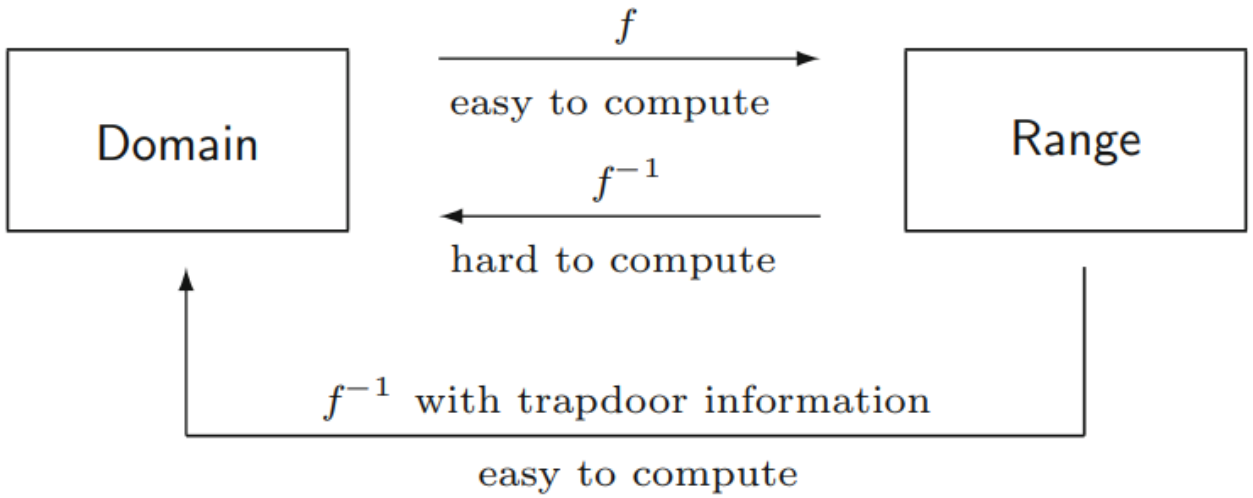
BÀI TOÁN LOG RỜI RẠC TRÊN HỆ MÃ HÓA CÔNG KHAI

18120019 – Nguyễn Hoàng Dũng

18120052 – Lê Hạnh Linh

18120134 – Nguyễn Hồ Thăng Long

Cơ sở của mã hóa bất đối xứng



Lấy các bài toán khó trong số học để làm cơ sở

- **Bài Toán Integer Factorization**
- **Bài Toán Discrete Logarithm**

VD: $1337^x = 27 \pmod{13729}$

Log rời rạc (Discrete Logarithm)

$Z^*_{19} = \{1,2,\dots,18\}$ có cơ số $\boldsymbol{g}$

BT1: xét $\boldsymbol{g} = 2$

$2^x \equiv 5 \pmod{19}$

x	1	2	3	4	5	6	7	8	9
2^x	2	4	8	16	13	7	14	9	18

10	11	12	13	14	15	16	17	18
17	15	11	3	6	12	5	10	1

BT2: xét $\boldsymbol{g} = 8$

$8^x \equiv 5 \pmod{19}$

x?

x	1	2	3	4	5	6
8^x	8	7	18	11	12	1

Log rời rạc (Discrete Logarithm)

$$Z^*_{19} = \{1, 2, \ldots, 18\}$$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1

x	1	2	3	4	5	6
8^x	8	7	18	11	12	1

- **Định nghĩa:**

  Bậc của $\boldsymbol{g}$ là con số $\boldsymbol{x}$ nhỏ nhất để $\boldsymbol{g^x \equiv 1 (\mathbf{mod}\ N)}$

- **Kí hiệu:** $\mathrm{ord}(g) = x$
- **Ví dụ:** Trên $\mathbb{Z}^*_{19}$: ord(2) = 18

  ord(8) = 6

Log rời rạc (Discrete Logarithm)

$$Z^*_{19} = \{1, 2, \ldots, 18\}$$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8^x	8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1

- Định lý Fermat nhỏ:

$$g^{N-1} \equiv 1 (\mathrm{mod}\, N)$$

- Nếu $\mathrm{ord}(g) = N - 1$ thì g được gọi là **Generator**
- Chọn $g = 2$ thì không gian tìm kiếm là lớn nhất

Log rời rạc (Discrete Logarithm)

$$Z^*_{19} = \{1, 2, \ldots, 18\}$$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
8^x	8	7	18	11	12	1	8	7	18	11	12	1	8	7	18	11	12	1

- **Tính chất:**

$$\text{ord}(g) \mid \text{N} - 1$$

- Kiểm tra các lũy thừa của g với ước của N – 1
- VD $g = 2$ xét $2^2, 2^3, 2^6, 2^9$

Tại sao log rời rạc khó?

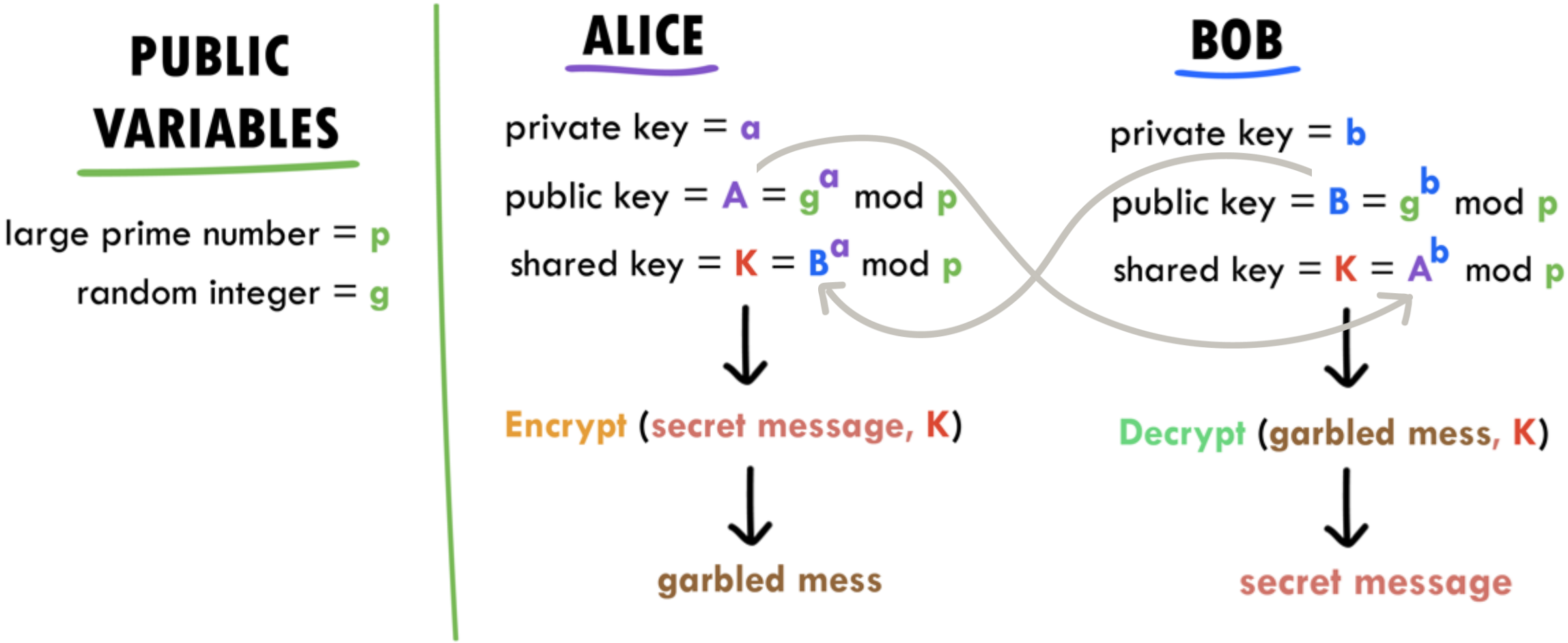
$$g^x = y \ (mod\ 170141183460469231731687303715884105727)$$

- **N là số lớn khoảng 1024 bits (309 chữ số)**
- **Không có quy luật**
- **Thuật toán Baby – Step Giant – Step** $\mathbf{O(\sqrt{N})}$
- **Thuật toán Pohlig – Hellman** $\mathbf{O(\sum_i e_i(logN + \sqrt{p_i}))}$

(với $N - 1 = \prod_i {p_i}^{e_i}$, p_i nhỏ)

⇨ Chọn $N = 2q + 1$ (q nguyên tố)

Thuật toán Diffie-Hellman



Bước 1: Tìm p nguyên tố

Algorithm 1 Generate Safe Prime

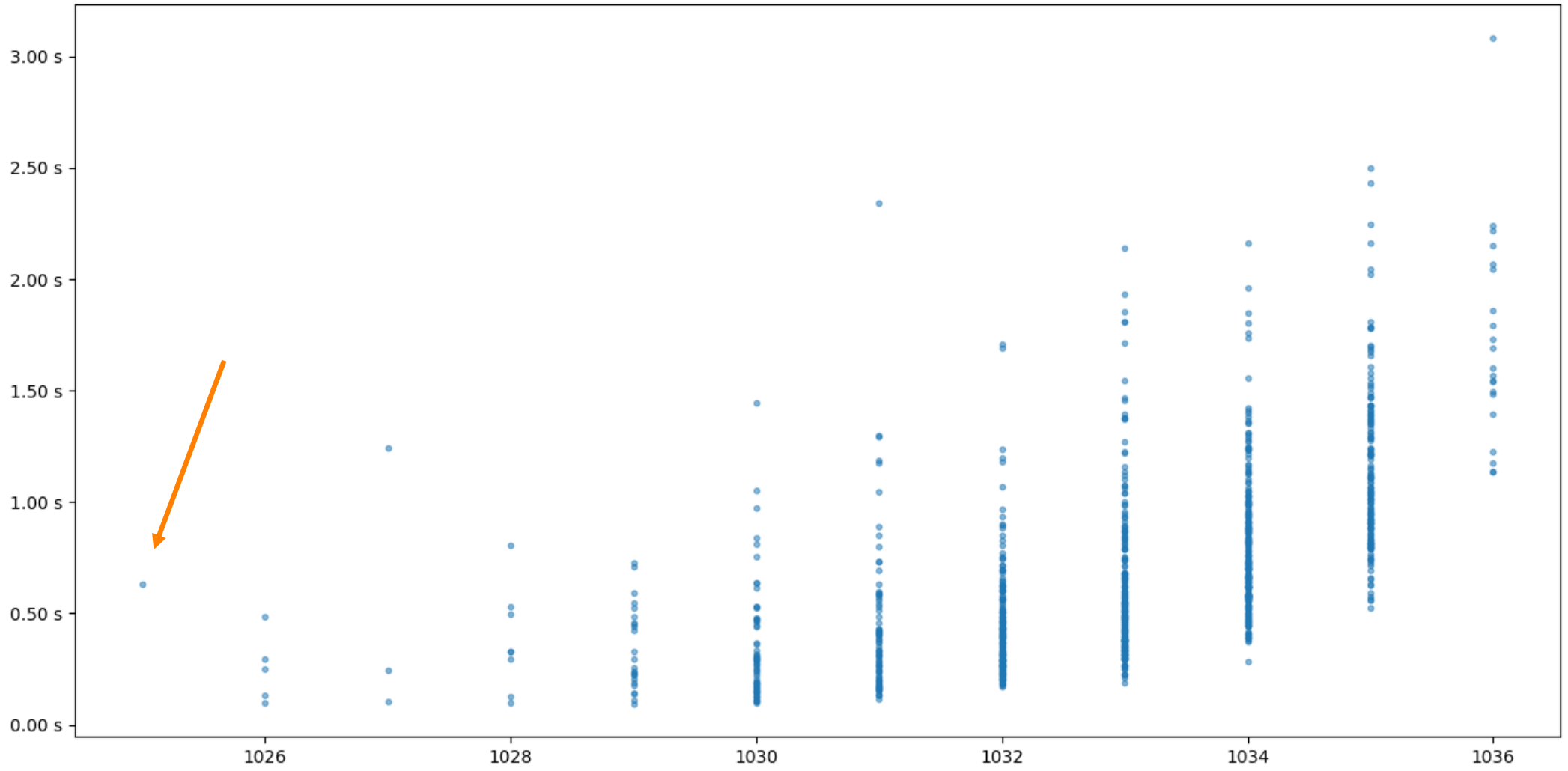
Input: number of bits n
Output: prime p with large factor q

```
1: function GetSafePrime
2:     q ← GetPrime(n)
3:     i ← 1
4:     while True do
5:         p ← 2q * i + 1
6:         i ← i + 1
7:         if IsPrime(p) then break
8: return p, q
```

Nhận Xét:

- Tìm $\boldsymbol{p}$ là số nguyên tố
- Và $\boldsymbol{p-1}$ có factor $\boldsymbol{q}$ đủ lớn
- Tìm $\boldsymbol{q}$ và kiểm tra bội của $\boldsymbol{2q}$
- Số bit của $\boldsymbol{p}$ sẽ nhiều hơn $\boldsymbol{n}$ bit

Sinh p nguyên tố (1024 bit) 1000 lần



Bước 2: Chọn g trên $\mathbb{Z}_p^*$

$$Z^*_{19} = \{1, 2, \ldots, 18\}$$

x	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
2^x	2	4	8	16	13	7	14	9	18	17	15	11	3	6	12	5	10	1

x	1	2	3	4	5	6
8^x	8	7	18	11	12	1

Tìm $\boldsymbol{g}$ có bậc lớn để tăng độ khó cho bài toán log rời rạc

$\Rightarrow$ Chọn $\boldsymbol{g}$ và kiểm tra $\boldsymbol{g^{(p-1)/q} \neq 1 \ (mod\ p)}$

Bước 2: Chọn g trên $\mathbb{Z}_p^*$

VD: Xét $\boldsymbol{p} = 103 \Rightarrow \boldsymbol{p} - 1 = 2 \times 3 \times 17$

$\boldsymbol{p} - 1$ có factor lớn là q = 17

Algorithm 2 Get Generator

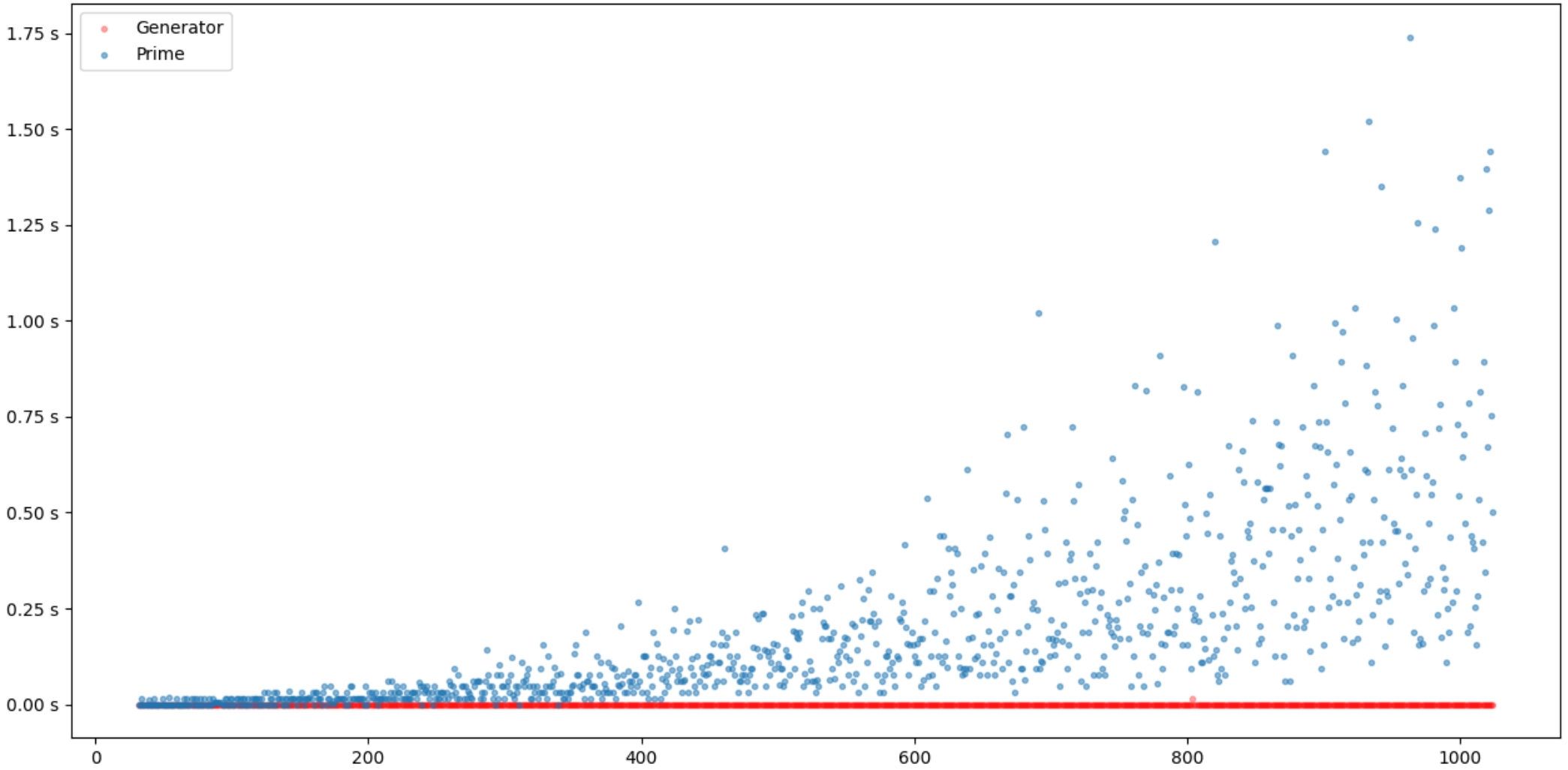
Input: prime p with large factor q
Output: a number g with large order

```
1: function GetGenerator
2:     k ← (p − 1)/q
3:     while True do
4:         g ← RANDOM(2, p − 2)
5:         if g^k ≠ 1 (mod p) then break
6: return g
```

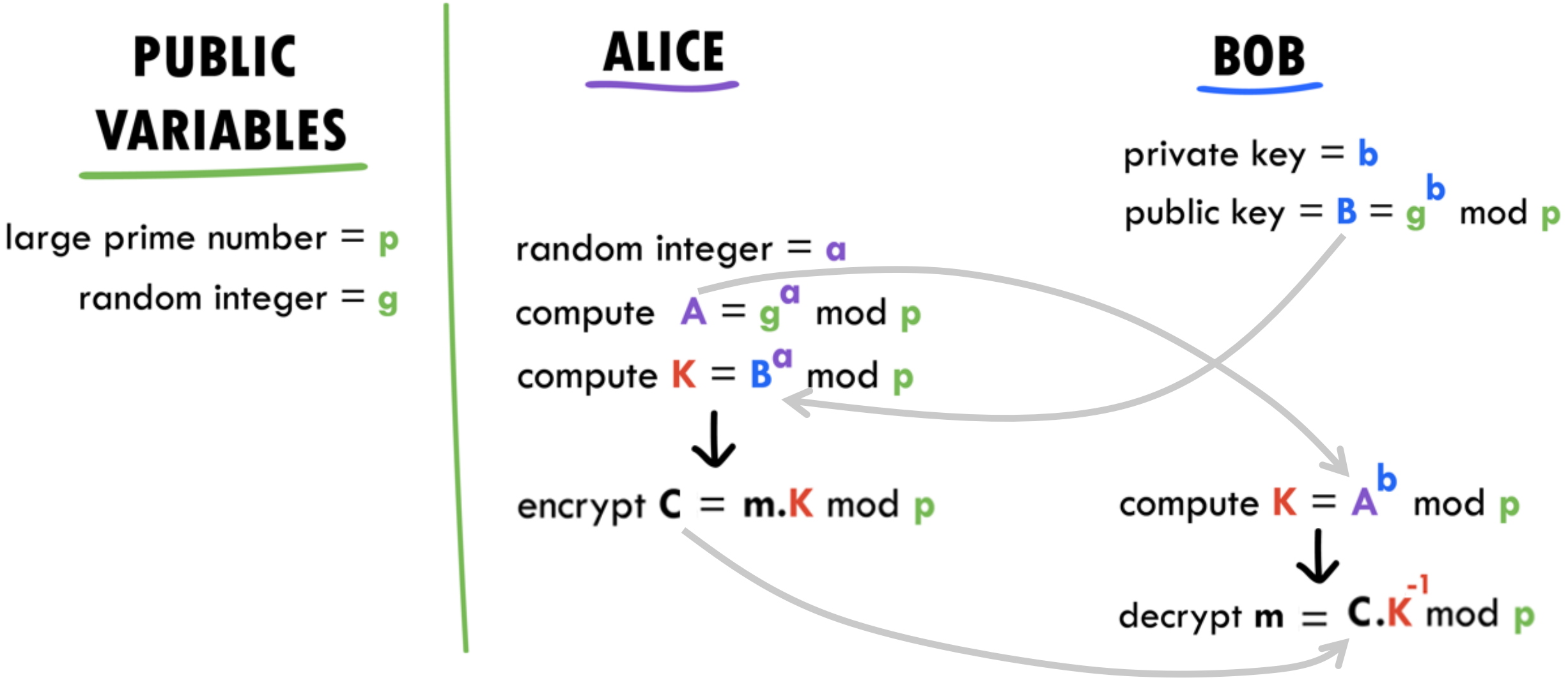
TH: $g^6 = 1 \ (mod\ p)$
$\Rightarrow$ không chọn g

TH: $g^6 \neq 1 \ (mod\ p)$
$\Rightarrow g^2 \neq 1, \ g^3 \neq 1$
$\Rightarrow g^q = 1$ or $g^{2q} = 1$ or $g^{3q} = 1$
or $g^{6q} = g^{p-1} = 1$
$\Rightarrow$ chọn g

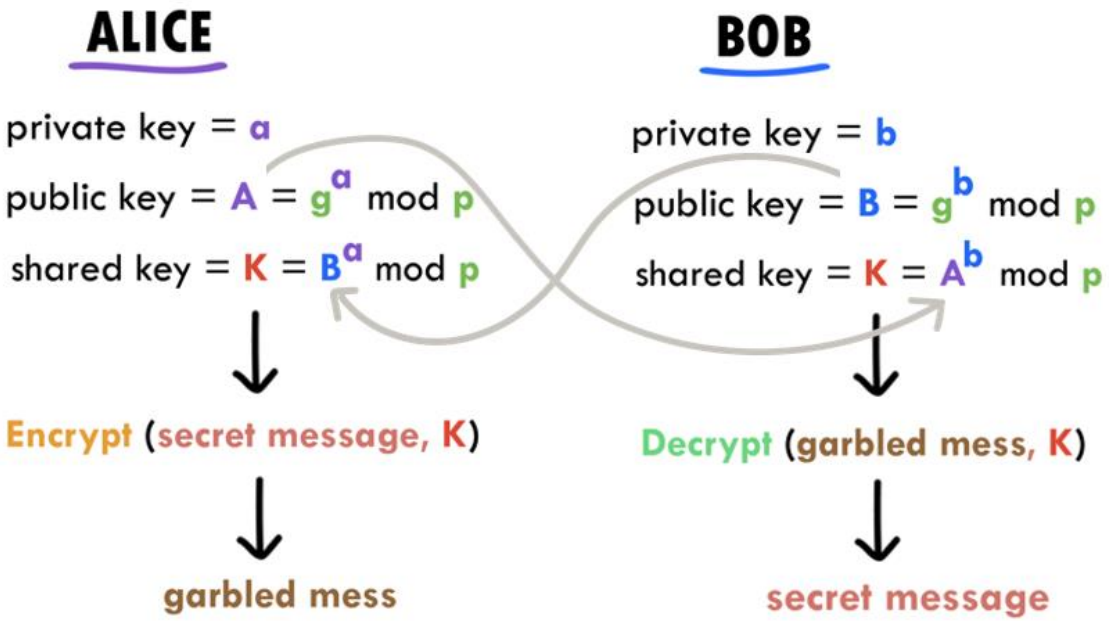
Kết quả chạy thử cả 2 bước



Thuật toán ElGamal



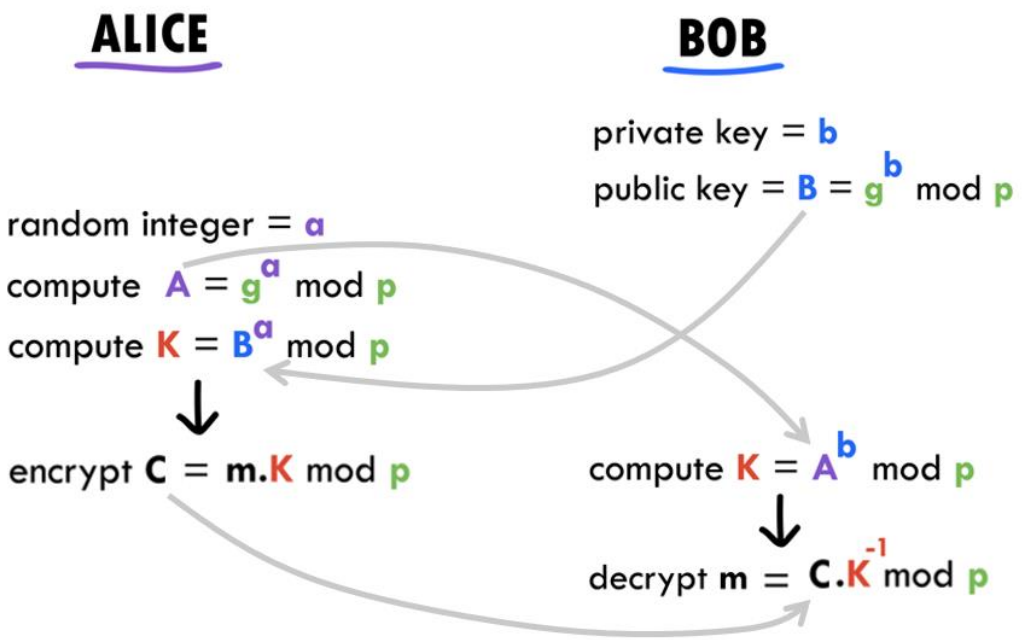
Diffie – Hellman



Interactive

Key Exchange Method

ElGamal



Non-Interactive

ElGamal Digital Signature

Digital Signature Algorithm

Tại sao quan tâm đến log rời rạc ?

Bài toán khó

Mới được nghiên cứu gần đây (50 năm)

Trường hợp nào thì tồn tại bài toán log rời rạc ?

- $\mathbb{Z}_{18}^*, \mathbb{Z}_{20}^*$?
- Phép cộng hoặc 1 phép toán khác ?

$\Rightarrow$ Có tồn tại những cấu trúc đại số khác có bài toán log rời rạc **khó**

$$Z^*_{19} = \{1, 2, \ldots, 18\}$$

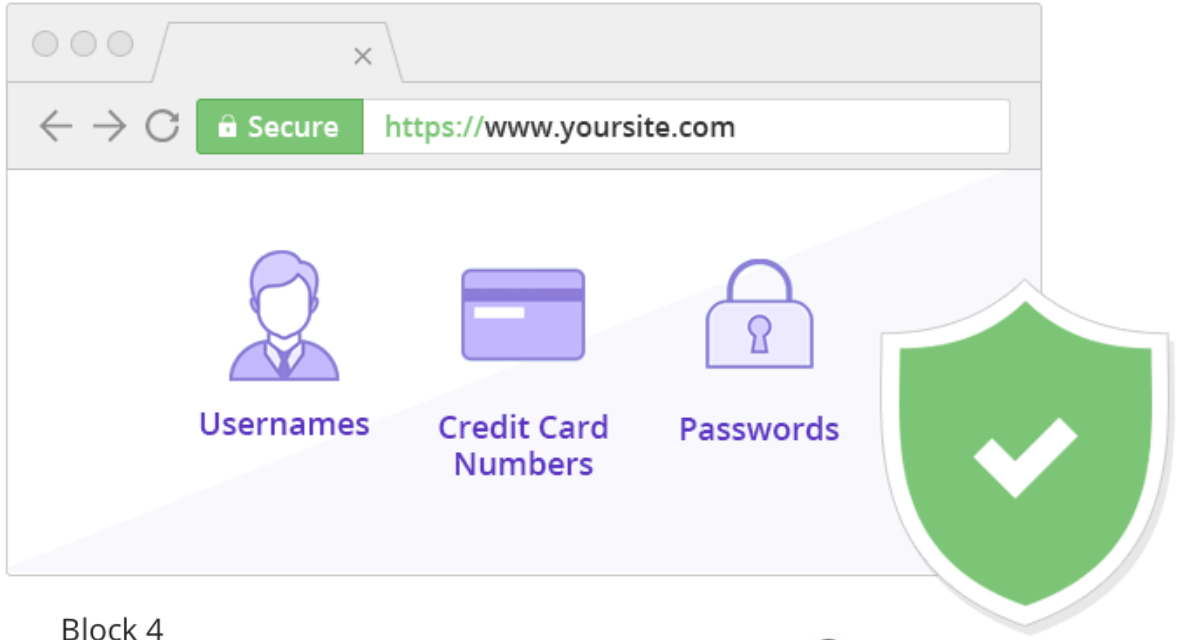
$$\underbrace{g \star g \star g \star \cdots \star g}_{x \text{ times}} = y$$

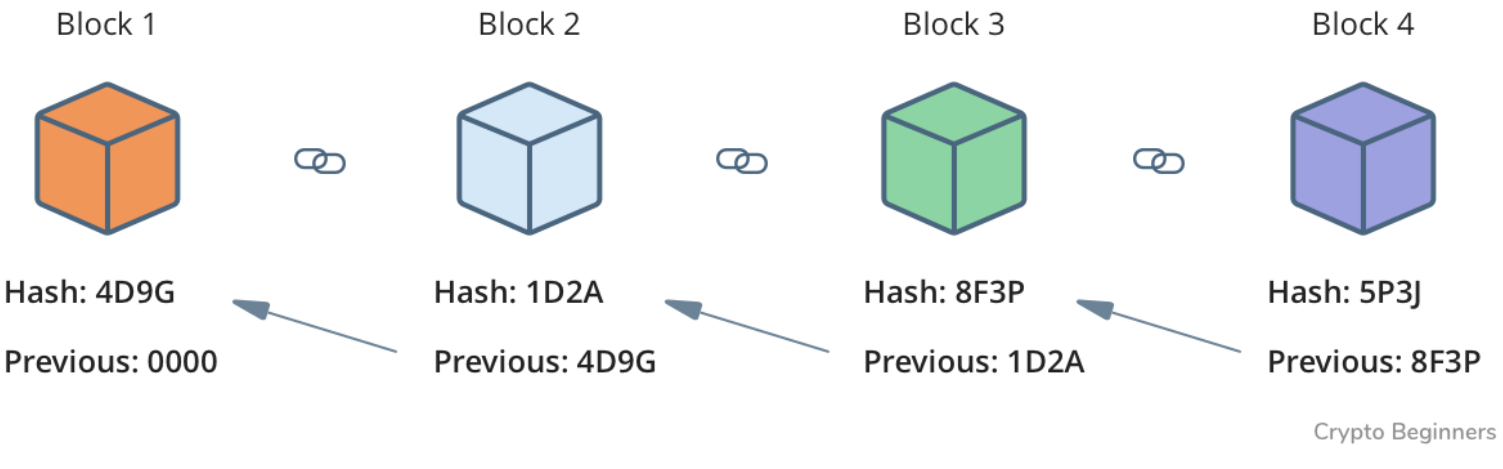
$g^x = y \quad \Rightarrow$ phép nhân

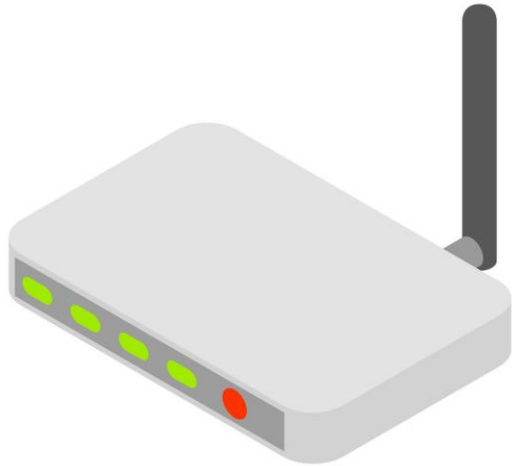
$gx = y \quad \Leftarrow$ phép cộng

$$u * v = \frac{u^2 + v}{u - v^2}$$

Ứng dụng của log rời rạc trên thực tế



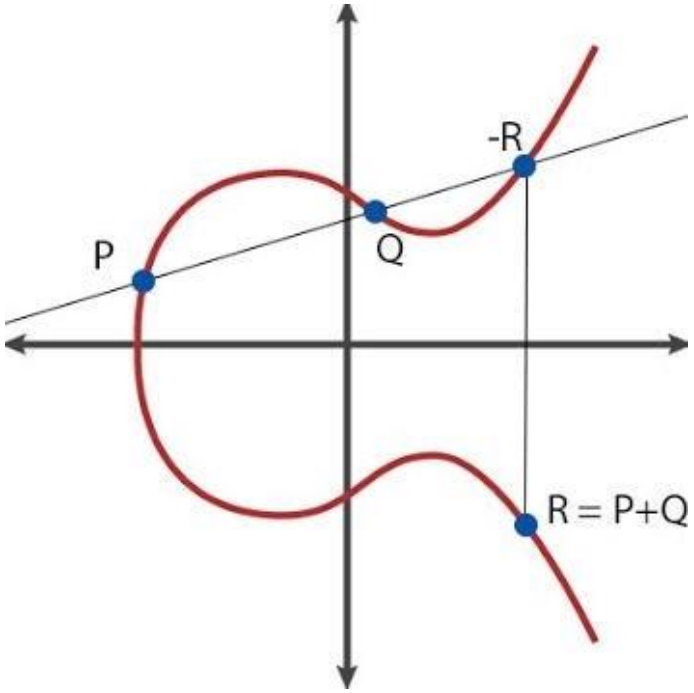




Ứng dụng của log rời rạc trên thực tế

Elliptic Curve Diffie – Hellman Key Exchange
Elliptic Curve Digital Signature Algorithm

Algorithm Family	Cryptosystems	Security Level (bit)			
		80	128	192	256
Integer factorization	RSA	1024 bit	3072 bit	7680 bit	15360 bit
Discrete logarithm	DH, DSA, Elgamal	1024 bit	3072 bit	7680 bit	15360 bit
Elliptic curves	ECDH, ECDSA	160 bit	256 bit	384 bit	512 bit
Symmetric-key	AES, 3DES	80 bit	128 bit	192 bit	256 bit



$E : y^2 = x^3 + Ax + B$

Câu hỏi ?

1. Tìm x, biết $2^x \equiv 3 \pmod 5$

2. Làm sao để tăng độ khó của bài toán log rời rạc ?

3. Để có độ an toàn 128 bit, mã hóa Diffie – Hellman cần làm việc trên bao nhiêu bit ?

4. Các bạn có câu hỏi gì thêm cho nhóm mình không ?

Tài liệu tham khảo

Cryptography and Network Security - William Stallings

An Introduction to Mathematical Cryptography - Jeffrey Hoffstein · Jill Pipher · Joseph H. Silverman

Understanding Cryptography - Christof Paar · Jan Pelzl