What is iO and its application

introduction to indistinguishability obfuscation

Pia Park
Research Engineer (grantee)
Machina iO, PSE'EF

Trust-less Applications

- Bitcoin and Ethereum Bridge
- Encrypted mempools
- Private Voting
- Private Auctions
- zk-TLS

Trust-less Applications Really?

- Bitcoin and Ethereum Bridge
- Encrypted mempools
- Private Voting
- Private Auctions
- zk-TLS

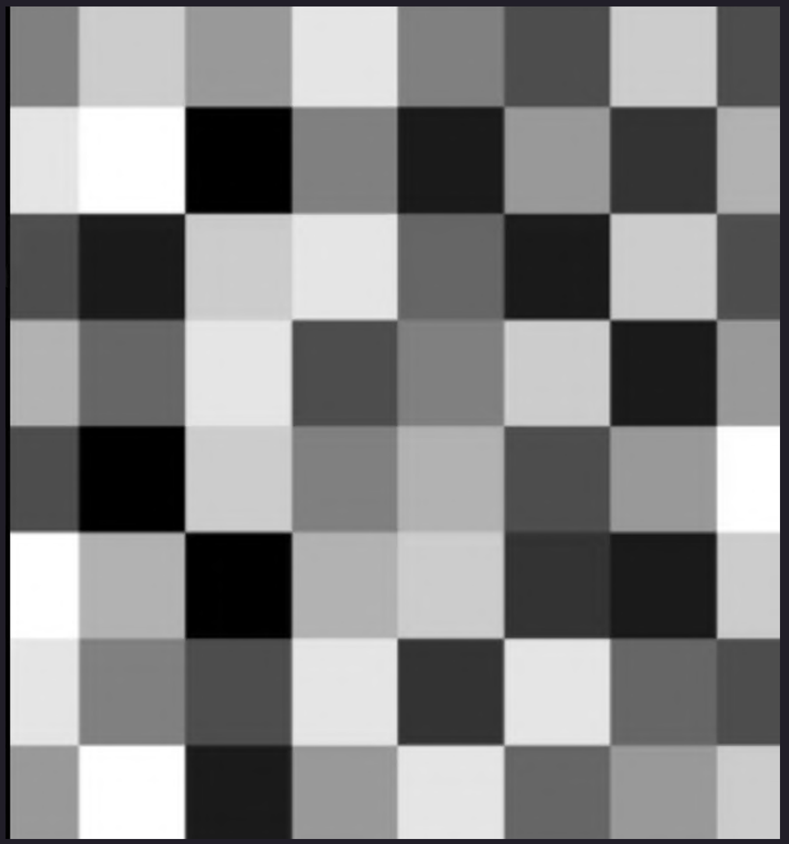
- Threshold-based cryptography
  - multi party computation
  - multi-key FHE

Try Pitch

Trust-less Applications Really!

- Bitcoin and Ethereum Bridge
- Encrypted mempools
- Private Voting
- Private Auctions
- zk-TLS

- indistinguishability obfuscation



Index

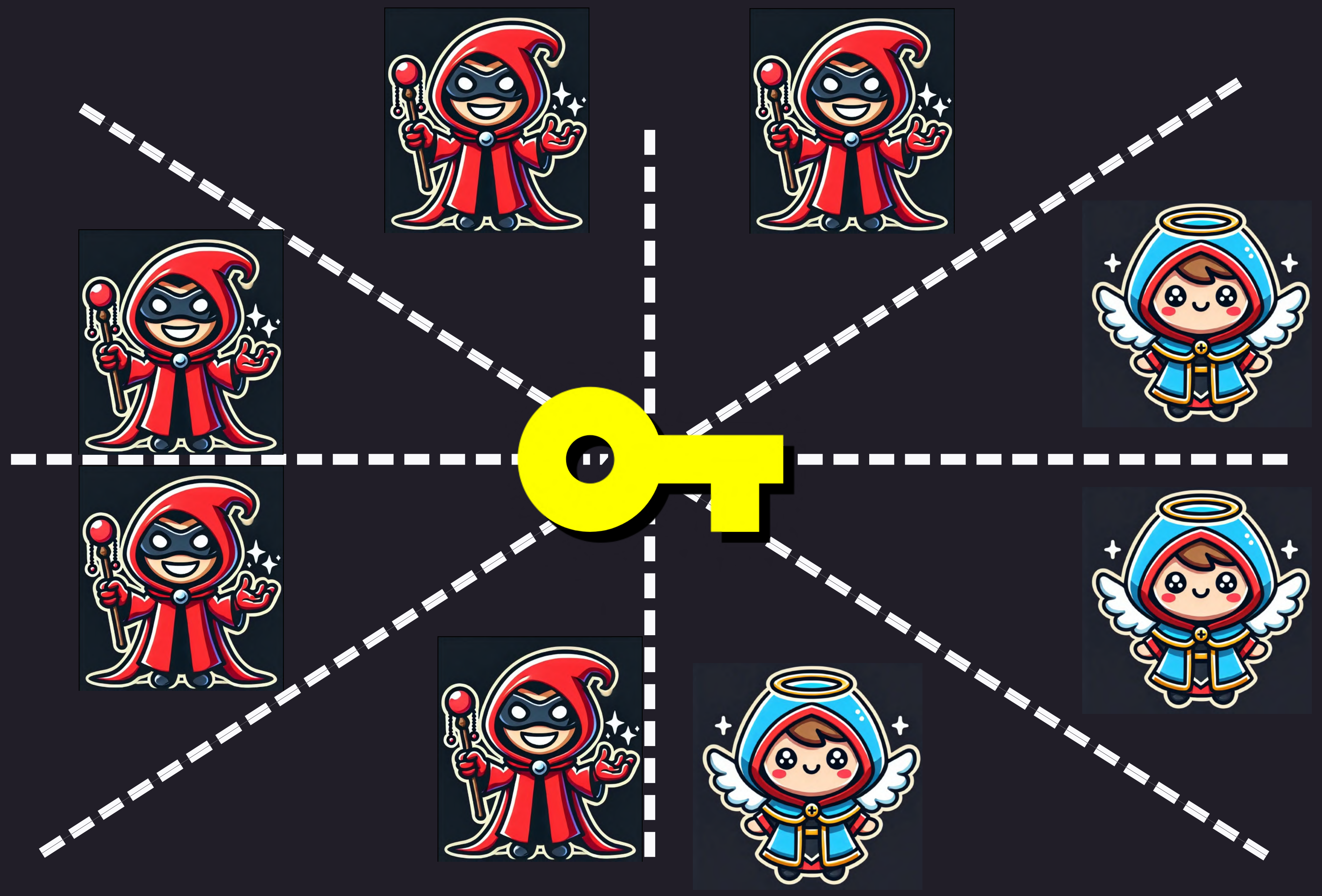
1. Limitation of current trust-less application
2. What is iO
3. Application of iO
4. Our contributions and progress

Bitcoin and Ethereum Bridge

signature

key

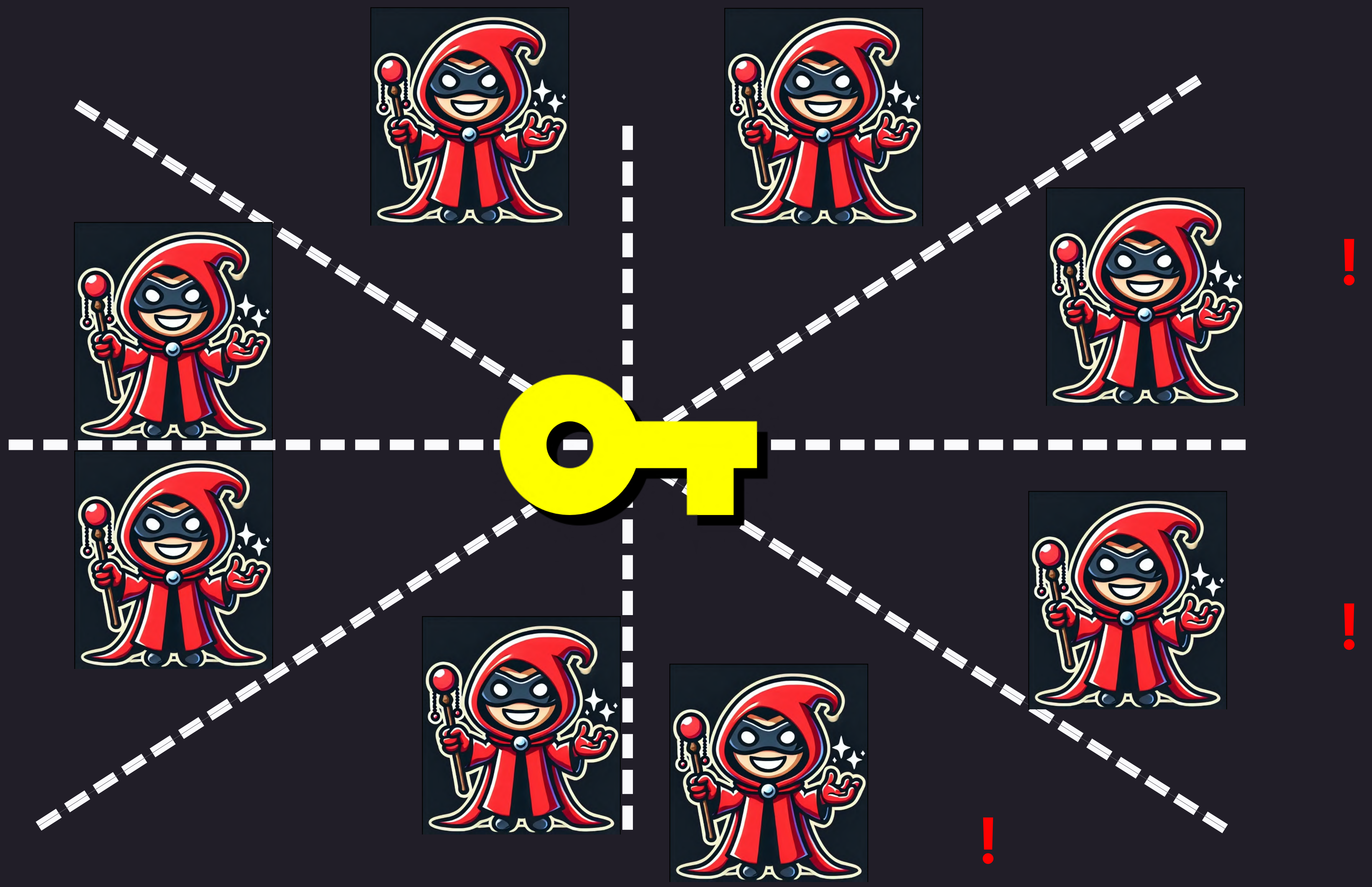
Multi-Party Computation



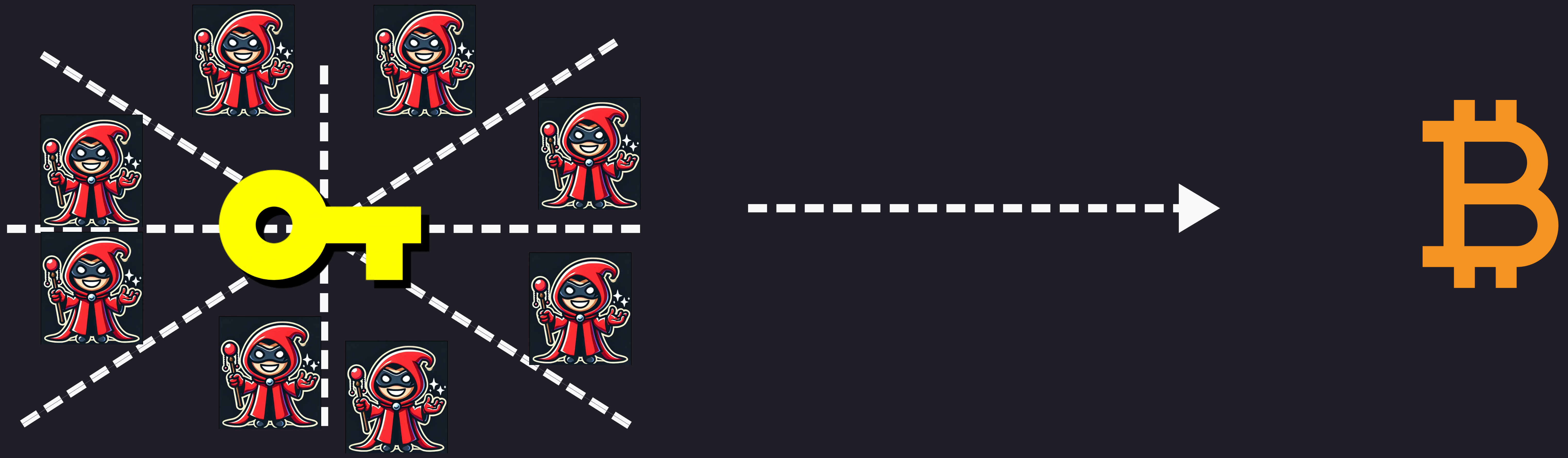
threshold

Try Pitch

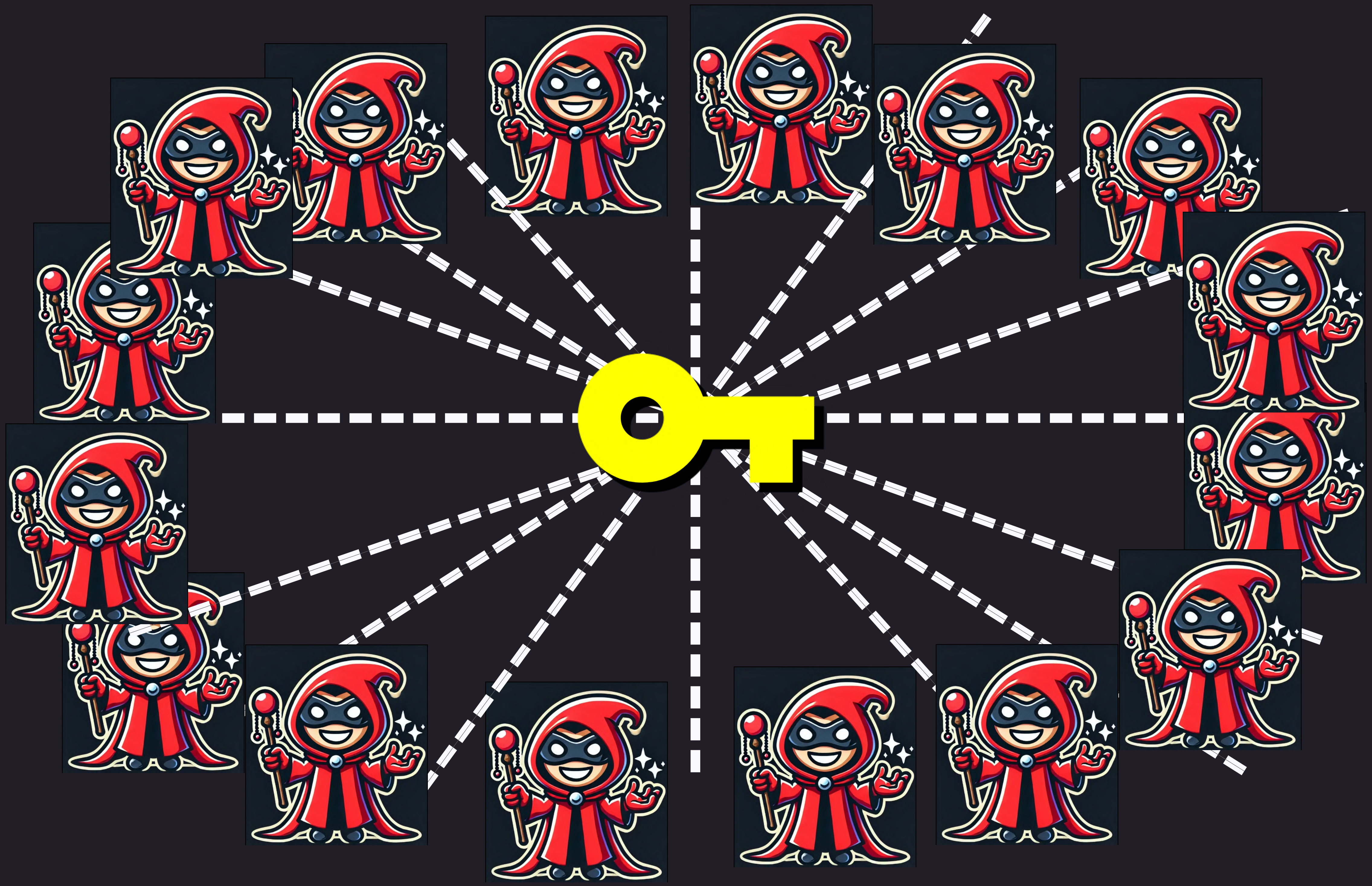
Limitation of Multi-Party Computation



Limitation of Multi-Party Computation



Limitation of Multi-Party Computation



Try Pitch

Obfuscation fixes this

What is Obfuscation?

Obfuscation as a program compiler

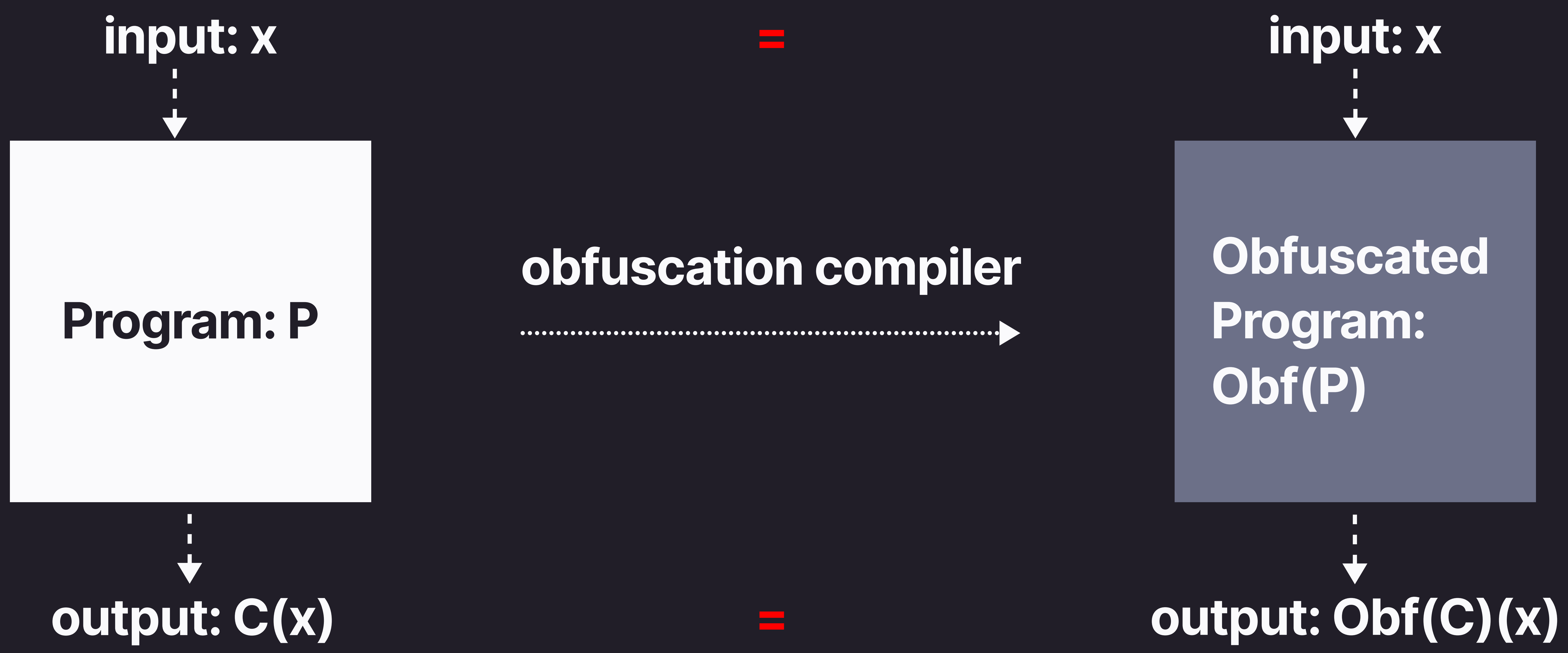
Program: P

obfuscation compiler

Obfuscated Program: Obf(P)

What is Obfuscation?

Obfuscation as a program compiler



Useful mental model of Obfuscation

$$\mathsf{Obf}(f(k,\bullet)) \rightarrow \widetilde{C}$$

obfuscation

$$\mathsf{Eval}(\widetilde{C}, x) \rightarrow f(k,x)$$

evaluation

Useful mental model of Obfuscation

- non interactive
- f and Obf(C) is public
- k is private
- binded to f

$$\mathrm{Obf}(f(k, \bullet)) \to \widetilde{C}$$

$$\mathrm{Eval}(\widetilde{C}, x) \to f(k, x)$$

obfuscation

evaluation

Useful mental model of Obfuscation

$\mathsf{Obf}(f(k, \bullet)) \rightarrow \widetilde{C}$

$\mathsf{Eval}(\widetilde{C}, x) \rightarrow f(k, x)$

e.g. Bitcoin and Ethereum Bridge

$f(k, x) =$

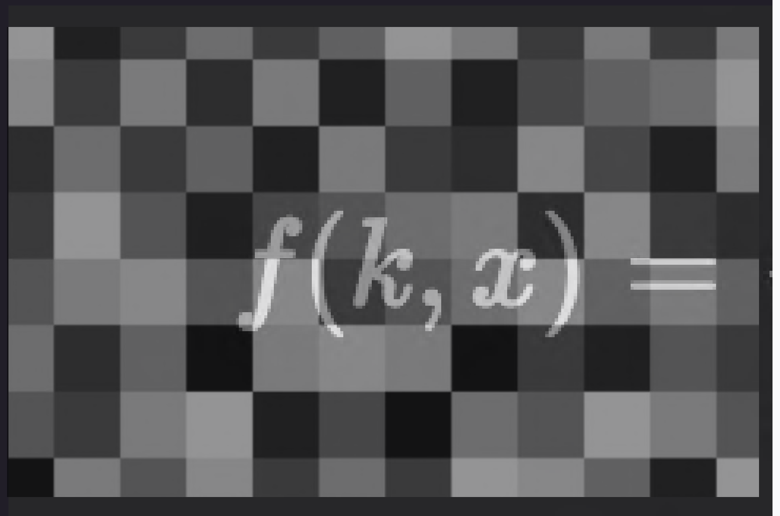
if given proof(=x) of burning WBTC on ethereum is valid
→ I want to get a signature to withdraw bitcoins from hardcoded wallet secret key (=k)

Useful mental model of Obfuscation

$\mathrm{Obf}(f(k, \bullet)) \rightarrow \widetilde{C}$

$\mathrm{Eval}(\widetilde{C}, x) \rightarrow f(k, x)$

e.g. Bitcoin and Ethereum Bridge



if given proof(=x) of burning WBTC on ethereum is valid
→ I want to get a signature to withdraw bitcoins from hardcoded wallet secret key (=k)

Applications of Obfuscation

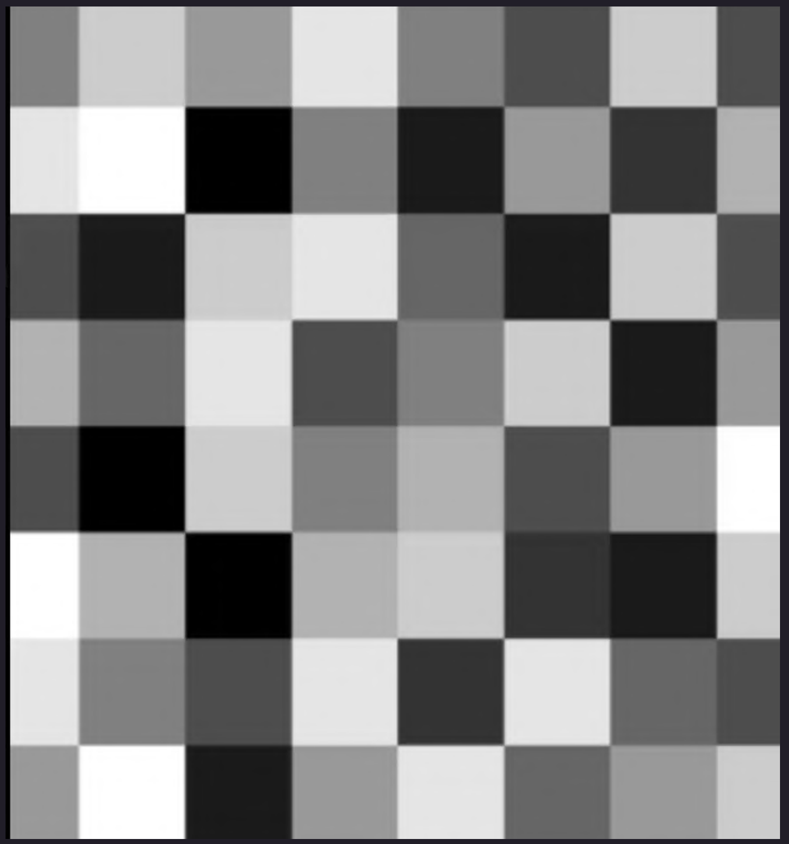
You want to encode specific secret within a specific program and allow others to use it

- Private non-interactive computation
    - Secrets are hardcoded in the programs then obfuscated
    - Anyone can execute the program and learn nothing about secret

Trust-less Applications Really!

- Bitcoin and Ethereum Bridge
- Encrypted mempools
- Private Voting
- Private Auctions
- zk-TLS

- indistinguishability obfuscation



Possible futures of the Ethereum protocol, part 6: The Splurge

2024 Oct 29 See all posts

manufacture, and are so powerful that they are not allowed in duels. Similarly, in cryptography, we have the trio of Egyptian god protocols:

Comments on iO

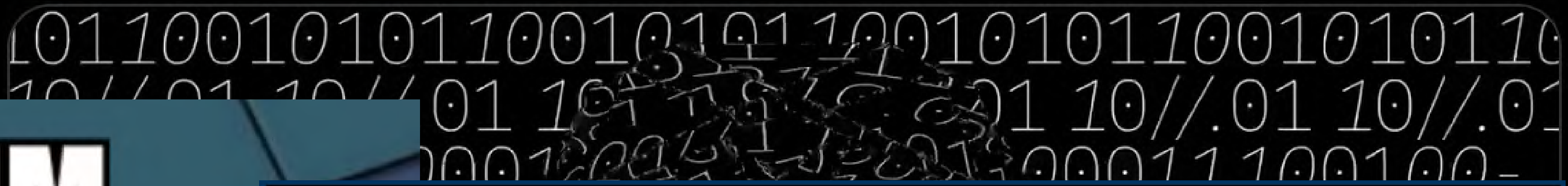
Is Indistinguishability Obfuscation Real?

Asked 4 years, 3 months ago   Modified 4 years, 2 months ago   Viewed 557 times

8

I've recently stumbled upon an interestin
indistinguishability obfuscation (iO) 's th
relatively recent paper by Jain, Lin, and S
the work presented in this article relies o
NC^0. The authors of the paper explicitly

Indistinguishability obfuscation was so difficult to attain that some computer scientists considered it impossible and abandoned it. In August, Huijia Lin, Amit Sahai and Aayush Jain finally achieved it.
quantamagazine.org/computer-scien...



ℵv @alpeh_v · Mar 17
I love io it's like "we really can do anything" and I'm like wow so cool I can imagine so many uses then they go "yeah it's so cool we encrypted a single addition and it takes 600 gb of ram"
4   42   2.2K

pia
@0xpiapark

you just can do things.. soon(tm)

A cryptographic master tool called i
years seemed too good to be true. Th
can work.

Our contributions and progress

README.md

Machina iO

Machina iO ("mah-kin-ah") a project within PSE, aims to move iO from theory to practice. We publish papers and write code.

Sora Suegami
Enrico Bottazzi
Pia Park

Our contributions and progress

✅ **Straightforward iO construction**

Diamond iO: A Straightforward Construction of Indistinguishability Obfuscation from Lattices

Sora Suegami*, Enrico Bottazzi†

February 2025

Abstract

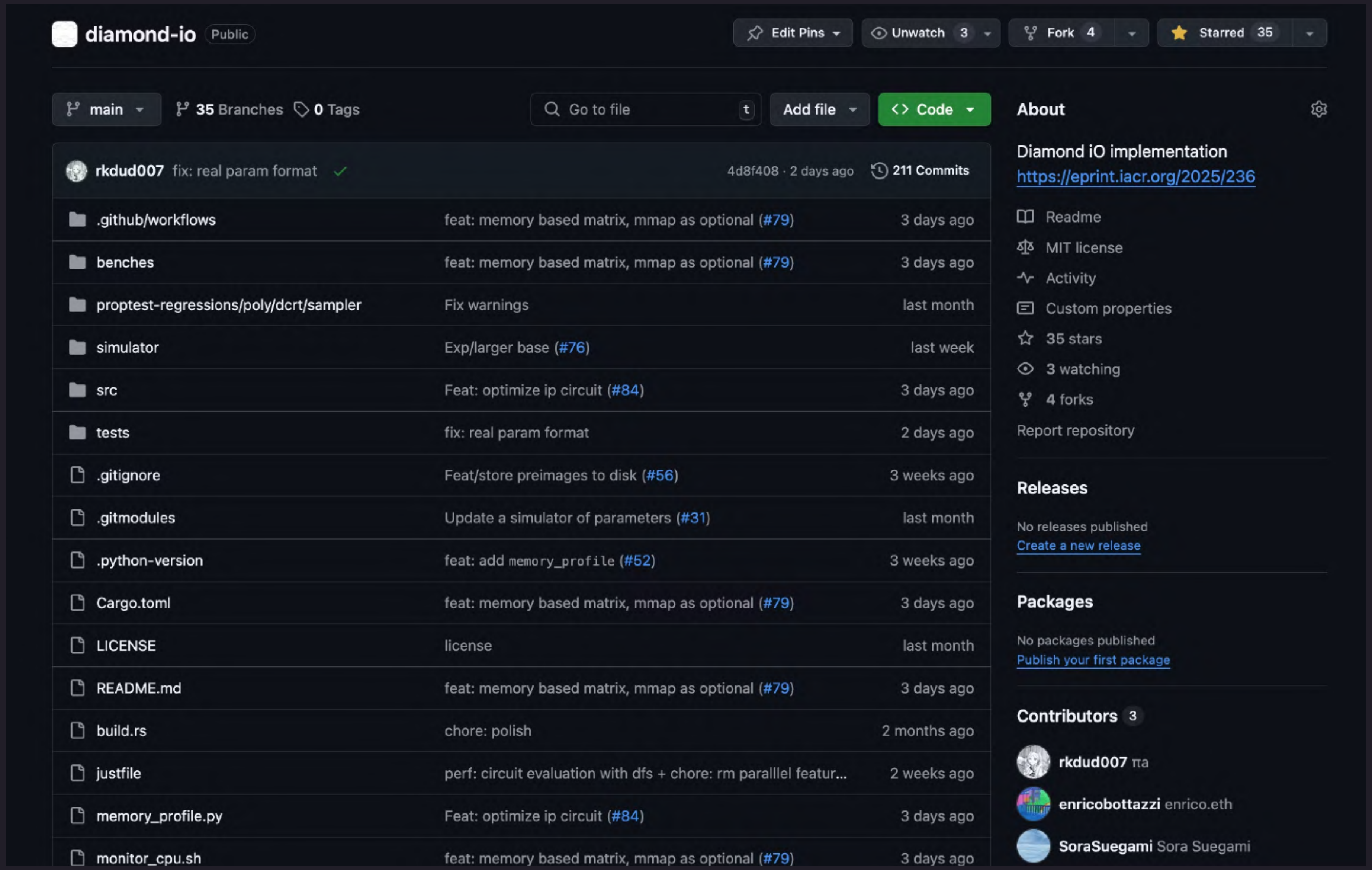
Indistinguishability obfuscation (iO) has seen remarkable theoretical progress, yet it remains impractical due to its high complexity and inefficiency. A common bottleneck in recent iO schemes is the reliance on bootstrapping techniques from functional encryption (FE) into iO, which requires recursively invoking the FE encryption algorithm for each input bit—creating a significant barrier to practical iO schemes.

In this work, we propose diamond iO, a new lattice-based iO construction that replaces the costly recursive encryption process with lightweight matrix operations. Our construction is proven secure under the learning with errors (LWE) and evasive LWE assumptions, as well as our new assumption—all-product LWE—in the pseudorandom oracle model. By leveraging the FE scheme for pseudorandom functionalities introduced by Agrawal et al. (ePrint'24) in a non-black-box manner, we remove the reliance on prior FE-to-iO bootstrapping techniques and thereby significantly reduce complexity. A remaining challenge is to reduce our new assumption to standard assumptions such as LWE, further advancing the goal of a practical and sound iO construction.

Our contributions and progress

✅ Straightforward iO construction

✅ End to End implementation of Diamond iO with secure param



Our contributions and progress

✅ **Straightforward iO construction**

✅ **End to End implementation of Diamond iO with secure param**

Existing simulated result (BOK+15, ESSoS):

More than 10^27 years
for obfuscation and evaluation

VS

Our sufficiently secure parameters (target security parameter 80, 1 input bit, around 60 gates):

Less than one hour
for obfuscation and evaluation
(we are double checking this result)

Our contributions and progress

✅ **Straightforward iO construction**

✅ **End to End implementation of Diamond iO with secure param**

🏗️ **Let's make iO practical!**

A. Larger input size
B. Complex circuit logic and depth

Existing simulated result (BOK+15, ESSoS):

More than 10^27 years
for obfuscation and evaluation

VS

Our sufficiently secure parameters (target security parameter 80, 1 input bit, around 60 gates):

Less than one hour
for obfuscation and evaluation
(we are double checking this result)

Q & A

Appendix

Indistinguishability Obfuscation (iO)

Obfuscations of two circuits with the same functionality, i.e., input-output relation, are **indistinguishable**.

Circuit C_0 $\neq$ Circuit C_1 $\Rightarrow$ Obfuscated Circuit $\mathrm{Obf}(C_0)$ $\approx$ Obfuscated Circuit $\mathrm{Obf}(C_1)$

$$C_0(x) = C_1(x)$$

Bootstrapping from FHE+ZKP to iO

Obfuscate: P_1 generates an obfuscated program $\widetilde{C} \leftarrow \mathsf{Obf}(f(k, \cdot))$. f and $\widetilde{C}$ is public, k is secret.

- Internally P_1 generate a ciphertext $\mathsf{Enc}(k)$ for secret k

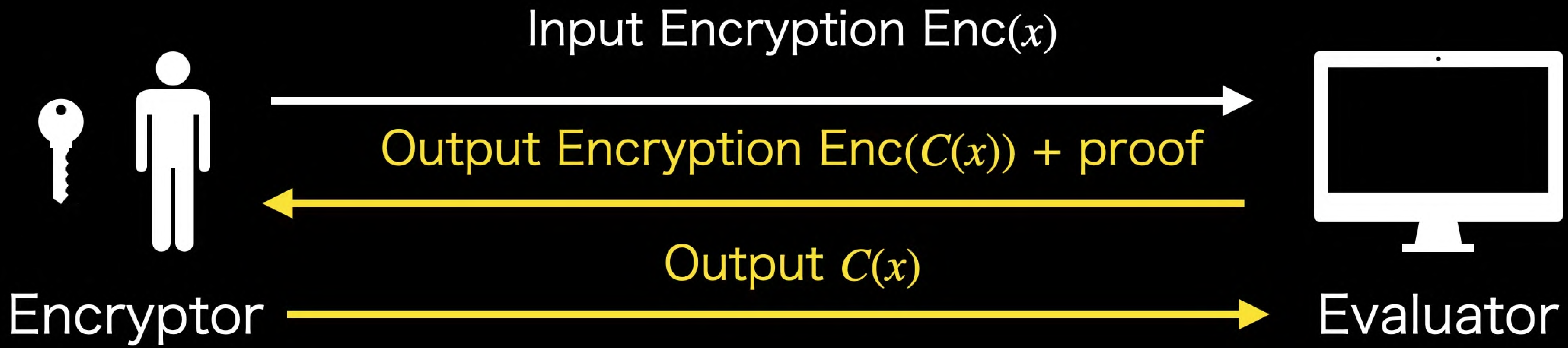
Evaluate: P_2 runs $\widetilde{C}$ over their input x to obtain $f(k, x)$.

- Internally P_2 dynamically choose an input x and homorphically evaluate f over x and $\mathsf{Enc}(k)$ to obtain a new ciphertext $\mathsf{Enc}(f(k, x))$
- P_2 decrypt $\mathsf{Enc}(f(k, x))$ and obtain $f(k, x)$ but not learning about k

Comparison with FHE

1. FHE is **malleable**: FHE itself cannot restricted a function being evaluated on given ciphertexts. A zk proof of homomorphic evaluation result is necessary to confirm the validity of the output encryption.

2. FHE requires **interaction**: An party holding a private key needs to remain online until receiving the output encryption and the proof.



Difference with FHE

HE (hormomorphic encryption)

What's different from FHE: if this evalutate, homomorphic operation can be perform over arbitrary addition and multiplication, we consider FHE(fully hormomorphic encryption)

What's different from FE: if you hold sk, can able to decrypt x as well. If encryptor(P_2) want to hide x from decryptor(P_1), you'd need functional encryption

- **Setup(KeyGen)**: P_1 setup key $(pk, sk) \leftarrow \text{Setup}(1^\lambda)$ and pk is public, sk is secret
- **Encrypt**: P_2 encrypt over public key pk and plain text x. $ct \leftarrow \text{Enc}(pk, x)$
- **Evaluate**: P_2 evaluate(homomorphic operation) over public key pk. $ct' \leftarrow \text{Eval}(pk, C, ct)$
- **Decrypt**: P_1 can decrypt using secret key $f(x) \leftarrow \text{Dec}(sk, ct')$.