Malicious Garbled Circuits CS 598 DH

Today's objectives

Understand Cut and Choose

Understand Information-Theoretic MACs

Sketch authenticated garbling protocol

<u>Setting</u>

Semi-honest Security

Malicious Security

Zero Knowledge

- **Primitives**
- **Oblivious Transfer**
- Commitments
- ORAM

General-Purpose Tools GMW Protocol Multi-party Multi-round

Garbled Circuit Constant Round Two Party

Pseudorandom functions/encryption

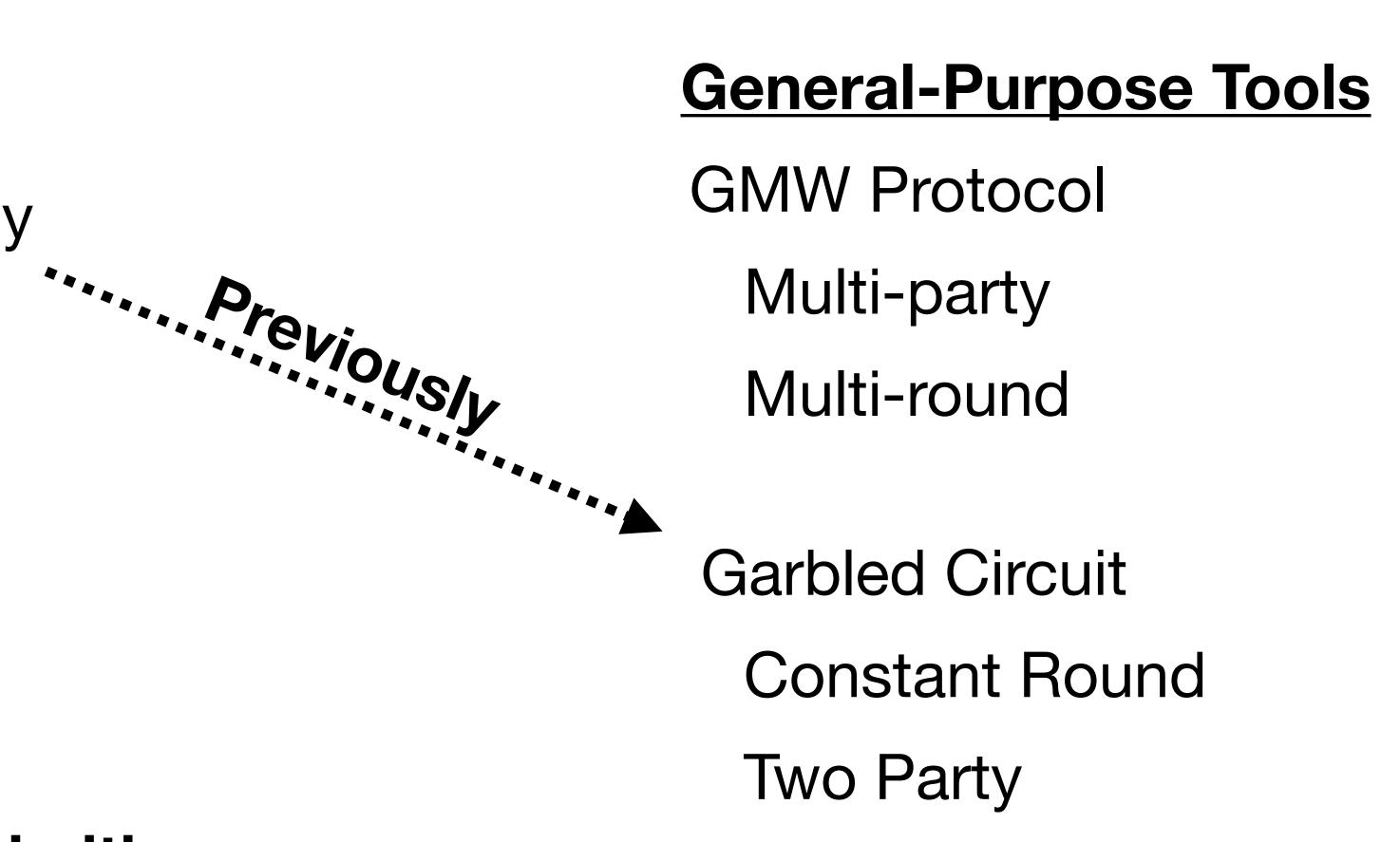
<u>Setting</u>

Semi-honest Security

Malicious Security

Zero Knowledge

- **Primitives**
- **Oblivious Transfer**
- Pseudorandom functions/encryption Commitments
- ORAM

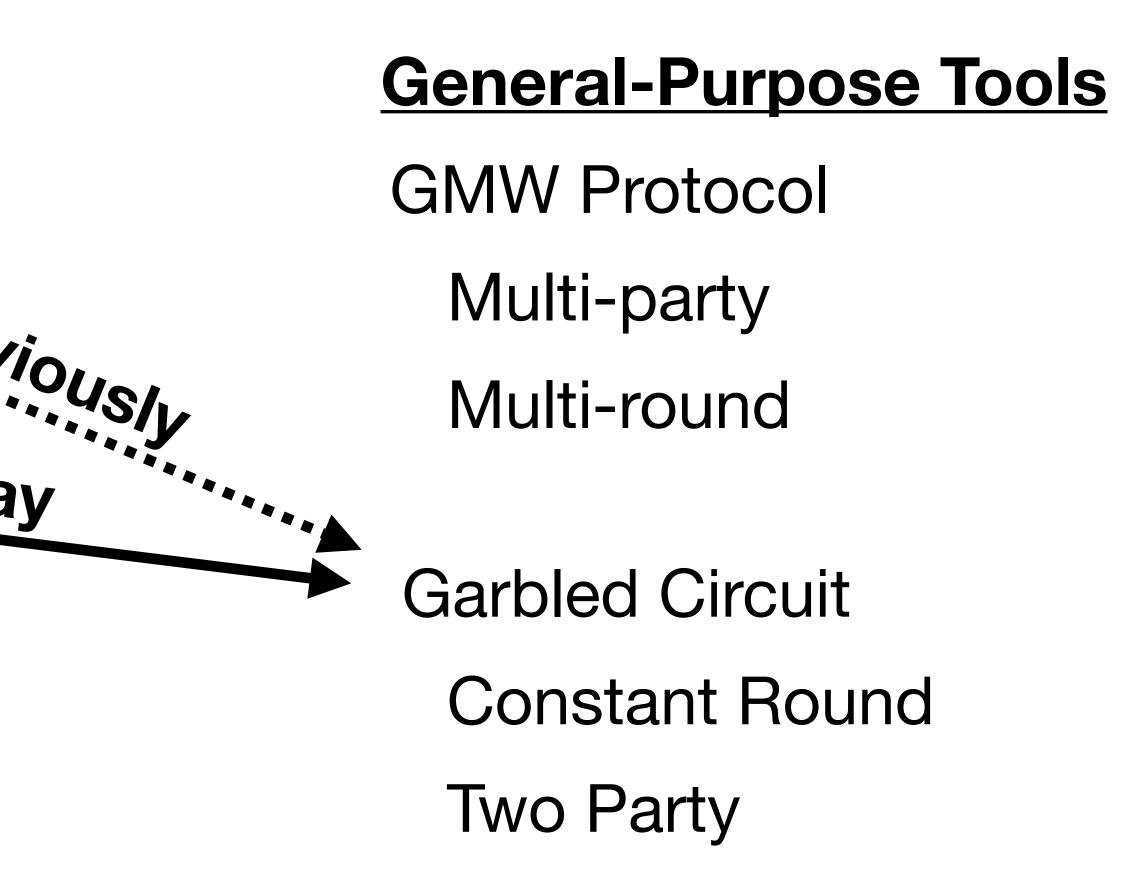


<u>Setting</u>

Semi-honest Security ···· Previously Today Malicious Security

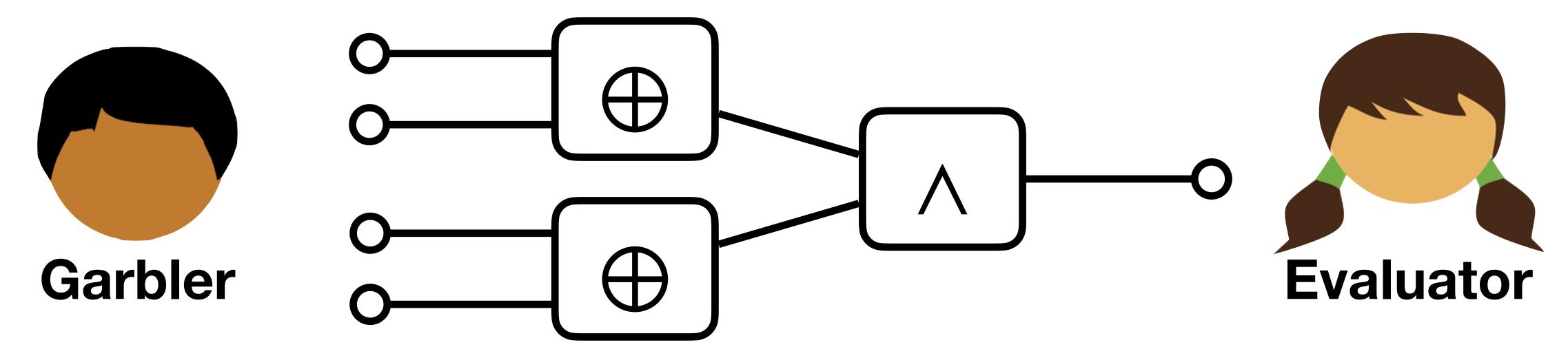
Zero Knowledge

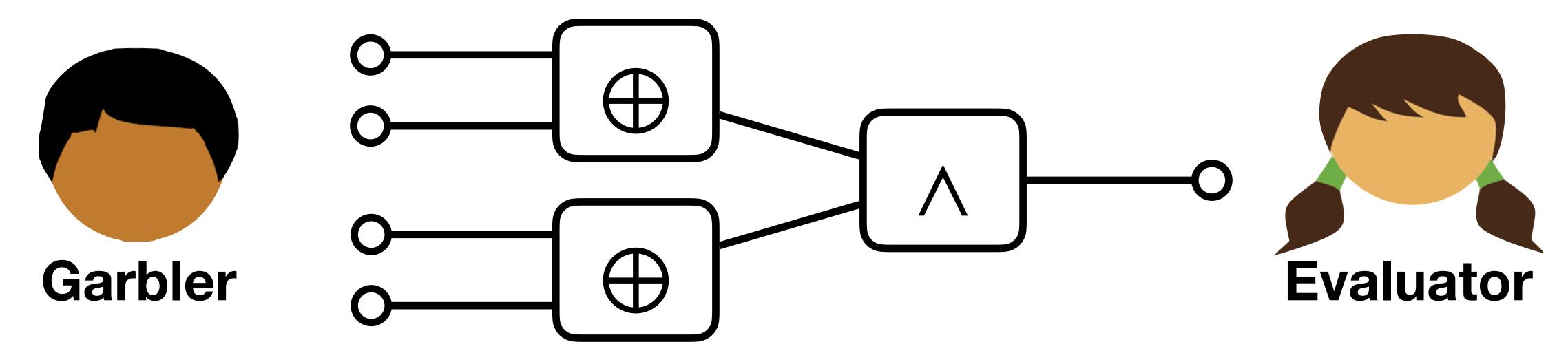
- **Primitives**
- **Oblivious Transfer**
- Pseudorandom functions/encryption Commitments
- ORAM

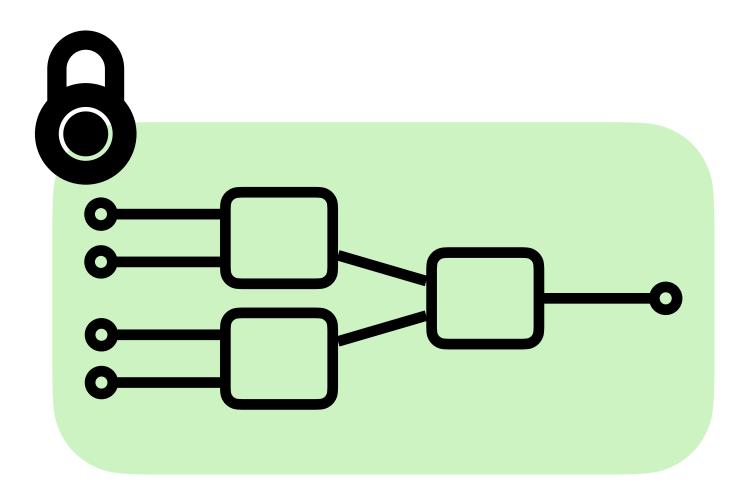


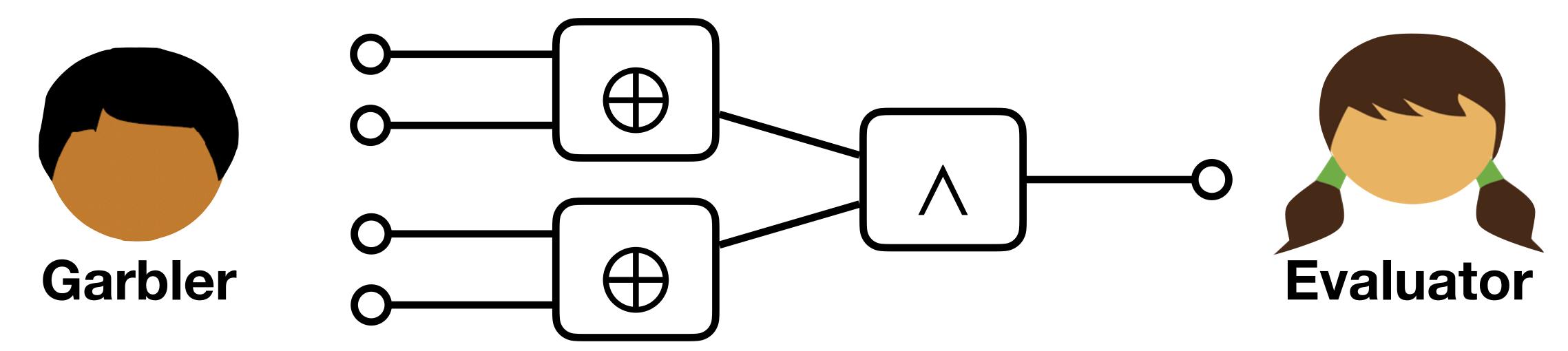


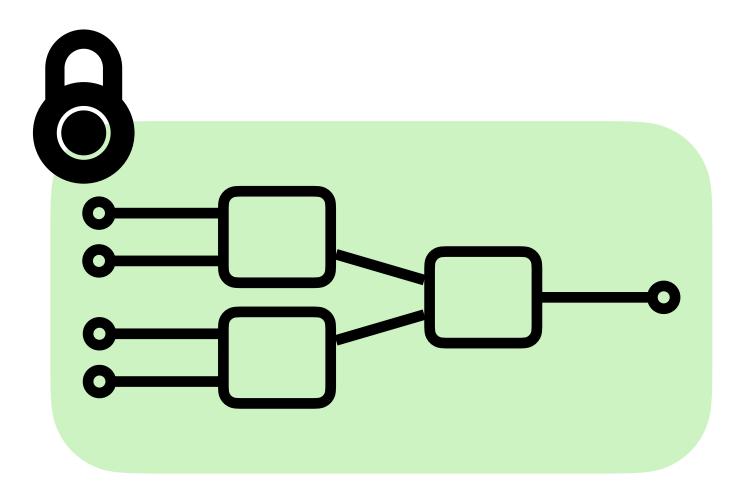


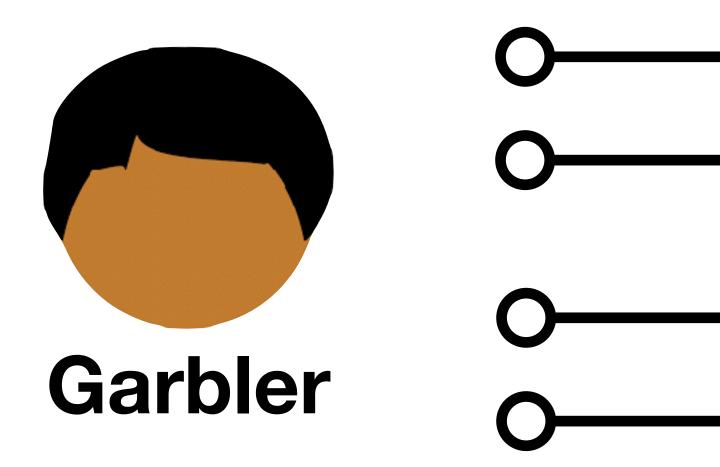






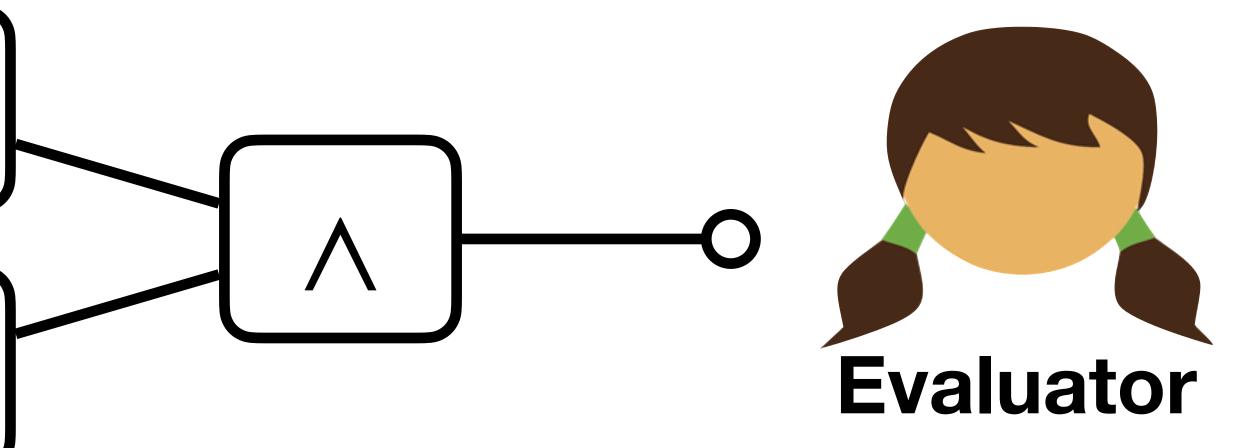


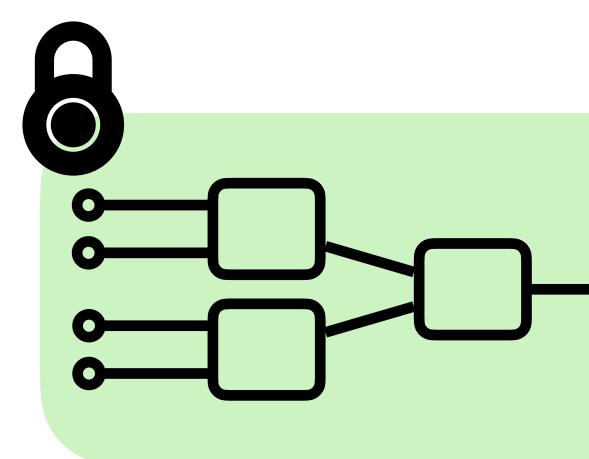




(OT)

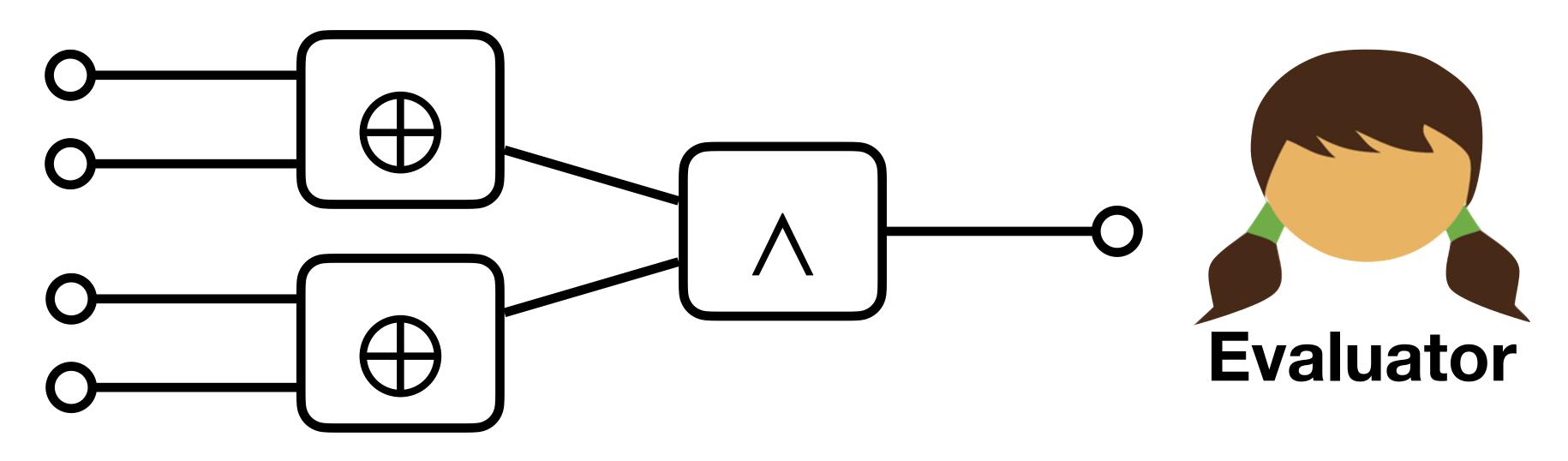
 \oplus

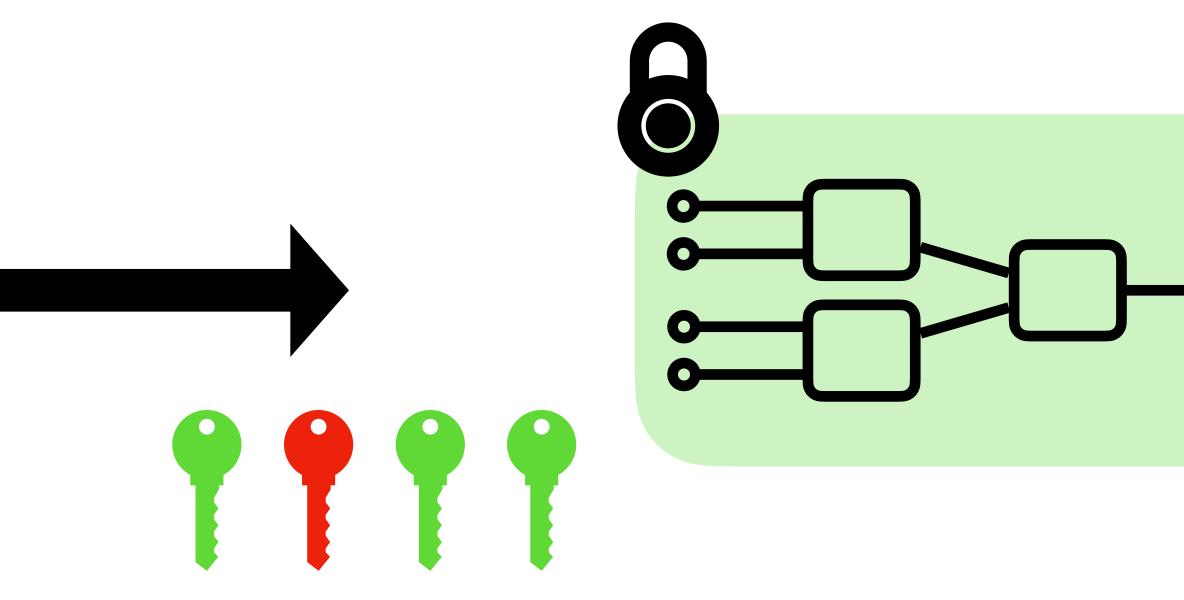






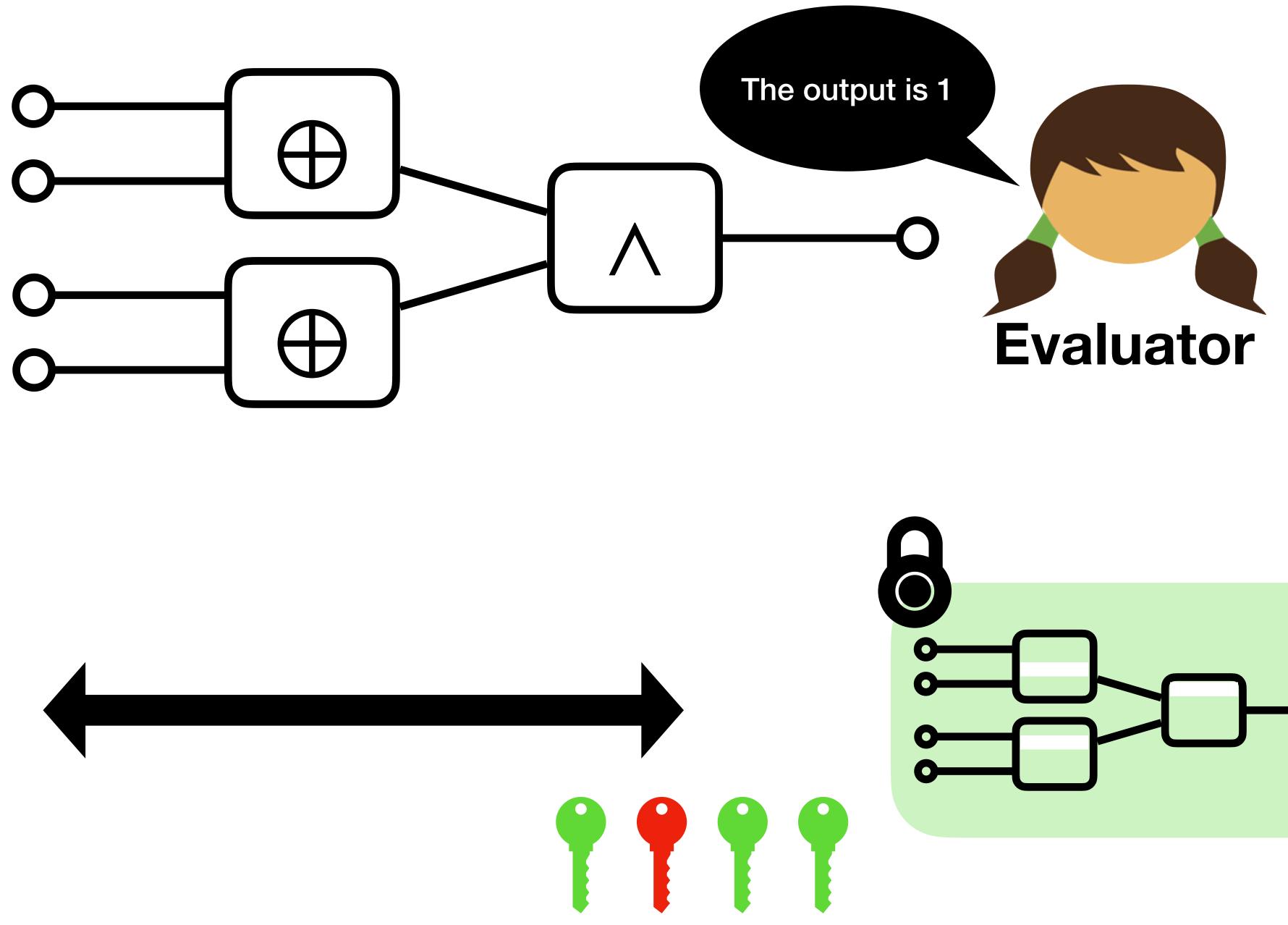






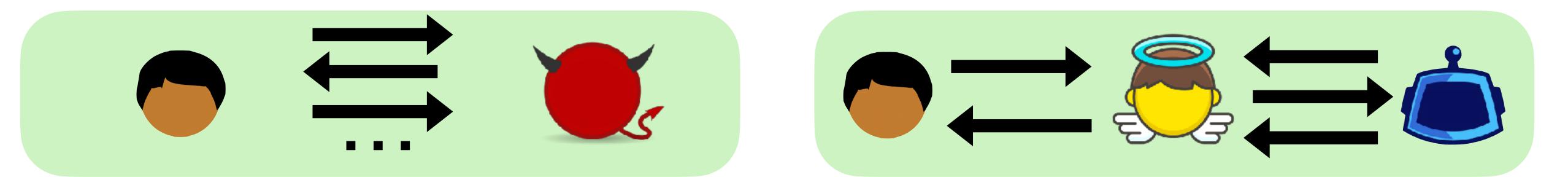






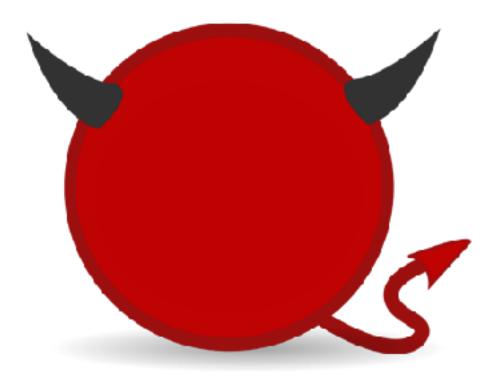


Malicious Security (with abort)



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 S_i (a simulator) such that for all inputs *x*, *y* the following holds: $\operatorname{Real}_{\mathscr{A}}^{\Pi}(x, y) \approx \operatorname{Ideal}_{\mathscr{S}_{i}}^{f}(x, y)$

Ensemble of outputs of each party



Garbler

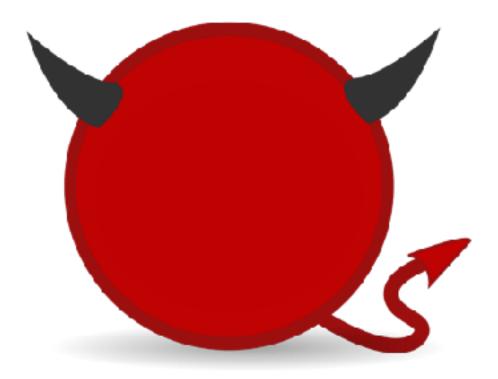
$\operatorname{Enc}(K_a^0, \operatorname{Enc}(K_b^0, K_c^0))$ $\operatorname{Enc}(K_a^0, \operatorname{Enc}(K_b^1, K_c^0))$ $\operatorname{Enc}(K_a^1, \operatorname{Enc}(K_b^0, K_c^0))$ $\operatorname{Enc}(K_a^1,\operatorname{Enc}(K_b^1,K_c^1))$

Why can't we simulate G?

G can encrypt each gate freely

E has no way to tell if gate it correctly garbled





Garbler

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

Why can't we simulate G?

G can encrypt each gate freely

E has no way to tell if gate it correctly garbled

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



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 Nielsen¹, Peter Sebastian Nordholt¹, Claudio Orlandi², Sai Sheshank Burra³

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.

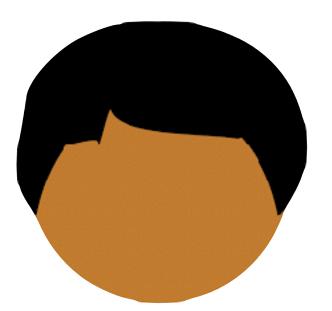
Problem with GC

Adversary can arbitrarily corrupt the GC encryptions

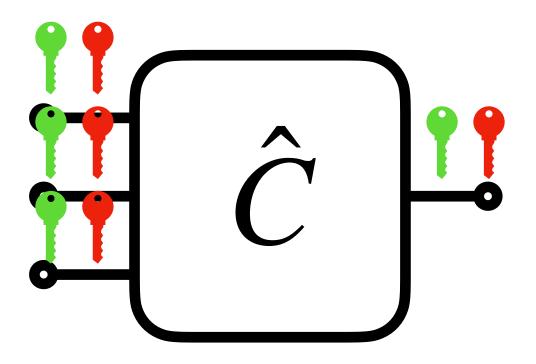
Solution Space

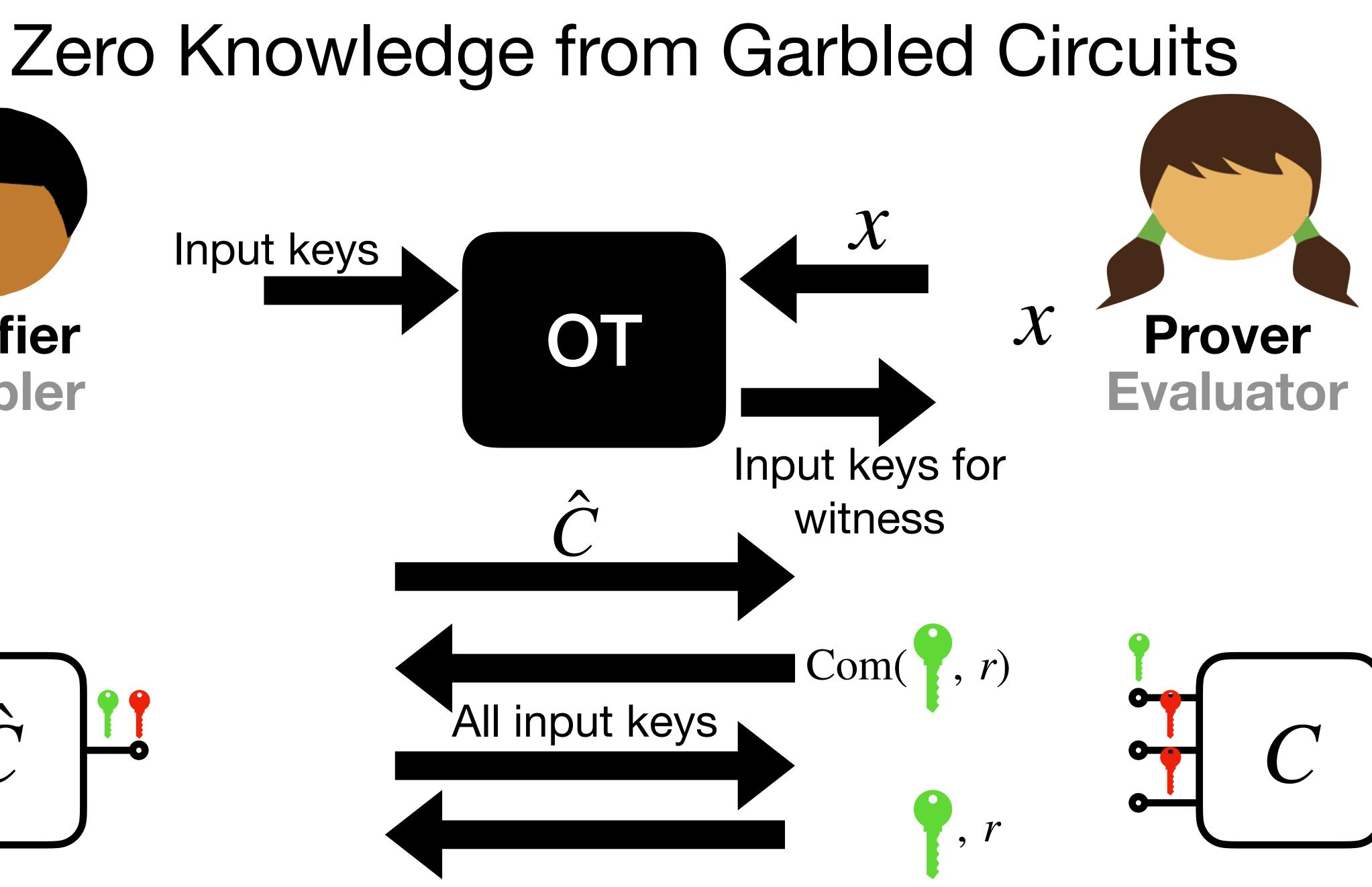
Evaluator can check a garbled circuit is well-formed

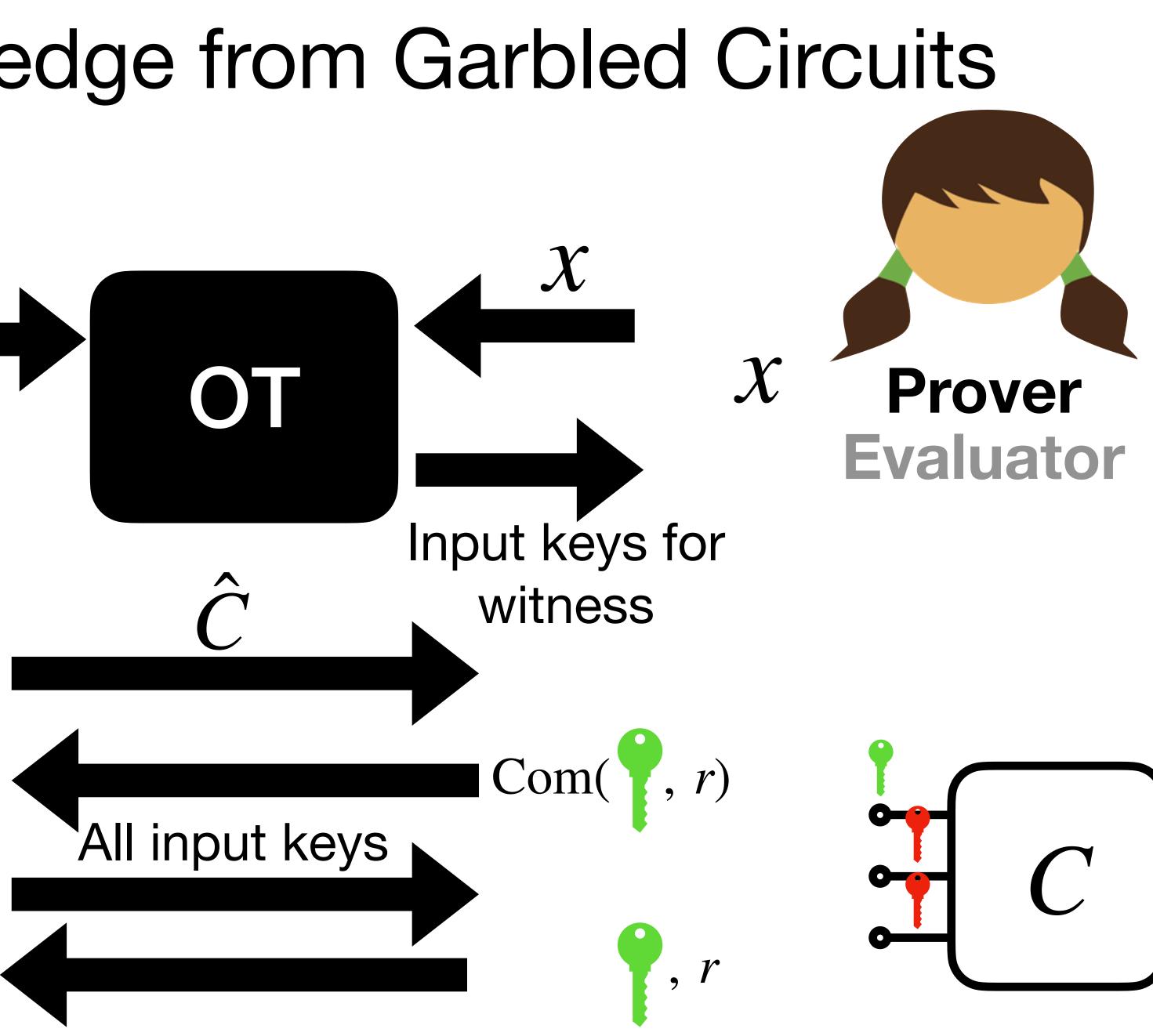
We can repeat multiple times

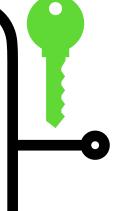


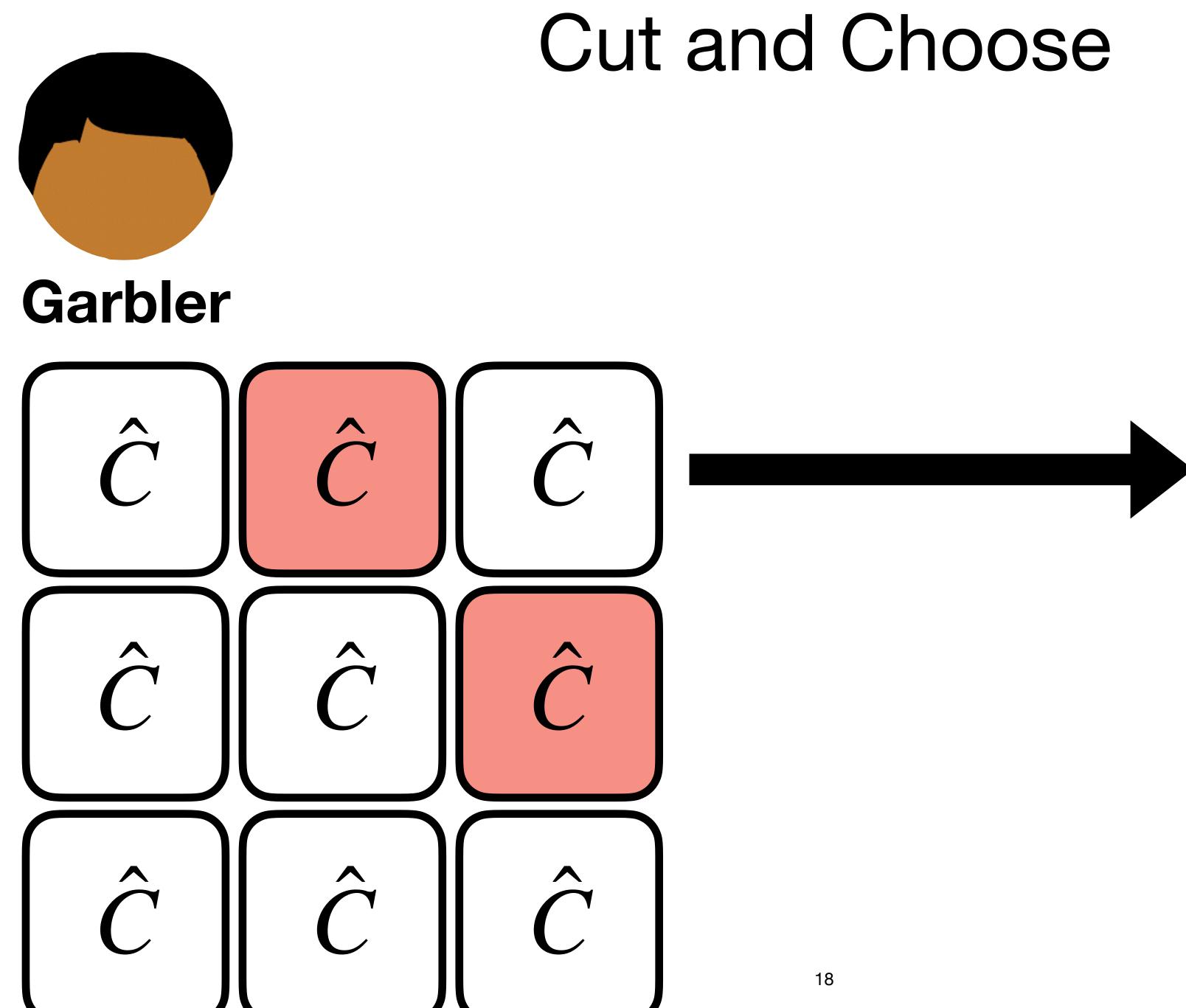
Verifier Garbler



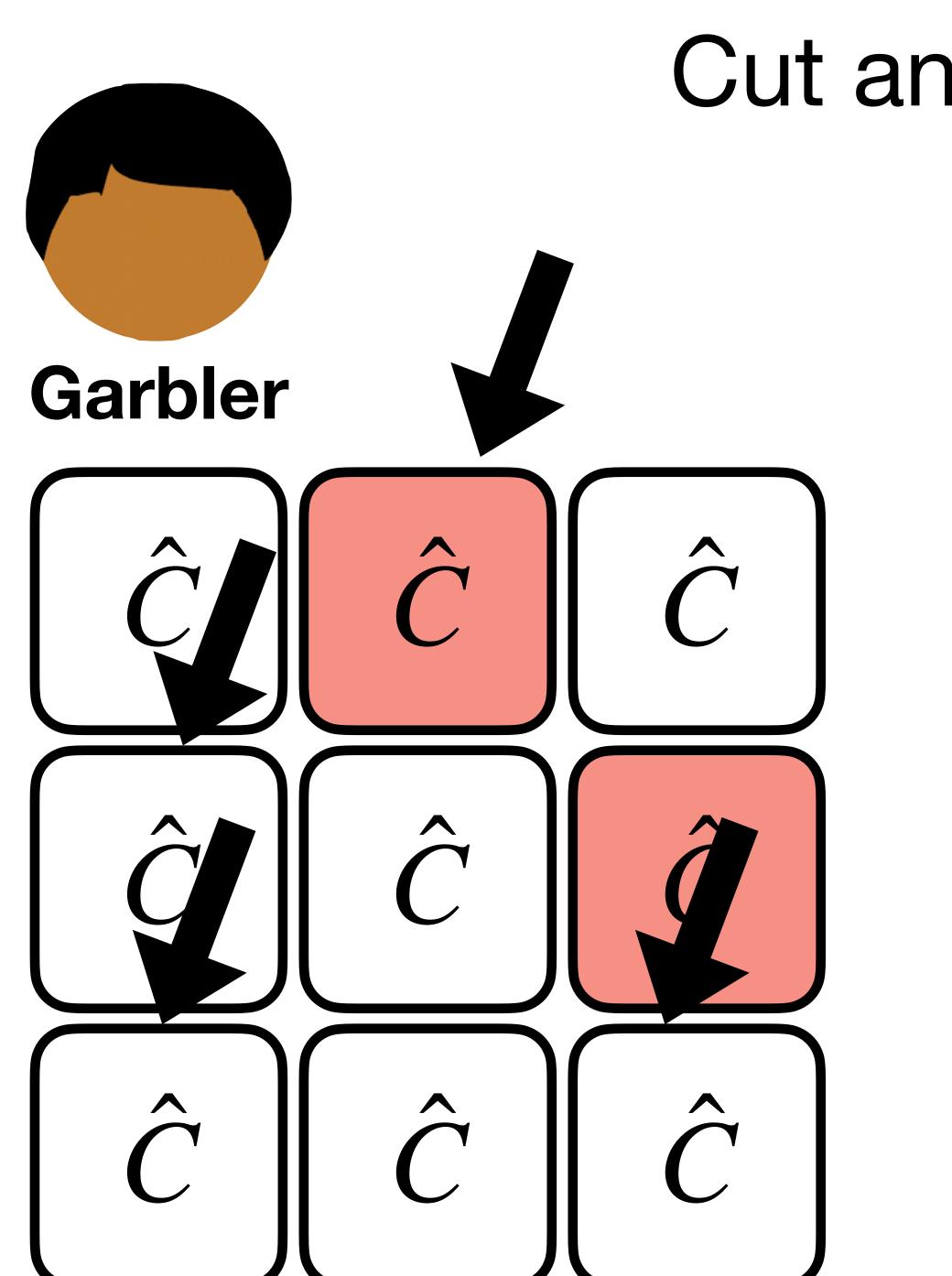






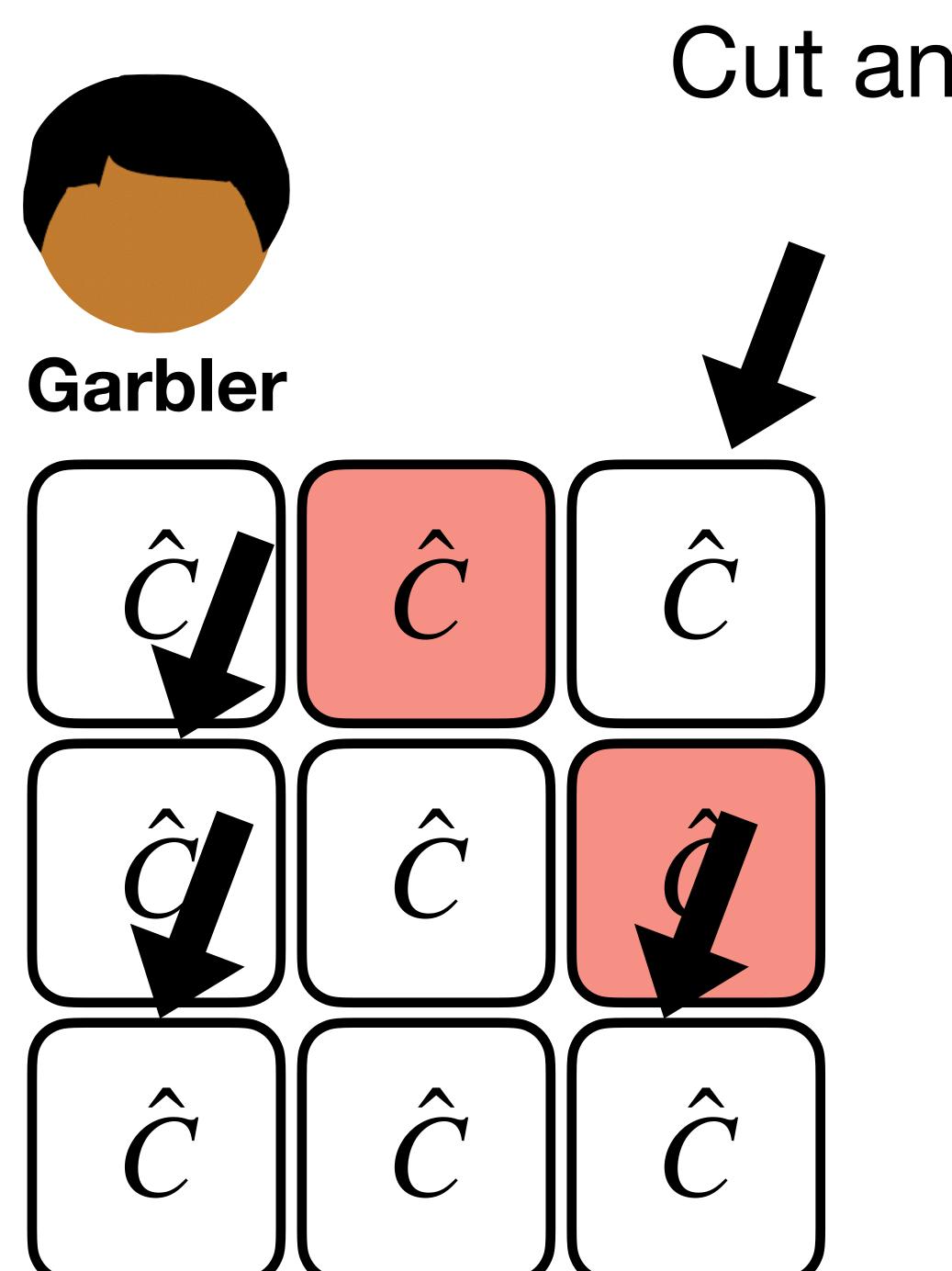






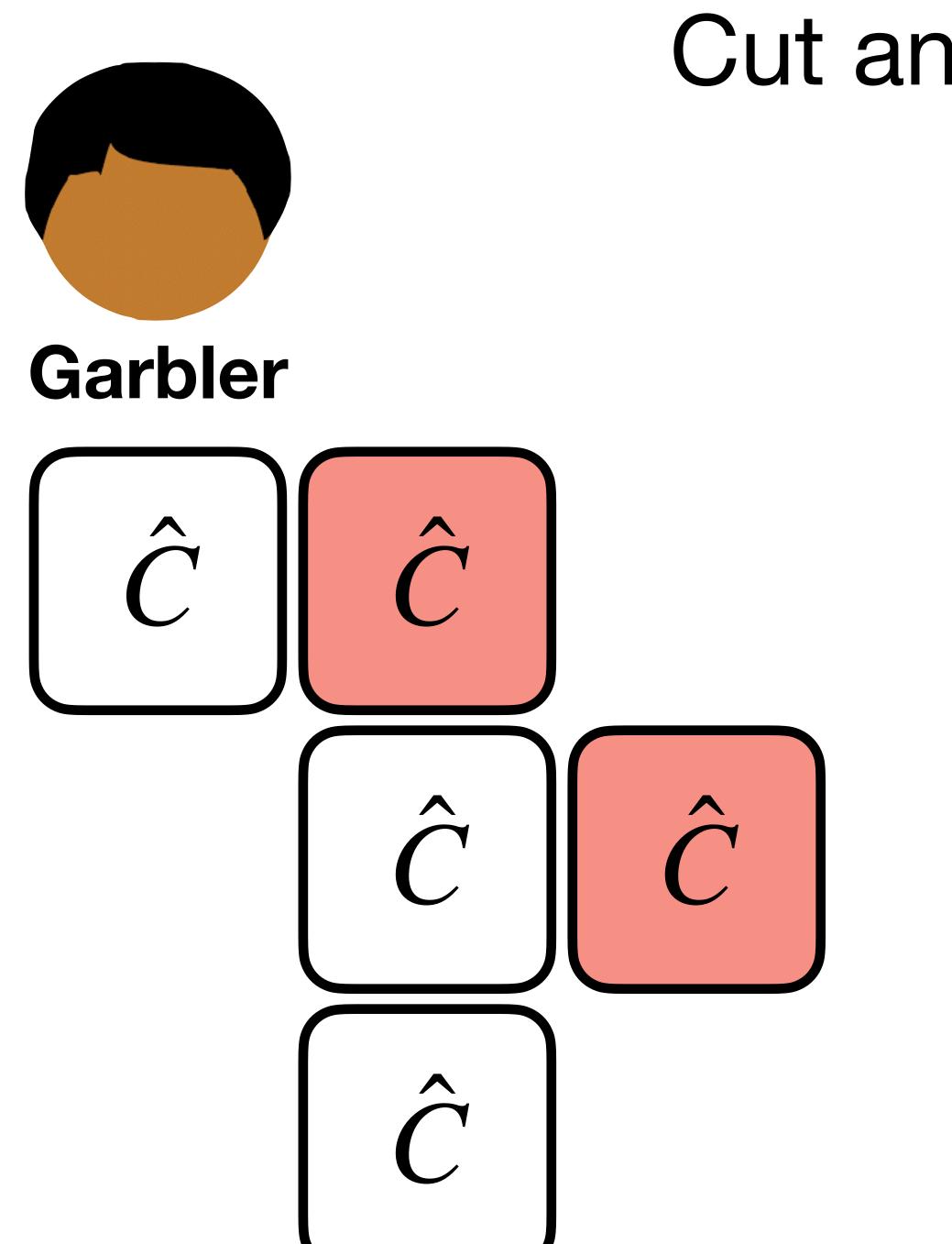


If any opened GC are ill-formed, E aborts



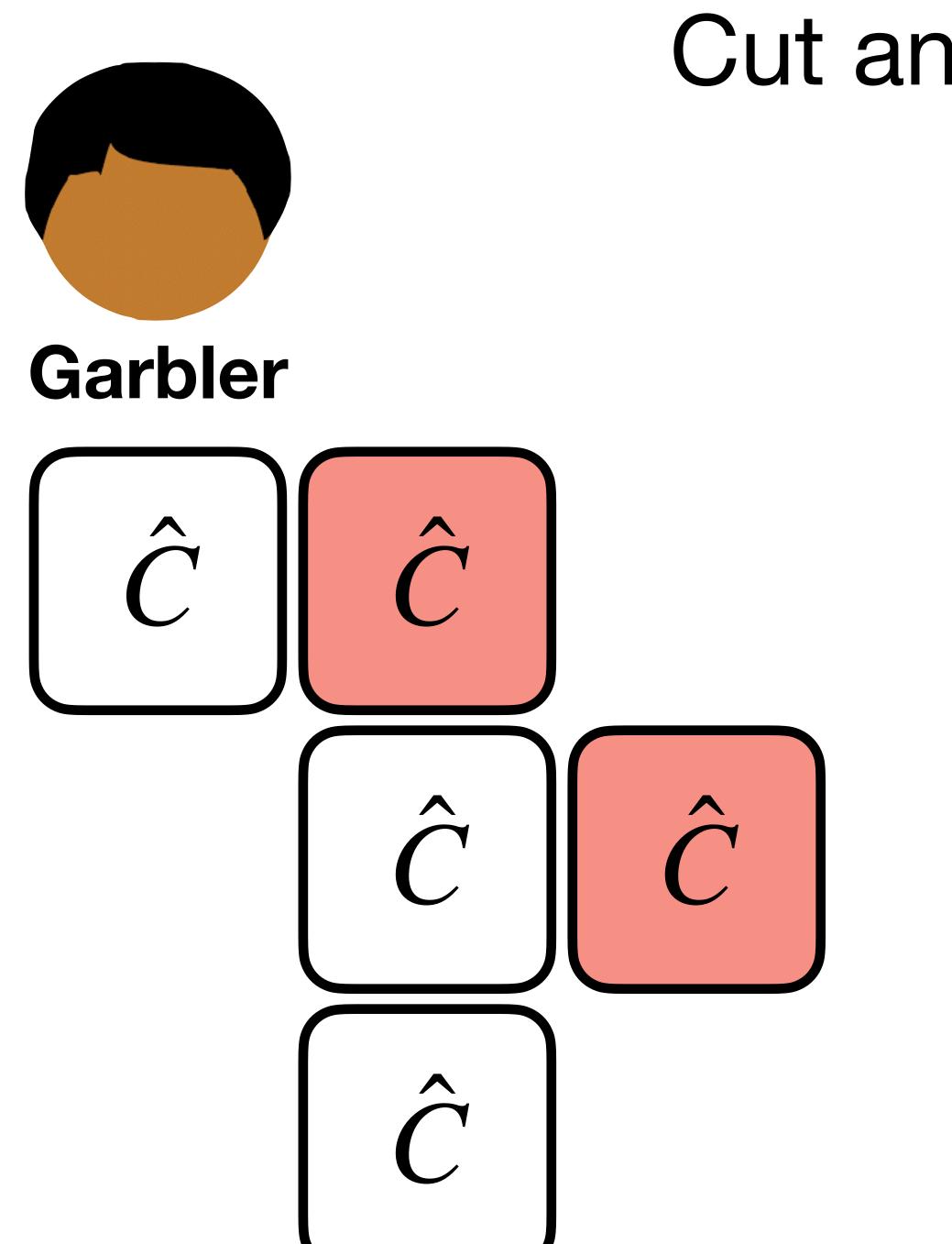


If all opened GC are well-formed, parties continue





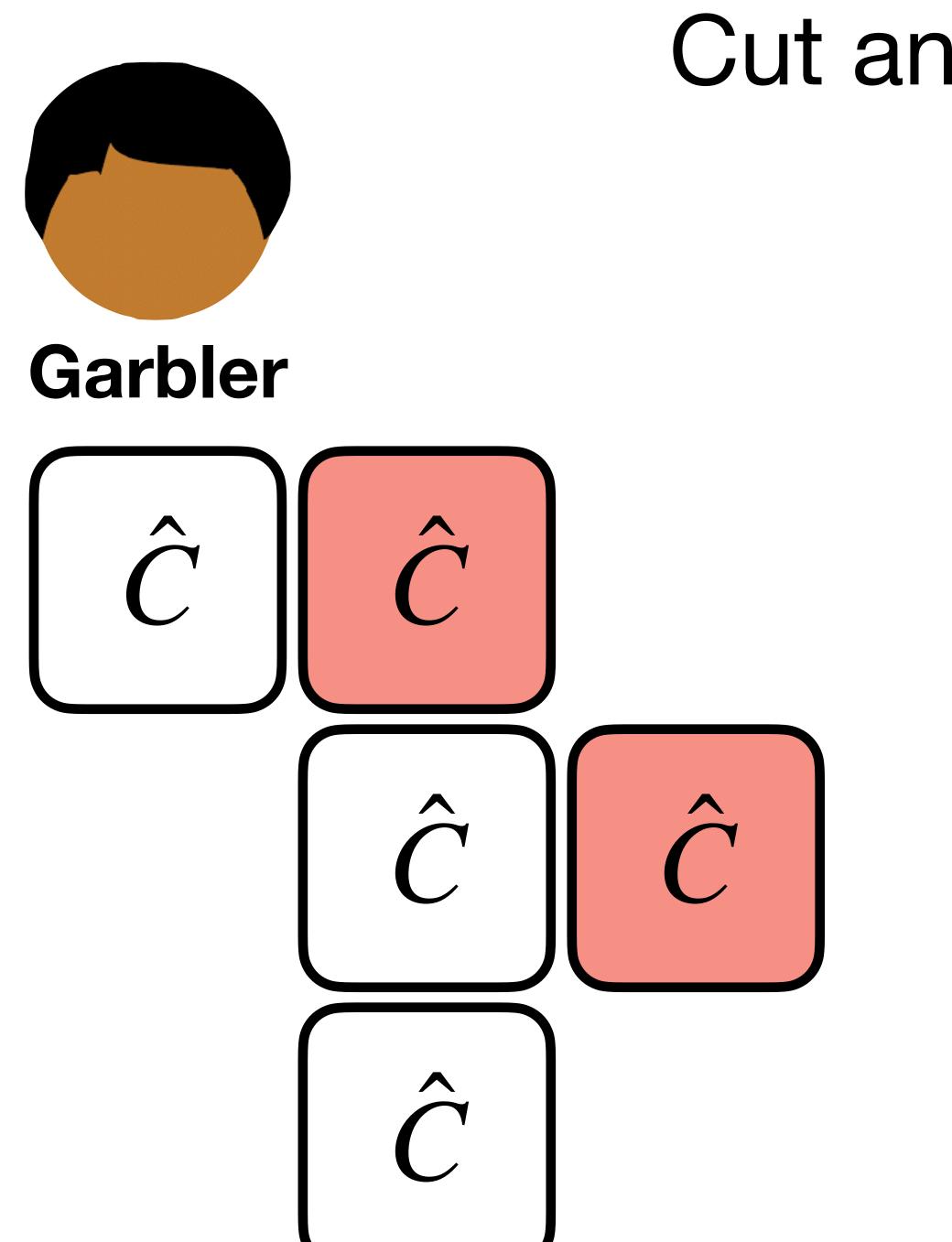
Parties evaluate remaining GCs, and E obtains outputs from each GC





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

Now what?





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.und.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 colline 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 vangxiao@cs.umd.edu

October 10, 2018

Authenticated Garbling from Simple Correlations

Samuel Dittmer^{1[0000-0003-0018-6354]}, Yuval Ishai², Steve $Lu^{1[0000-0003-1837-8854]}$, 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,

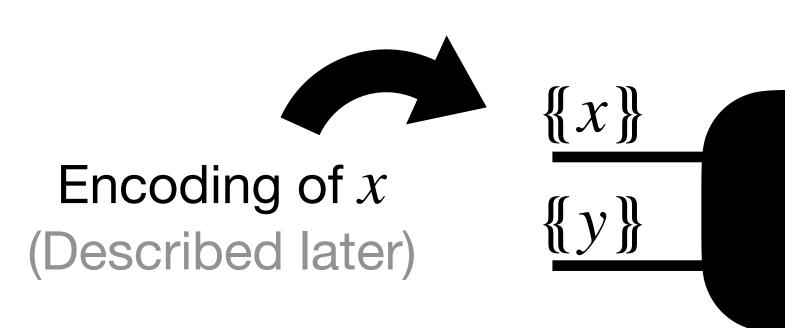
we get the following reduced costs in the random oracle model: – Using variants of both vector oblivious linear evaluation (VOLE)

- 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., 2message) protocol with cost comparable to 8 GCs.

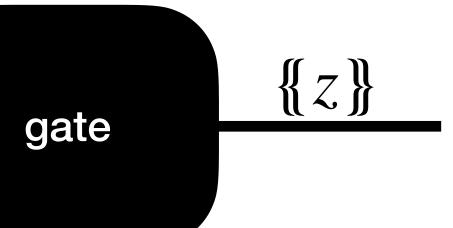
Finally, we show that by using recent constructions of pseudorandom correlation generators (Boyle et al., CCS 2018, Crypto 2019, 2020), the simple correlations consumed by our protocols can be securely realized without forming an efficiency bottleneck.

1 Introduction

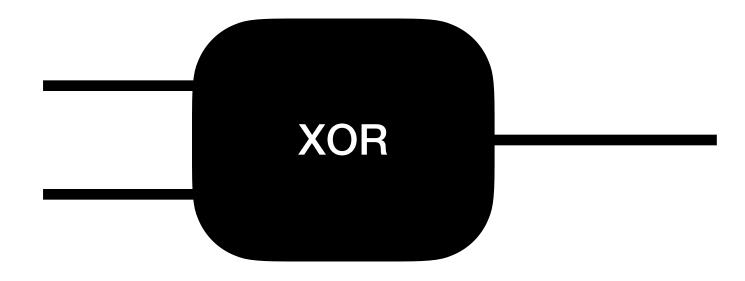
E can tell if a the revealed value is corrupted.



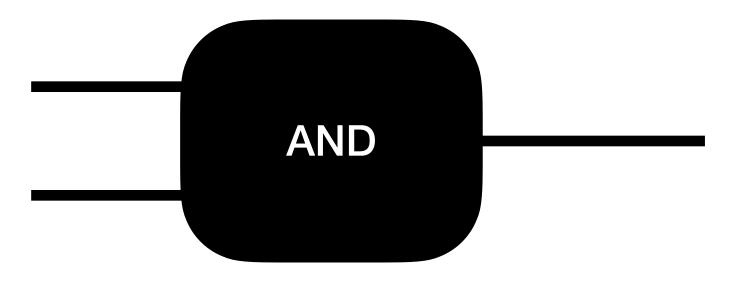
Crucial Insight: use information-theoretic MACs on each wire so that GC can reveal internal values to E.



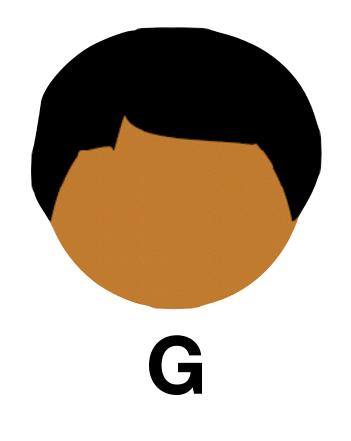
Just like classic GC, gate-by-gate evaluation in constant rounds



However, the technique prevents G from cheating



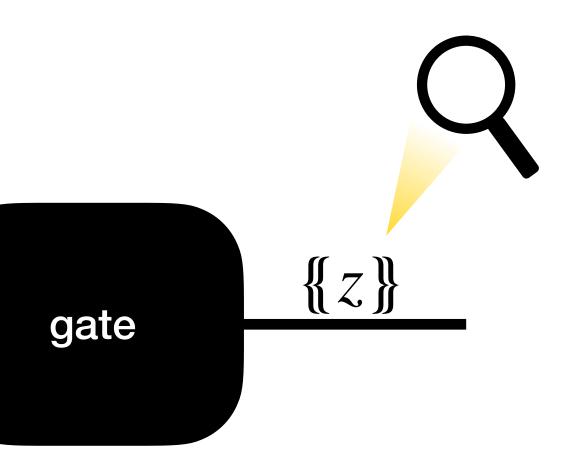
Crucial Insight: add a mechanism by GC can reveal internal values to E. E can tell if a the revealed value is corrupted.





 $\{\!\!\{X\}\!\!\}$

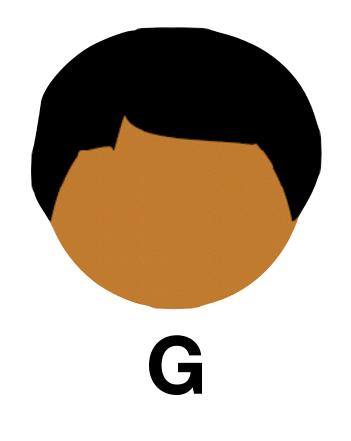
Encoding of *x* (Described later)







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

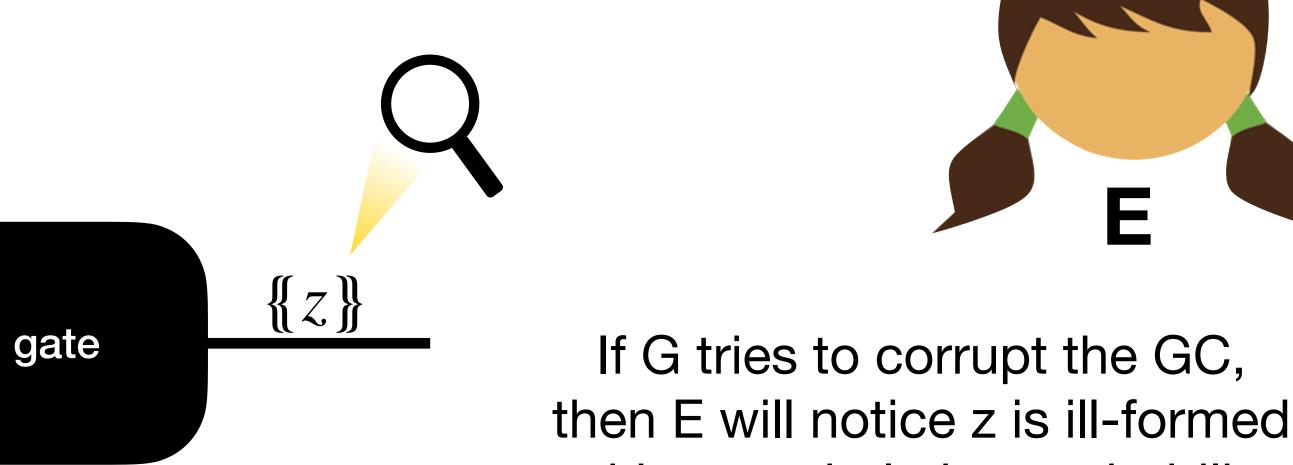




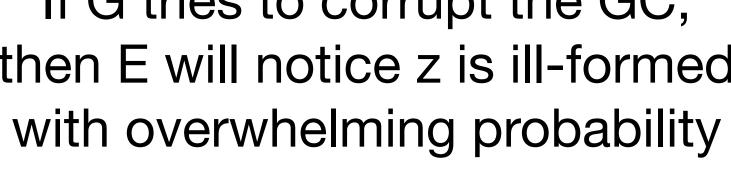
 $\{\!\!\{\boldsymbol{X}\}\!\!\}$

∜ *Y* }}

Encoding of *x* (Described later)





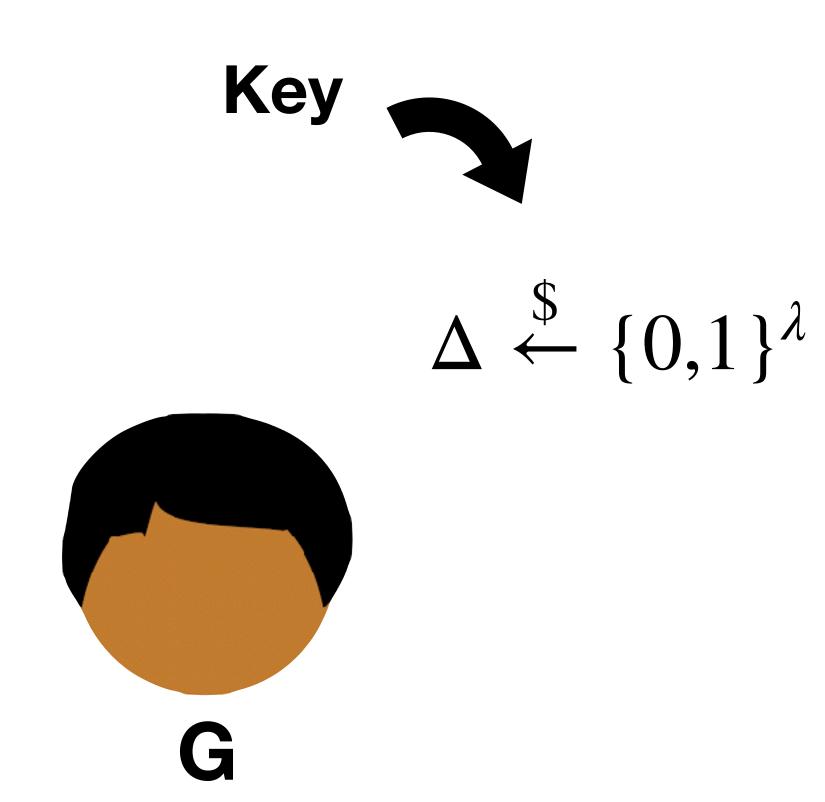


 $\{\!\!\{X\}\!\!\}$

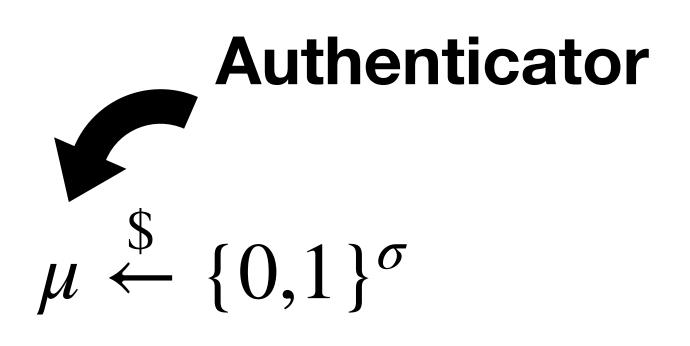






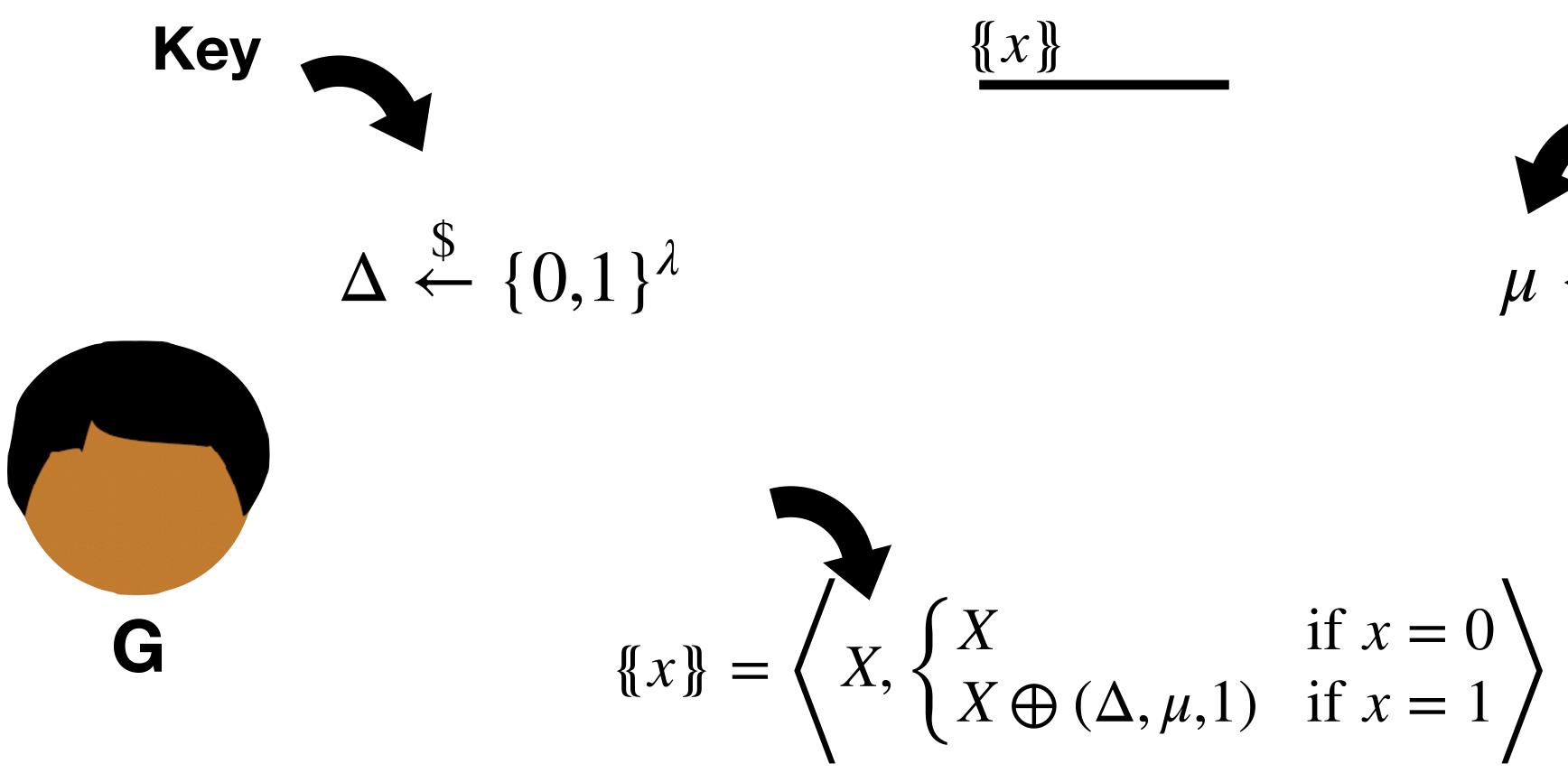


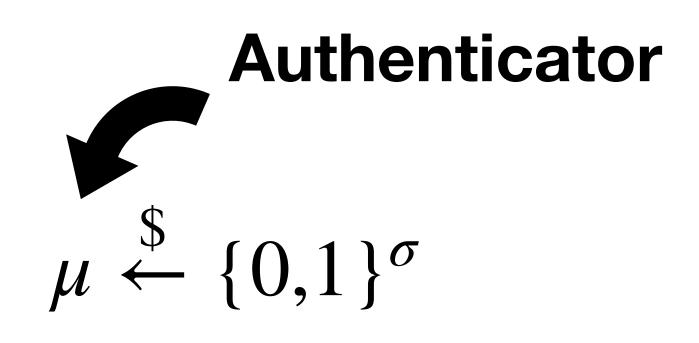
 $\{\!\!\{X\}\!\!\}$





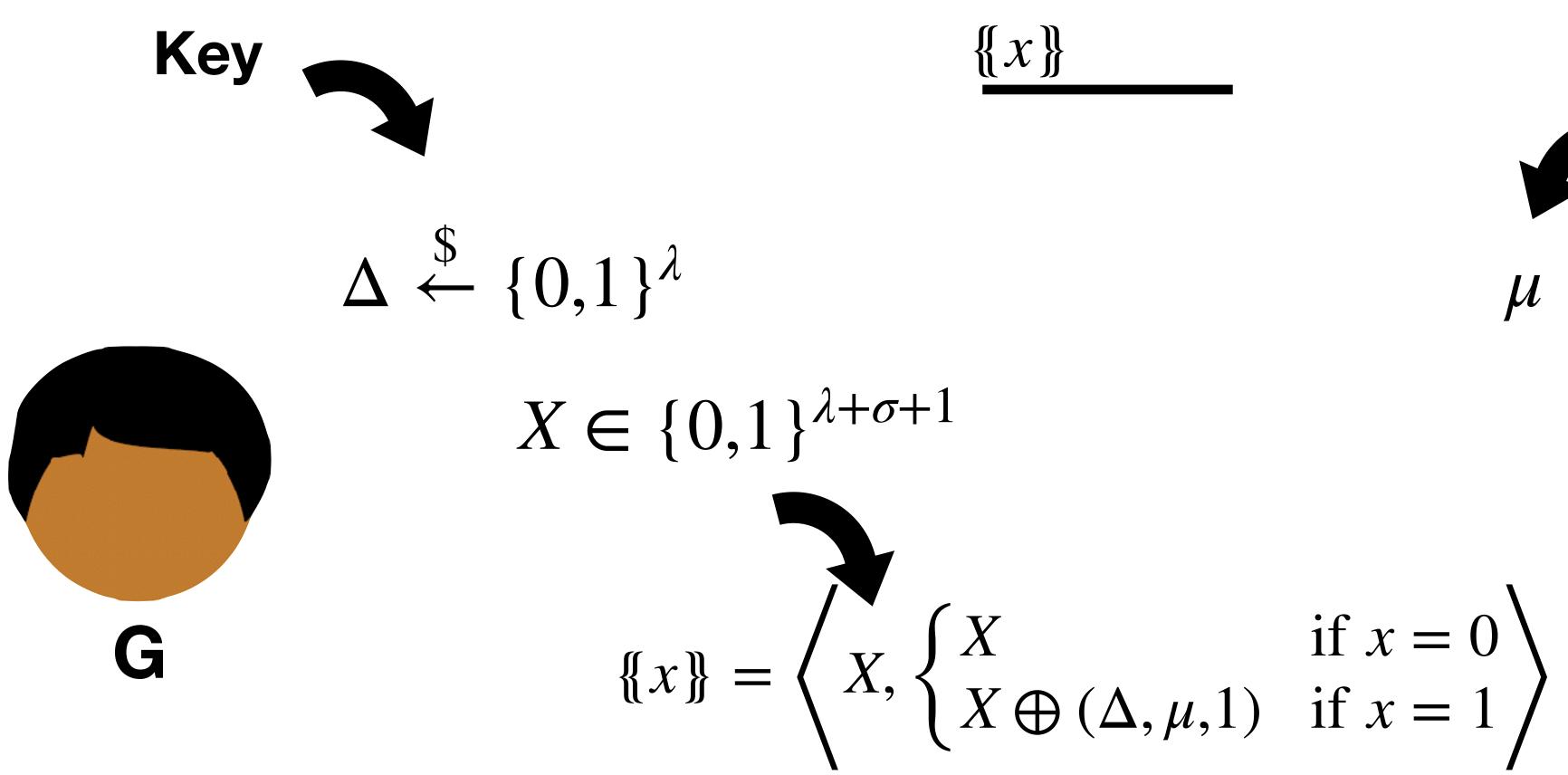


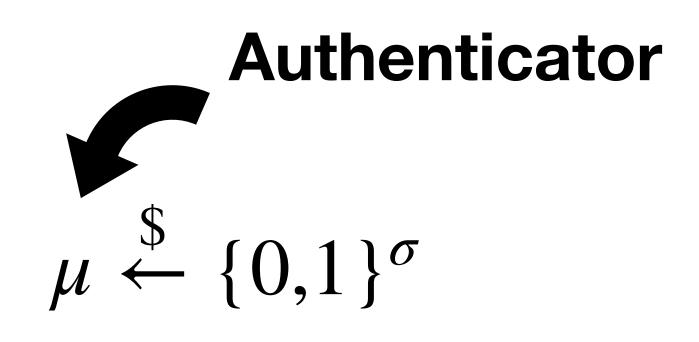






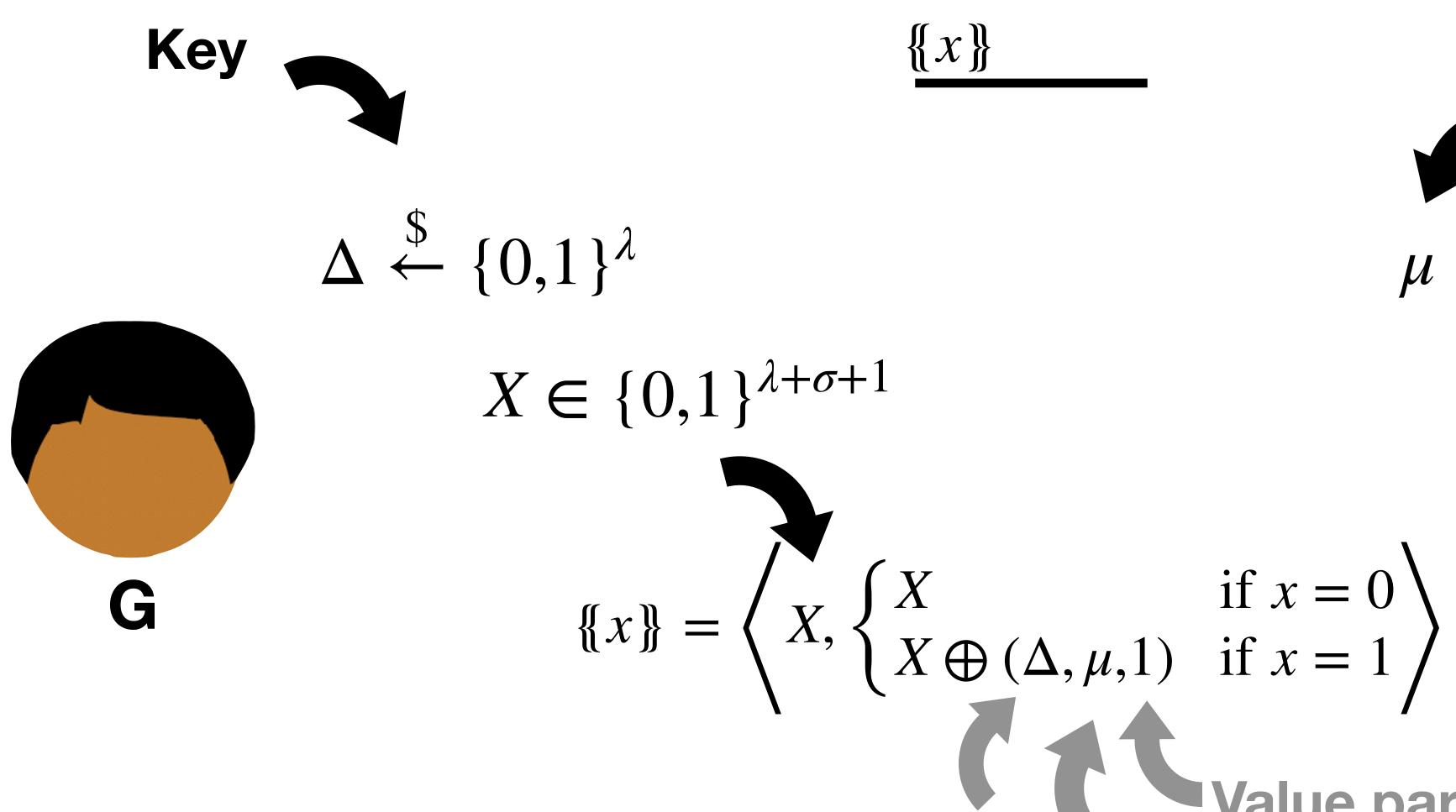




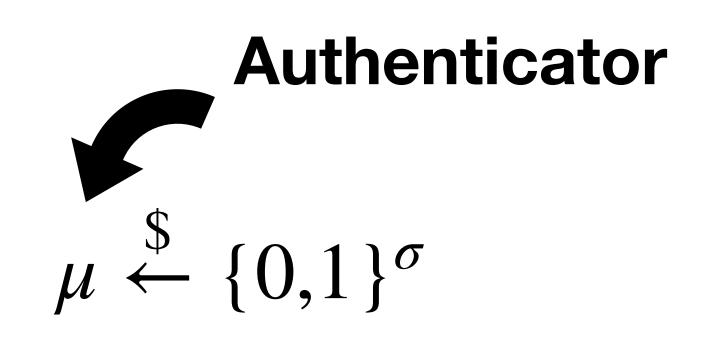








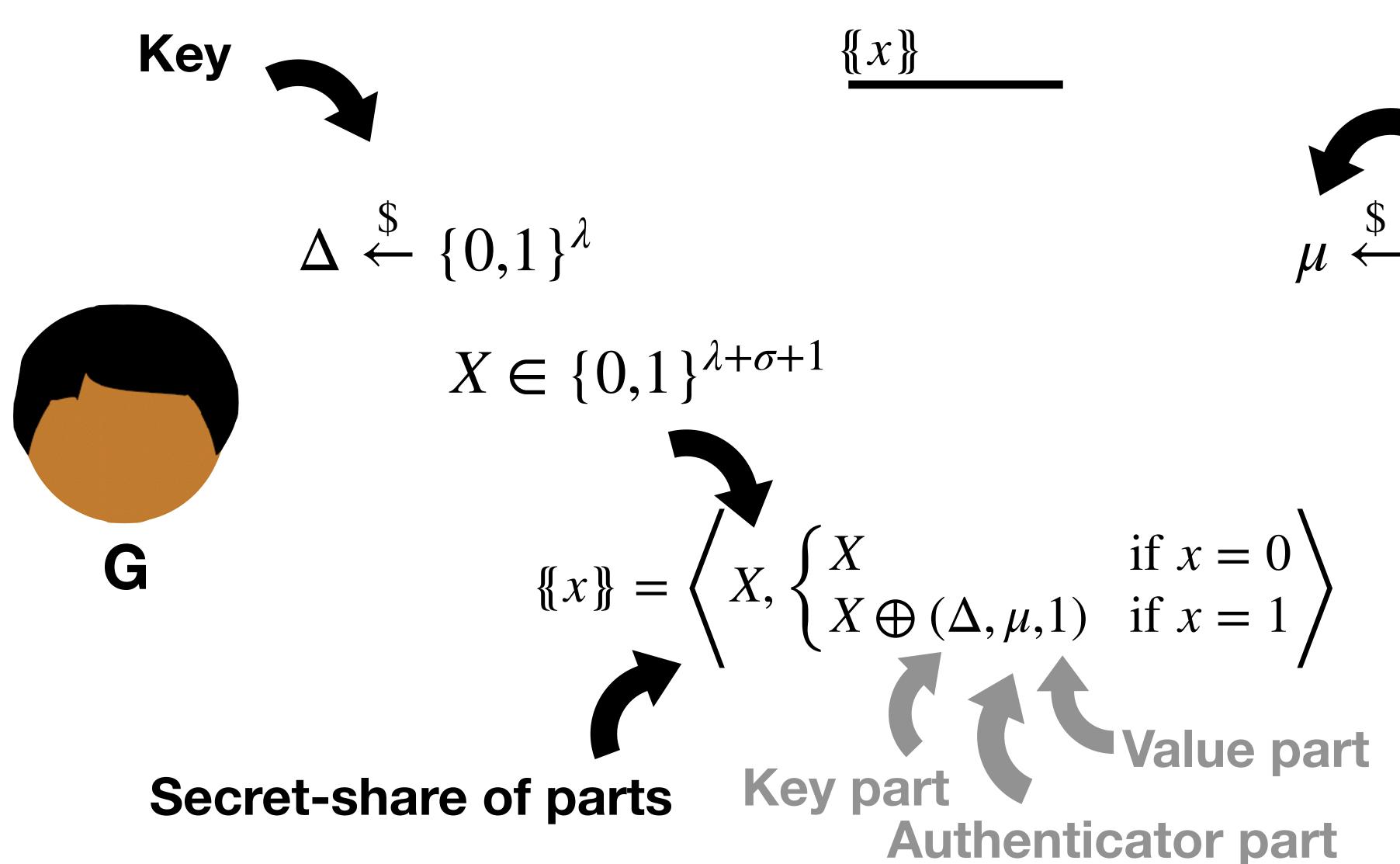
Key part





Value part **Authenticator part**



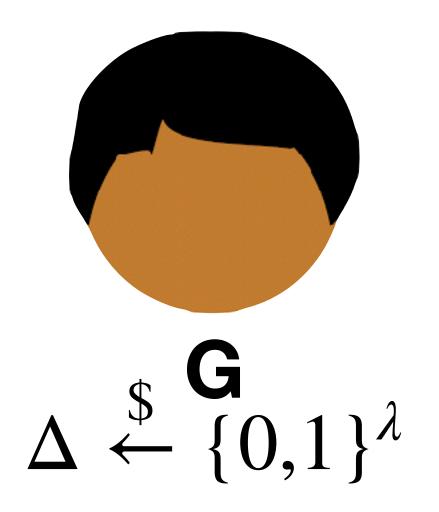






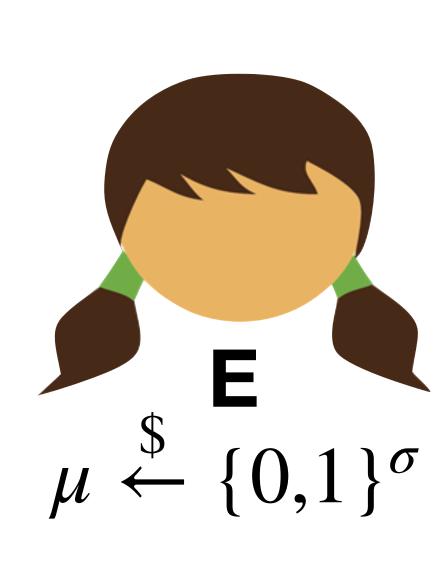
F

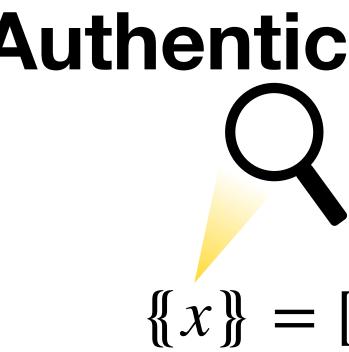
 $\{\!\!\{x\}\!\!\} = [x \cdot \Delta, x \cdot \mu, x]$



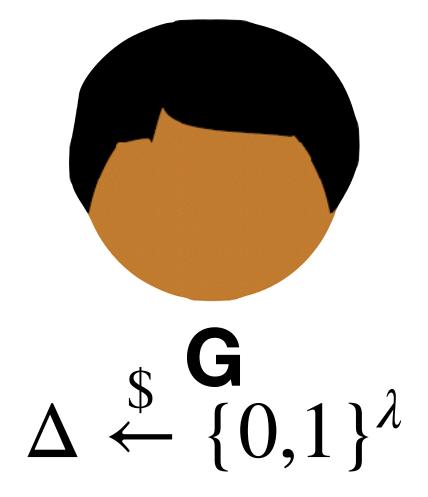
Key part

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





open authenticator, value



Key part

Authenticated Garbling

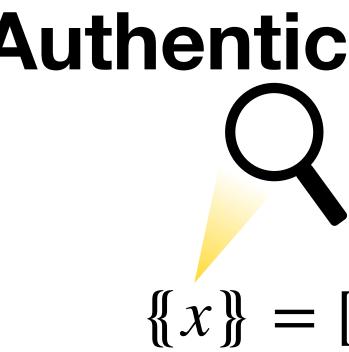
 $\{\!\!\{x\}\!\!\} = [x \cdot \Delta, x \cdot \mu, x]$



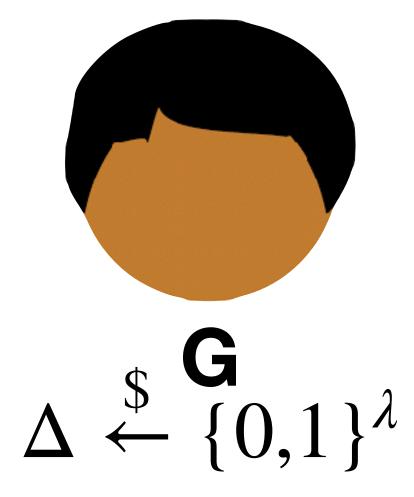
 $x \cdot \mu, x$

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





open authenticator, value

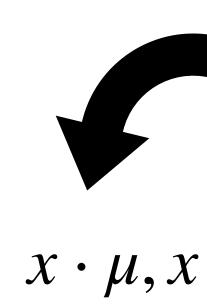


Key part

Authenticated Garbling

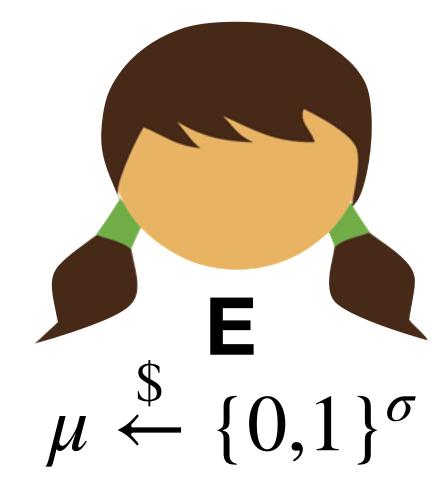
 $\{\!\!\{x\}\!\!\} = [x \cdot \Delta, x \cdot \mu, x]$

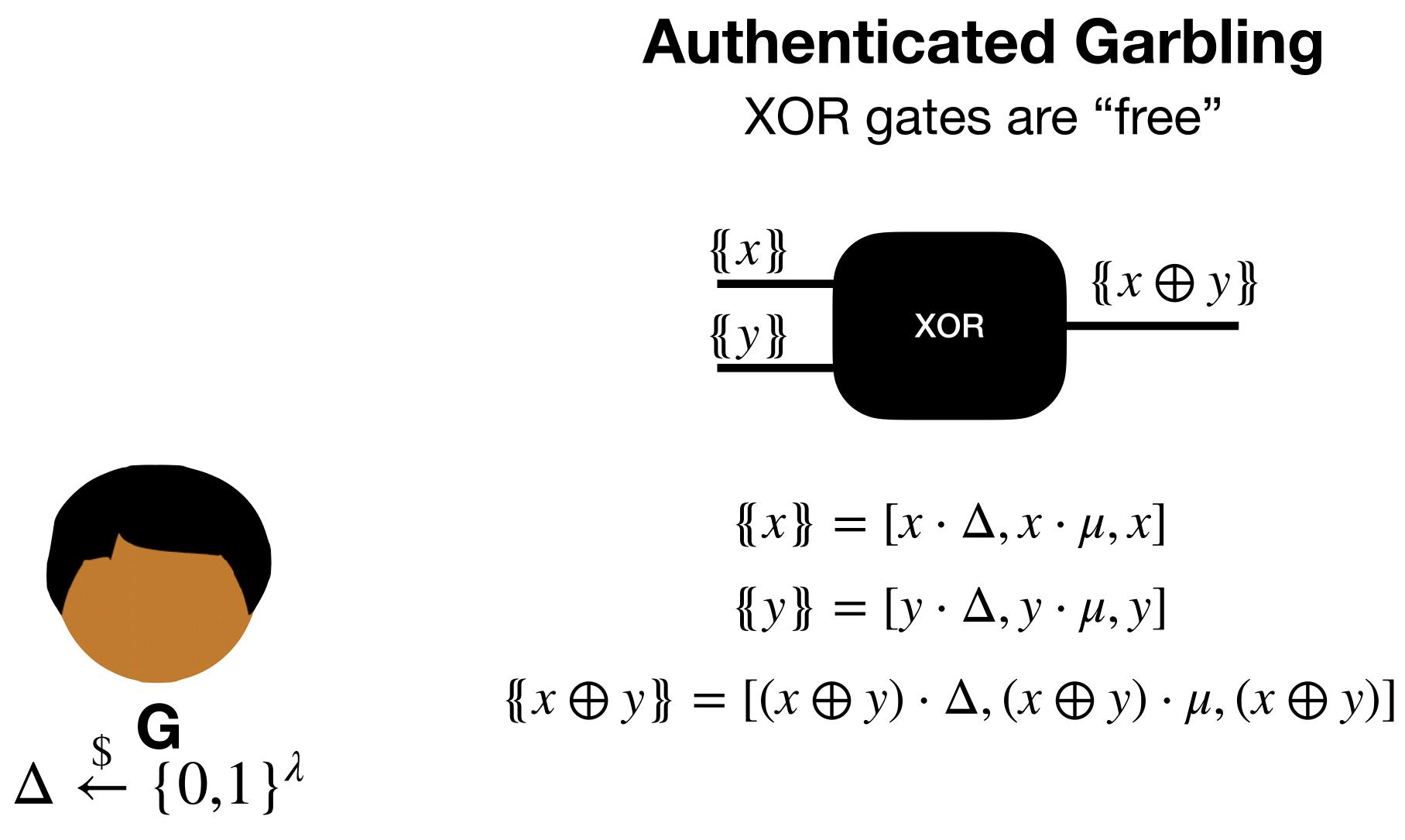


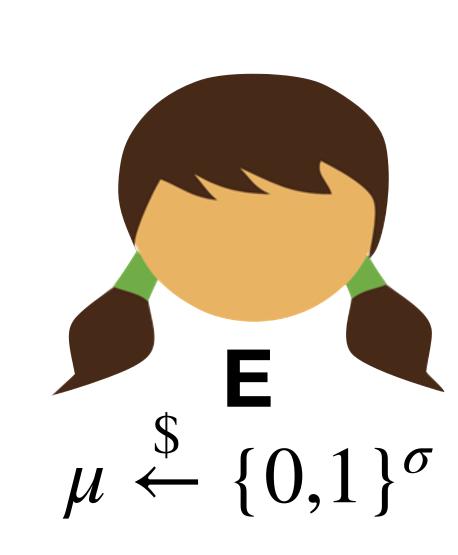


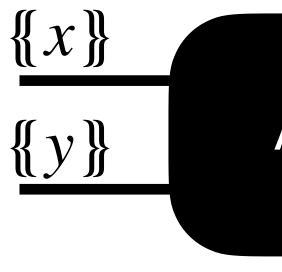
G cannot flip bit, because G does not know μ

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





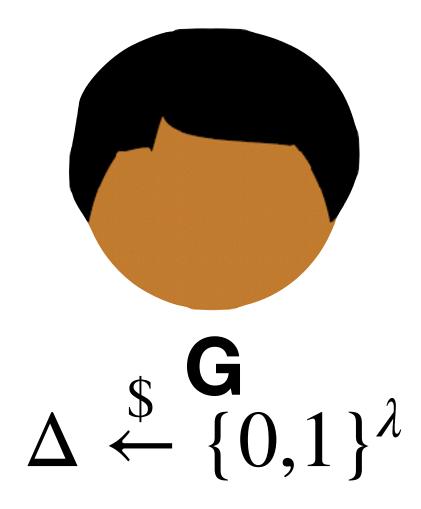




Suppose G and E have access to a **doubly** authenticated multiplication triple

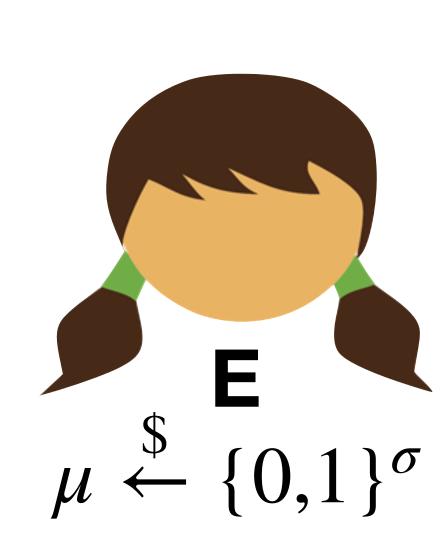
 $\{\!\!\{\alpha\}\!\!\},\{\!\!\{\mu\}\!\!\}$

where α ,

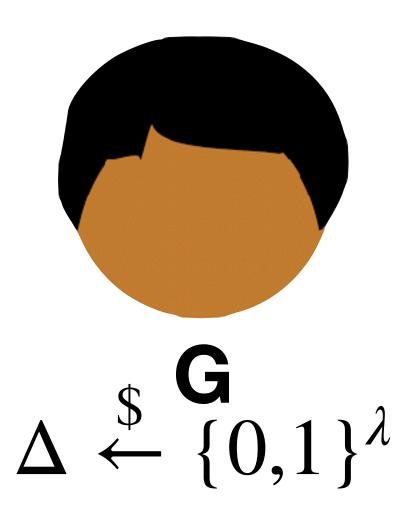


AND
$$\{\!\{x \cdot y\}\!\}$$

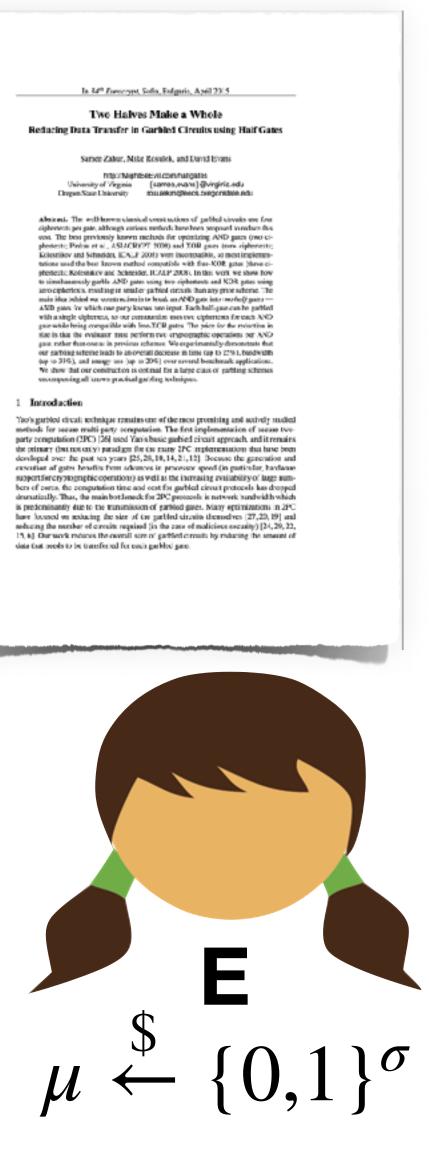
$$\beta \}, \{ \{ \alpha \cdot \beta \} \}$$
$$\beta \stackrel{\$}{\leftarrow} \{ 0, 1 \}$$



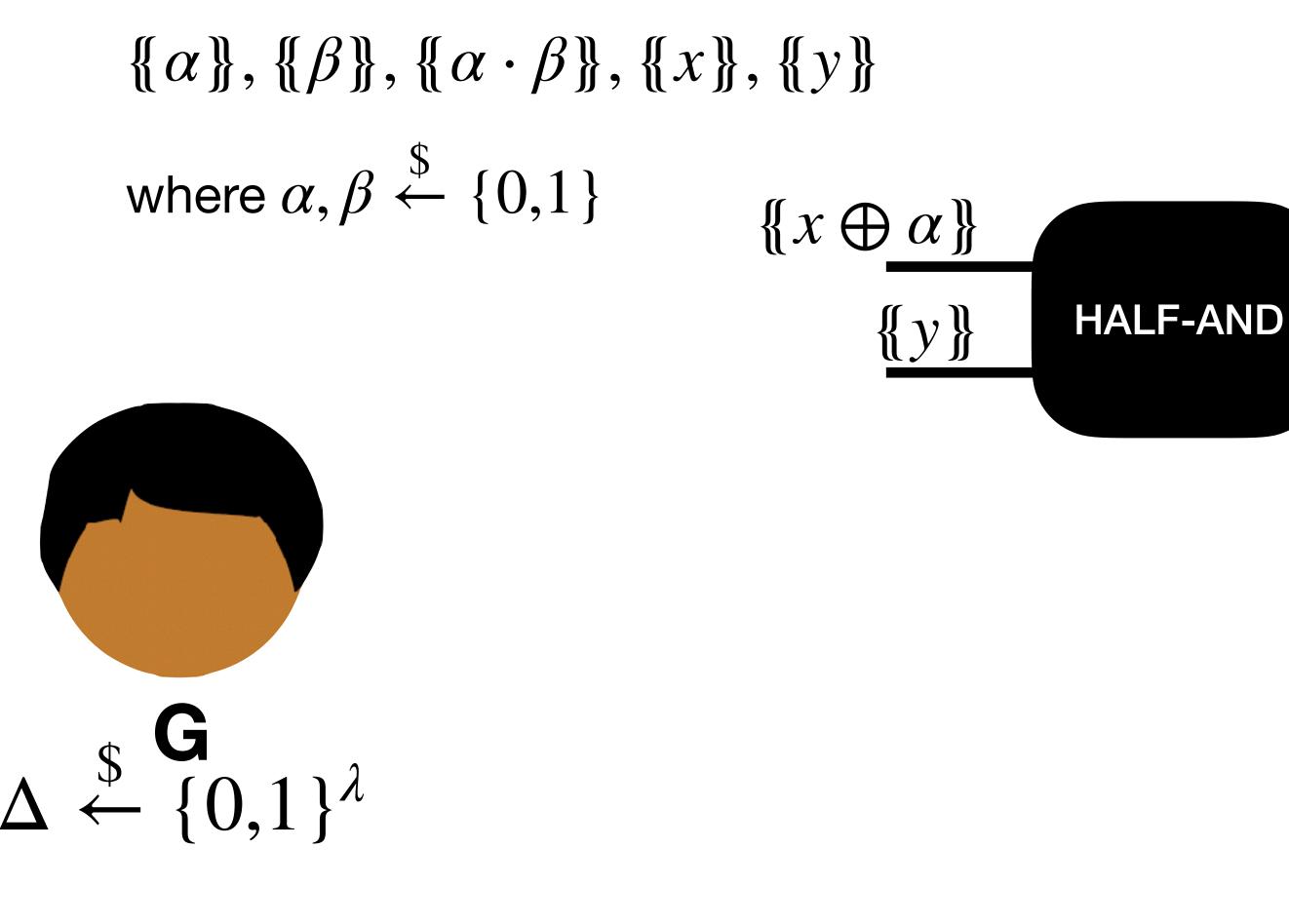
$\{\!\{\alpha\}\!\}, \{\!\{\beta\}\!\}, \{\!\{\alpha \cdot \beta\}\!\}, \{\!\{x\}\!\}, \{\!\{y\}\!\}\}$ where $\alpha, \beta \stackrel{\$}{\leftarrow} \{0, 1\}$



Abstract. The well-heaven classical constructions of publical circuits use for tations used the best known method compatible with free-KOR gates (dave a min idea (which we construction is to bead, an AND gate into two hely gate. AND gates for which one party knows one input. Each half-gate can be gathle



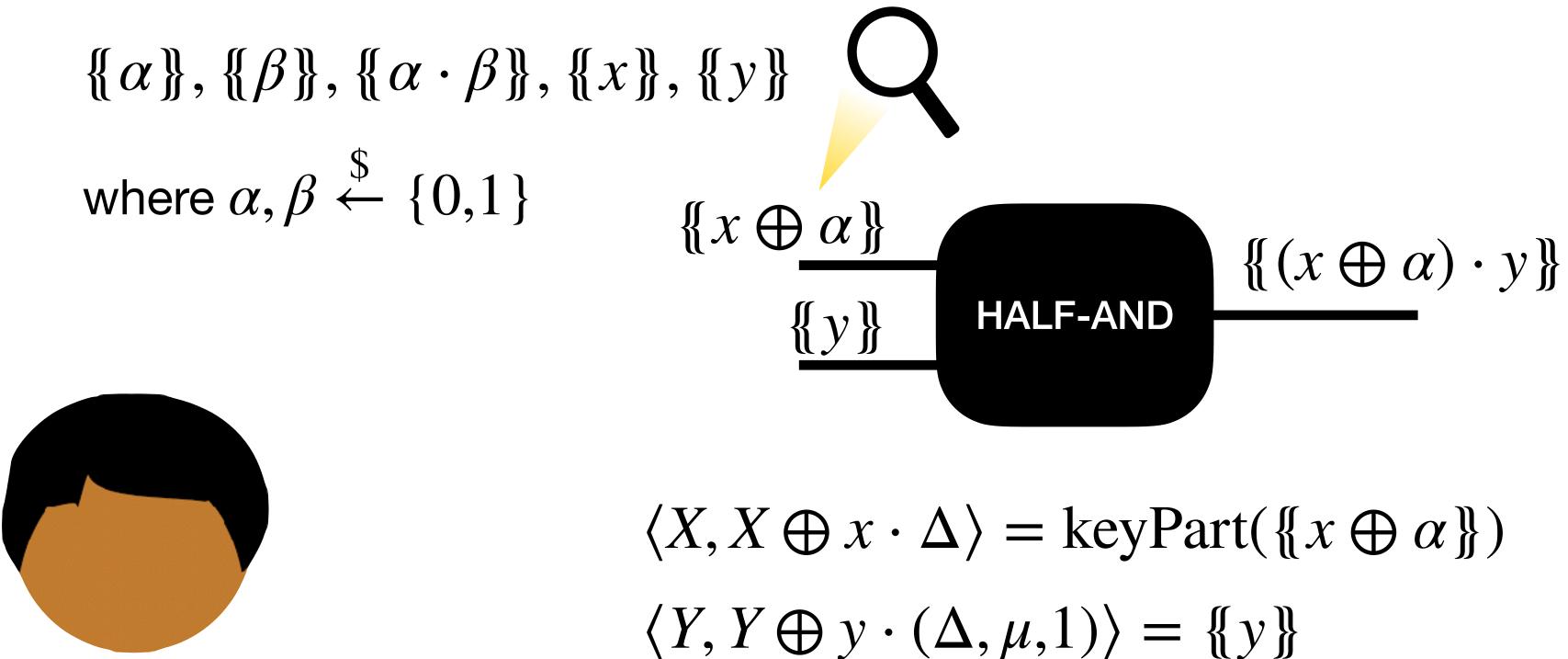
Observe: $(x \oplus \alpha) \cdot y \oplus (y \oplus \beta) \cdot \alpha \oplus \alpha \cdot \beta = x \cdot y$



abstract. The well-brown classical constructions of publical circuits use for main idea tehind wa constructivatis te boah an AND gate into iwo/holy gates — AND gates for which one party knows are input. Each holf-gate can be gathed



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

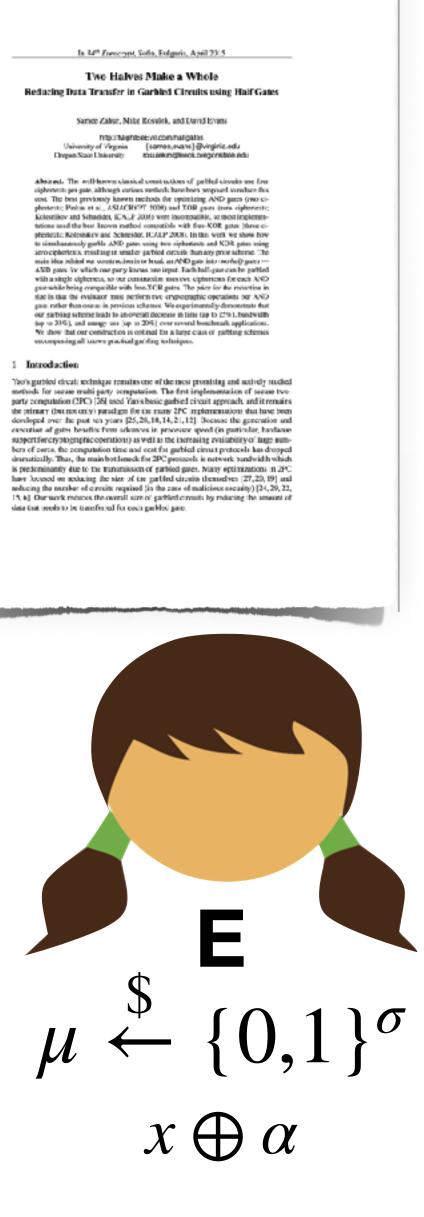


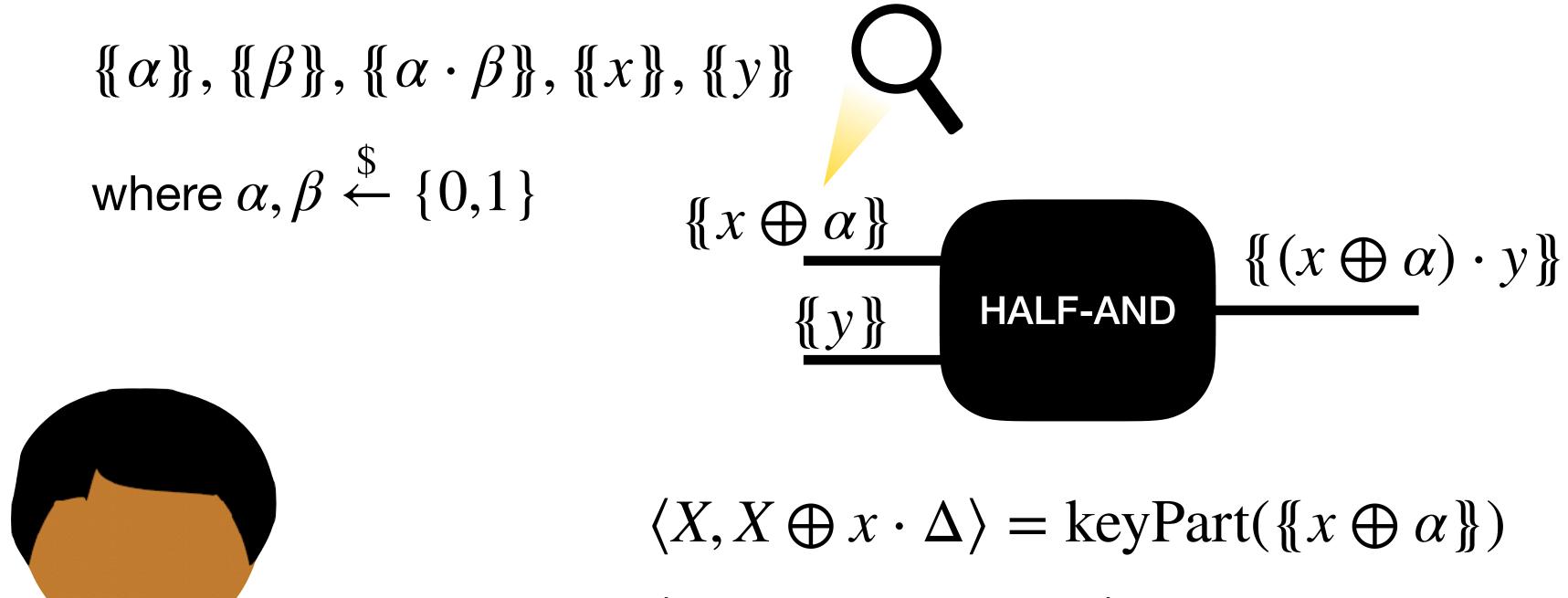


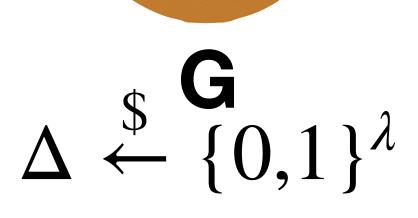
In 34th Euroceppt, Solia, Eulgaria, April 23:5

http://www.ever.com/harcatic

abstract. The well-herews classical constructions of publical circuits use for tations used the best known method compatible with free-KOR gates (dave a tain idea (whited war construction is to break an AND gate into morbed) gate AND gates for which one party knows are input. Each half-gate can be gathle







Enc(X, Z) $Enc(X \oplus \Delta, Y \oplus Z)$ In 34th Euroceppt, Sofia, Eulgaria, April 2315

Two Halves Make a Whole Reducing Data Transfer in Garbled Circuits using Haif Gates

> Samee Zahur, Nitle Rosalek, and David Evana http://wehtset/il.com/hargalist versity of Virginia [sames,evans]@virginia.edu State Dakenity IbiLakkingReeck.bergonsteeu

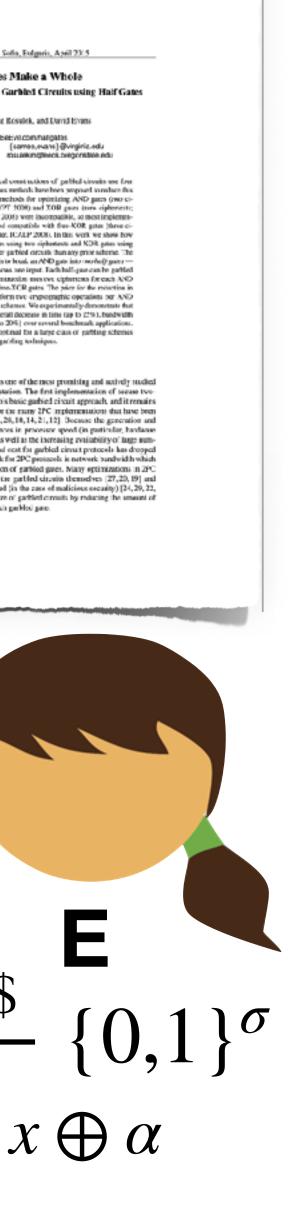
Abstract. The well-haven classical creat actions of pathtal circuits use for ightetests per gate, although cariaus methods have been proposed increduce this cost. The best proviously known methods for optimizing AND gates (two ei-destests: Probas et al., ASLACROPT 1000) and XOB gates (torus eighterests; colestility and Schneider, ICALF 2005) were incompatible, so mest implem stions used the best known method compatible with five-KOR gates three Perfects: Rotestatics and Schreider, ICALP 2003. In this werk we show how in simultaneously garble AND pates using two siphestexts and NDR pates using arrocepterfects, resulting it smaller packed circuits than any pror scheme. The ain idea (whind was construction in to break an AND gate into morbelly gat AND gates for which one party knows are input. Each half-gate can be gath with a single ciphertext, so our construction uses two ciphertents for e-pate while being comparible with lose.XCR gates. The paice for the reize is that the evaluator mass perform two ergpographic operations per AN sie rather than one as in provious schemes. We experimentally demonstrate that or partning scheme heads to an overall decrease in time (ap to 12%), bundwidth top to 33/63, and amonge use (ap to 20%) over reveal benchmark applications. No show that our construction is optimal for a large class of pathing schemes ncompassing all known practical garbling techniques

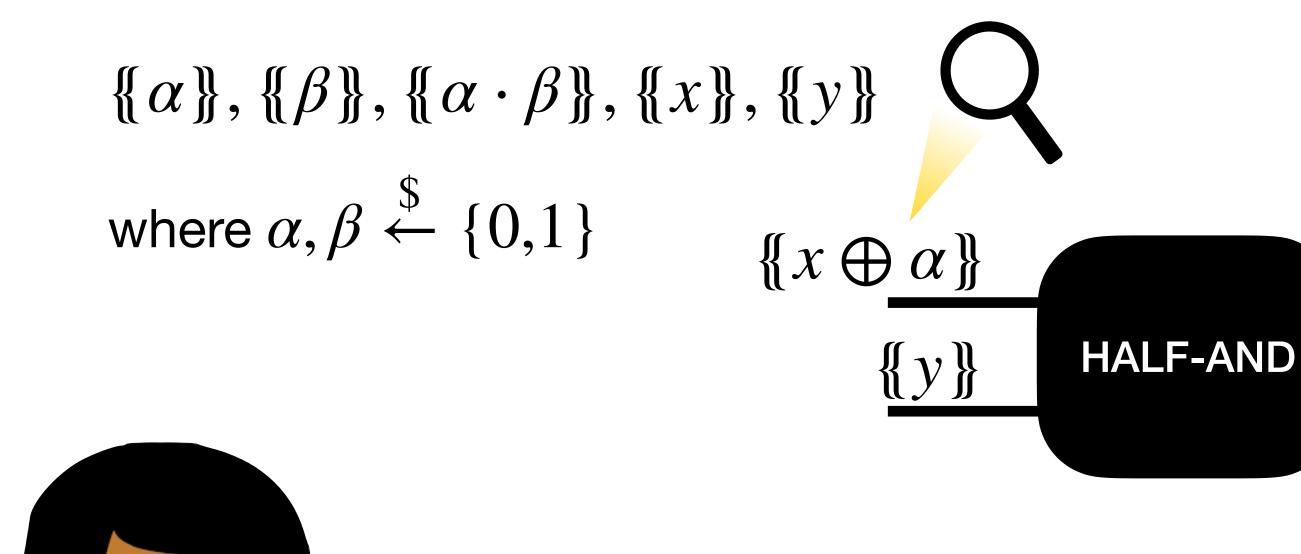
1 Introduction

Yao's garbled circuit technique remains one of the most promiting and actively studied nortioda for secue multi party comparation. The first implementation of secue two party computation (2PC) [26] used Yao's basic gashied circuit approach, and it remains te primary (burnoron'y) paradigm for the many 2PC implementation; that have been evidoped over the past yeary [25, 28, 19, 14, 21, 12]. Docume the generation and executian of gates benefits from advances in processor speed (in particular, hardware opertifier cryptographic operations) as well as the increasing availability of large numers of perce, the computation time and cost for garbled circuit protocels has dropped dramatically. Thus, the main botheneck for 2PC protocols is network sandwidth which is predominantly due to the transmission of parbled gates. Many optimizations in 2PC have locused on nodacing the size of the partield situation themselves (27,22), (9) and notacing the number of curvaits required (in the cases of multicleus sociality) [24,29,22, 15, 4]. Our work measure the ownall sam of garbled cureants by reducing the amount of ata that needs to be transferred for each parkled gate

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







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

Enc(X, Z) $Enc(X \oplus \Delta, Y \oplus Z)$ In 34th Euroceppt, Sofia, Eulgaris, April 2015

http://wehtset/il.com/hargalist

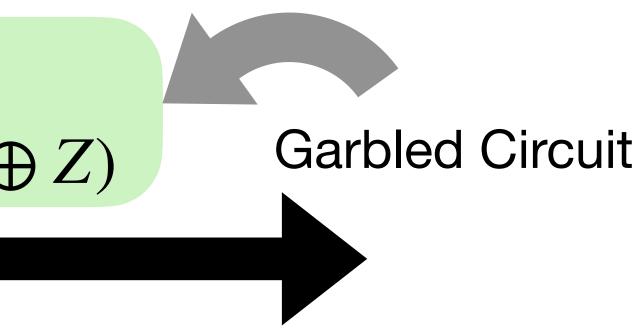
detent. The will have classical constructions of pathed circuits use for colestility and Schneider, ICALF 2005) were incompatible, so mest implem tions used the best known method compatible with free-KOR gates (dave Perfectly: Rolessmacy and Schreider, ICALP 2008, In this work we show how is simultaneously garlia AND pates using two riphentexts and KDR pates using ero-ephertexis, resulting in smaller garbied circuits, humany prior scheme. The in idea (whind war construction is to break an AND gate into morbelly gat AND gates for which one party knows are input. Each half-gate can be gath with a single eighteness, so our construction uses two eight pase while being comparible with loss XCR gates. The pa ize is that the evaluator must perform two ergpographic operations per AN sie rather faan one as in provious schemes. We experimentally demonstrate fa

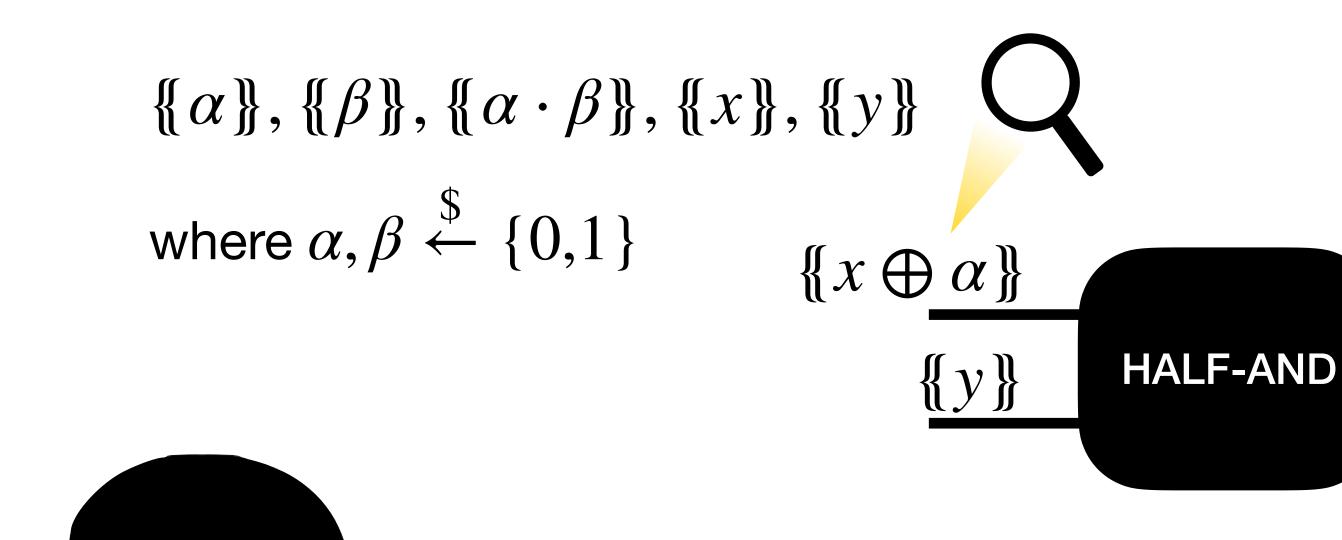
ata that needs to be transferred for each parkled gate



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

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





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

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

http://wehtset/il.com/hargalist

Abstract. The well-haven classical creat actions of pathtal circuits use for colestility and Schneider, ICALF 2005) were incompatible, so mest implem stions used the best known method compatible with five-KOR gates three ain idea (whind was construction in to break an AND gate into morbelly gat AND gates for which one party knows one input. Each half-gate can be garble with a single cipherness, so our construction uses over ciphe pase while being comparible with loss XCR gates. The pair ize is that the evaluator mass perform two ergpographic operations per AN sie rather than one as in provious schemes. We experimentally demonstrate that

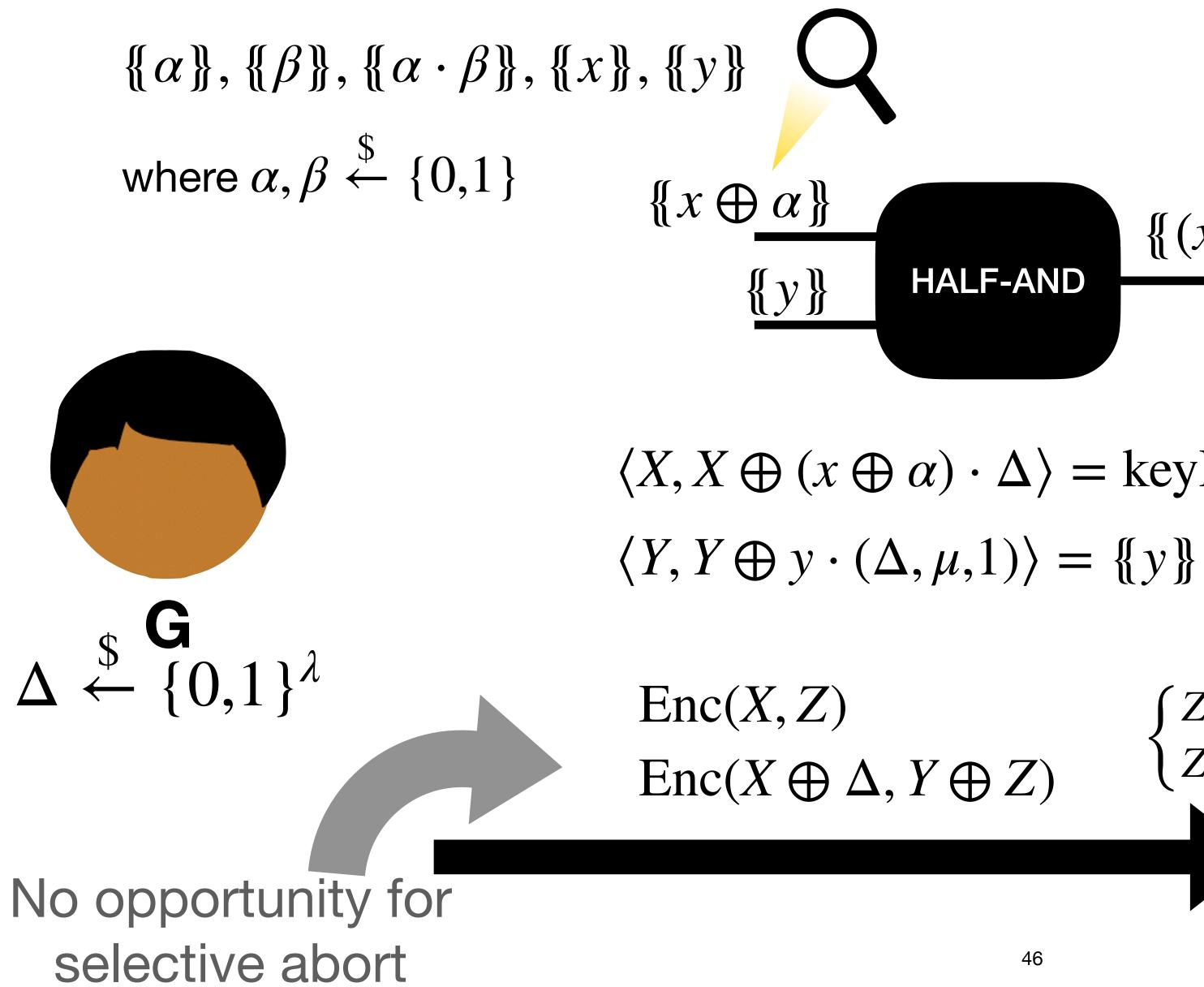
ata that needs to be transferred for each parkled gate

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

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

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





In 34th Euroceppt, Sofia, Eulgaris, April 2015

Two Halves Make a Whole Data Transfer in Garbled Circuits using Half Gate

> samee Zahur, Nilse Rosulek, and David Evana http://wehtset/il.com/hargalist sky of Virginia (samos,evans)@virginia.edu Taur Eniversity totukers@excl.biegcreatee.or

detent. The will have classical constructions of pathed circuits use for ightetests per gate, although cariaus methods have been proposed increduce this cost. The best proviously known methods for optimizing AND gates (two ei-destests: Probas et al., ASLACROPT 1000) and XOB gates (torus eighterests; colestility and Schneider, ICALF 2005) were incompatible, so mest implem tions used the best known method compatible with free-KOR gates (dave Perfectly: Rolessmacy and Schreider, ICALP 2008, In this work we show how is simultaneously garlia AND pates using two riphentexts and KDR pates using ero-ephertexis, resulting in smaller garbied circuits, humany prior scheme. The ain idea (whind was construction in to break an AND gate into morbelly gat AND gates for which one party knows are input. Each half-gate can be gath with a single eighteness, so our construction uses two eight pase while being comparible with loss XCR gates. The pa ize is that the evaluator mass perform two ergpographic operations per AN sie rather than one as in provious schemes. We experimentally demonstrate that or partning scheme heads to an overall decrease in time (ap to 12%), bundwidth top to 33/63, and amonge use (ap to 20%) over reveal benchmark applications. No show that our construction is optimal for a large class of pathing schemes ncompassing all known practical garbling techniques

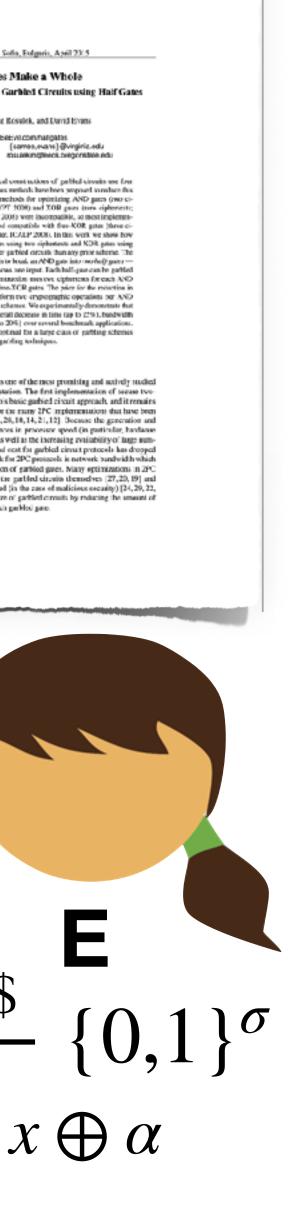
1 Introduction

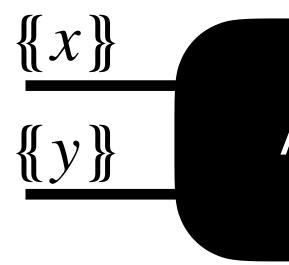
too's garbled circuit technique remains one of the most promiting and actively studied nortioda for secue multi party comparation. The first implementation of secue two party computation (2PC) [26] used Yao's basic gashied circuit approach, and it remains e primary (but not only) paradigm for the many 2PC implementations that have been oveloped over the past ten years [25,28,19,14,21,12]. Because the generation and execution of gates benefits from advances in processor speed (in particular, hardware opertforceyptographic operations) as well as the increasing availability of large sum ers of parts, the computation time and cost for garbled circuit protocols has drooped deursatically. Thus, the main botheneck for 2PC protocols is network sandwidth which is predominantly due to the transmission of parblod gates. Many optimizations in 2PC have locused on nodacing the size of the partield situation themselves (27,22), (9) and notacing the number of curvaits required (in the cases of multicleus sociality) [24,29,22, 15, 4]. Our work measure the ownall sam of garbled cureants by reducing the amount of ata that needs to be transferred for each parkled gate

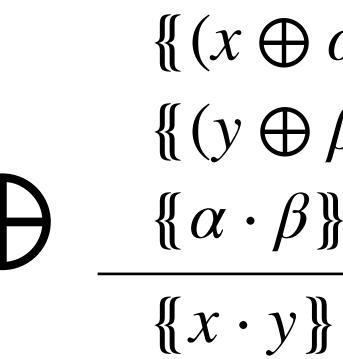
 $\{\!\!\{(x \oplus \alpha) \cdot y\}\!\!\}$

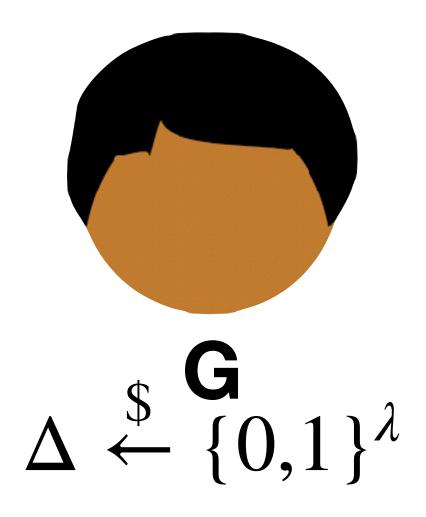
 $\langle X, X \oplus (x \oplus \alpha) \cdot \Delta \rangle = \text{keyPart}(\{\!\!\{x \oplus \alpha\}\!\})$

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





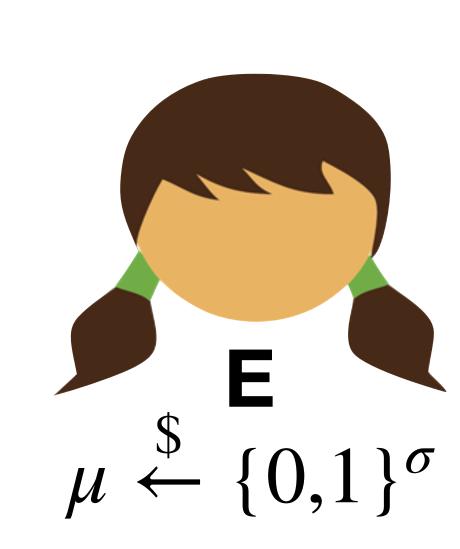


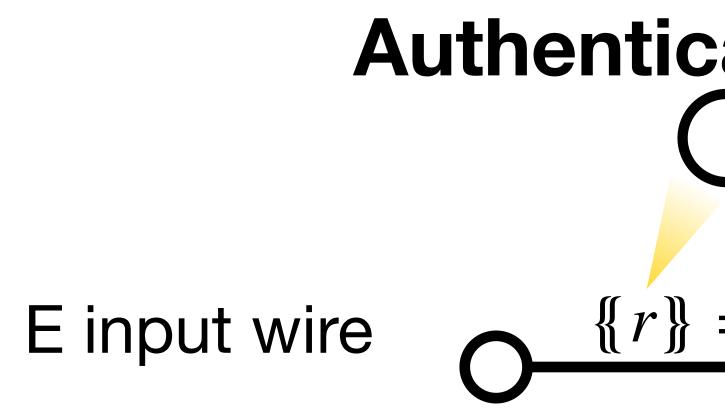


$$\{\!\!\{x \cdot y\}\!\!\}$$

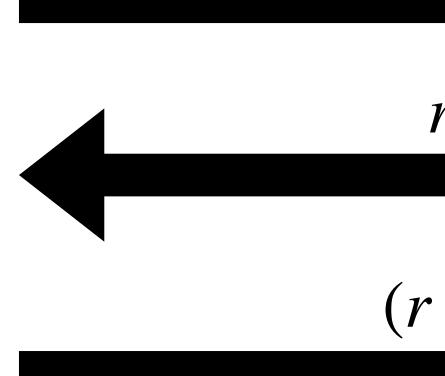
$$\left. \begin{array}{c} \left. \left. \left. \left. \left. \right\} \right\} \right\} \right\} \right\} \right\}$$

$$\left. \left. \left. \left. \left. \right\} \right\} \right\} \right\}$$





Open authenticator part, value part $r \cdot \mu, r$ $r \bigoplus x$ ${\mathcal X}$ $(r \oplus x)\Delta$ μ

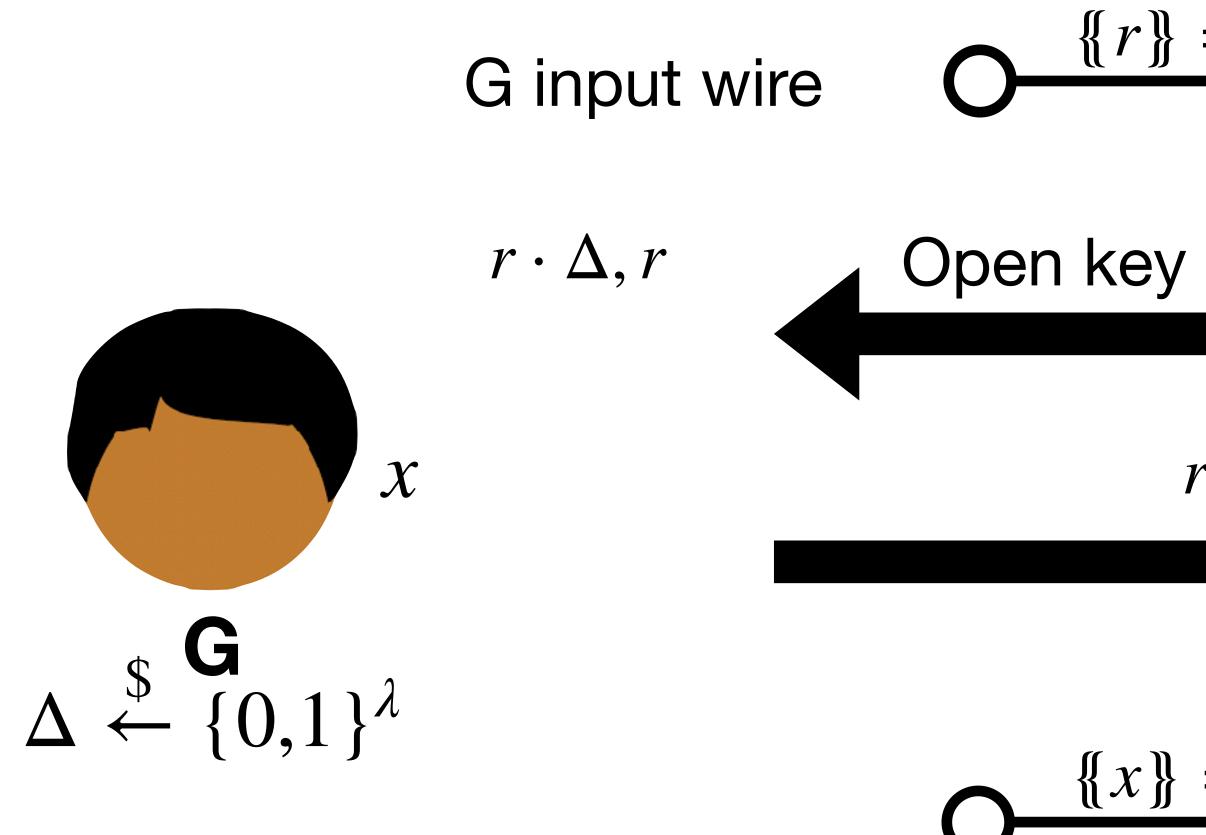




Authenticated Garbling $\{\!\!\{r\}\!\!\} = [r \cdot \Delta, r \cdot \mu, r]$



Ε

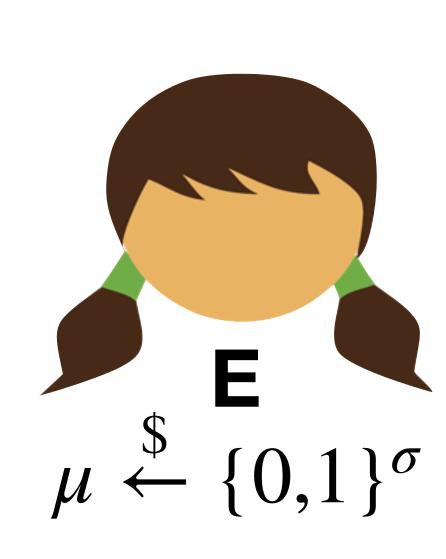


 $\{\!\!\{r\}\!\!\} = [r \cdot \Delta, r \cdot \mu, r]$

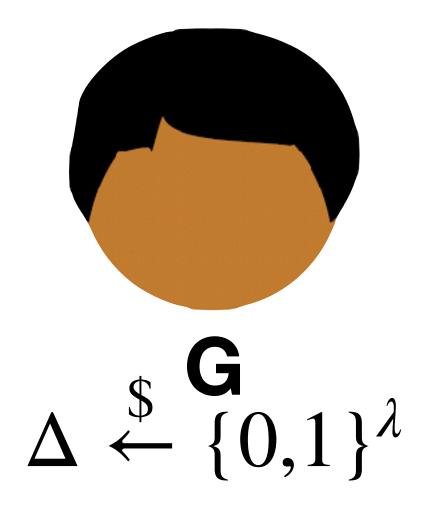
Open key part, value part

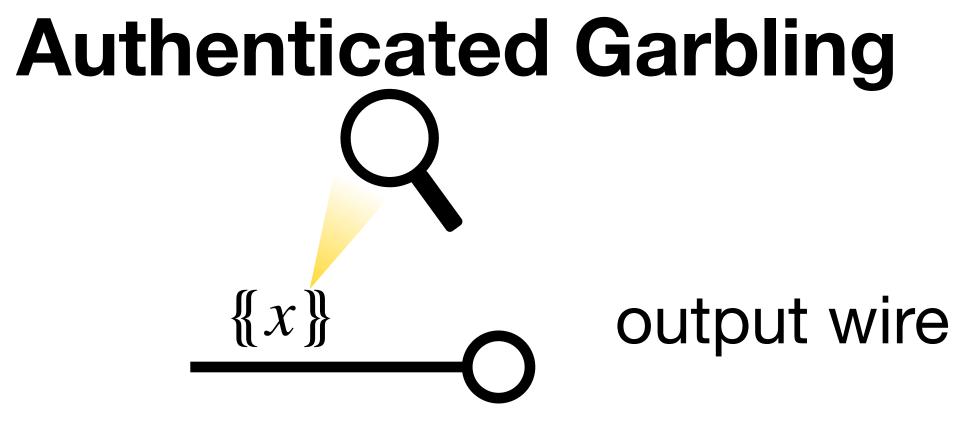
 $r \bigoplus x$

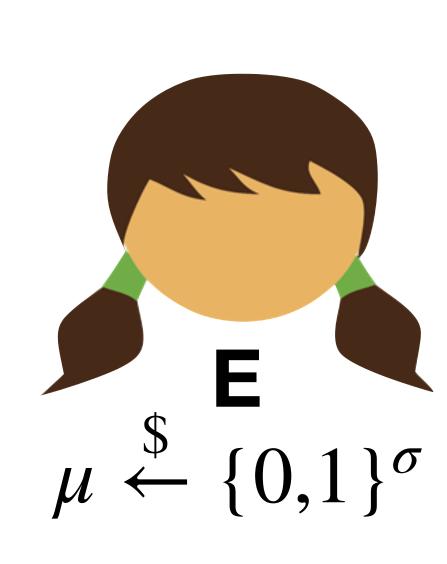
 $\{\!\!\{x\}\!\!\} = [x \cdot \Delta, x \cdot \mu, x]$

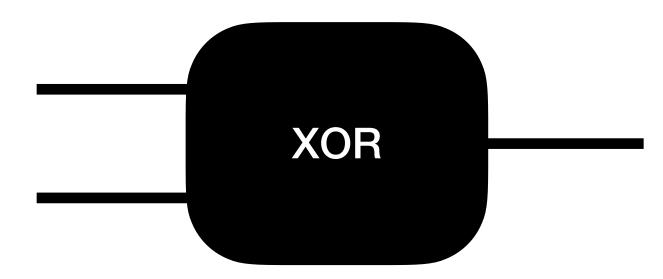


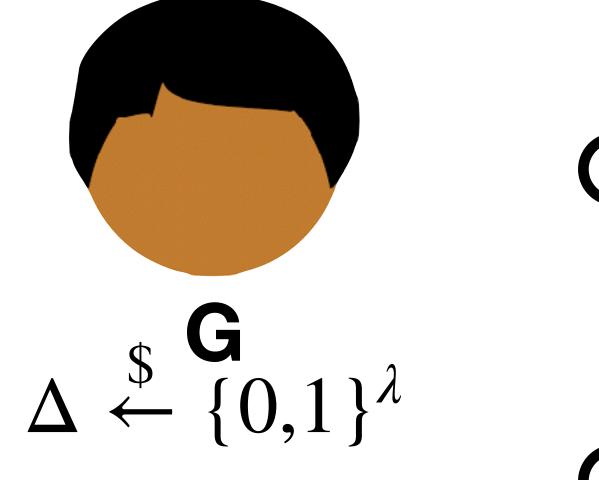
$\{\!\!\{X\}\!\!\}$

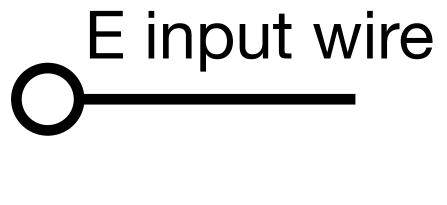


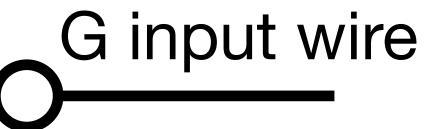


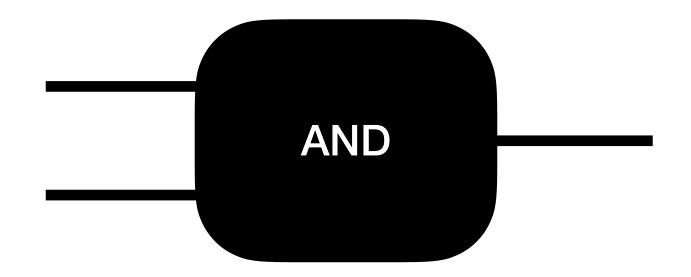


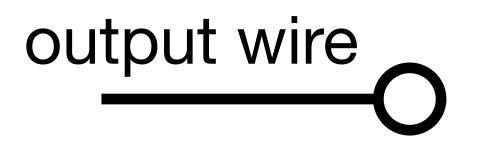


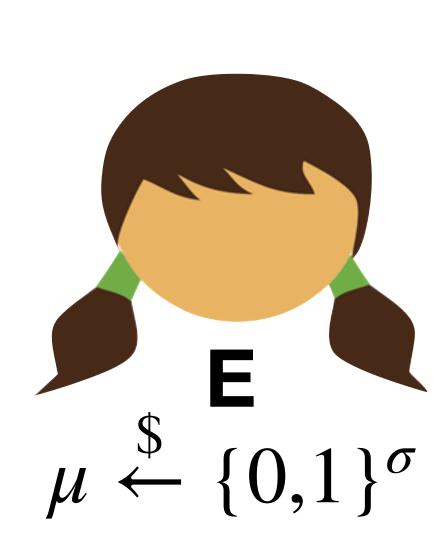












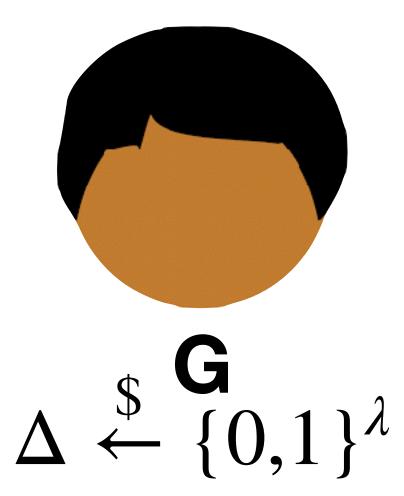
Authenticated Garbling Preprocessing Functionality

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

 $\{\!\!\{\alpha\}\!\!\}, \{\!\!\{\alpha\}\!\!\}$

where a

Is this an easier problem?



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

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



Authenticated Garbling Preprocessing Functionality

 $\{\!\!\{\alpha\}\!\!\}, \{\!\!\{\alpha\}\!\!\}$

where a

Is this an easier problem?

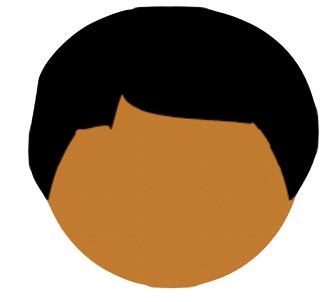
Yes! Random bits only; not dependent on inputs Can be computed all at once; no circuit topology

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

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

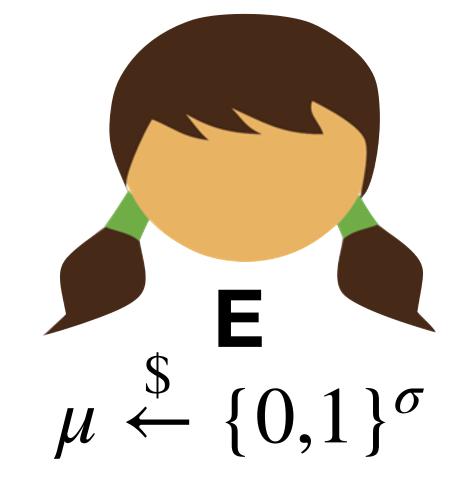
$$\alpha, \beta \stackrel{\$}{\leftarrow} \{0,1\}$$

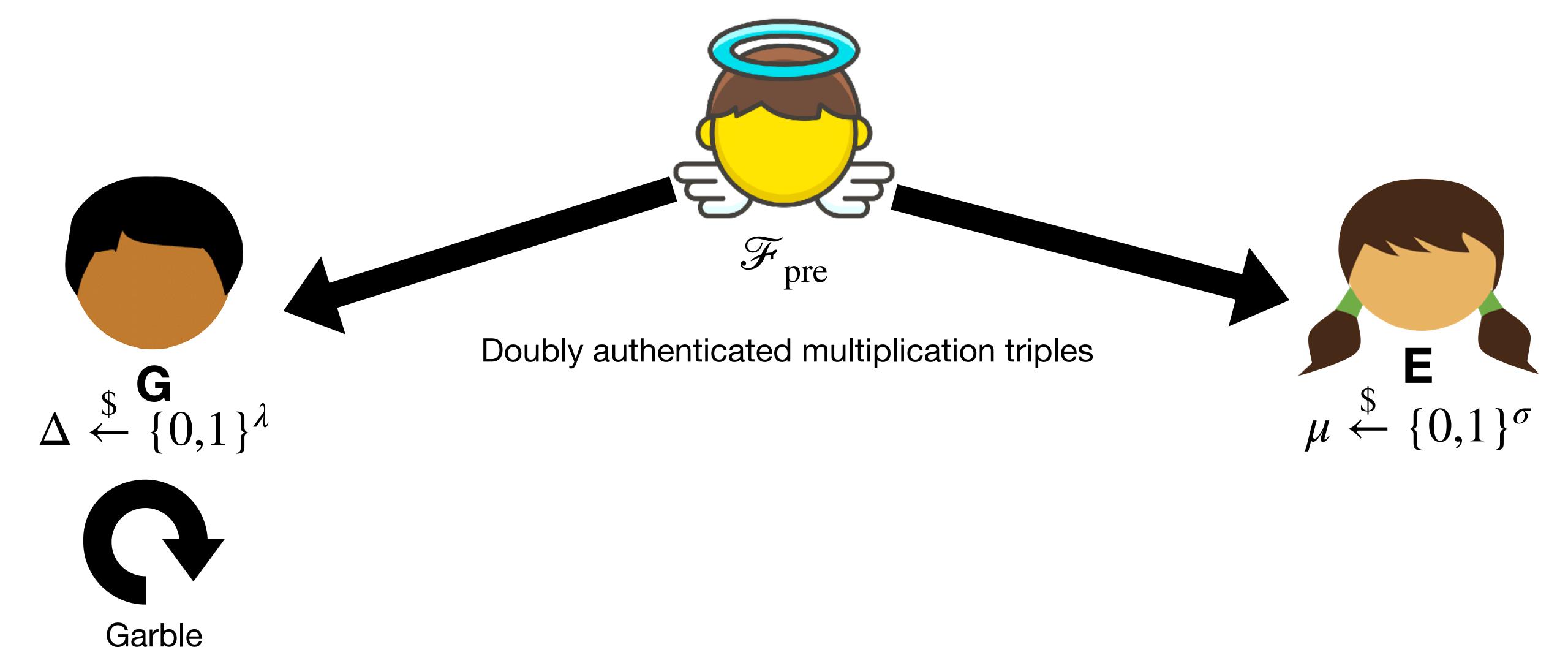


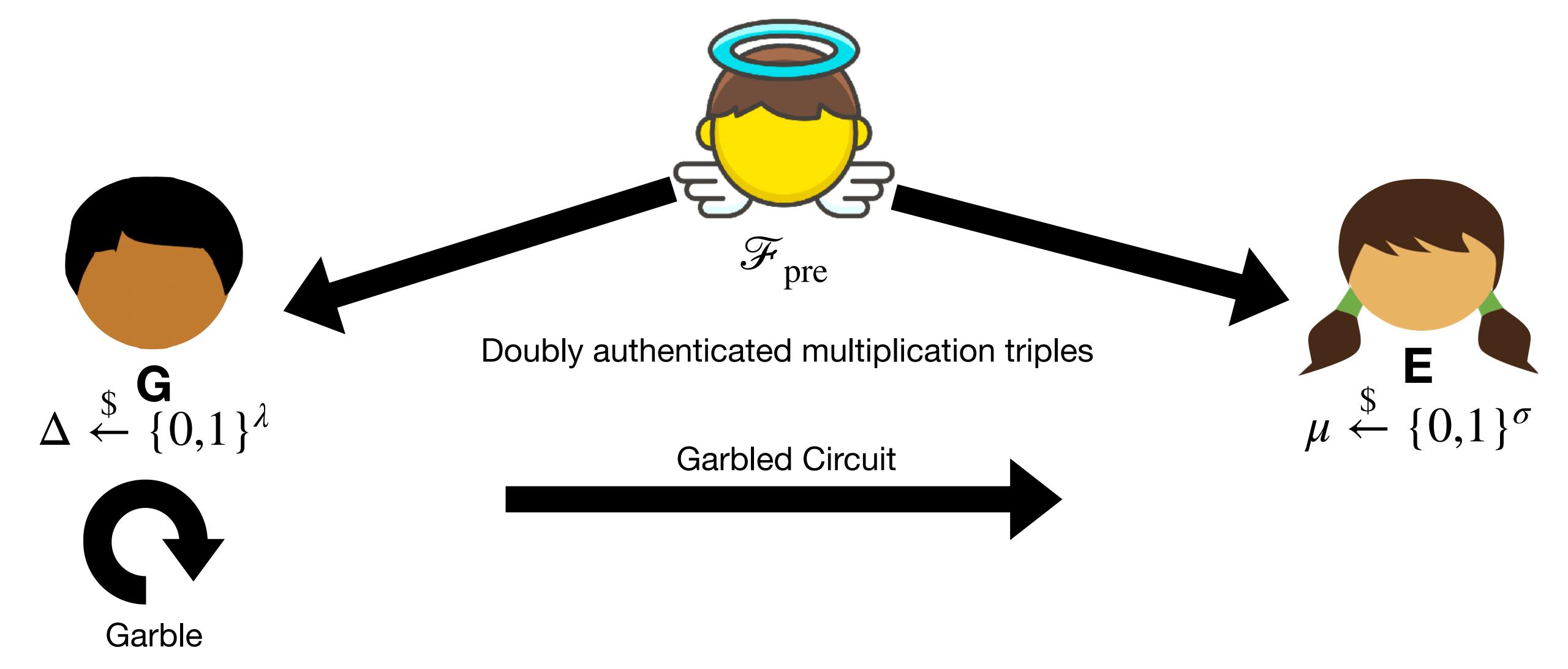


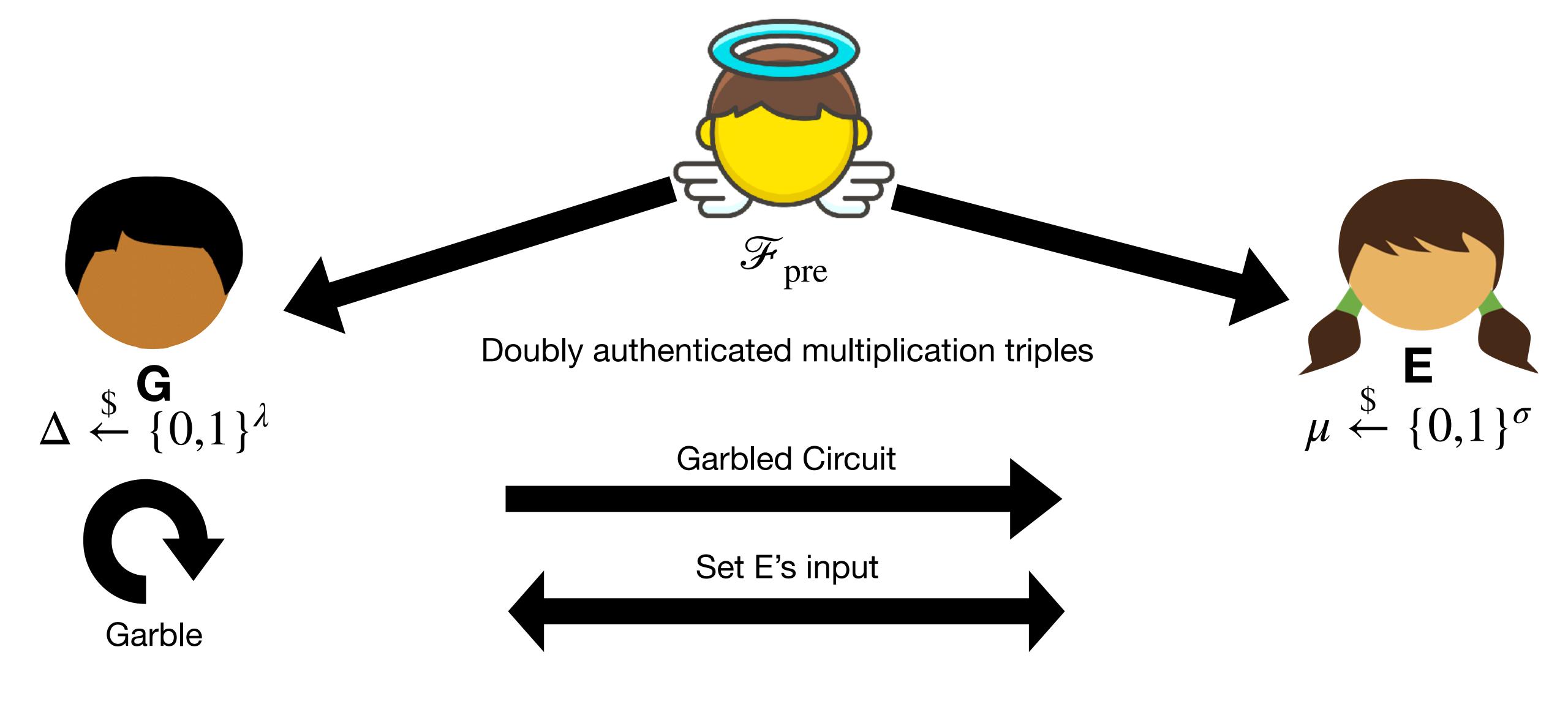
 $\Delta \stackrel{\mathbf{G}}{\leftarrow} \{0,1\}^{\lambda}$

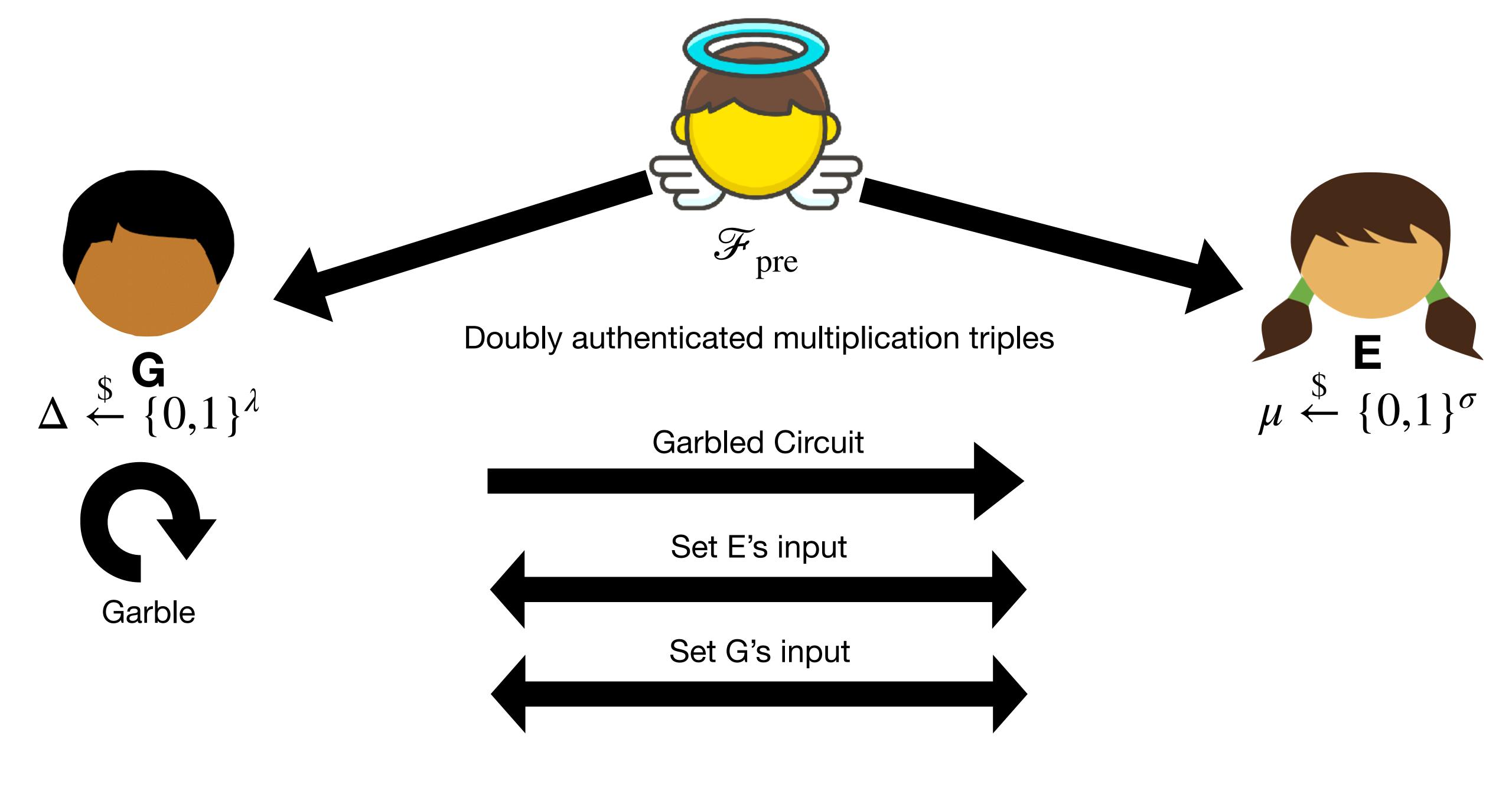


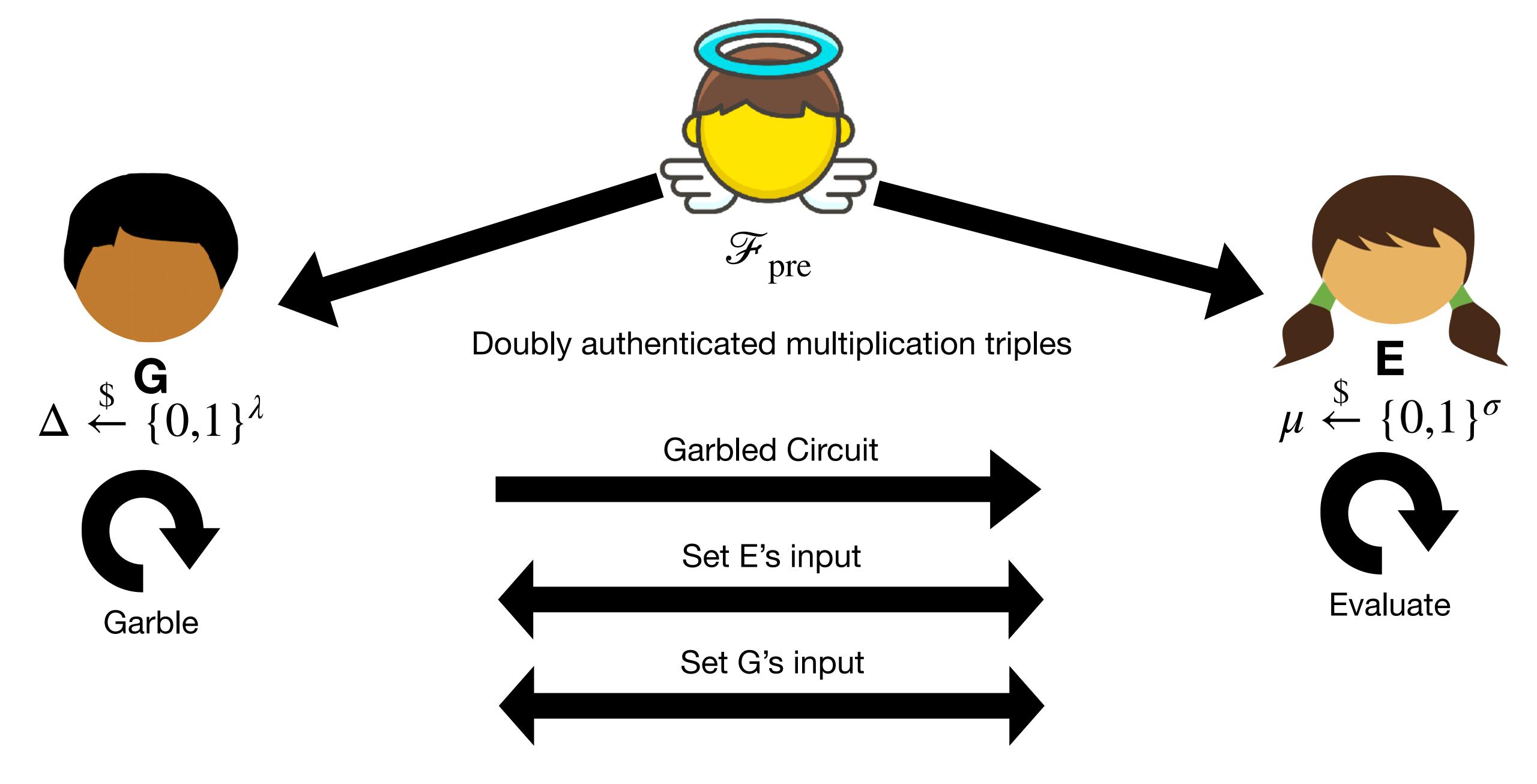












Authenticated Garbling and Efficient Maliciously Secure **Two-Party Computation**

Xiao Wang University of Maryland wangxiao@cs.und.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.

Introduction 1

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, $\rho \ge 40$). 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