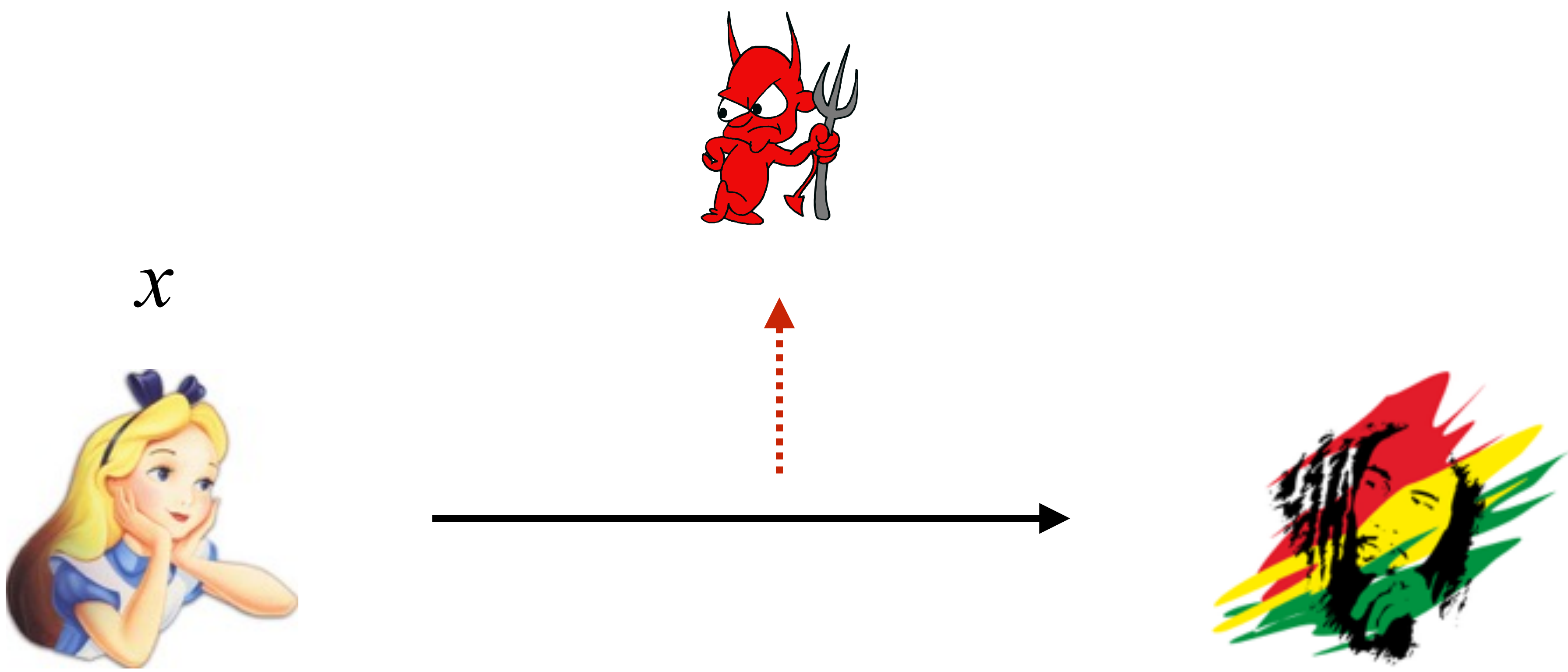
Secure Computation - I

(Introduction and Definitions)

Lecture 13

Data Privacy, so far



How can Alice send x privately to Bob?

Data Privacy, so far

Public-key Encryption

PK, x

SK

Data Privacy, so far

Public-key Encryption

PK, x

x

SK

Data Privacy, so far

Public-key Encryption

PK, x

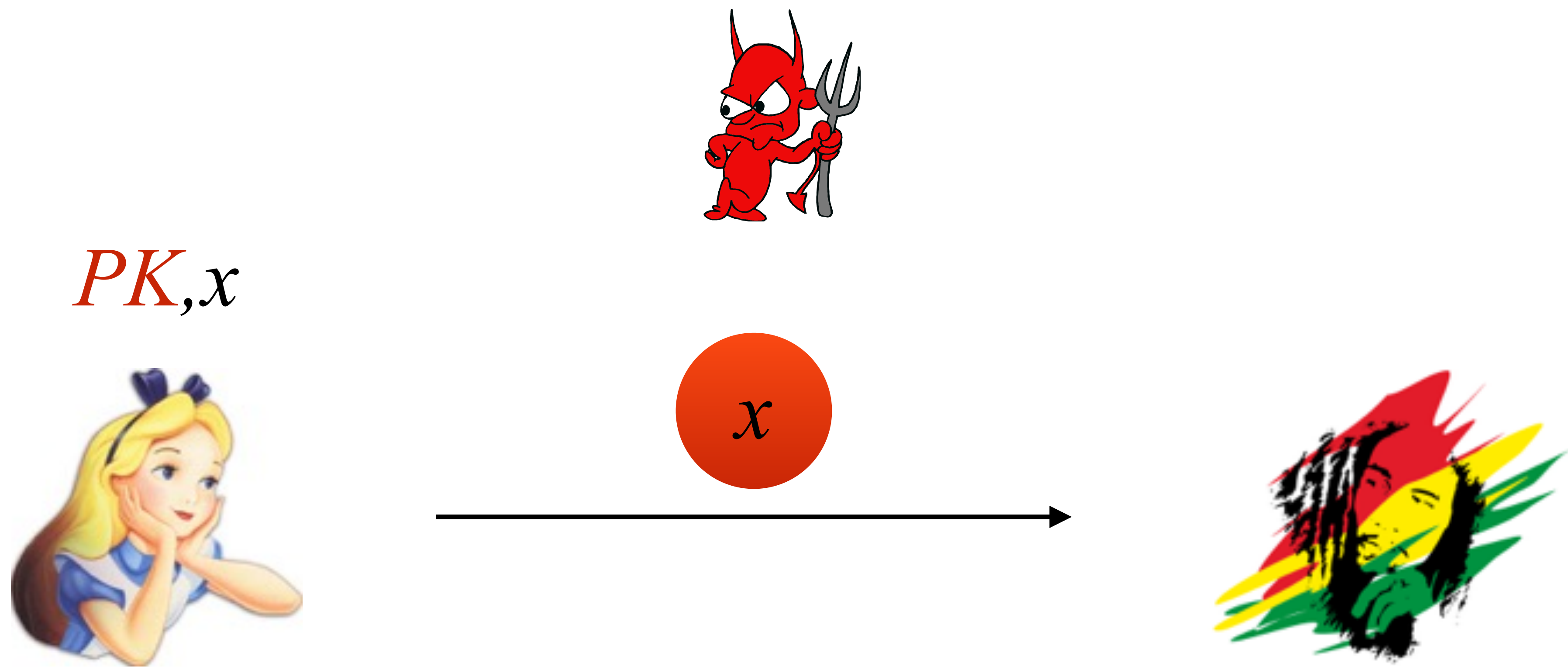
x

SK

- If Bob has the secret-key, then he learns **entire** x

Data Privacy, so far

Public-key Encryption



- If Bob has the secret-key, then he learns **entire** x
- Else, he learns **nothing** about x

“All-or-nothing” Privacy

(either learn the entire secret or nothing about it)

Today's Lecture

"Controlled" Privacy

(release partial information about the secret)

Example from earlier

<u>Zero-Knowledge Proofs</u>

(x, w)

Prover

$x \in L$

(x)

Verifier

- Bob learns that $x \in L$ but nothing else about w

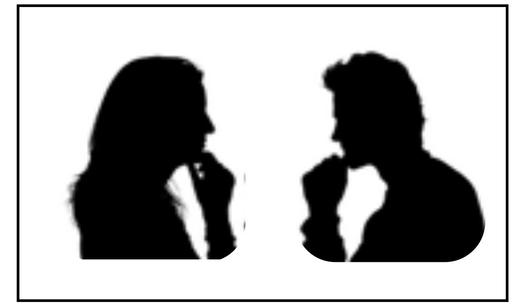
Matchmaking

Matchmaking

Matchmaking

Problem: Tinder not only learns that the players matched, but also their entire profiles

Matchmaking

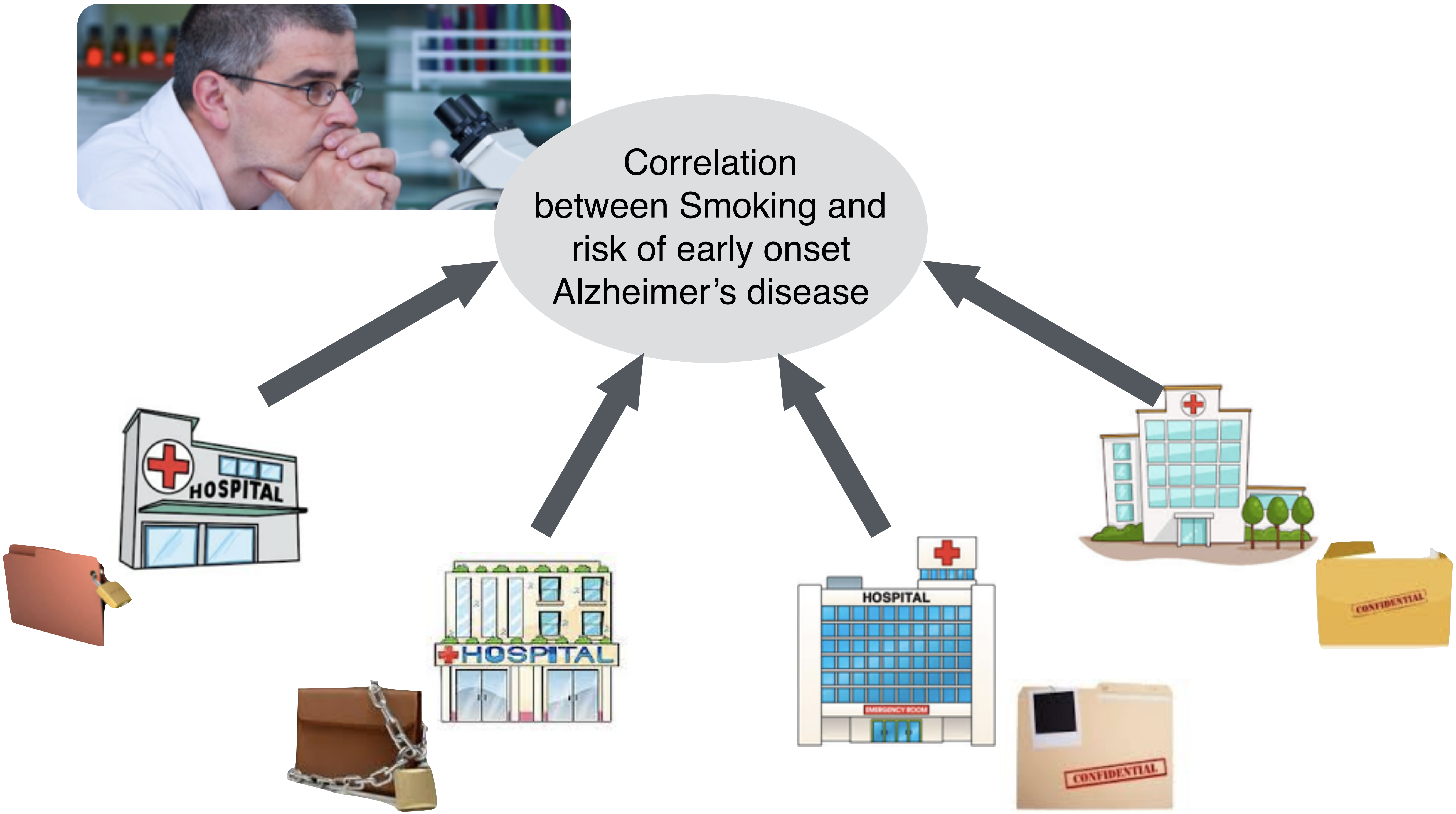


<u>Problem</u>: Tinder not only learns that the players matched, but also their entire profiles

Matchmaking

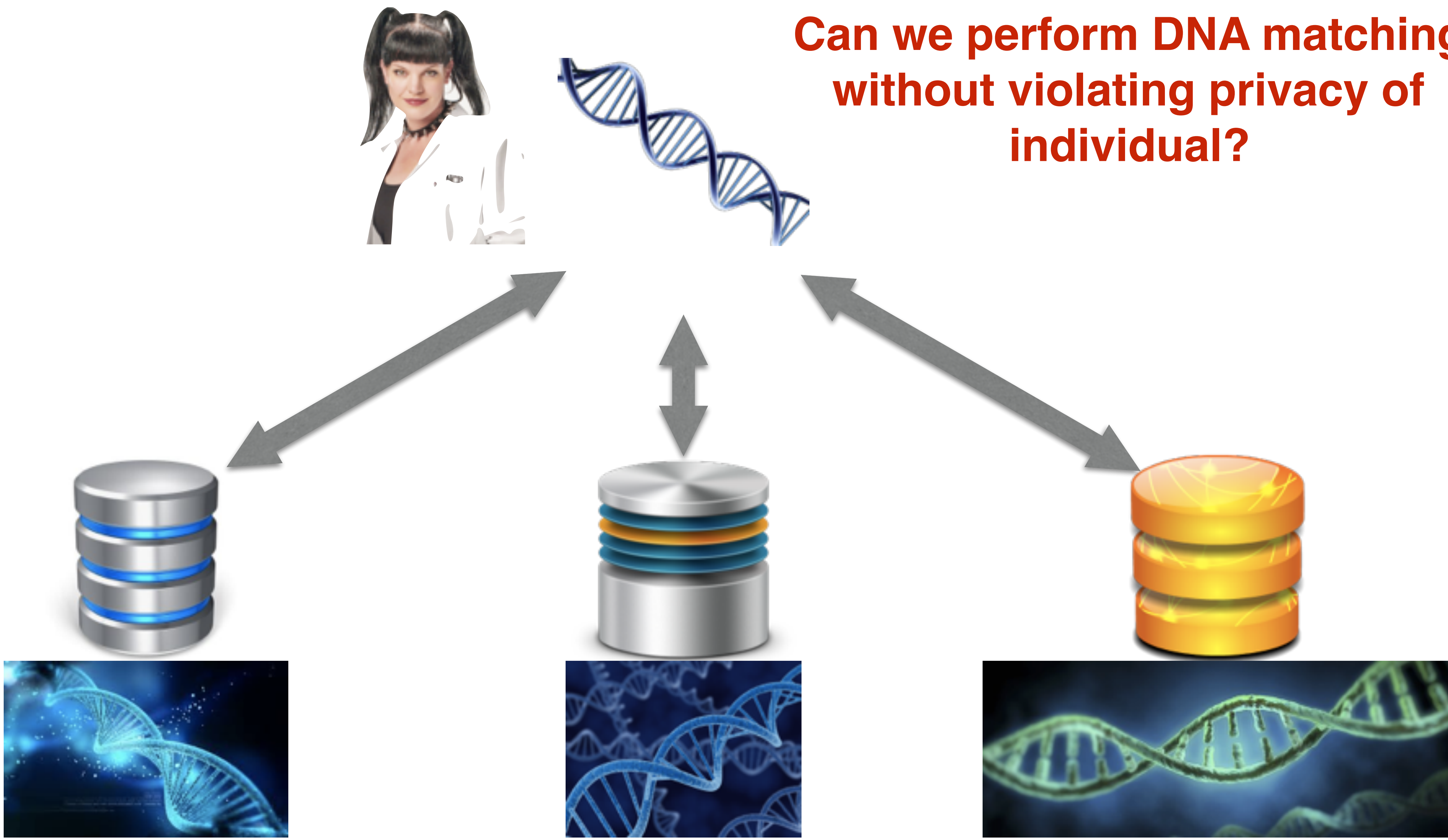
Want: The only information revealed is that there was a match, no more

Data Mining Medical Databases



Law Enforcement

Can we perform DNA matching without violating privacy of individual?



(In)Secure Skies

Feb 10, 2009: Two satellites, Iridium 33 and Kosmos-2251, collided

Unlikely that Governments will share Location and Trajectory of Military Satellites

How can governments compute "safe" trajectories without sharing private data?

General Problem

Common Input: f

x

y

General Problem

<u>Goals</u>:

- **Correctness**: Both parties learn $f(x,y)$

<u>Common Input</u>: f

x

y

General Problem

Goals:

- **Correctness**: Both parties learn $f(x,y)$
- **Security**: Each party only learns $f(x,y)$

Common Input: f

x

y

Remarks

Wlog, we will consider:

- **Symmetric functions**: $f(x,y) = (z_1,z_2)$ where $z_1=z_2$

Remarks

Wlog, we will consider:

- **Symmetric functions**: $f(x,y) = (z_1,z_2)$ where $z_1=z_2$
  - Think: Asymmetric functions?

Remarks

Wlog, we will consider:

- **Symmetric functions**: $f(x,y) = (z_1,z_2)$ where $z_1=z_2$
  - Think: Asymmetric functions?
  - $g((x,r),(y,s))$: $(z_1,z_2) = f(x,y)$. Output z_1+r, z_2+s

Remarks

Wlog, we will consider:

- **Symmetric functions**: $f(x,y) = (z_1,z_2)$ where $z_1=z_2$
    - <u>Think</u>: Asymmetric functions?
    - $g((x,r),(y,s))$: $(z_1,z_2) = f(x,y)$. Output z_1+r, z_2+s
- **Deterministic functions**

Remarks

Wlog, we will consider:

- **Symmetric functions**: $f(x,y) = (z_1,z_2)$ where $z_1=z_2$
    - Think: Asymmetric functions?
    - $g((x,r),(y,s))$: $(z_1,z_2) = f(x,y)$. Output z_1+r, z_2+s
- **Deterministic functions**
    - Think: Randomized functions?

Remarks

Wlog, we will consider:

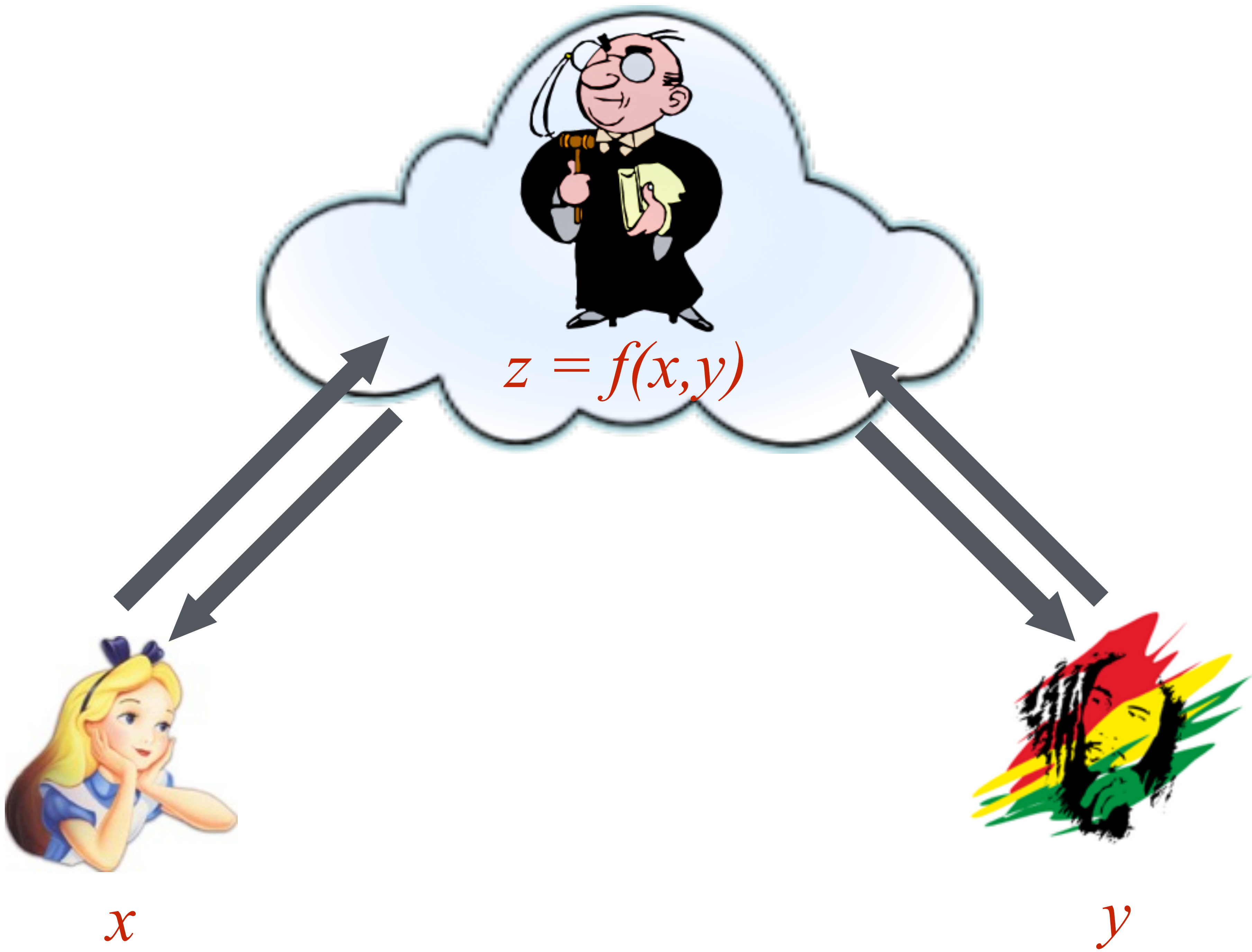
- **Symmetric functions**: $f(x,y) = (z_1,z_2)$ where $z_1=z_2$
  - Think: Asymmetric functions?
  - $g((x,r),(y,s))$: $(z_1,z_2) = f(x,y)$. Output z_1+r, z_2+s
- **Deterministic functions**
  - Think: Randomized functions?
  - $g((x,r),(y,s))$: Output $f(x,y;r+s)$

General Problem

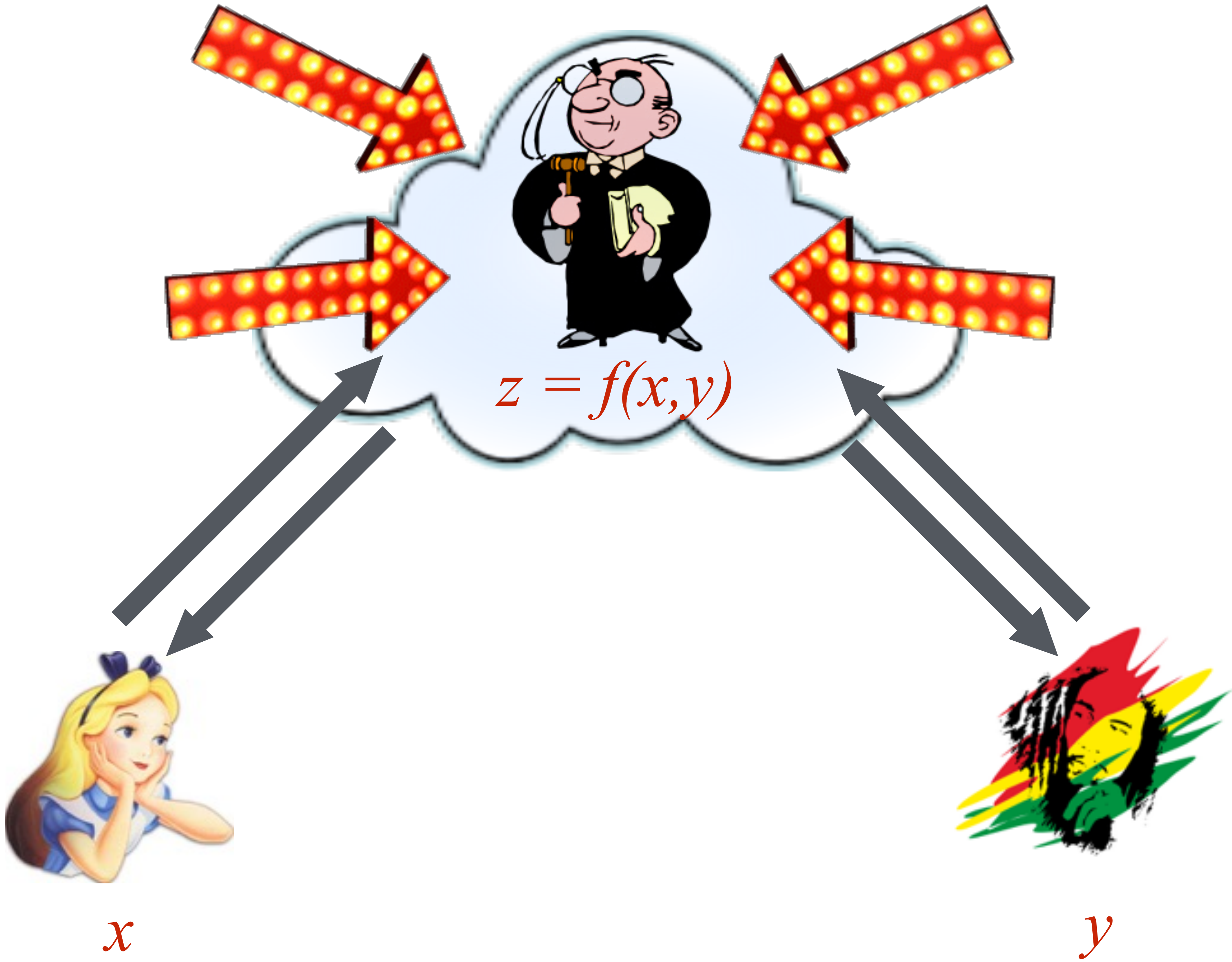
x

y

General Problem



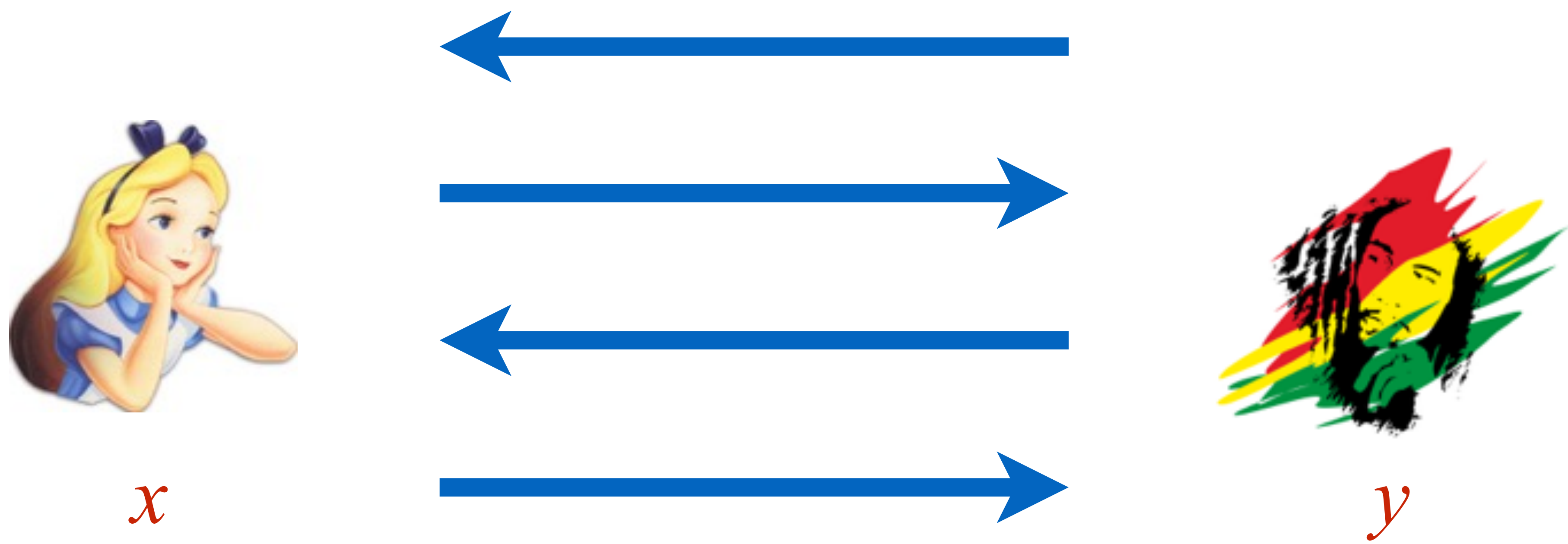
General Problem



<u>Problem</u>: Where to find this trusted party?

Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]

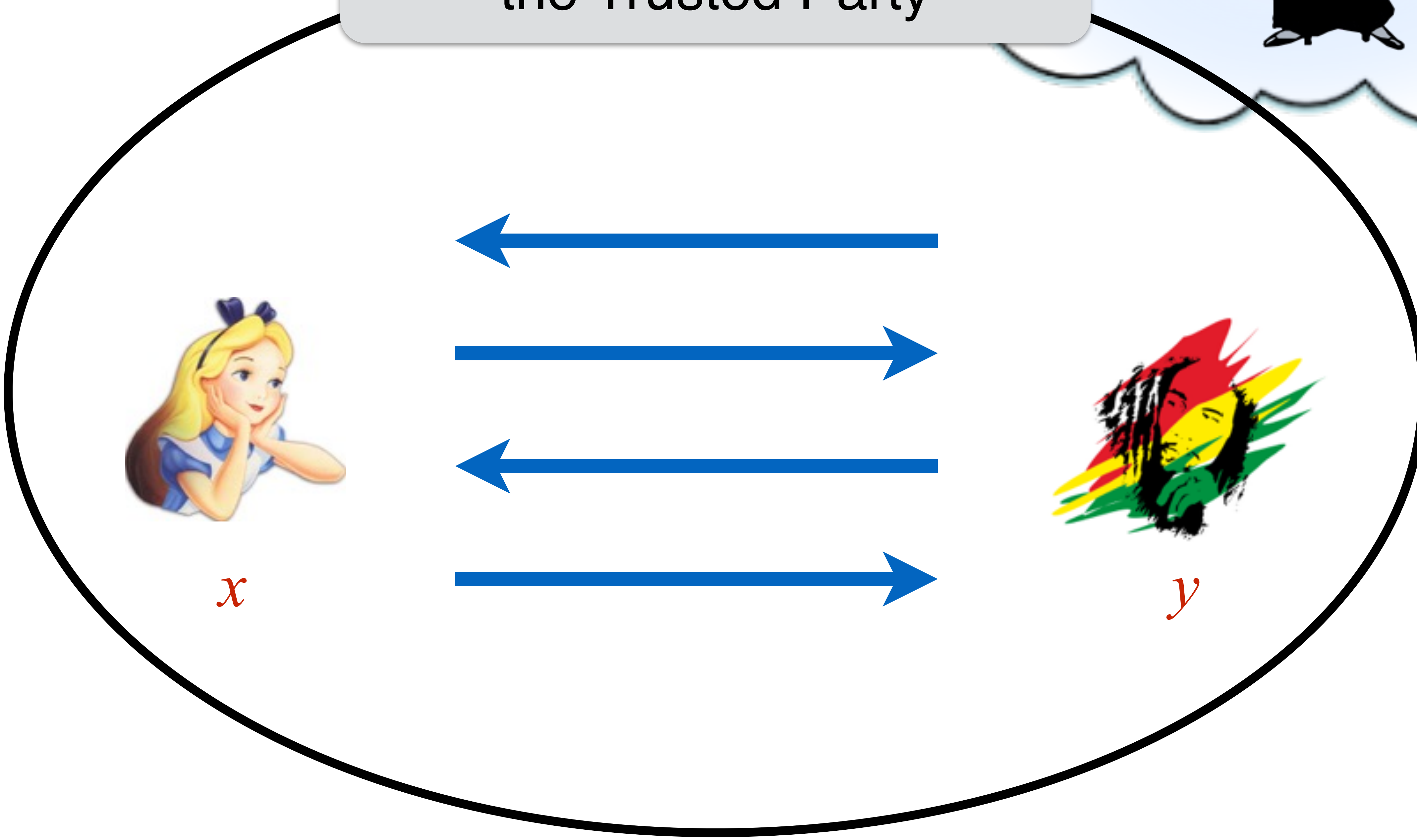


<u>Goal</u>: Compute $f(x,y)$

Secure Computation

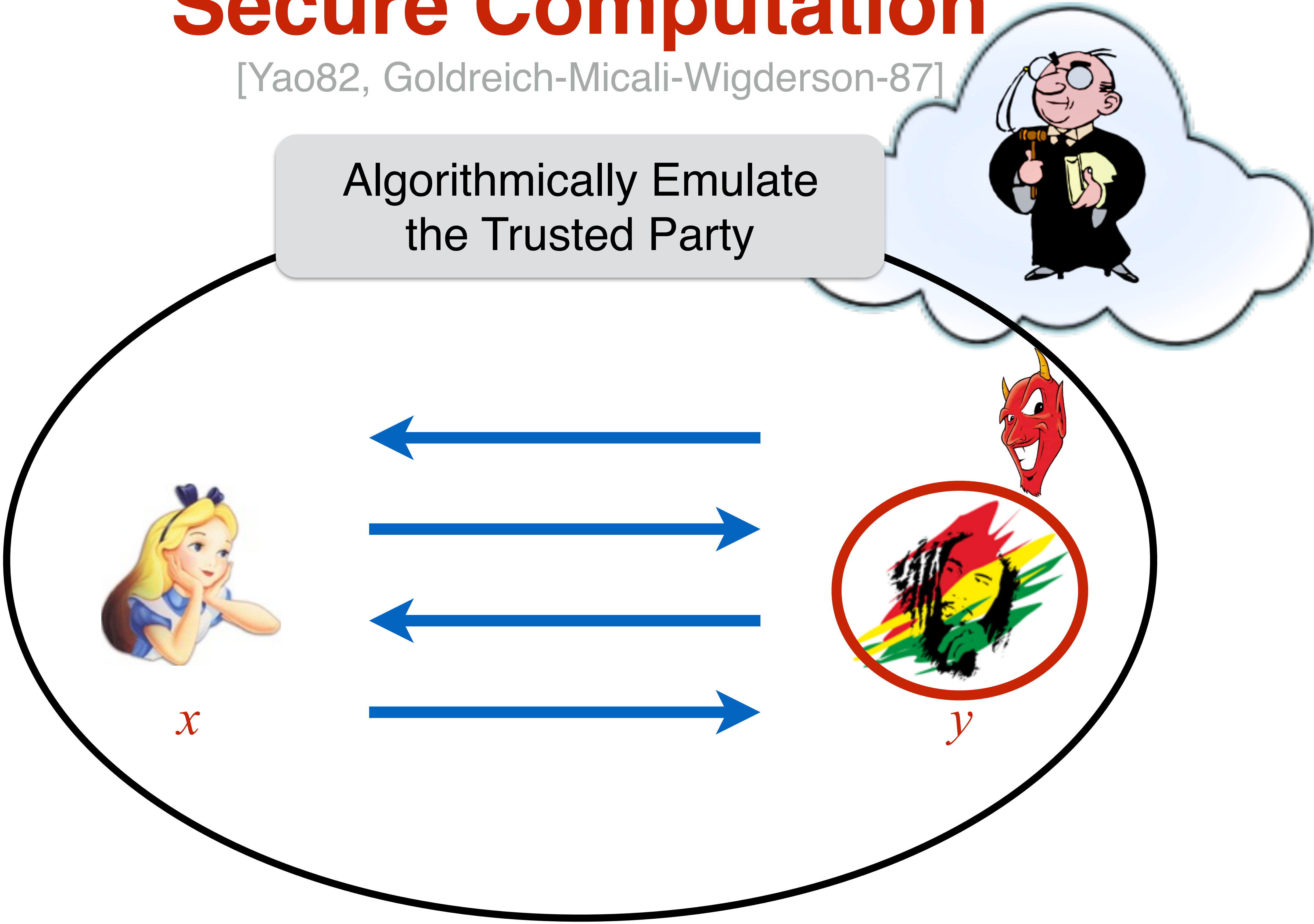
[Yao82, Goldreich-Micali-Wigderson-87]

Algorithmically Emulate
the Trusted Party



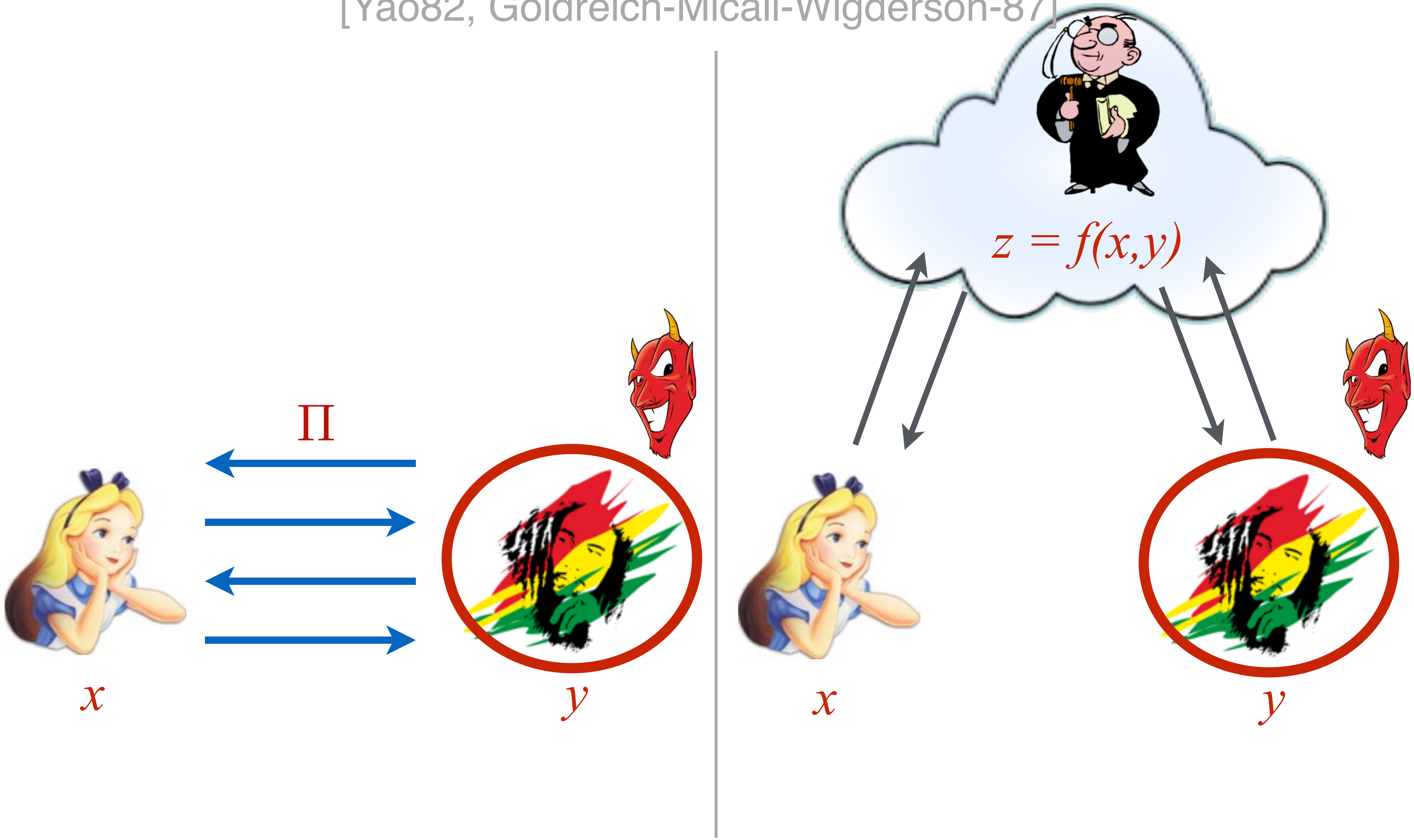
Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]



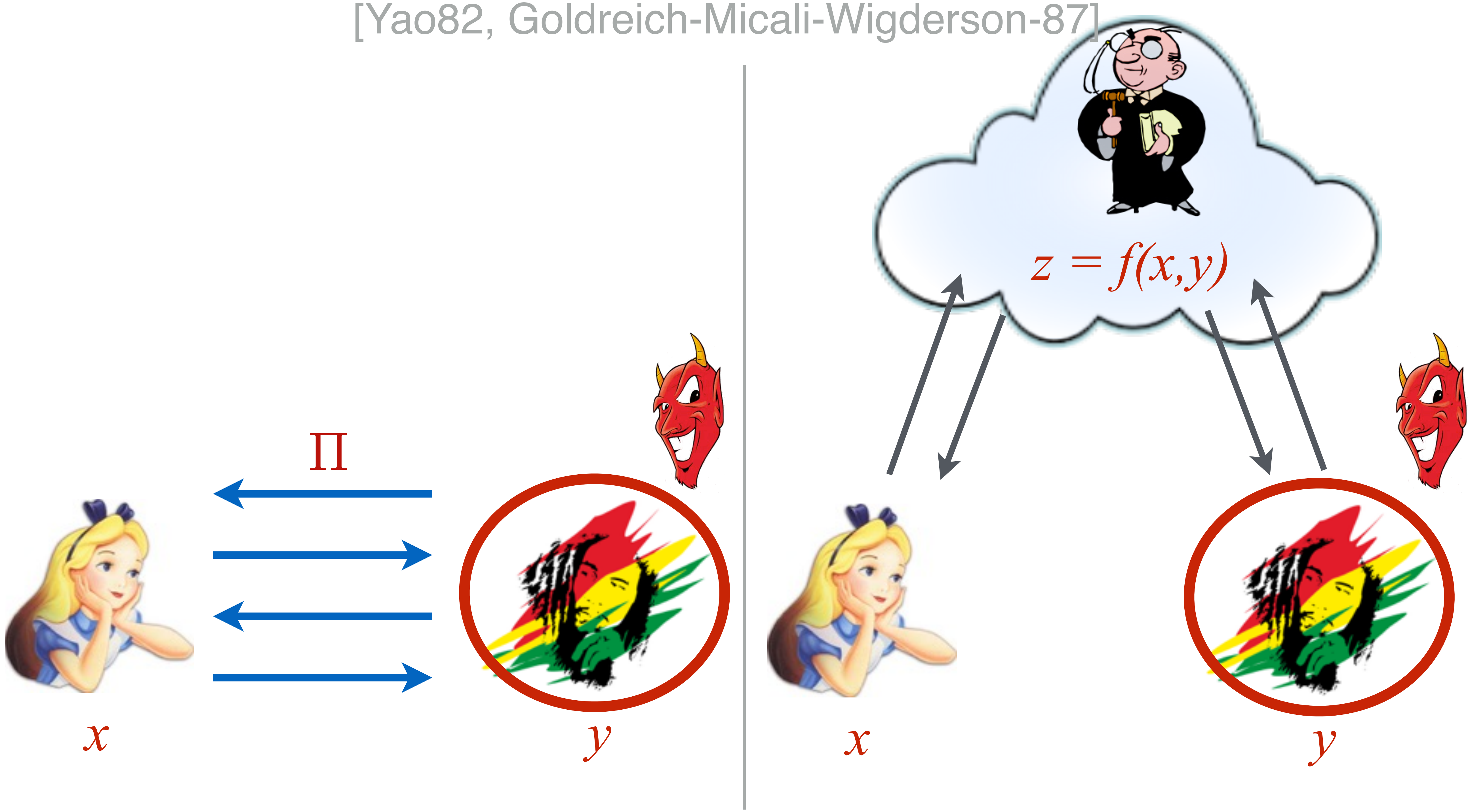
Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]



Secure Computation

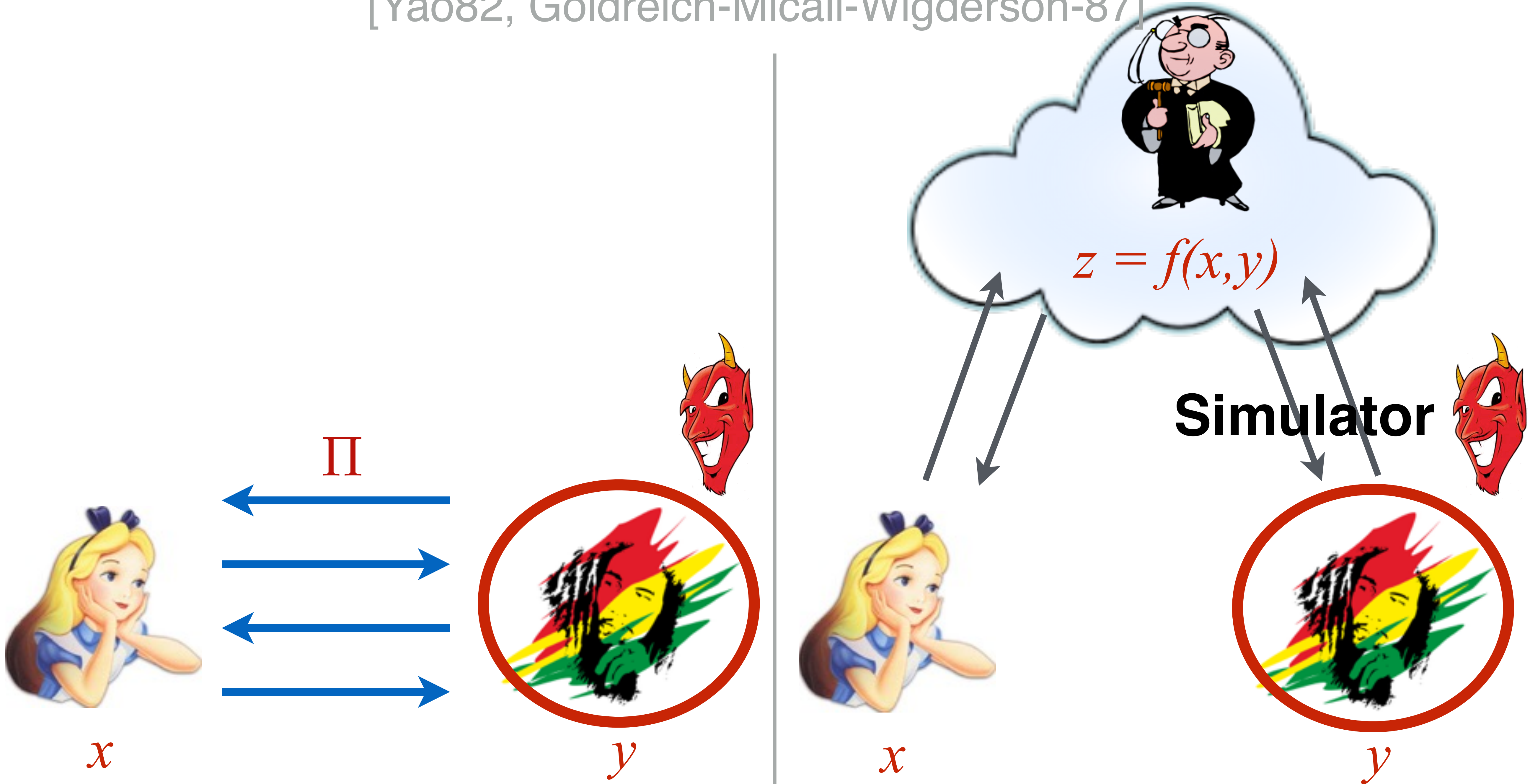
[Yao82, Goldreich-Micali-Wigderson-87]



Protocol Π securely computes f if adversary learns same information in left and right worlds

Secure Computation

[Yao82, Goldreich-Micali-Wigderson-87]



Security defined formally via simulation

Secure (Two-Party) Computation

[Yao82, Goldreich-Micali-Wigderson-87]

Def (Secure Computation): Protocol Π securely computes f if for every PPT adversary $\mathbf{A}$, there exists a PPT simulator $\mathbf{S}$ s.t. for all inputs (x,y) to f, and all auxiliary information z,

$$\mathbf{View}_{\mathrm{real}}(x,y,z) \sim \mathbf{View}_{\mathrm{ideal}}(x,y,z)$$

where,

- $\mathbf{View}_{\mathrm{real}}$: = everything seen by $\mathbf{A}$ (including input, random tape, aux input and protocol messages) and output of honest party
- $\mathbf{View}_{\mathrm{ideal}}$: = output of $\mathbf{S}$ and output of honest party

Remarks

- **Passive vs Active adversaries**
  - Passive adversaries follow the protocol. Active adversaries may use arbitrary strategy

- Must modify ideal world to capture active adv
  - $\mathbf{S}$ can send any y^* to trusted party
  - $\mathbf{S}$ can tell trusted party whether honest party should get output or not