Circuit Garbling and Yao's 2-party Computation

School on Secure Multiparty Computation

Arpita Patra

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Computer Science and Automation

Roadmap

- Yao’s millionaire’s problem- triggered fundamental area of secure computation
- Generic secure 2-party computation (2PC)
  - Security goal
- Yao’s 2PC
  - Garbled circuit
  - Oblivious Transfer
- Tracing the journey of garbled circuits and some open questions

Yao's Millionaires' Problem

Protocols for Secure Computations (Extended Abstract). FOCS 1982: 160-164

Turing award winner Andrew Yao

Yao's millionaires' problem

₹ X

?

<

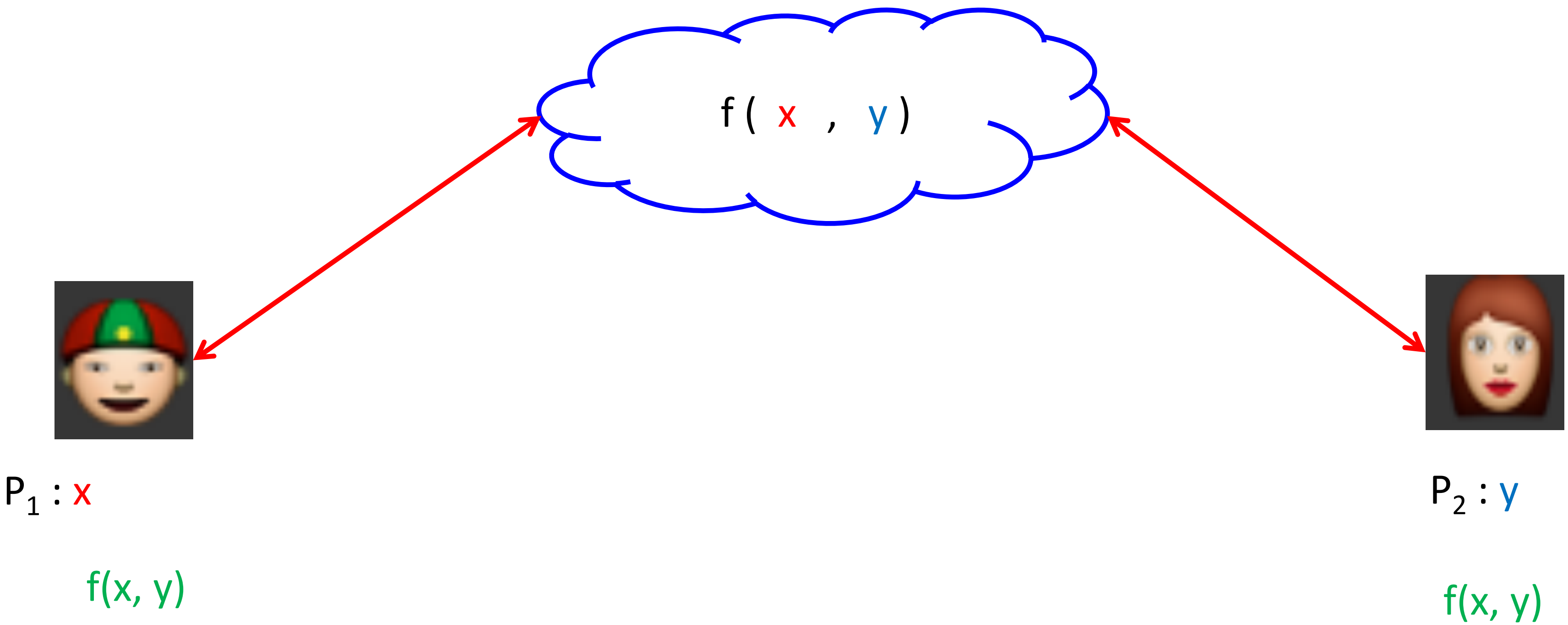
=

>

₹ Y

Find the richer without disclosing exact value of individual assets

Secure 2-PC

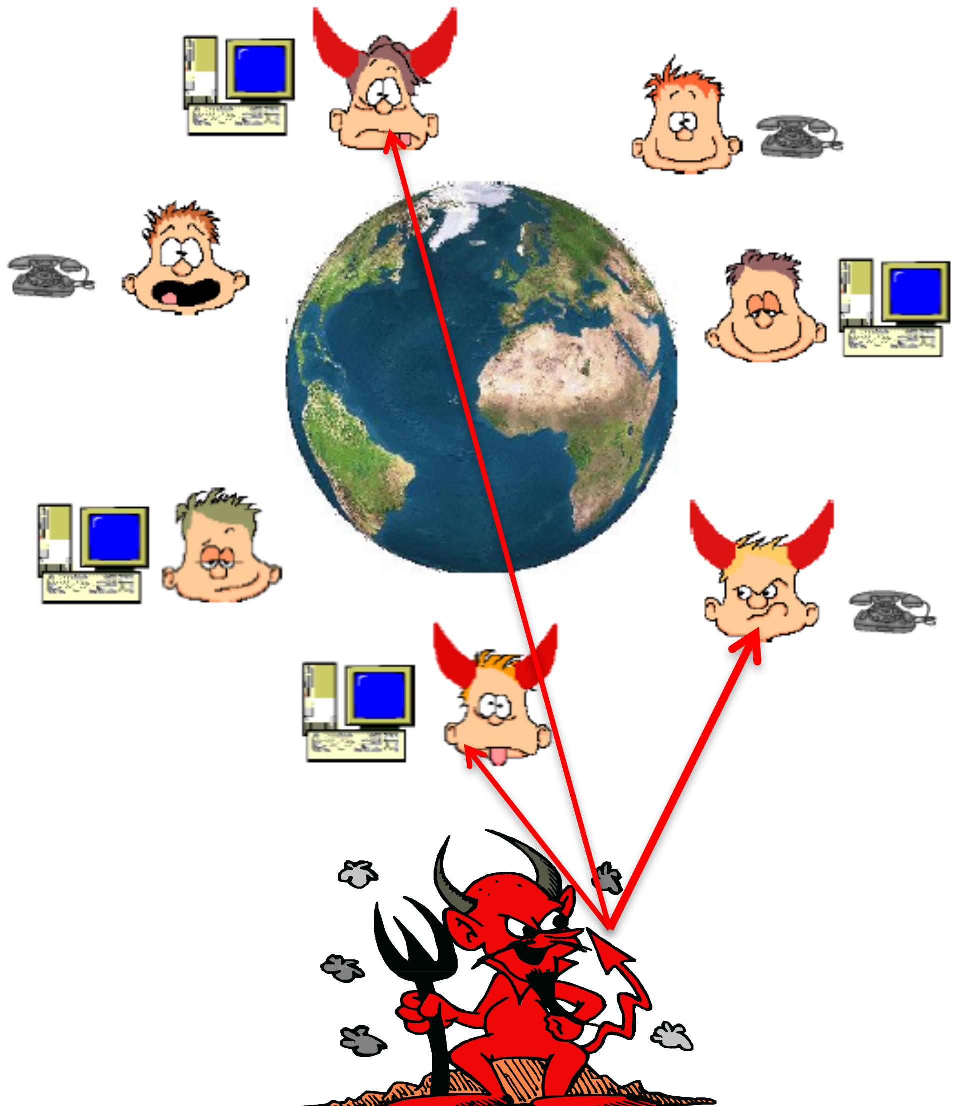


- Mutually distrusting entities with individual private data

- Want to compute a joint function of their inputs without revealing anything beyond

Secure Multiparty Computation (MPC)

MPC – holy grail



Setup:

- n parties $P_1,\ldots,P_n$; 'some' are corrupted
- P_i has private input x_i
- A common n-input function f

Goals:

- **Correctness:** Compute $f(x_1,x_2,..x_n)$
- **Privacy:** Nothing beyond function output must be leaked

Applications: (Dual need of data privacy & data usability)

Preventing Satellite Collision

E-auction

Data Analytics

Privacy-preserving ML

Outsourcing

E-voting

Application of 2PC- Privacy-preserving Data mining

- How many patients suffering from AIDS in total ?
- Are there any common patient registered for disease X in all the hospitals ?
- Varieties of other statistics …

AIIMS
(All India Institute of Medical Sciences)

Apollo
HOSPITALS

How to solve 2PC?

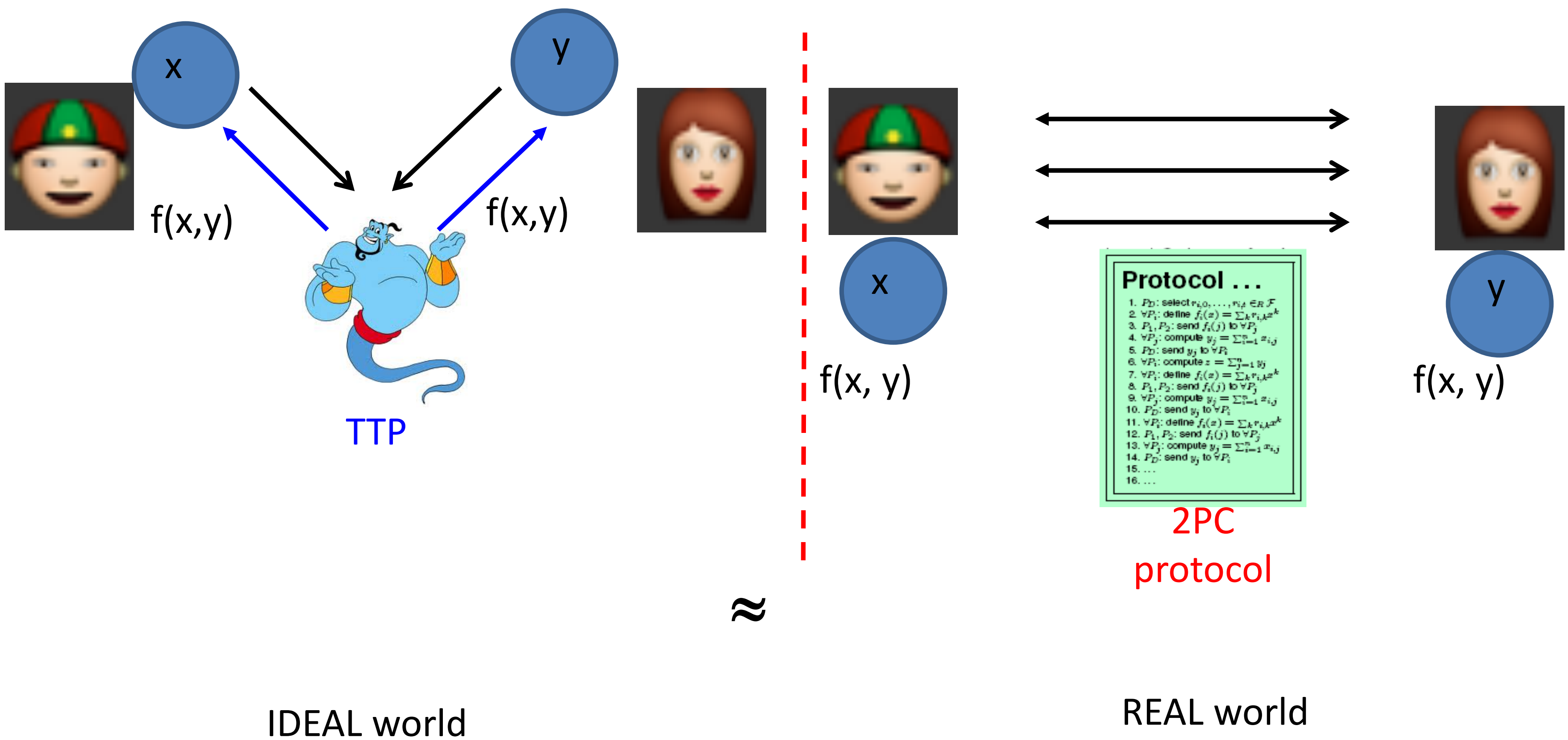
- Trusted third party (TTP) $\rightarrow$ solution for secure 2PC
  - ➢ Send input to TTP, obtain function output : Ideal solution

IDEAL world secure 2PC protocol

TTPs exist only in fairy tales!!

Security goal of 2PC

- Goal of a secure 2PC protocol : emulate the role of a TTP
  - ➢ De-centralizing the trust

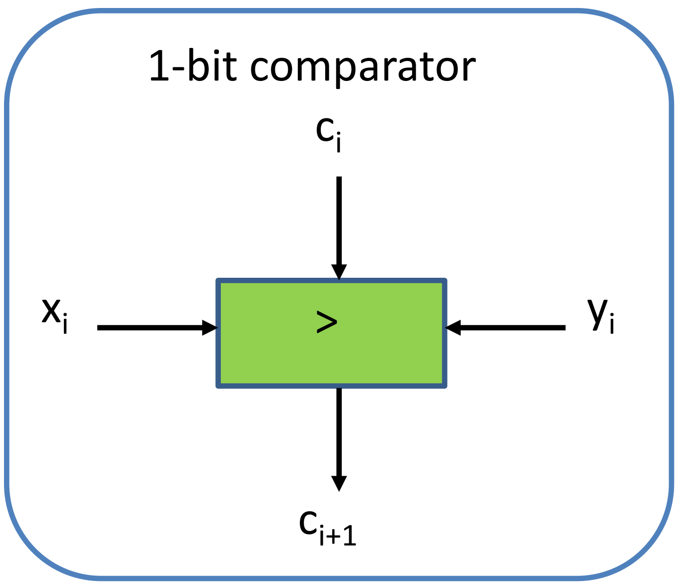


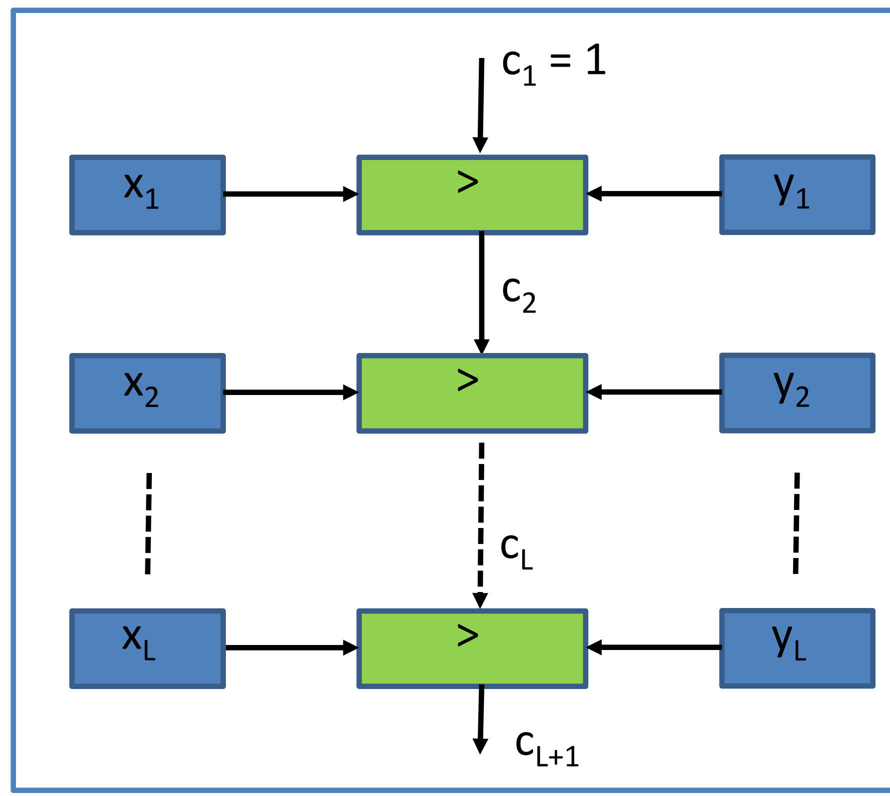
Circuit Representation of function

- Circuit abstraction
  - f : represented as a Boolean circuit C
  - Any efficiently computable f can be represented as a C
  - C: DAG with input gates, output gates and internal Boolean gates ((AND, OR, NOT), (NAND), (NOR): universal gates)

Circuit Abstraction Example: $\geq$

- X, Y: L-bit non-negative integers





- $c_{i+1} = 1 \leftrightarrow (x_i > y_i)$ OR
  $([x_i = y_i]$ AND $[c_i = 1])$

- $c_{i+1} = x_i \oplus [(x_i \oplus c_i) \wedge (y_i \oplus c_i)]$

- $X \geq Y \leftrightarrow c_{L+1} = 1$

Circuit Garbling

What we do?

- Encode/Garble the circuit
- Encode input
- Evaluate encoded circuit on encoded input and get encoded output
- Decode output using decoding information

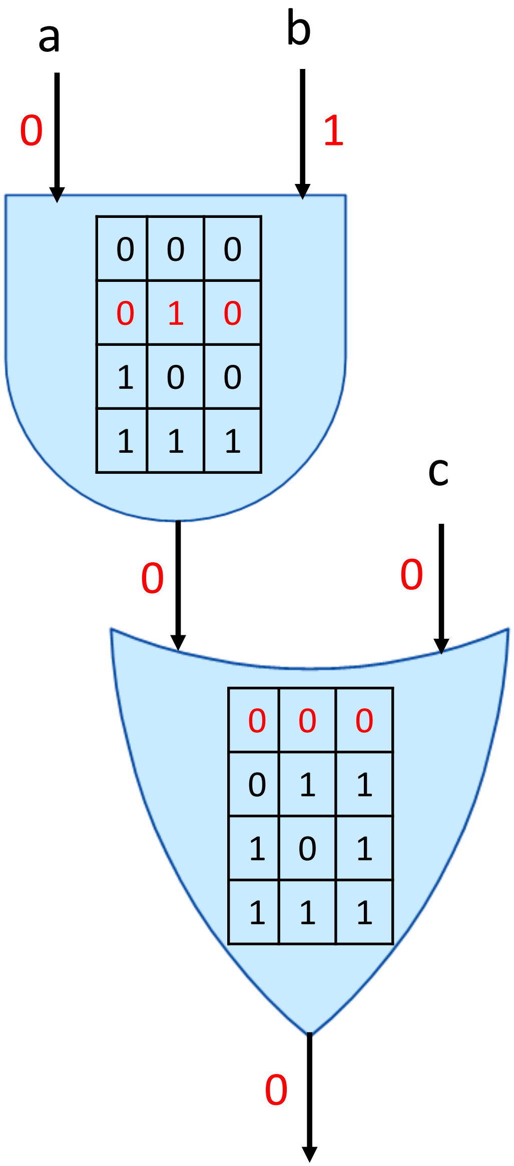
What is the goal?

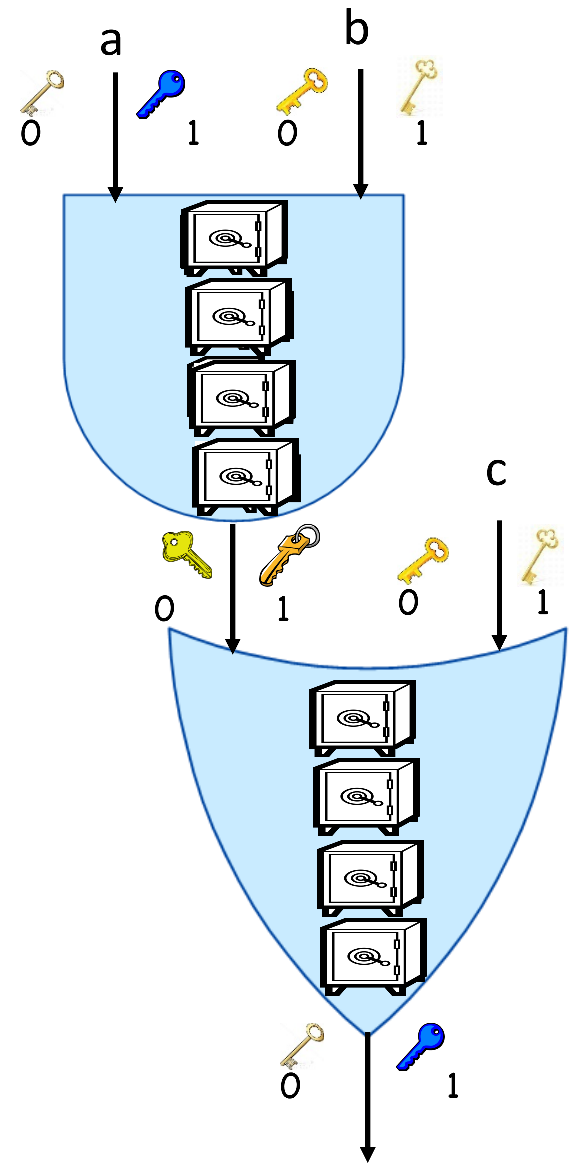
- Nothing beyond function output is leaked
  - ✓ Preserves input privacy
  - ✓ No leaking of intermediate gate outputs
  - ✓ No leaking of output if decoding info is withheld

Yao: secure circuit evaluation

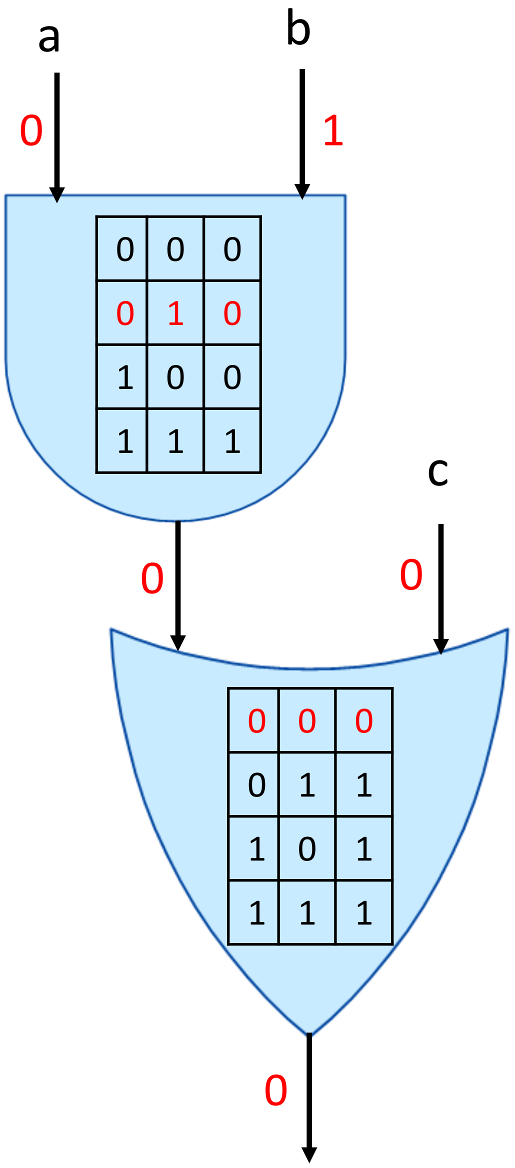
- ➢ Parties jointly evaluate the circuit securely
- ➢ Only final outcome revealed during evaluation
- ➢ Intermediate values remain private

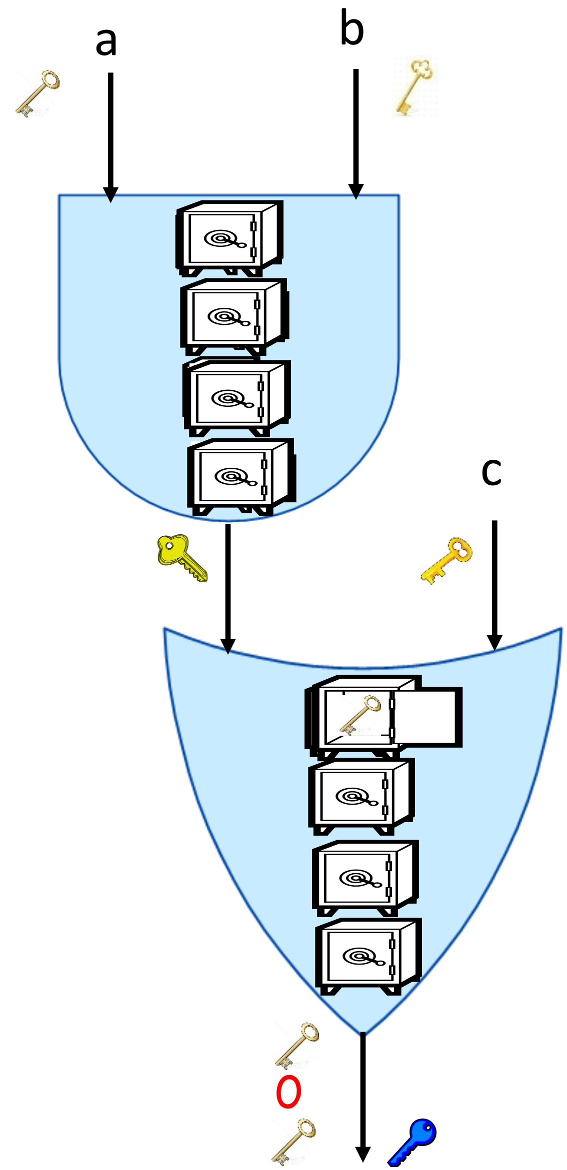
The making of Garbled Circuit



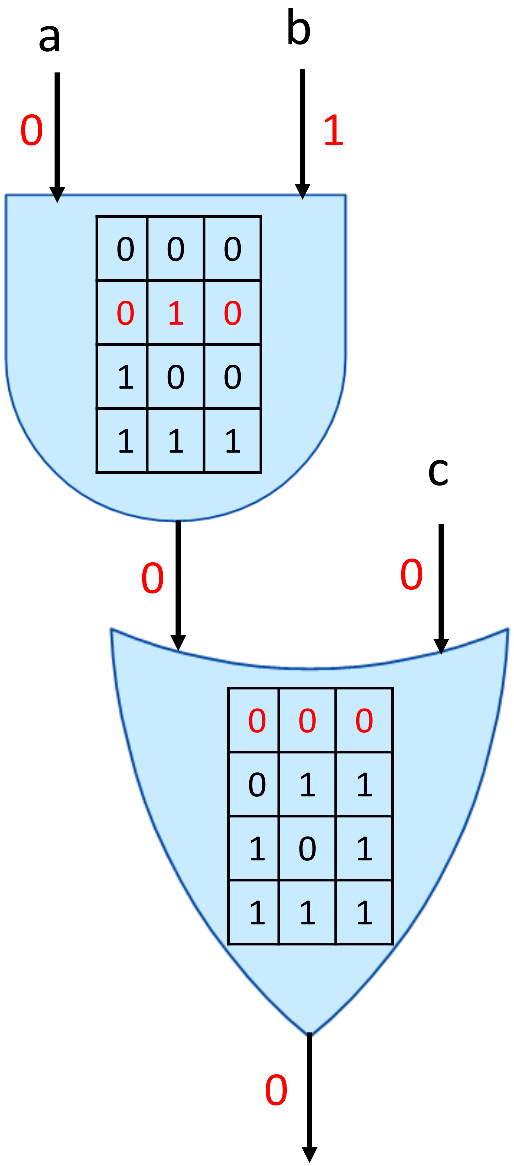


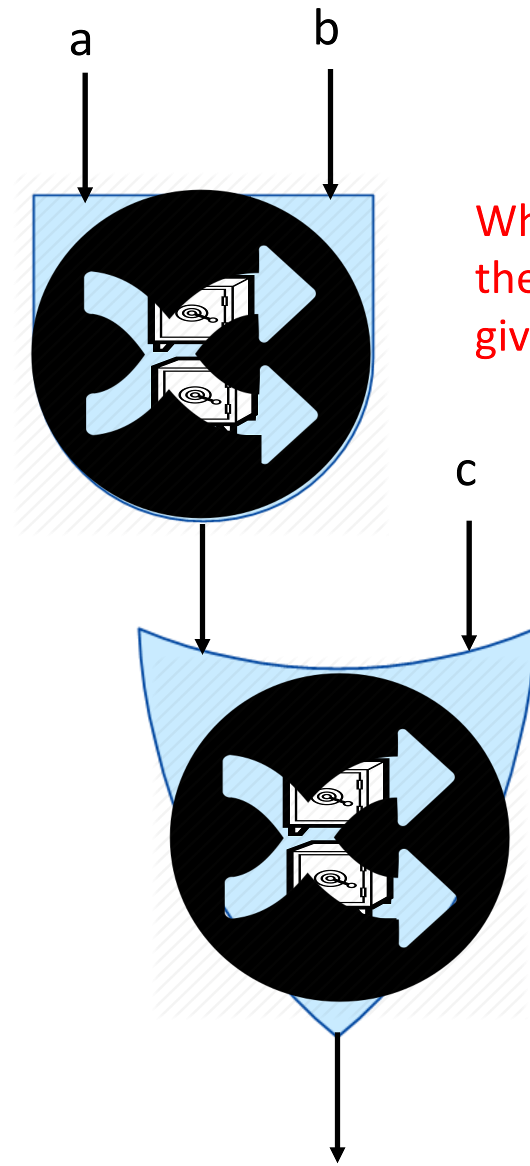
Evaluating a Garbled circuit vs. Evaluating a circuit





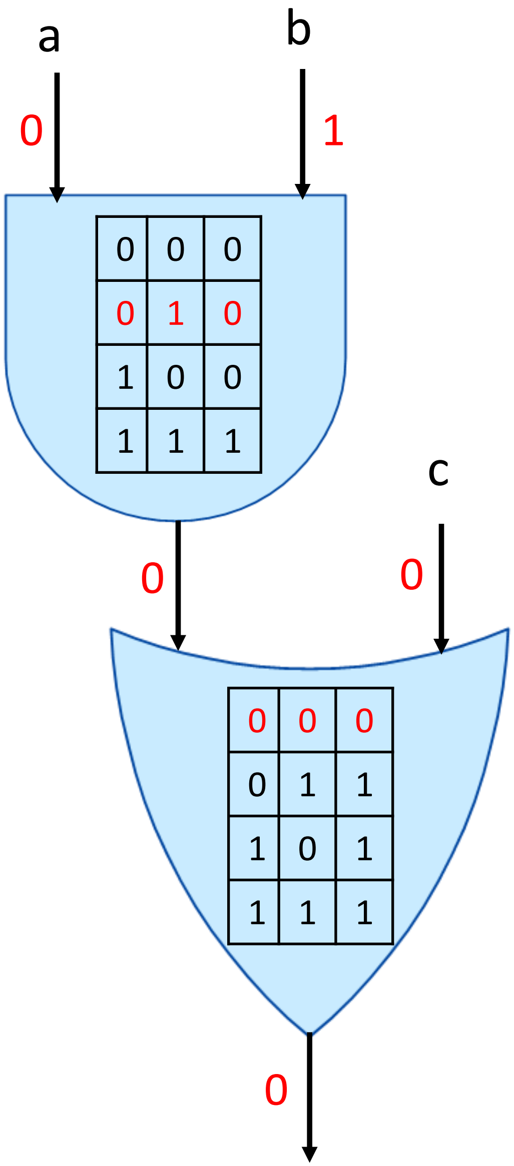
Is all Okay?

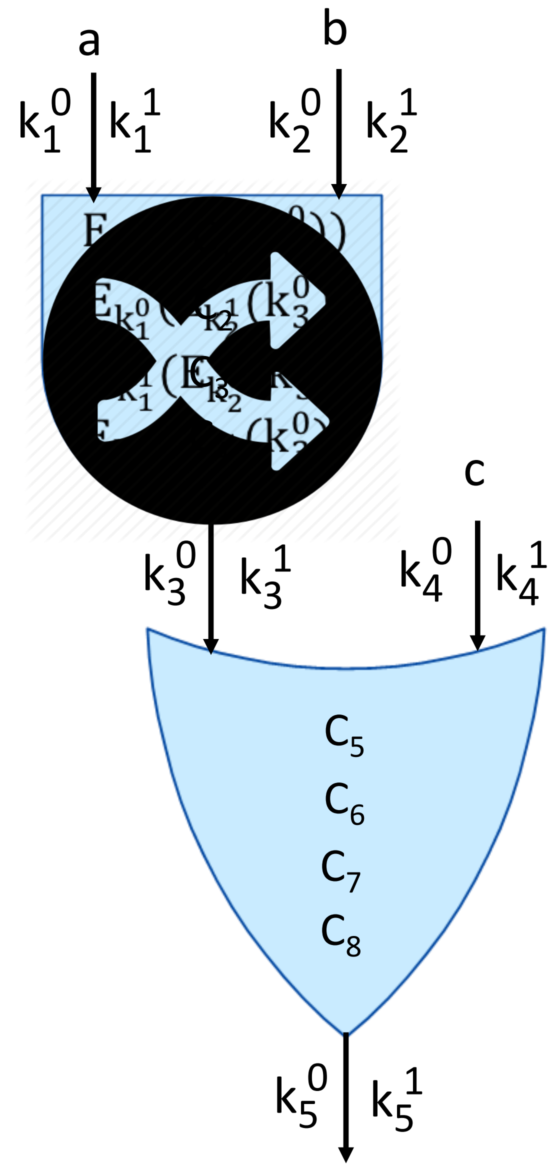




What happens if the ciphertexts are given in the order?

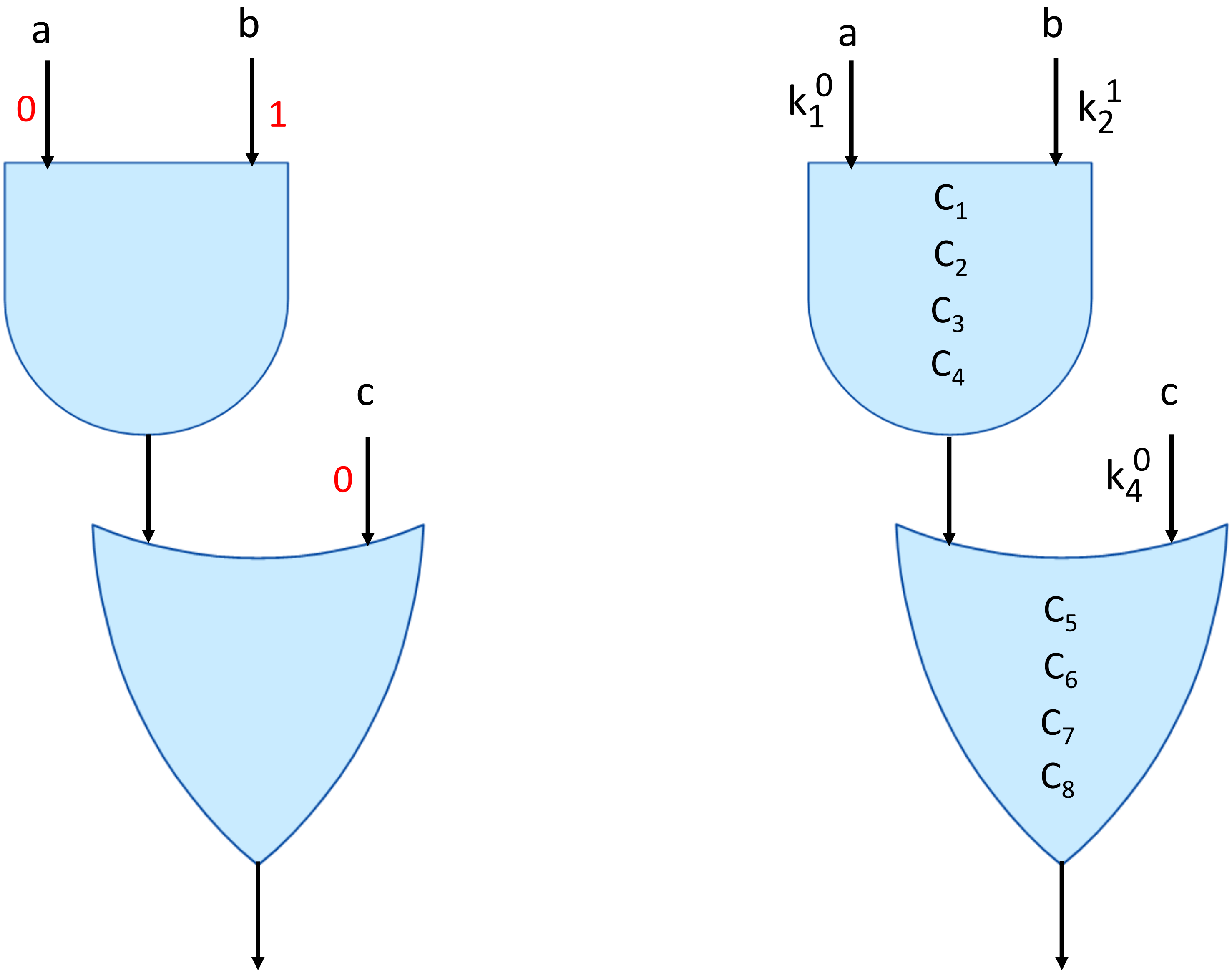
Replacing key-box with Cryptographic Mechanisms





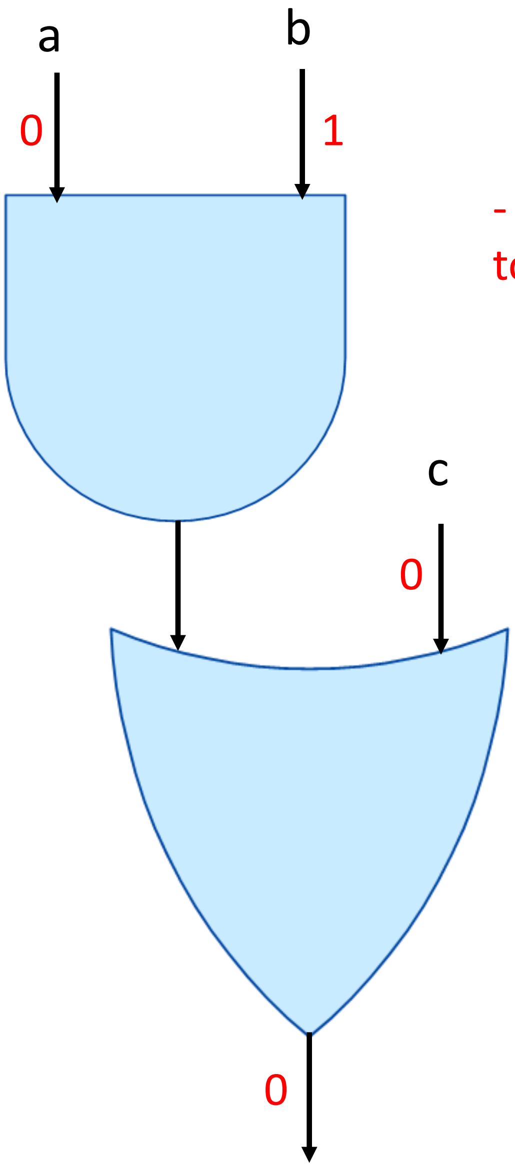
(G, E, D) = Symmetric Key Encryption (SKE)

Evaluating a Garbled circuit vs. Evaluating a circuit



(G, E, D) = Symmetric Key Encryption (SKE)

Something may be wrong…

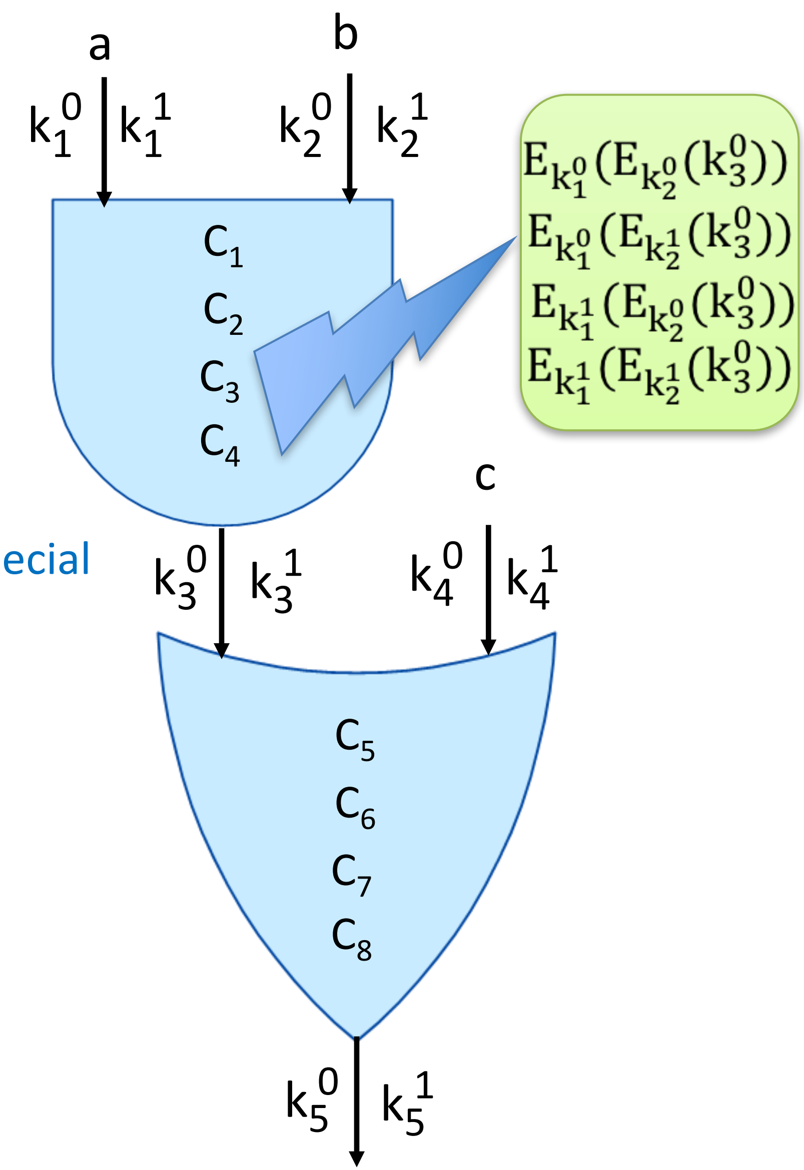


- Which ciphertext to decrypt?

  - Try all

- Which decrypted value to go for?

  - SKE with `special correctness'



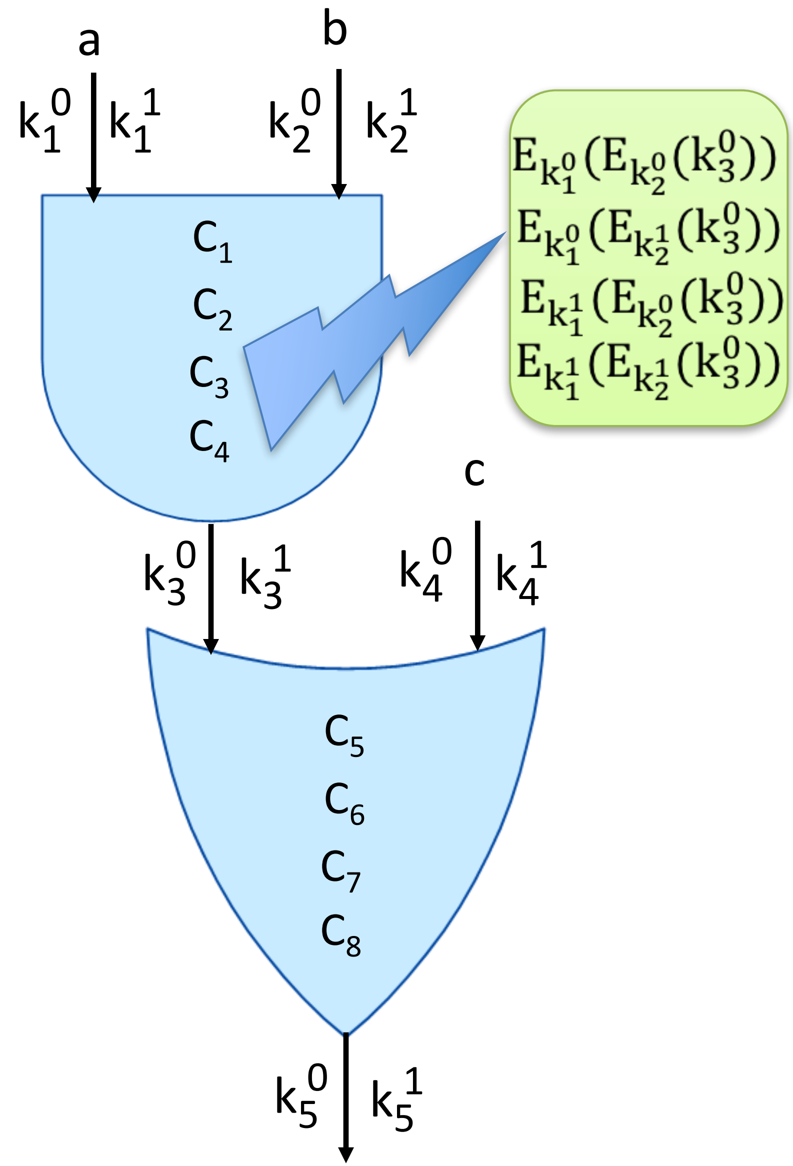
(G, E, D) = Symmetric Key Encryption (SKE)

Making things all right…

(G,E,D) has `special correctness'

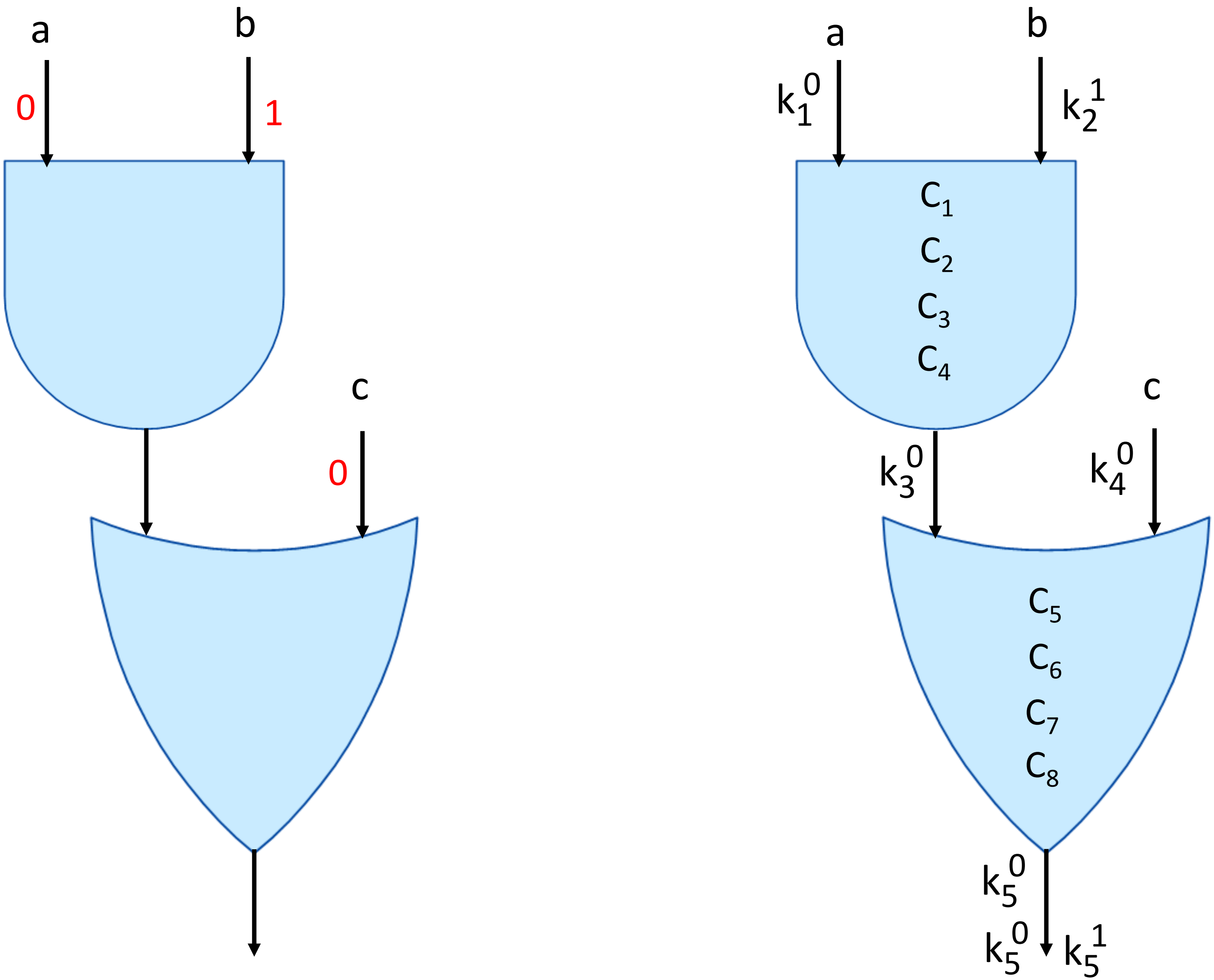
- for two distinct keys (k_1,k_2), encryption under k_1 will result in ⊥ when decrypted under k_2 (with overwhelming probability)

$$\Pr\left[D_{k_2}\left(E_{k_1}(m)\right) \neq \perp\right] \leq \mathcal{E}(n) \quad \forall m$$



(G, E, D) = Symmetric Key Encryption (SKE)

Evaluating Garbled circuit vs. Evaluating a circuit



(G, E, D) = Symmetric Key Encryption (SKE) with **`special correctness'**

What security from SKE is needed?

- an bad evaluator should have no info about what the three unopened ciphertext contain

- if it can guess the unopened message are same for an AND gate, then it knows the meaning of the key it decrypted!

k_0, k_1 $(x_0,y_0,z_0), (x_1,y_1,z_1)$ →

$c_0 \leftarrow Enc_{k0}(Enc_{k'1}(x_b))$

$c_1 \leftarrow Enc_{k'0}(Enc_{k1}(y_b))$

$c_2 \leftarrow Enc_{k'0}(Enc_{k'1}(z_b))$

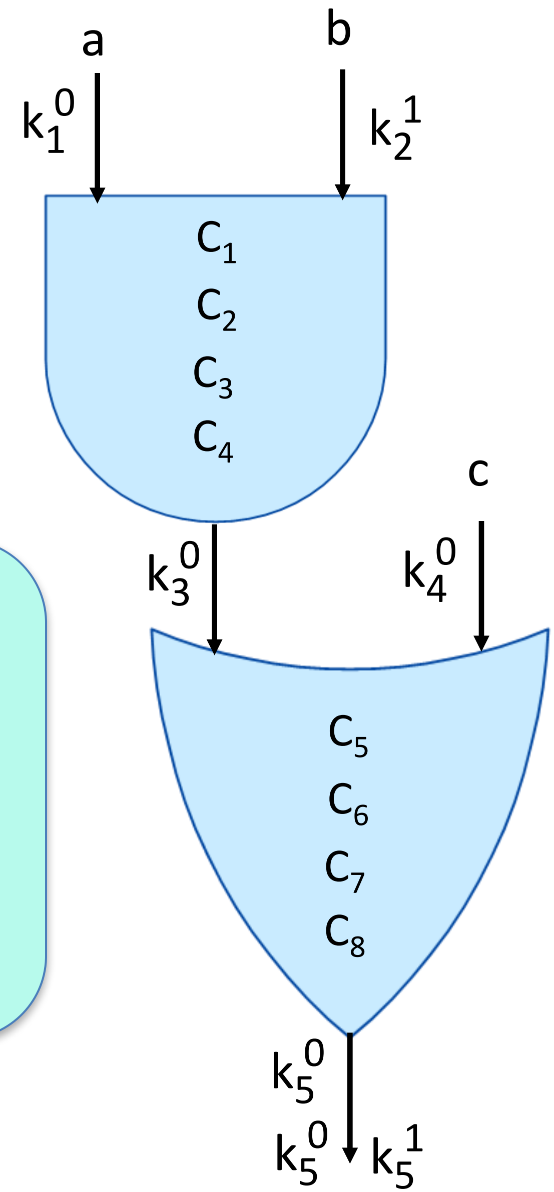
←

$b' \in \{0, 1\}$ →

k'_0, k'_1

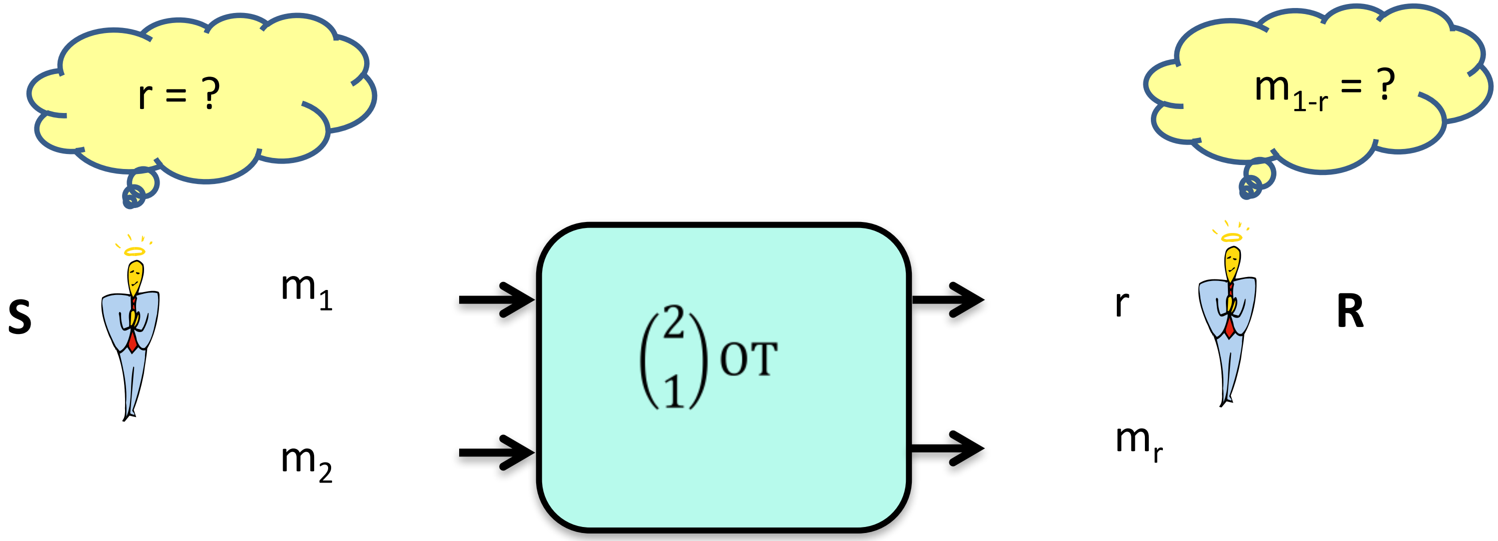
b

+ `chosen double ciphertext security'

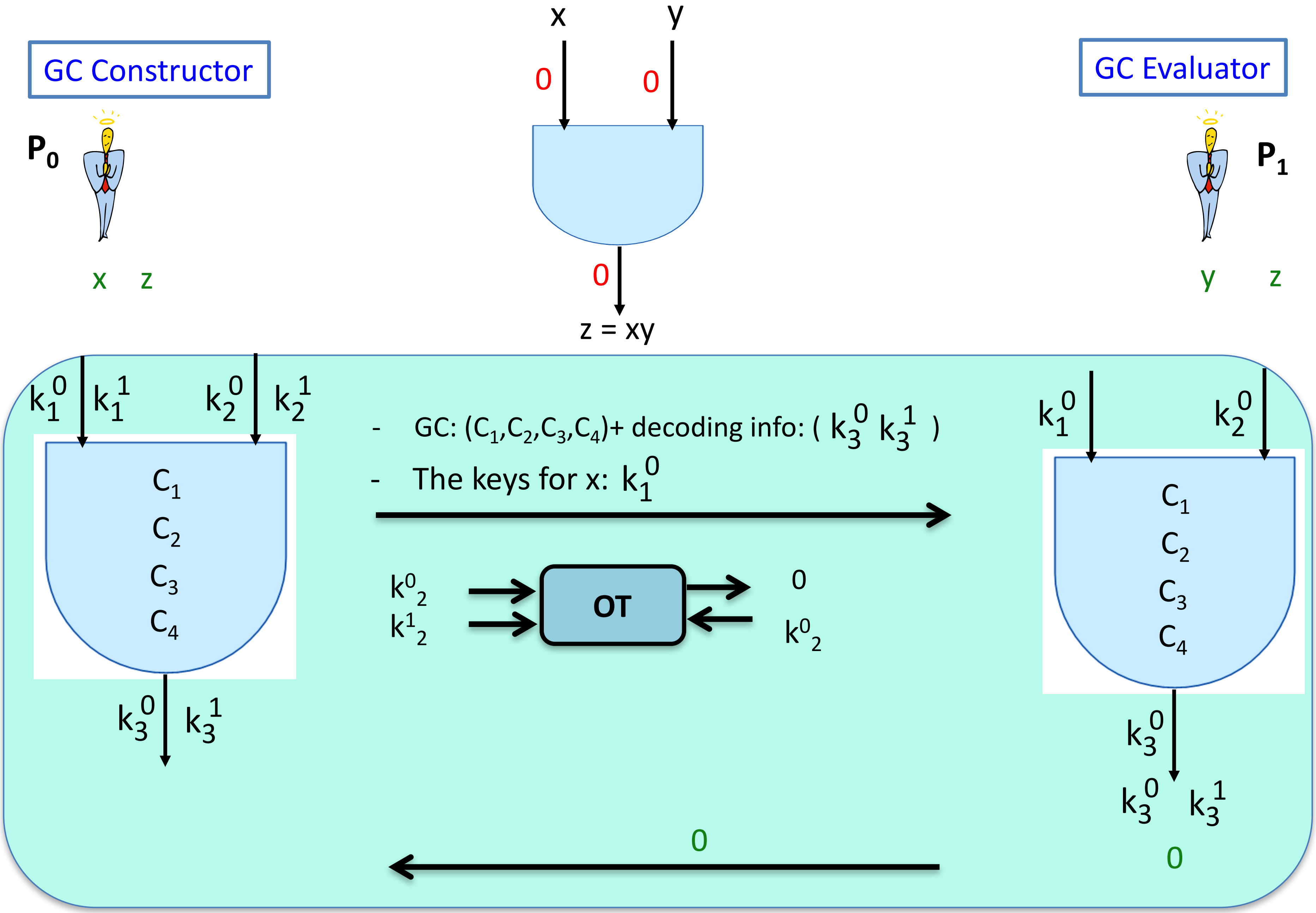


(G, E, D) = Symmetric Key Encryption (SKE) with **`special correctness'**

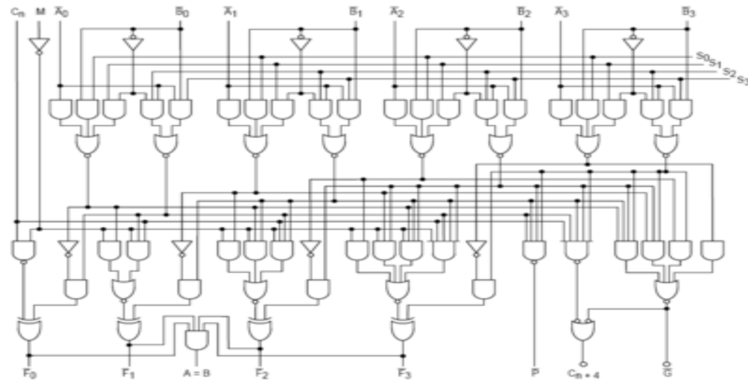
Oblivious Transfer



Yao's 2-Party Protocol

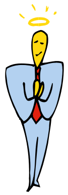


Yao's 2-Party Protocol



GC Constructor

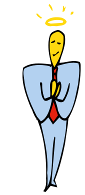
P_0



$X = (x_1, x_2, \ldots x_k)$

Z

GC Evaluator



P_1

$Y = (y_1, y_2, \ldots y_k)$

Z

- Garbled Circuit + decoding information
- The keys for X

→

k^0_1, k^1_1 → OT_1 → y_1; ← k^{y1}_1

k^0_k, k^1_k → OT_k → y_k; ← k^{yk}_k

Z

←

Circuit Garbling- Tracing the history

- Point-and-permute [NPS99]: - No `special correctness' needed
- Only one ciphertext needs to be decrypted

- Garbled Row Reduction:

- [NPS99]: 4-to-3 ciphertexts
- [PSSW09,GNLP15,ZRE15]: 4-to-2 ciphertexts (optimal for AND)
- [KKKS15]: 4 bits (for formulaic circuits)
- [Kol05]: 0 bits (for formulaic circuits + key length dependent on depth)

- Free XOR/FleXOR [KS08,KMR14]: No ciphertext and no crypto operations for XOR gates

- From technique to primitive [BHR12a,BHR12b]: Privacy, Obliviousness, Authenticity and verifiability

- Applications in ZK, outsourcing computation [JKO13]: Privacy-free GC

Stay tuned to our reading group

Blogging/Reading   Travelling   Know India Campaign!   My Co-conspirator

Arpita Patra

HOME   RESEARCH   TEACHING   PROFESSIONAL ACTIVITIES   STUDENTS   RECOGNITIONS   PHOTOGRAPHY

Session 1

- Speaker: Yash
- Logistics: 8 March 2017, 3:30-6 pm, CrIS Lab (Room 329, CSA, IISc)
- Theme: Foundations.
- Description: Recap of notation and language of garbled circuits
- References: [BHR12a, BHR12b].

Session 2

- Speaker: Divya, Swati
- Logistics: 12 March 2017, 10:00 am - 1 pm, CrIS Lab (Room 329, CSA, IISc)
- Theme: Yao's scheme and proof
- Description: Consistent notation for Yao's garbling scheme and simulation
- References: [LP09]

Session 3

- Speaker: Pratik, Swati, Rishabh
- Logistics: 16 March 2017, 3:00 - 6 pm, CrIS Lab (Room 329, CSA, IISc)
- Theme: Optimizations
- Description: Historical GC optimizations
- References: [NPS99,KS08, PSSW09, KMR14, KKKS15]

Circuit Garbling- Recent Results

- **Size-zero Privacy-free Garbled circuits for Formulas [KP17]:** Under submission

- **Zero knowledge Protocols from Garbled circuits [GKPS17]:** Under submission
    - 3,2 and 1 round protocols
    - Any private garbled circuits is also authentic

- **Non-interactive Secure Computation [PS17]:** Under submission

Thank You!