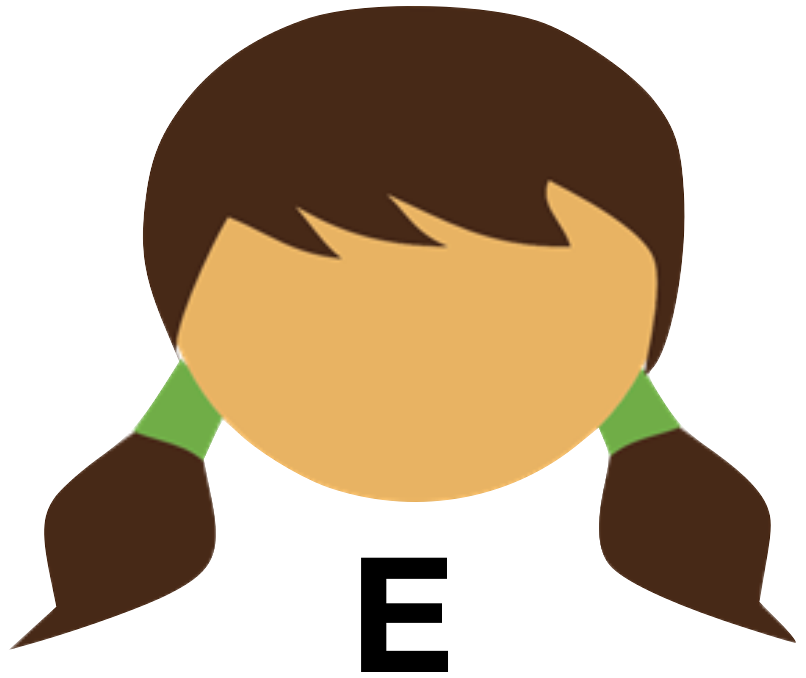
where $\alpha, \beta \stackrel{\$}{\leftarrow} \{0,1\}$

Is this an easier problem?



G

$\Delta \stackrel{\$}{\leftarrow} \{0,1\}^\lambda$



E

$\mu \stackrel{\$}{\leftarrow} \{0,1\}^\sigma$

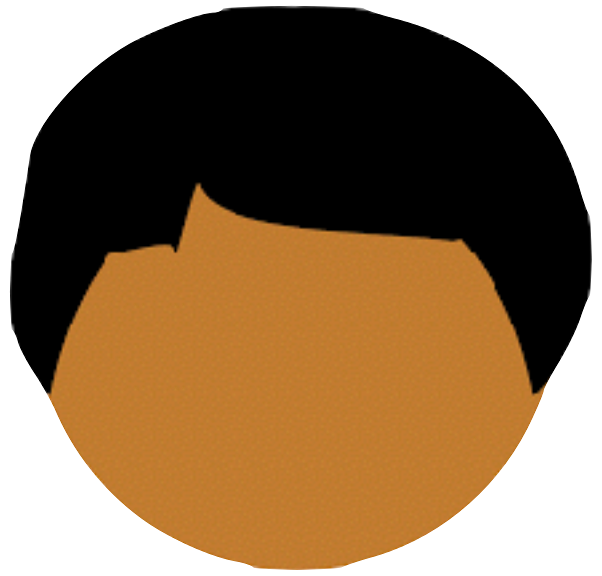
Authenticated Garbling

Preprocessing Functionality

Suppose G and E have access to a **doubly authenticated multiplication triple**

$$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}$$

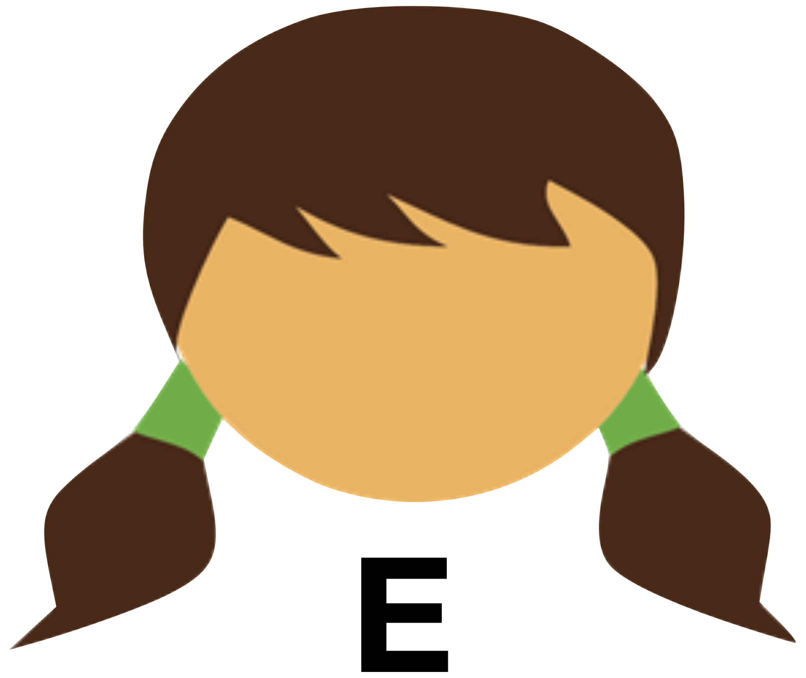
where $\alpha, \beta \xleftarrow{\$} \{0,1\}$



G

$\Delta \xleftarrow{\$} \{0,1\}^\lambda$

Is this an easier problem?



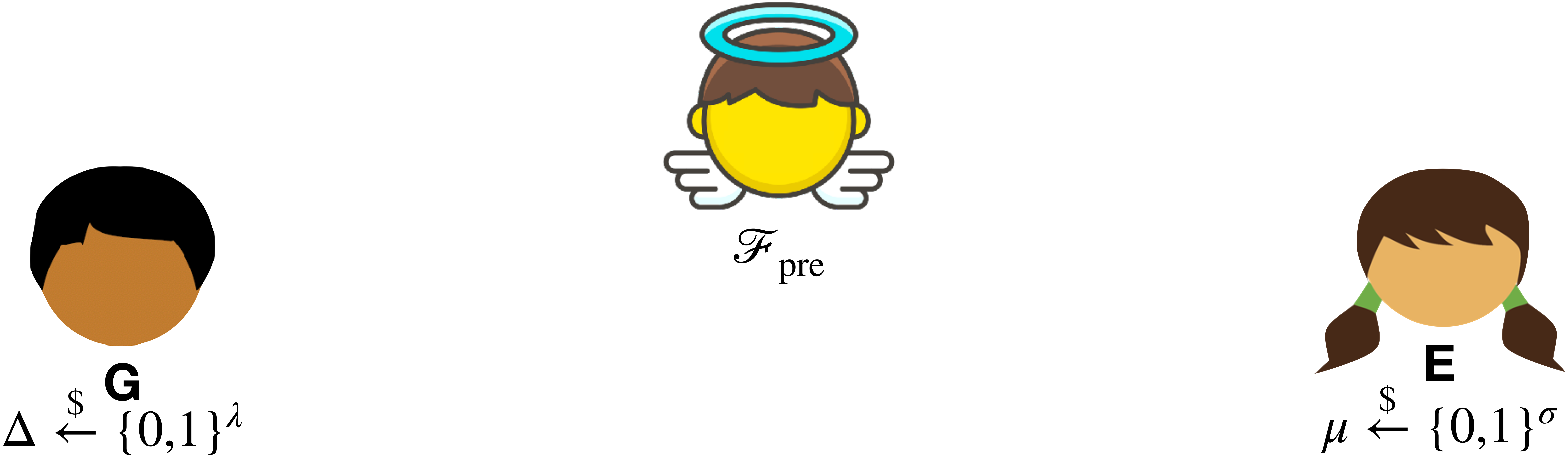
E

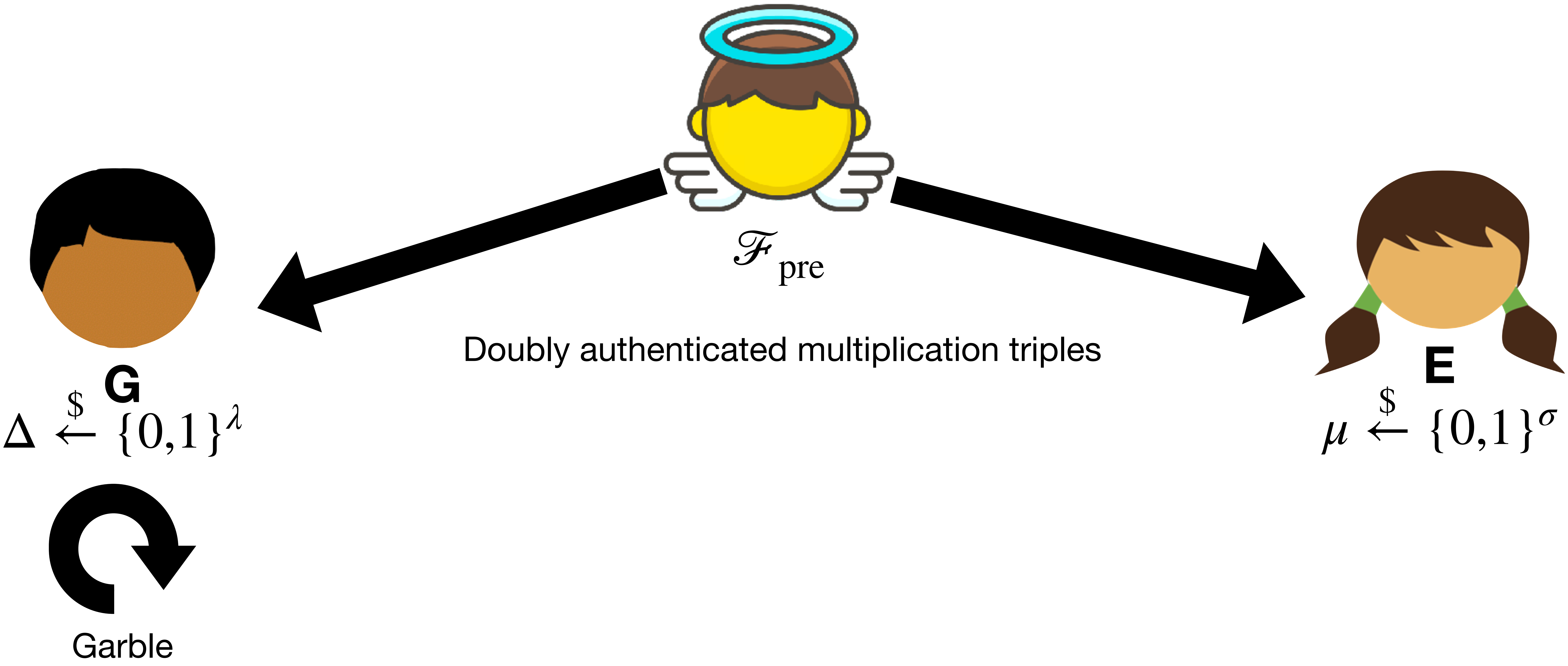
$\mu \xleftarrow{\$} \{0,1\}^\sigma$

Yes!

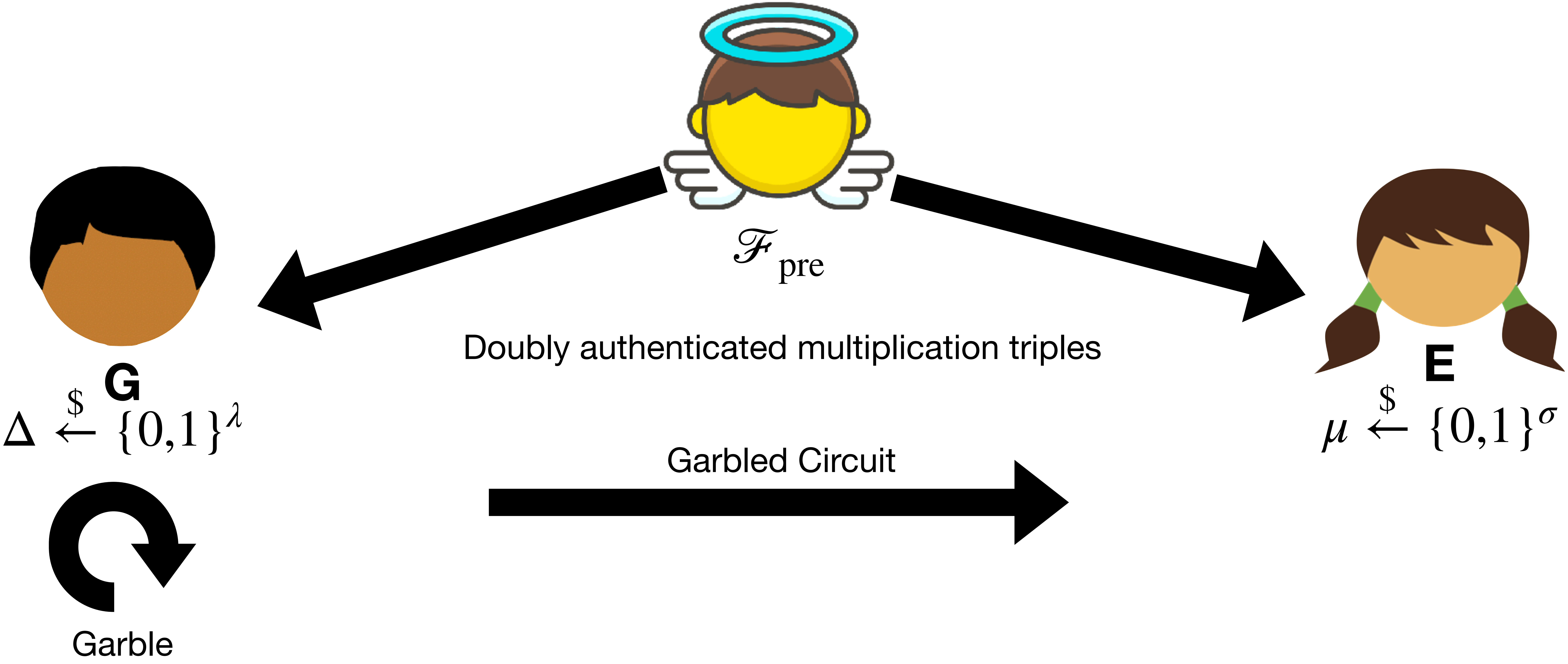
Random bits only; not dependent on inputs

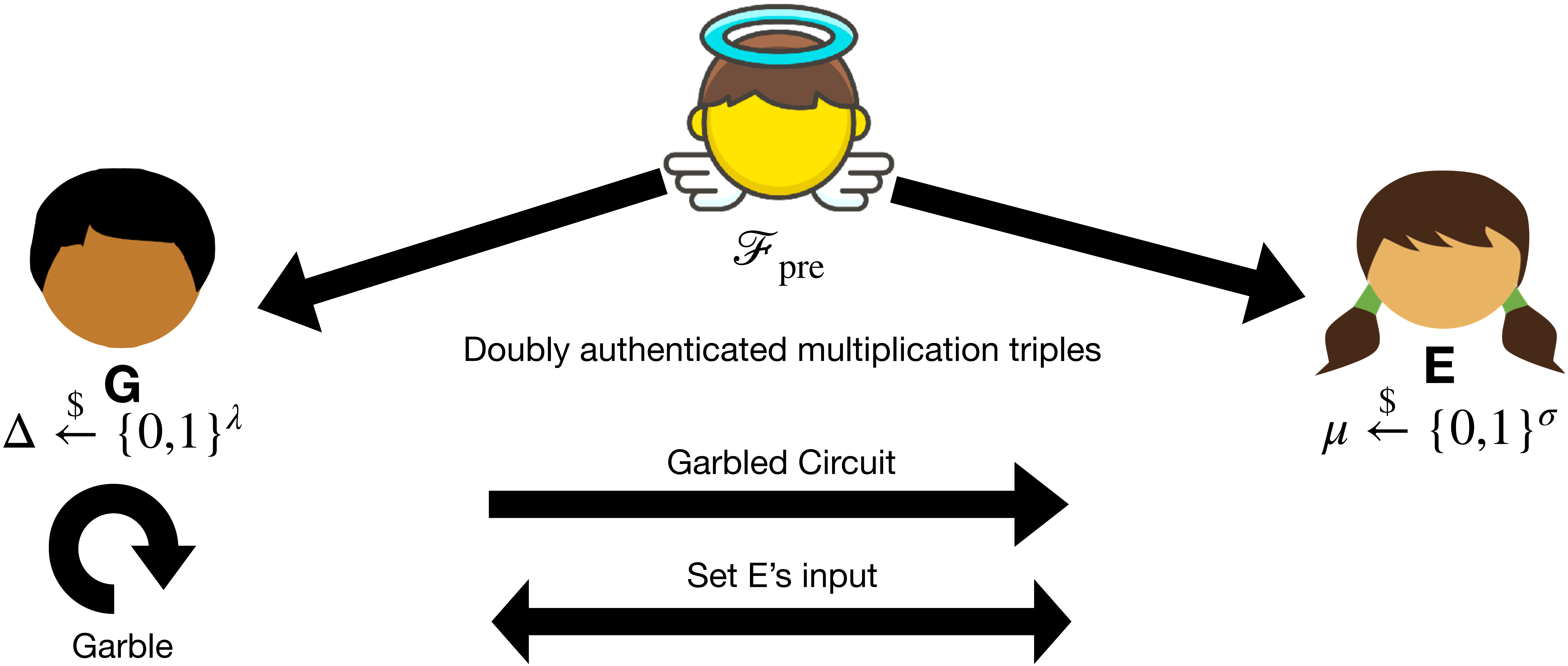
Can be computed all at once; no circuit topology


$\mathcal{F}_{\text{pre}}$
G
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$
E
$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$

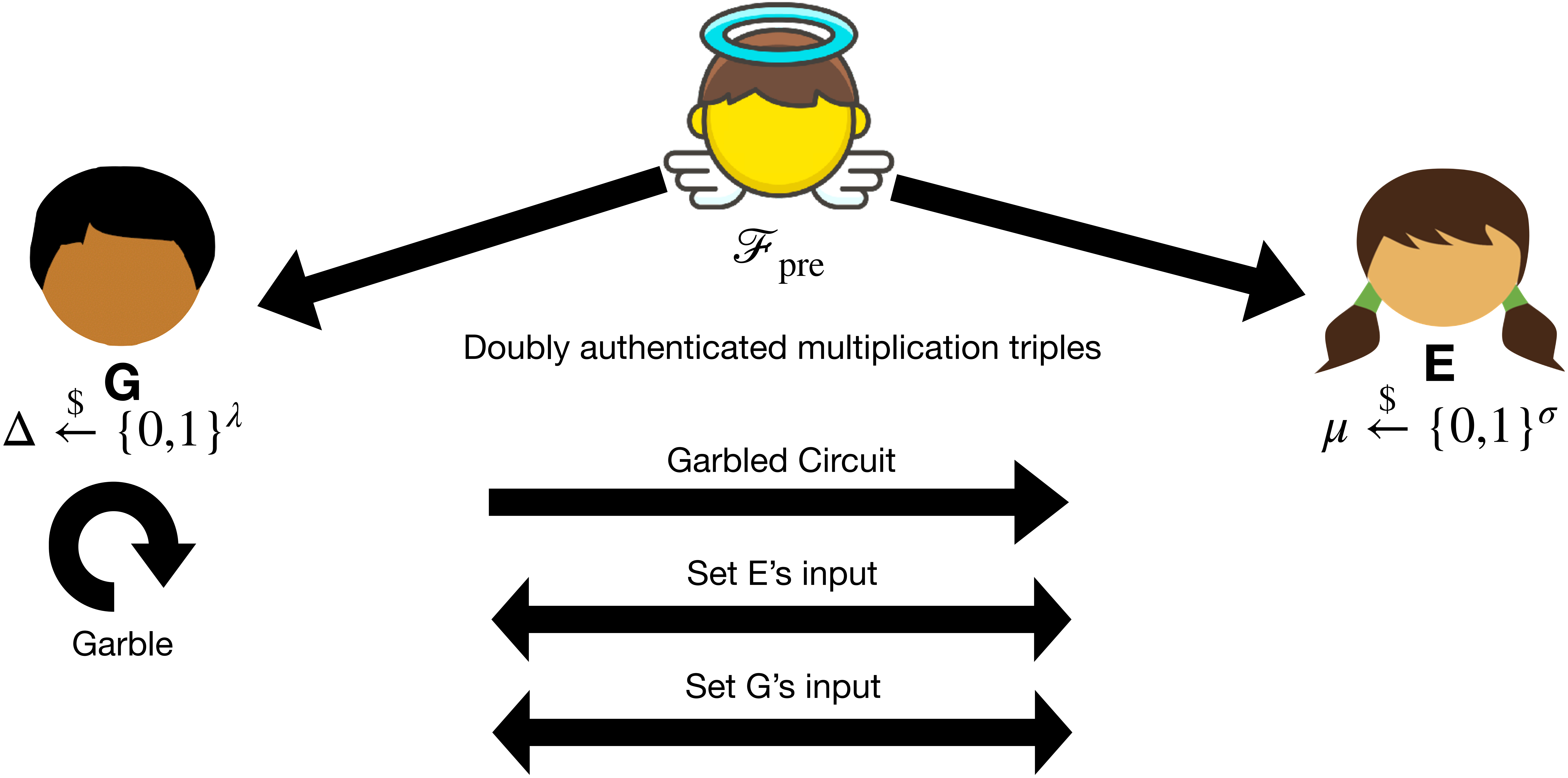


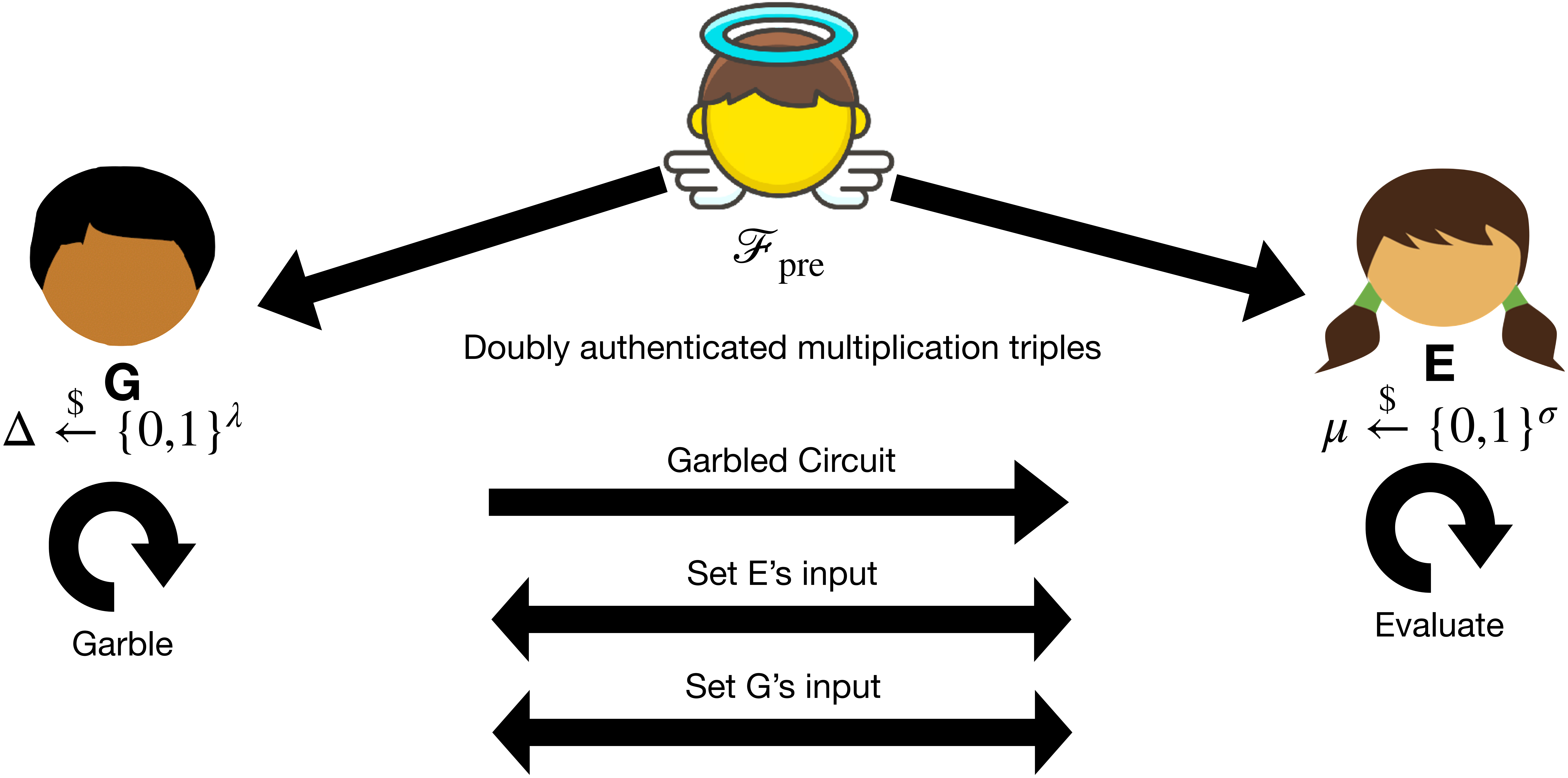
$\mathcal{F}_{\text{pre}}$
Doubly authenticated multiplication triples
G
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$
Garble
E
$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$


$\mathcal{F}_{\text{pre}}$
Doubly authenticated multiplication triples
G
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$
E
$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$
Garbled Circuit
Garble



$\mathcal{F}_{\text{pre}}$
Doubly authenticated multiplication triples
G
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$
E
$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$
Garbled Circuit
Set E's input
Garble


$\mathcal{F}_{\text{pre}}$
Doubly authenticated multiplication triples
G
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$
E
$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$
Garble
Garbled Circuit
Set E's input
Set G's input


$\mathcal{F}_{\text{pre}}$
Doubly authenticated multiplication triples
G
$\Delta \xleftarrow{\$} \{0,1\}^{\lambda}$
E
$\mu \xleftarrow{\$} \{0,1\}^{\sigma}$
Garbled Circuit
Set E's input
Set G's input
Garble
Evaluate

Authenticated Garbling and Efficient Maliciously Secure Two-Party Computation

Xiao Wang
University of Maryland
wangxiao@cs.umd.edu

Samuel Ranellucci
University of Maryland
George Mason University
samuel@umd.edu

Jonathan Katz
University of Maryland
jkatz@cs.umd.edu

Abstract

We propose a simple and efficient framework for obtaining efficient constant-round protocols for maliciously secure two-party computation. Our framework uses a function-independent preprocessing phase to generate authenticated information for the two parties; this information is then used to construct a *single* "authenticated" garbled circuit which is transmitted and evaluated.

We also show how to efficiently instantiate the preprocessing phase by designing a highly optimized version of the TinyOT protocol by Nielsen et al. Our overall protocol outperforms existing work in both the single-execution and amortized settings, with or without preprocessing:

- In the single-execution setting, our protocol evaluates an AES circuit with malicious security in 37 ms with an online time of just 1 ms. Previous work with the best online time (also 1 ms) requires 124 ms in total; previous work with the best total time requires 62 ms (with 14 ms online time).
- If we amortize the computation over 1024 executions, each AES computation requires just 6.7 ms with roughly the same online time as above. The best previous work in the amortized setting has roughly the same total time but does not support function-independent preprocessing.

Our work shows that the performance penalty for maliciously secure two-party computation (as compared to semi-honest security) is much smaller than previously believed.

1 Introduction

Protocols for secure two-party computation (2PC) allow two parties to compute an agreed-upon function of their inputs without revealing anything additional to each other. Although originally viewed as impractical, protocols for generic 2PC in the semi-honest setting based on Yao's garbled-circuit protocol [Yao86] have seen tremendous efficiency improvements over the past several years [MNPS04, HEKM11, ZRE15, KS08, KMR14, ALSZ13, BHKR13, PSSW09].

While these results are impressive, semi-honest security—which assumes that both parties follow the protocol honestly yet may try to learn additional information from the execution—is clearly not sufficient for all applications. This has motivated researchers to construct protocols achieving the stronger notion of *malicious* security. One popular approach for designing constant-round maliciously secure protocols is to apply the "cut-and-choose" technique [LP07, sS11, sS13, KSS12, LP11, HKE13, Lin13, Bra13, FJN14, AMPR14] to Yao's garbled-circuit protocol. For statistical security $2^{-\rho}$, the best approaches using this paradigm require ρ garbled circuits (which is optimal); the most efficient instantiation of this approach, by Wang et al. [WMK17], securely evaluates an AES circuit in 62 ms.

The cut-and-choose approach incurs significant overhead when large circuits are evaluated precisely because ρ garbled circuits need to be transmitted (typically, $\rho \geq 40$). In order to mitigate this, recent works have explored secure computation in an *amortized* setting where the same function is evaluated multiple times

Constant round protocol secure against malicious adversaries for arbitrary Boolean circuits

Used doubly-authenticated multiplication triples to allow E to check values are well-formed, prevent G from performing selective abort attack

Doubly-authenticated multiplication triples can be efficiently constructed using multiplication triples

Today's objectives

Understand Cut and Choose

Understand Information-Theoretic MACs

Sketch authenticated garbling protocol