

Nhập môn An toàn Thông tin

Chứng minh không để lộ tri thức
(Zero Knowledge Proof)

StarkWare Industries

Article [Talk](#)

From Wikipedia, the free encyclopedia

StarkWare Industries is an Israeli software company that specializes in [cryptography](#). It develops [zero-knowledge proof](#) technology that compresses information to address the [scalability](#) problem of the [blockchain](#), and works on the [Ethereum](#) platform.^[1] In May 2022 the company's estimated value was \$8 billion, an increase from \$2 billion six months earlier.^[2]

Nội dung trình bày

ZKP là gì?

Chứng minh tương tác

Chứng minh không lộ
tri thức

Yêu cầu của ZKP

Ví dụ

Tô màu đồ thị

Giao thức định danh
Schnorr

ZK-SNARK

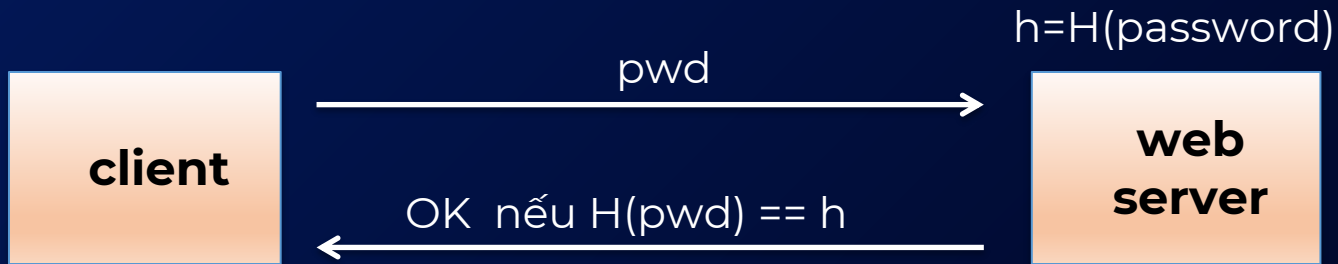
Chứng minh không
tương tác

SNARK là gì?

Một số kiểu SNARK

Công nghệ ZK-SNARK

Chứng minh tương tác



- Client tương tác với server để chứng minh: *"client biết password"*.
- Nhưng sau khi tương tác, server cũng biết password!

Chứng minh không để lộ tri thức (ZKP)



Prover **P**



Verifier **V**

Chấp nhận
hoặc Bác bỏ

- **P** tương tác với **V** để chứng minh một *khẳng định* là đúng
- **V** hoàn toàn bị thuyết phục
- Quá trình tương tác không để lộ tri thức

Yêu cầu của Chứng minh không để lộ tri thức

- **Tính đầy đủ** (Completeness). Nếu \mathbf{P} là trung thực, thì \mathbf{P} cuối cùng cũng sẽ thuyết phục được \mathbf{V} .
- **Tính đúng đắn** (Soundness). \mathbf{P} chỉ thuyết phục được \mathbf{V} nếu khẳng định là đúng.
- **Tính không để lộ tri thức** (Zero-knowledgeness). \mathbf{V} không có được thông tin gì ngoài thông tin rằng khẳng định là đúng.

Ví dụ: Trà Sữa hay Sữa Trà?

- **P:** **Đổ trà trước, sữa sau** uống sẽ ngon hơn **đổ sữa trước, trà sau**
- **V:** Có thể phân biệt được à? Chứng minh đi.


Liệu một chứng minh bằng cách chỉ ra

- vị ngon tinh tế của việc “**đổ trà trước, sữa sau**” và
- vị kém ngon của “**đổ sữa trước, trà sau**”

có thuyết phục được bạn?

Ví dụ: Trà Sữa hay Sữa Trà?


Prover **P**

Nếm thử 
và quyết định
c là trà sữa hay
sữa trà



c

Verifier **V**

Chọn ngẫu nhiên
 là trà sữa hoặc
sữa trà

Chấp nhận nếu c đúng
Bác bỏ nếu c sai

Chứng minh không để lộ tri thức của Trà Sữa

- **Tính đầy đủ.** Nếu **P** biết cách phân biệt, thì **V** sẽ bị thuyết phục.
- **Tính đúng đắn.** Nếu **P** không biết cách phân biệt thì
 $\text{Prob}(\mathbf{V} \text{ chấp nhận}) \approx \frac{1}{2}$
nếu lặp lại 100 lần, và **V** chỉ chấp nhận nếu **P** trả lời đúng cả
 $\text{Prob}(\mathbf{V} \text{ chấp nhận}) \approx 1 / 2^{100}$
- **Tính không để lộ tri thức.** **V** không có thông tin gì ngoài **P** biết cách phân biệt Trà sữa và Sữa trà.

Nội dung trình bày

ZKP là gì?

Chứng minh tương tác

Chứng minh không lộ
tri thức

Yêu cầu của ZKP

Ví dụ

Tô màu đồ thị

Giao thức định danh
Schnorr

ZK-SNARK

Chứng minh không
tương tác

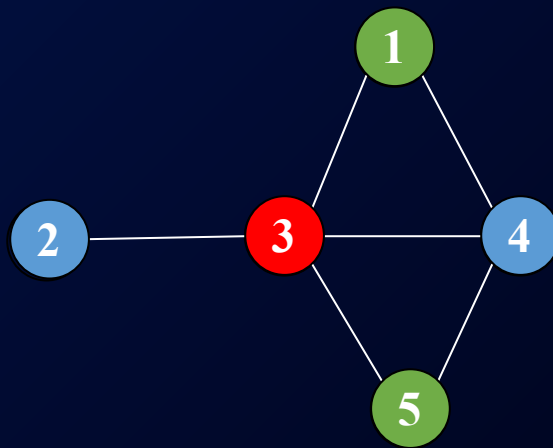
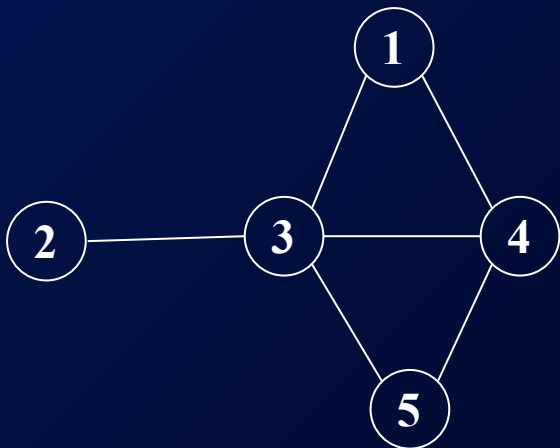
SNARK là gì?

Một số kiểu SNARK

Công nghệ ZK-SNARK

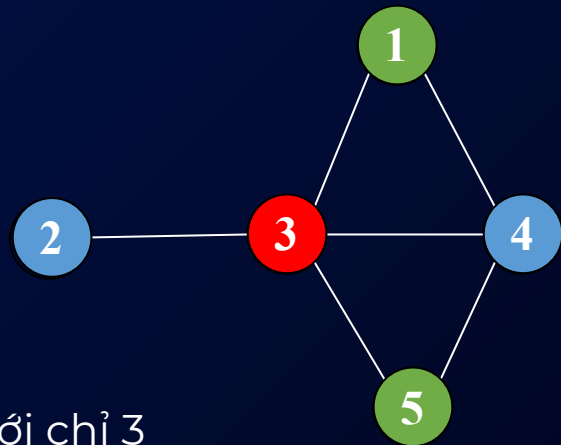
Tô đồ thị bằng 3 màu

- **Dữ liệu vào:** Đồ thị $G = (V, E)$
- **Câu hỏi:** liệu có cách gán màu lên mỗi đỉnh của G
 $V \rightarrow \{\text{R}, \text{G}, \text{B}\}$
thỏa mãn hai đỉnh kề nhau có màu khác nhau?

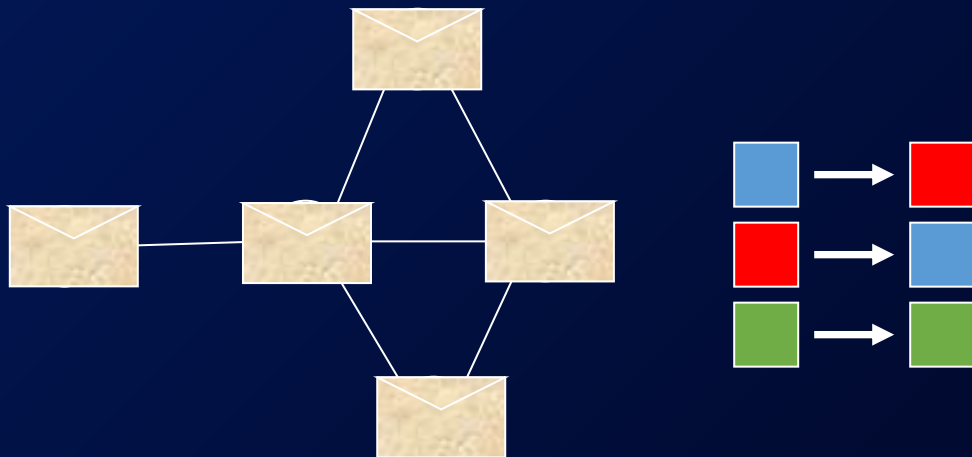


Tô màu đồ thị bằng 3 màu

- **Dữ liệu vào:** Một đồ thị G .
- Prover \mathbf{P} chứng minh rằng \mathbf{P} có thể tô màu G với chỉ 3 màu.
- Prover \mathbf{P} giữ bí mật cách tô màu.

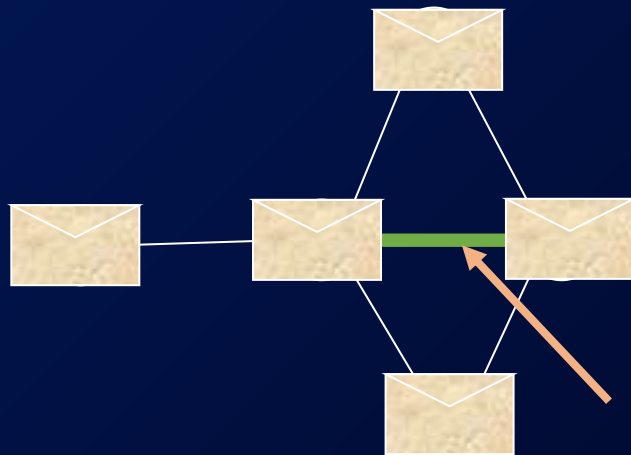


ZKP cho bài toán tô 3 màu



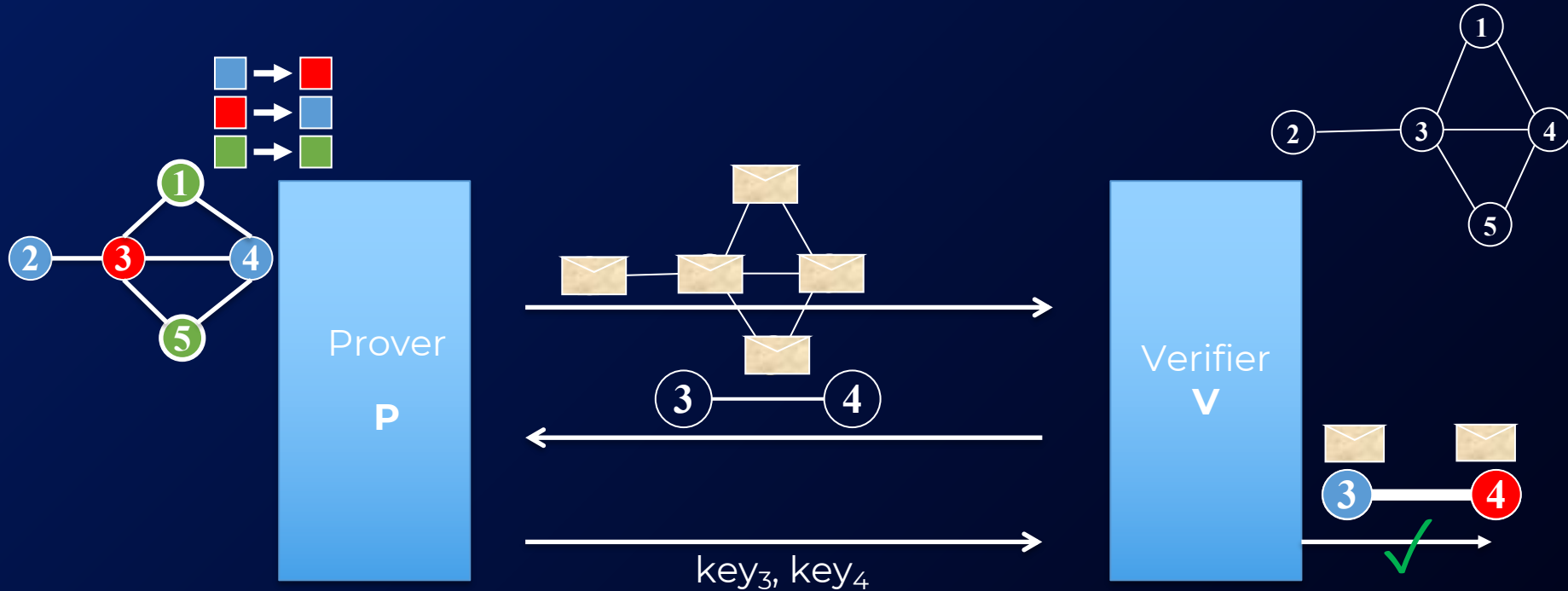
- **P** hoán vị ngẫu nhiên tập 3 màu, và gán lại màu đỉnh
- **P** giấu màu của mỗi đỉnh đồ thị trong phong bì, và gửi các phong bì này cho **V**.

ZKP cho bài toán tô 3 màu



- **V** chọn một cạnh bất kỳ
- **P** mở 2 phong bì chứa màu của 2 đỉnh đầu mút của cạnh này,
- **V** kiểm tra tính hợp lệ: 2 màu của 2 đỉnh này phải khác nhau

ZKP cho bài toán tô 3 màu: Hình thức hóa



Nếu đồ thị không thể tô 3 màu, thì **V** bác bỏ với xác suất $1/(\#cạnh)$

Chứng minh không để lộ tri thức

- **Tính đầy đủ.** Nếu đồ thị có thể tô bằng 3 màu $\Rightarrow \mathbf{V}$ sẽ bị thuyết phục
- **Tính đúng đắn:** Nếu đồ thị không thể tô bằng 3 màu \Rightarrow
 \forall Prover \mathbf{P} , Verifier \mathbf{V} bác bỏ với xác suất $\geq 1/(\#\text{cạnh})$
- **Tính không để lộ tri thức:** Có thể tô 3 màu $\Rightarrow \mathbf{V}$ không thu được thông gì, vì mọi thông tin \mathbf{V} nhận được, \mathbf{V} có thể tự tạo ra được. Cụ thể, \mathbf{V} có thể mô phỏng quá trình tương tác với \mathbf{P} .

Nội dung trình bày

ZKP là gì?

Chứng minh tương tác

Chứng minh không để lộ tri thức

Yêu cầu của ZKP

Ví dụ

Tô màu đồ thị

Giao thức định danh Schnorr

ZK-SNARK

Chứng minh không tương tác

SNARK là gì?

Một số kiểu SNARK

Công nghệ ZK-SNARK

Giao thức định danh Schnorr

- Mọi người đều biết khóa công khai pk của Alice.
- Alice muốn chứng minh rằng cô ấy có khóa bí mật sk_A tương ứng với pk_A mà không muốn lộ khóa sk_A

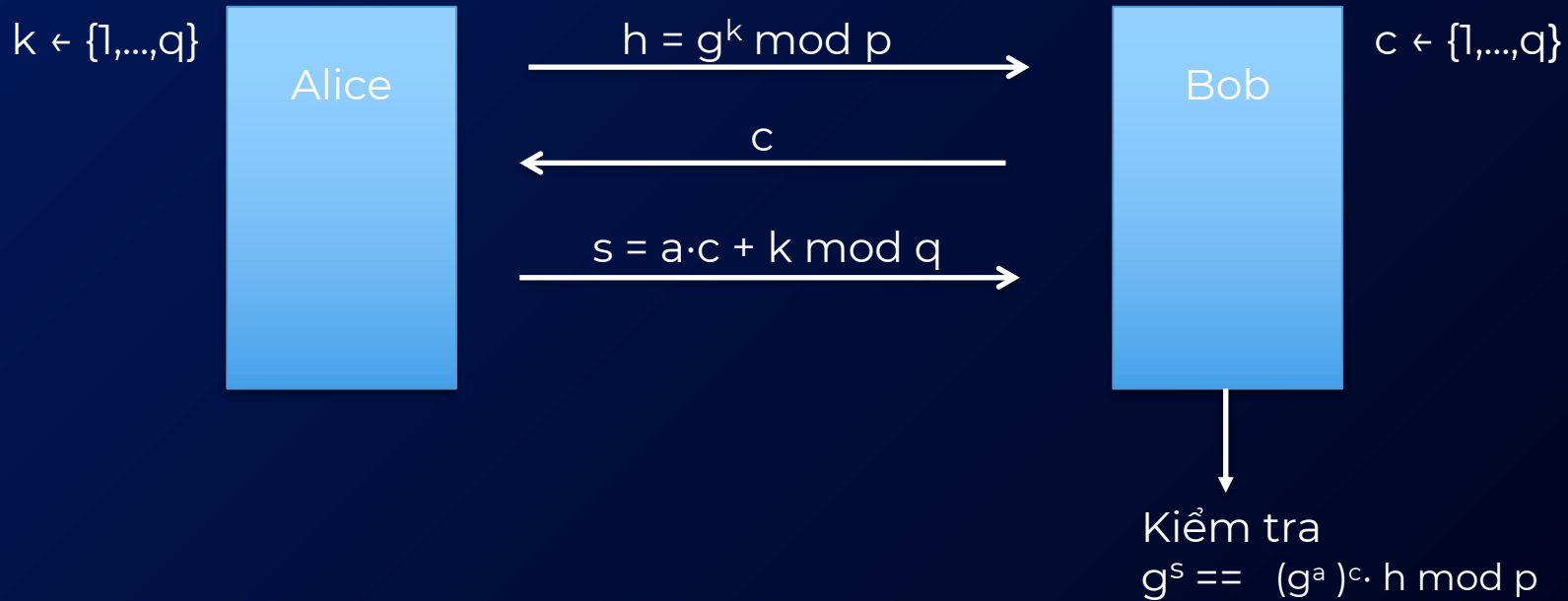
Khởi tạo ():

- Chọn p là số nguyên tố lớn, và lấy g là một phần tử sinh của nhóm vòng cấp q .

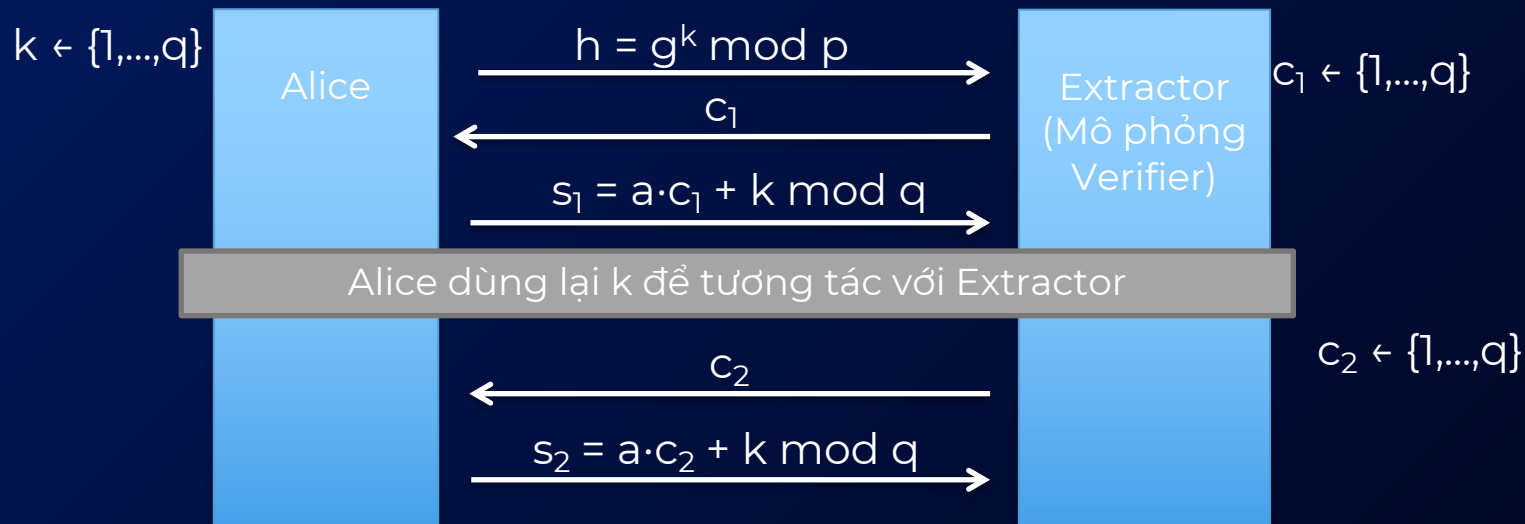
Sinh khóa ():

- Chọn ngẫu nhiên $a \leftarrow \{1, \dots, q\}$
- $pk_A = g^a \bmod p$ và $sk_A = a$

Giao thức định danh Schnorr



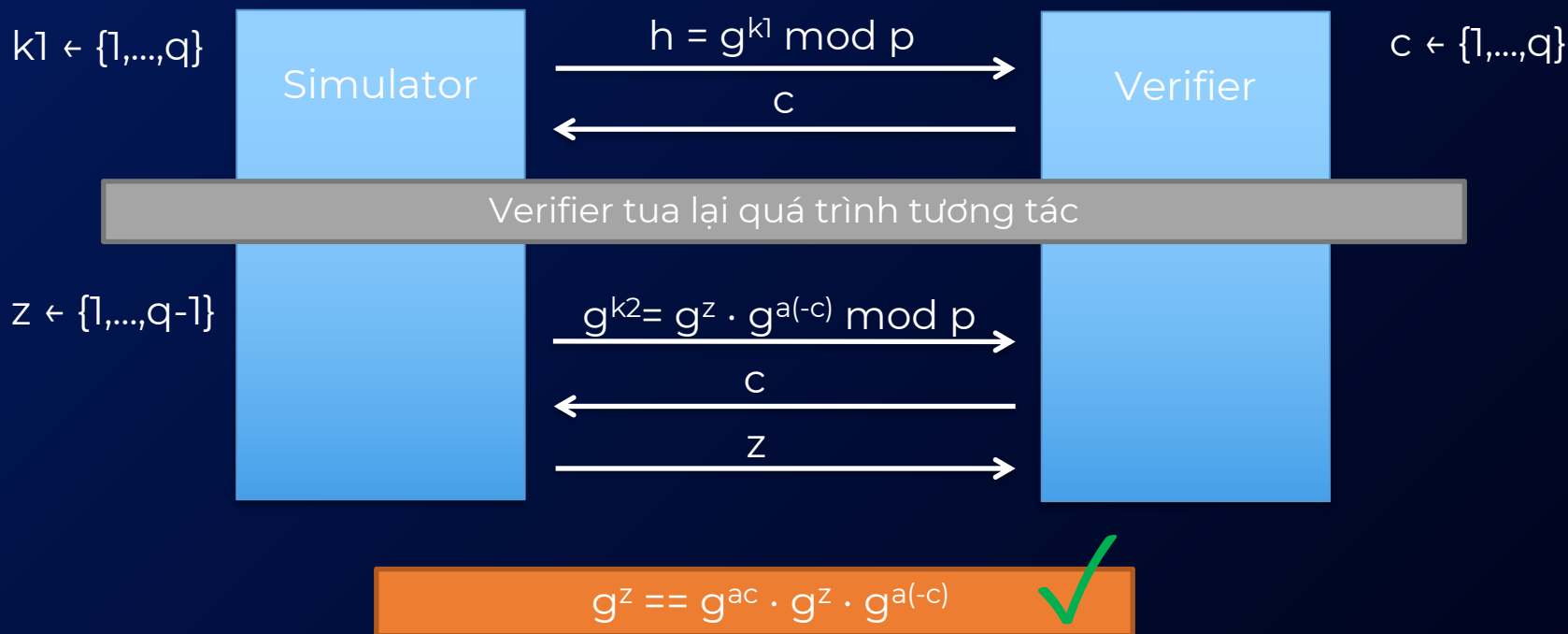
Liệu Alice có biết khóa bí mật?



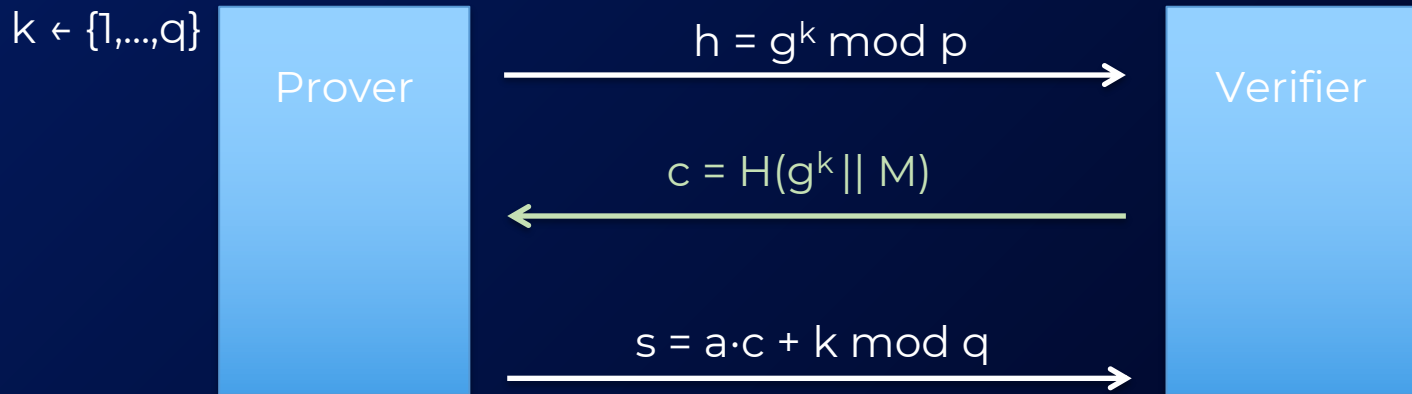
$$\begin{aligned} & (s_1 - s_2) / (c_1 - c_2) \bmod q \\ &= ((a c_1 + k) - (a c_2 + k)) / (c_1 - c_2) \bmod q \\ &= a(c_1 - c_2) / (c_1 - c_2) \\ &= a \end{aligned}$$

Không để lộ tri thức

Verifier có thể tự tạo ra dữ liệu tương tác mà không cần Prover!



Chữ ký Schnorr: Tương tác → Không tương tác



Hàm băm $H()$ với mã băm giống như ngẫu nhiên.

- M là thông điệp cần ký và s là chữ ký của M
- Chỉ người có khóa bí mật mới tạo ra được chữ ký

Nội dung trình bày

ZKP là gì?

Chứng minh tương tác

Chứng minh không lộ
tri thức

Tính chất cần thiết

Ví dụ

Tô màu đồ thị

Giao thức định danh
Schnorr

ZK-SNARK

Chứng minh không
tương tác

SNARK là gì?

Các kiểu SNARK

Công nghệ ZK-SNARK

Chứng minh tương tác hay không tương tác?

Tương tác



Chấp nhận / Bác bỏ

Không tương tác



Chứng minh π



Chấp nhận / Bác bỏ

zk-SNARK là gì?

SNARK (Succinct Non-Interactive ARgument of Knowledge):
không cần tương tác cùng với chứng minh cực ngắn

Ví dụ: “tôi biết m thỏa mãn $\text{SHA256}(m) = 0$ ”



- **SNARK:** chứng minh π phải “cực ngắn” và cho phép kiểm tra “cực nhanh”

Nếu m kích thước 1GB thì không thể dùng m làm chứng minh
vì m không ngắn và tính $\text{SHA256}(m)$ không nhanh

- **zk-SNARK:** chứng minh phải không để lộ tri thức về m .

Ứng dụng zk-SNARK trong Blockchain

- Tornado cash, Zcash
- Zk-rollup, zk-vn
- Zk-Bridge

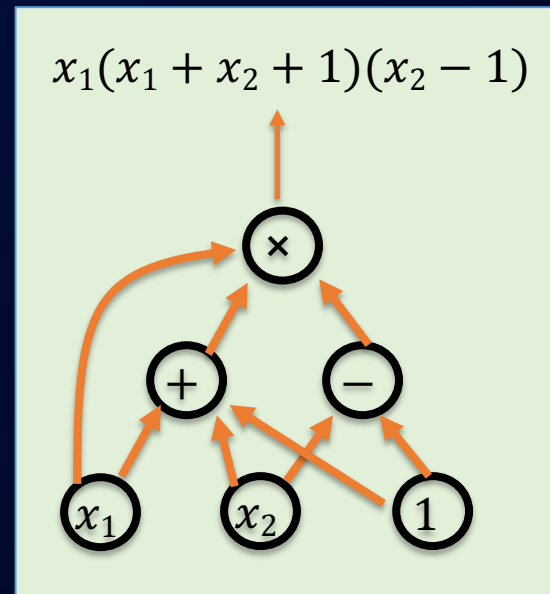
Mạch số học

Trường hữu hạn $\mathbb{F} = \{0, \dots, p-1\}$ với $p > 2$ là số nguyên tố.

Mạch số học: $C: \mathbb{F}^n \rightarrow \mathbb{F}$

- Đồ thị phi chu trình (DAG) với các nút trong được gán nhãn $+$, $-$, hoặc \times
các đầu vào được gán nhãn $1, x_1, \dots, x_n$
- Nó xác định một đa thức n biến

$|C| = \#$ số lượng cổng của mạch C



Một số mạch hay gặp

- $C_{\text{hash}}(h, m)$: return 0 nếu $\text{SHA256}(m) = h$, và $\neq 0$ nếu ngược lại

$$C_{\text{hash}}(h, m) = (h - \text{SHA256}(m)) , \quad |C_{\text{hash}}| \approx 20\text{K cổng}$$

- $C_{\text{sig}}(\text{pk}, m, \sigma)$: return 0 nếu σ là một chữ ký ECDSA hợp lệ của m ứng với khóa công khai pk

SNARK: Succinct Non-interactive ARgument of Knowledge

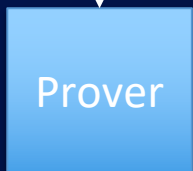
Mạch số học công khai: $C(\mathbf{x}, \mathbf{w}) \rightarrow \mathbb{F}$

Giá trị công khai thuộc \mathbb{F}^n

Giá trị bí mật thuộc \mathbb{F}^m

Preprocessing (setup): $\mathbf{S}(C) \rightarrow$ tham số công khai (pp, vp)

$pp, \mathbf{x}, \mathbf{w}$



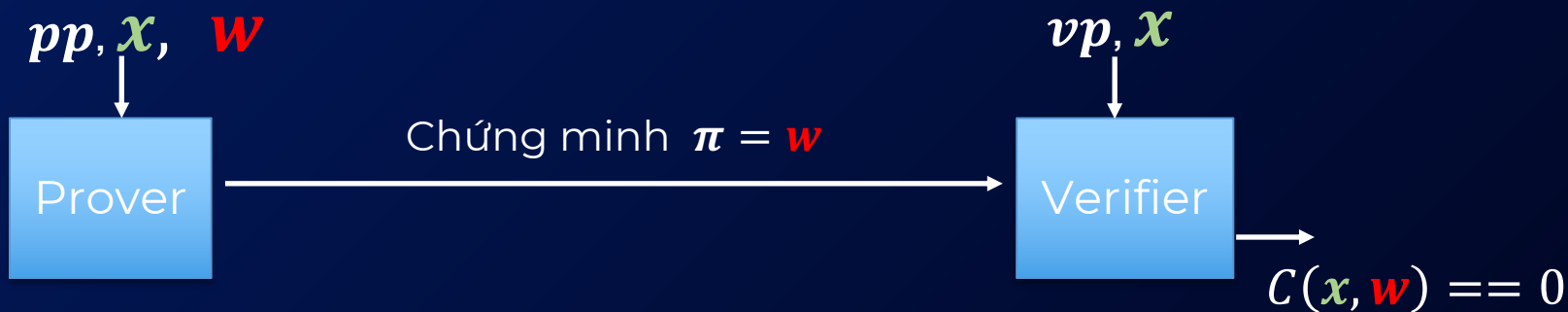
Chứng minh ngắn π cho $C(\mathbf{x}, \mathbf{w}) = 0$

vp, \mathbf{x}



Chấp nhận /
bác bỏ

Hệ chứng minh tầm thường



Vấn đề

- Giữ bí mật w : Prover không muốn tiết lộ w cho Verifier
- w có thể rất dài: mong muốn có một chứng minh "cực ngắn"
- Tính $C(x, w)$ có thể rất lâu: mong muốn kiểm tra "cực nhanh"

SNARK (Succinct Non-Interactive ARgument of Knowledge)

Một *preprocessing* SNARK là bộ ba thuật toán (S, P, V) :

- $S(C) \rightarrow$ các tham số công khai (pp, vp) cho prover và verifier
- $P(pp, x, w) \rightarrow$ chứng minh “cực ngắn” π
- $V(vp, x, \pi)$ hàm kiểm tra chạy “cực nhanh”

SNARK: (S, P, V) phải đầy đủ, đúng đắn, và chứng minh phải cực ngắn

zk-SNARK: (S, P, V) là một SNARK và không để lộ tri thức

Các kiểu tiền xử lý (preprocessing)

Setup cho mạch C : $\mathbf{S}(C; r) \rightarrow$ tham số công khai (pp, vp)
Các bit ngẫu nhiên

Kiểu setup:

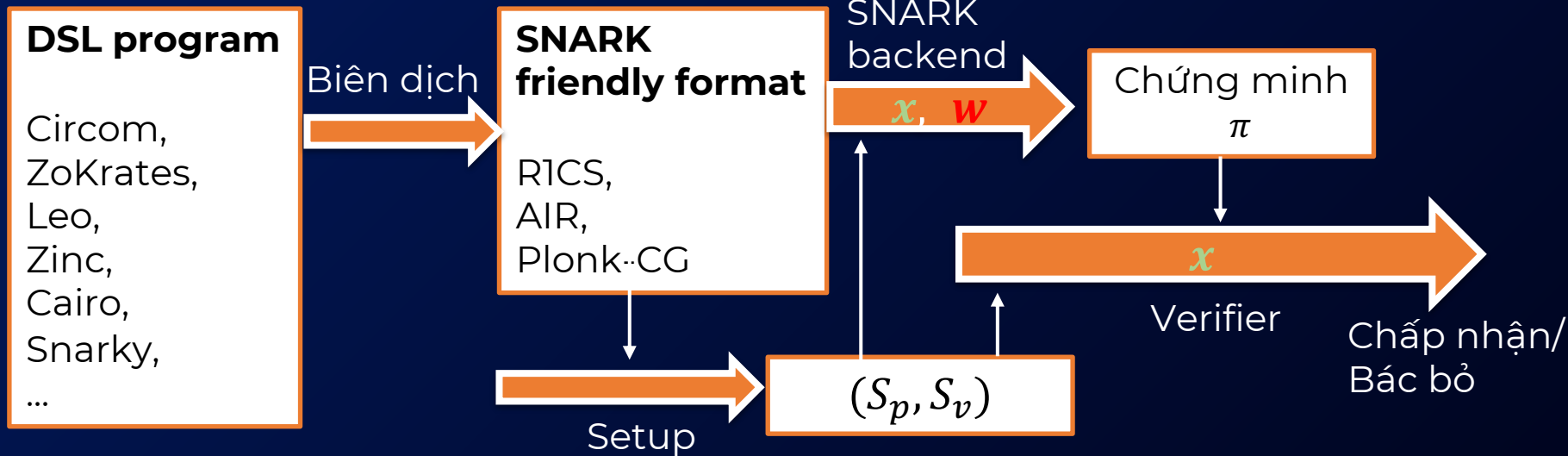
- **trusted setup per circuit:** $\mathbf{S}(C; r)$ số ngẫu nhiên r phải bí mật với Prover
Prover biết $r \Rightarrow$ có thể chứng minh cả khẳng định sai
- **trusted but universal (updatable) setup:** giá trị bí mật r độc lập với mạch C
 $\mathbf{S} = (S_{init}, S_{index}):$ $\underbrace{S_{init}(\lambda; r) \rightarrow gp,}_{\text{chỉ một lần}} \underbrace{S_{index}(gp, C) \rightarrow (pp, vp)}_{\text{Không bí mật với Prover}}$
- **transparent setup:** $\mathbf{S}(C)$ không dùng giá trị ngẫu nhiên (no trusted setup)

So sánh một số hệ SNARK gần đây

	Kích thước chứng minh π	Thời gian kiểm tra	Trusted Setup?
Groth'16	≈ 200 Bytes $O(1)$	≈ 1.5 ms $O(1)$	Có/cho mỗi mạch
Plonk / Marlin	≈ 400 Bytes $O(1)$	≈ 3 ms $O(1)$	Có/universal
Bulletproofs	≈ 1.5 KB $O(\log C)$	≈ 3 sec $O(C)$	Không
STARK	≈ 100 KB $O(\log^2 C)$	≈ 10 ms $O(\log C)$	Không

(cho mạch với 2^{20} cổng)

Công nghệ SNARK



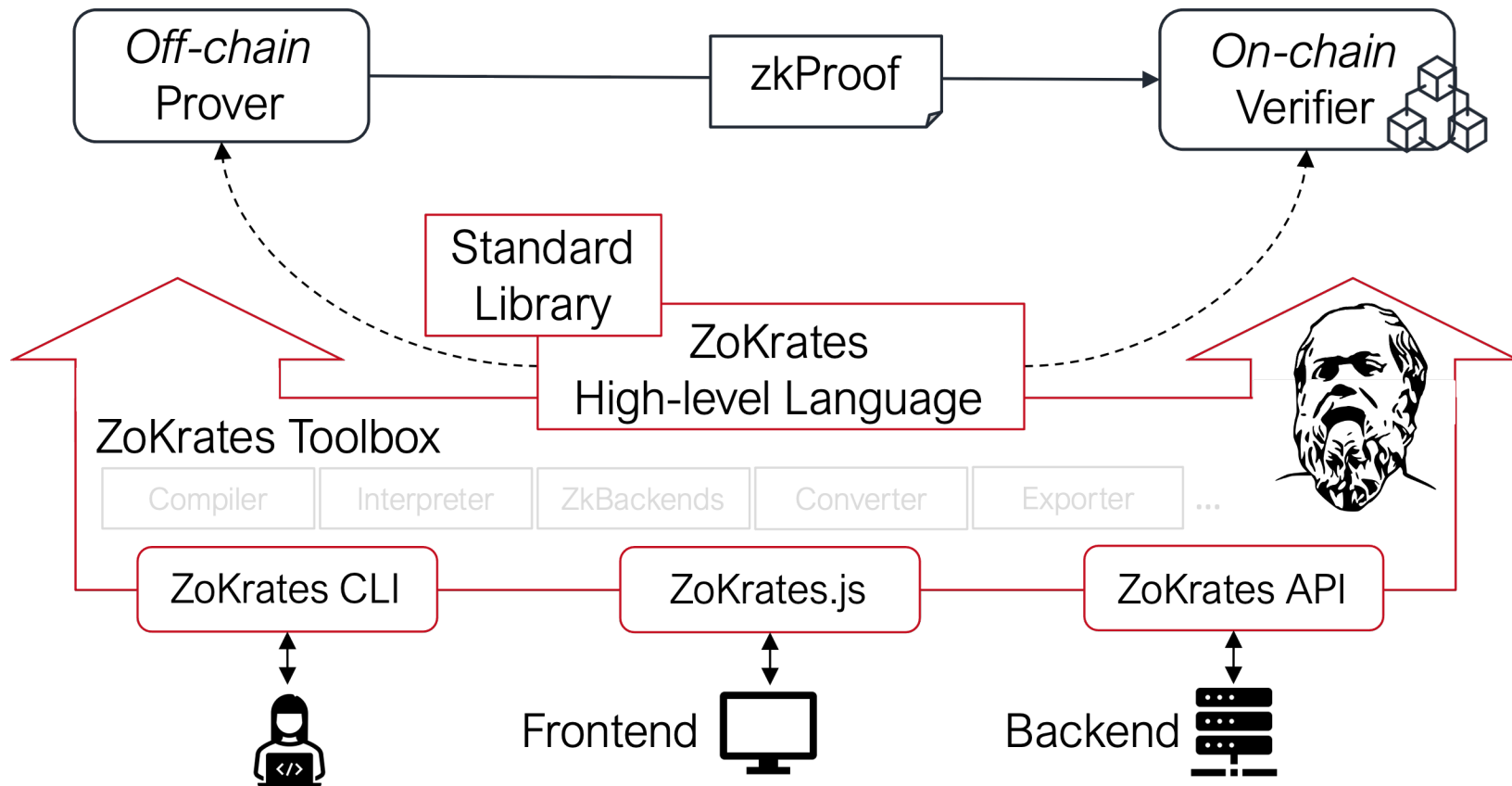
Ví dụ: Sử dụng Zokrates

Chứng minh không để lộ tri thức cho bài toán sau:

- Cho giá trị công khai x , Prover biết $w \in F_p$ thỏa mãn $\text{SHA256}(w)=x$
- F_p là trường nguyên tố 256-bit

Sẽ được biên dịch thành
mạch số học trên F_p

```
def main(field x[2], private field w) -> (field):  
    h = sha256packed( w )  
    h[0] == x[0] // check top 128 bits  
    h[1] == x[1] // check bottom 128 bits  
    return 1
```



<https://github.com/ZoKratesPlus>

Các bước chạy Zokrates

```
# compile
zokrates compile -i root.zok
# perform the setup phase
zokrates setup
# execute the program
zokrates compute-witness -a 337 113569
# generate a proof of computation
zokrates generate-proof
# export a solidity verifier
zokrates export-verifier
# or verify natively
# where input = (proof.json, verification.key)
zokrates verify
```

```
def main(private field a, field b):
    assert(a * a == b)
    return
```

XIN CẢM ƠN!

Tài liệu tham khảo

- Zero Knowledge Proofs: <https://zk-learning.org>
- Decentralized Finance: <https://defi-learning.org>

Câu hỏi

Tính chất “Nếu khẳng định S là đúng, thì Verifier có thể thuyết phục Prover rằng S đúng” là

- ✓ Tính đầy đủ
- ❑ Tính đúng đắn
- ❑ Tính không lộ tri thức

Câu hỏi

Tính chất “Mọi Verifier đều không rút ra thông tin gì ngoài khẳng định vừa được chứng minh là đúng” là

- ❑ Tính đầy đủ
- ❑ Tính đúng đắn
- ✓ Tính không lộ tri thức

Câu hỏi

Trong một hệ chứng minh SNARK *những* tính chất nào phải được đảm bảo:

- ✓ Tính đầy đủ
- ✓ Tính đúng đắn
- Tính không lộ tri thức
- ✓ Chứng minh phải “cực ngắn”
- ✓ Thời gian kiểm tra chứng minh phải “cực nhanh”
- Thời gian tạo ra chứng minh phải “cực nhanh”
- ✓ Quá trình chứng minh phải không tương tác