ZK SHANGHAI
零知识证明工作坊

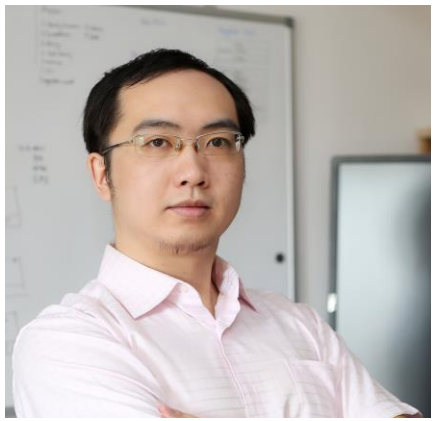
CIRCOM实用电路

现代零知识密码学

Hosted by SutuLabs & Kepler42B-ZK Planet

课程资源：zkshanghai.xyz

个人介绍



梁爽

区块链 架构师

上海交大 计算机博士生
（休学创业中）

微信：icerdesign
微博：@wizicer
Github: @wizicer
Twitter: @icerdesign
LinkedIn: www.linkedin.com/in/icerdesign

1999年	• 正式开始学习写程序
2009年	• 在新媒传信（飞信）做高性能服务器程序架构及开发
2012年	• 在Honeywell工业控制部门做PLC、RTU上位机组态软件架构及开发
2017年	• 接触区块链，并开始创业开发区块链数据库
2020年	• 入学上海交大攻读博士学位，研究零知识证明数据库
2022年	• 获Chia全球开发大赛第一名，并开始Pawket钱包的开发
2023年	• 获得零知识链Mina的项目资助

今日课程内容

- 简单的ZK签名方案
- 简单的群签名方案
- 使用Merkle树支持更大的群
- snarkjs编译流程

简单签名方案

- KeyGen → (sk, pk): 选择一个随机密钥 sk 和对应的公钥 pk
- Sign(m, sk) → s: 给定消息 m 和密钥 sk，输出签名 s
- Verify(m, s, pk) → 1/0: 给定消息 m、签名 s 和公钥 pk，验证签名是否有效

问题：消息 m 未受到约束，

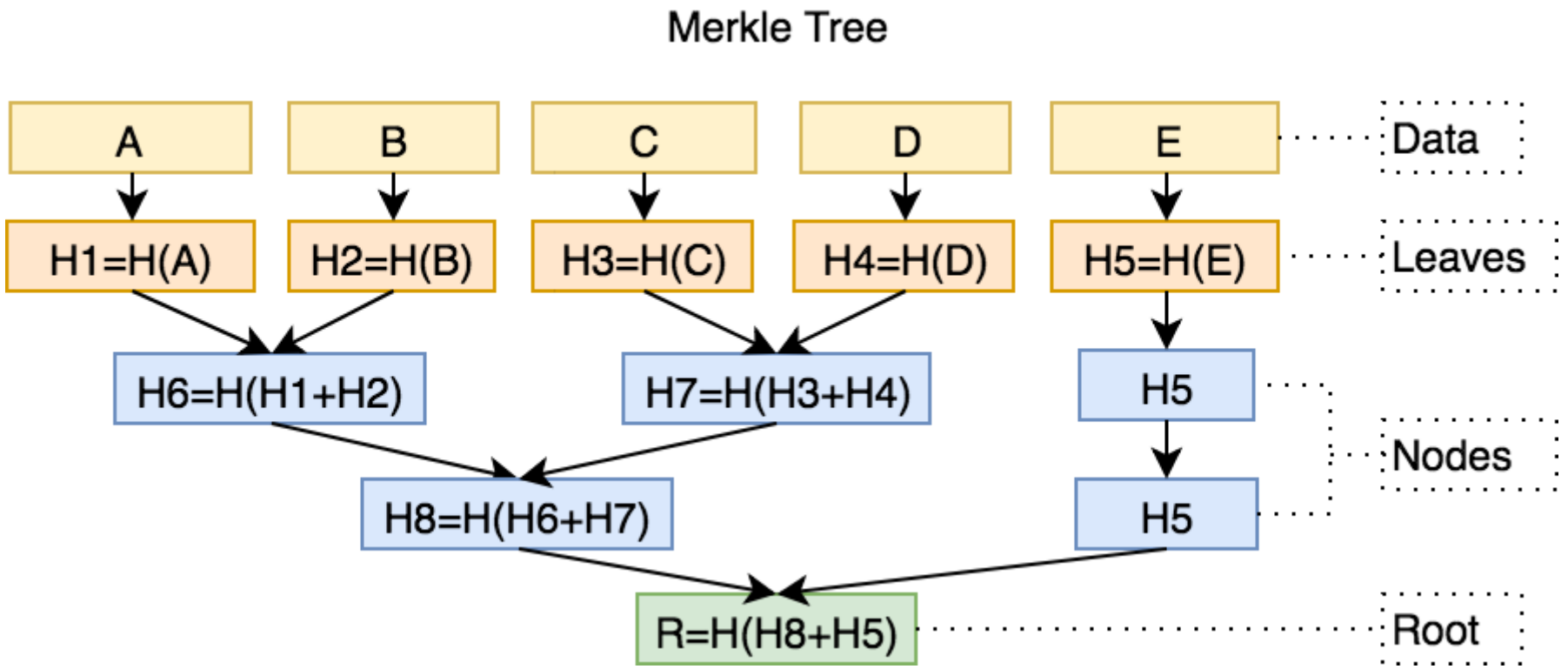
是否影响签名的可靠性?

简单群签名

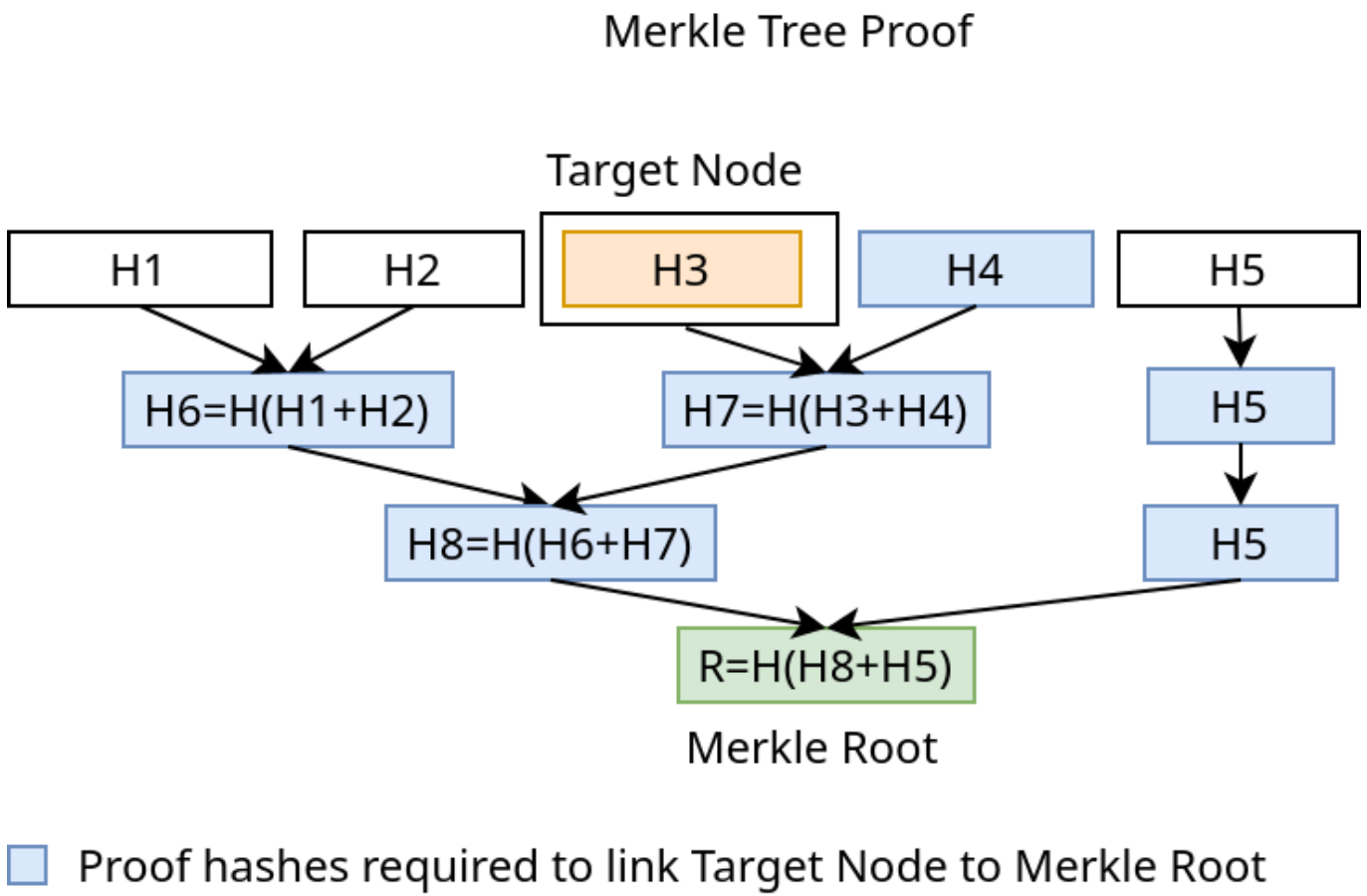
- KeyGen → (ski, pki): 为组中的每个成员选择一组随机的秘密密钥 ski 和相应的公钥 pki
- GroupSign(m, ski, G) → s: 给定消息 m 和密钥，输出组签名 s
- GroupVerify(m, s, G) → 1/0: 给定消息 m、组签名 s 和组 G，验证签名是否来自组

问题：如果群里面的成员

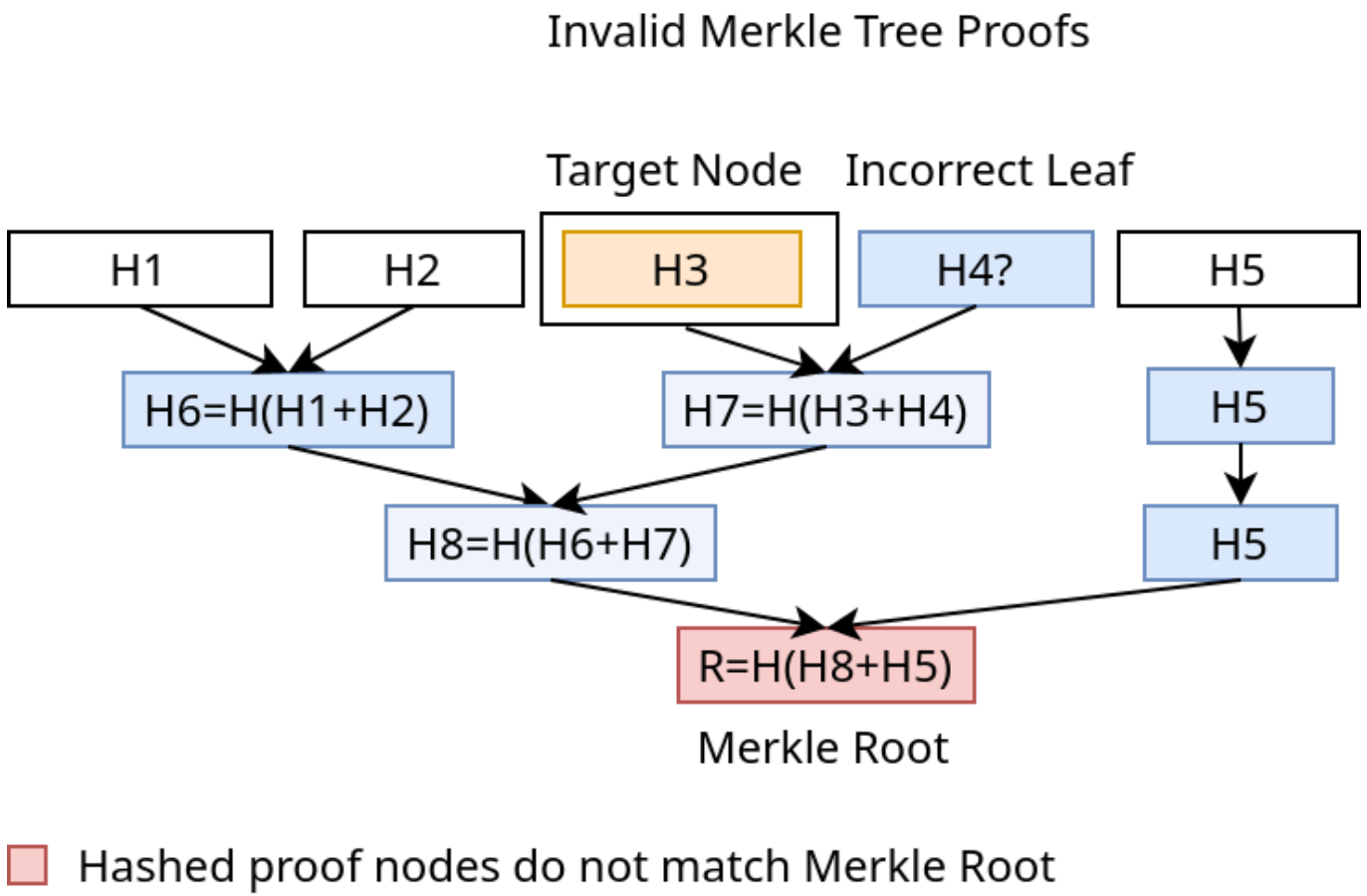
非常多并且数量可变，怎么办？



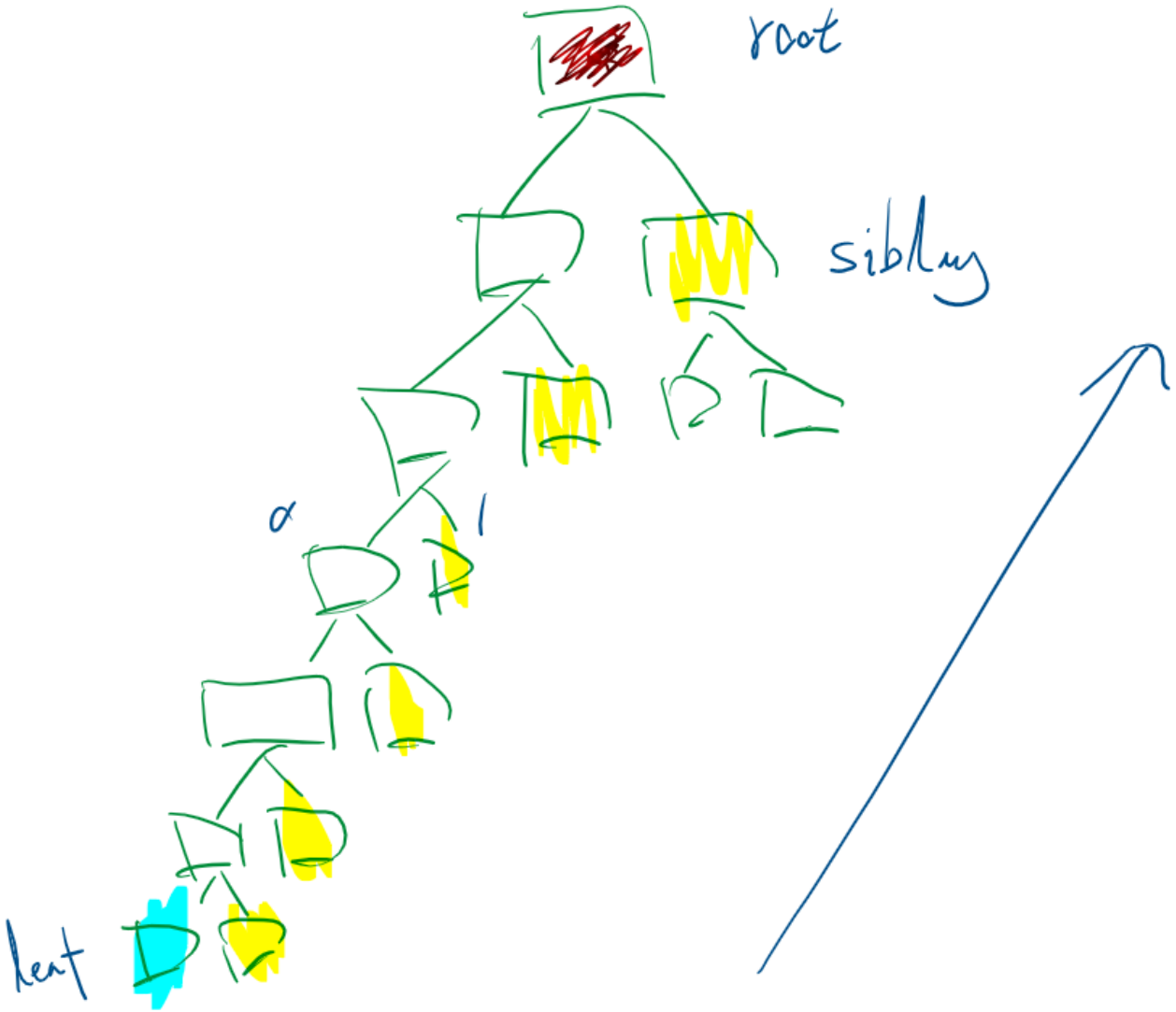
https://github.com/merkletreejs/merkletreejs



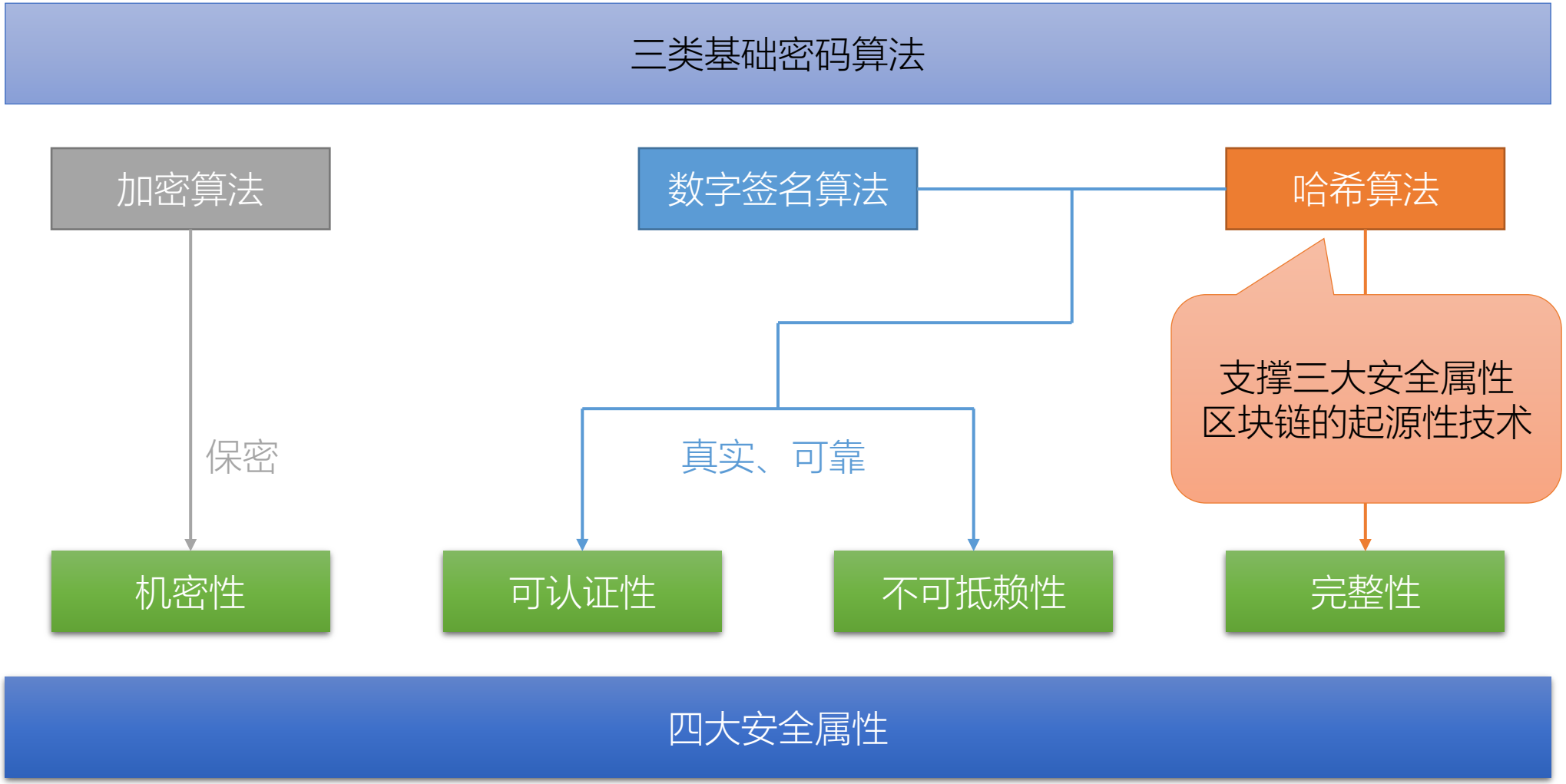
https://github.com/merkletreejs/merkletreejs



https://github.com/merkletreejs/merkletreejs


root
sibling
a
leaf

密码学


三类基础密码算法
加密算法
数字签名算法
哈希算法
保密
真实、可靠
支撑三大安全属性
区块链的起源性技术
机密性
可认证性
不可抵赖性
完整性
四大安全属性

密码学——哈希计算

密码学——哈希计算

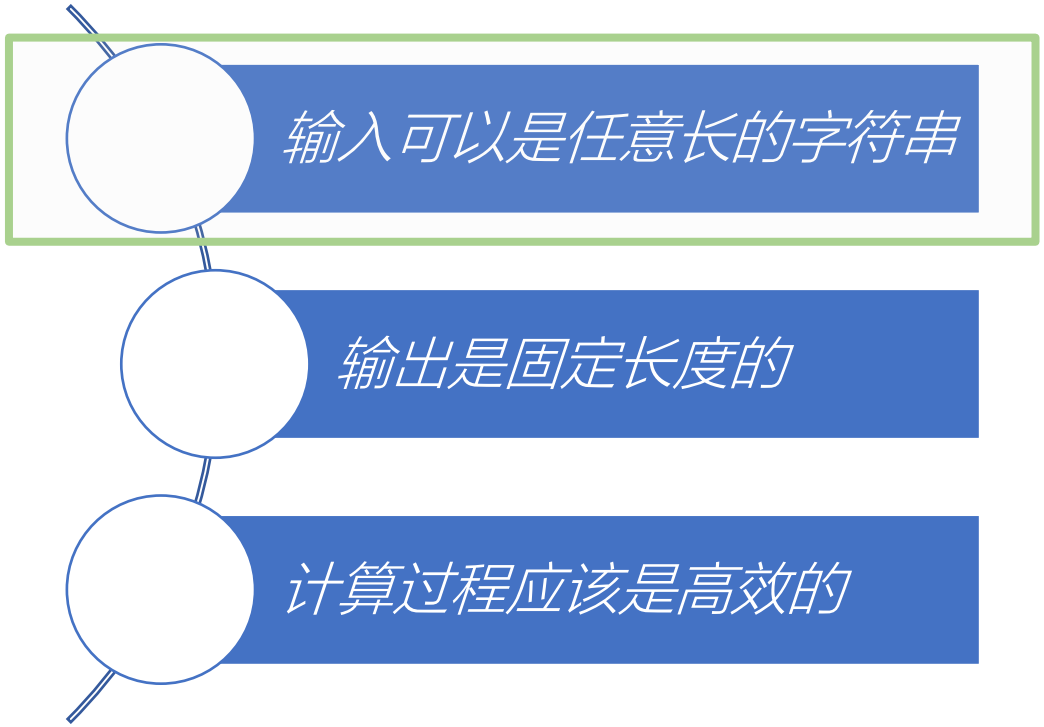
特点

- *输入可以是任意长的字符串*
- *输出是固定长度的*
- *计算过程应该是高效的*

安全特性

- 抗第一原像攻击
- 抗第二原像攻击
- 抗碰撞

哈希计算——特点



哈希计算

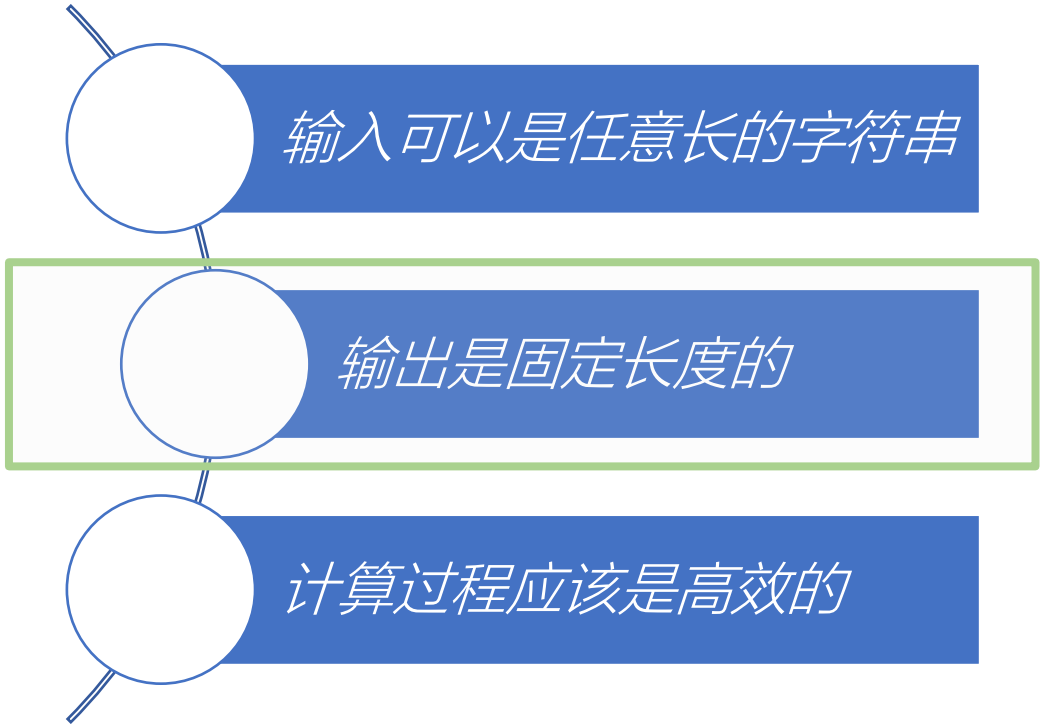
SHA256(“素图科技”)=

输入信息

```
00110101011100000101101001011101
01111001000111010101011011100010
11101011000110110111110001110000
00010010100100110111011011110001
10010010010010110011100110001111
11000101101110100100011010011110
00011001000010111000001001110000
01001001010010110000111100001110
```

哈希值

哈希计算——特点



哈希计算

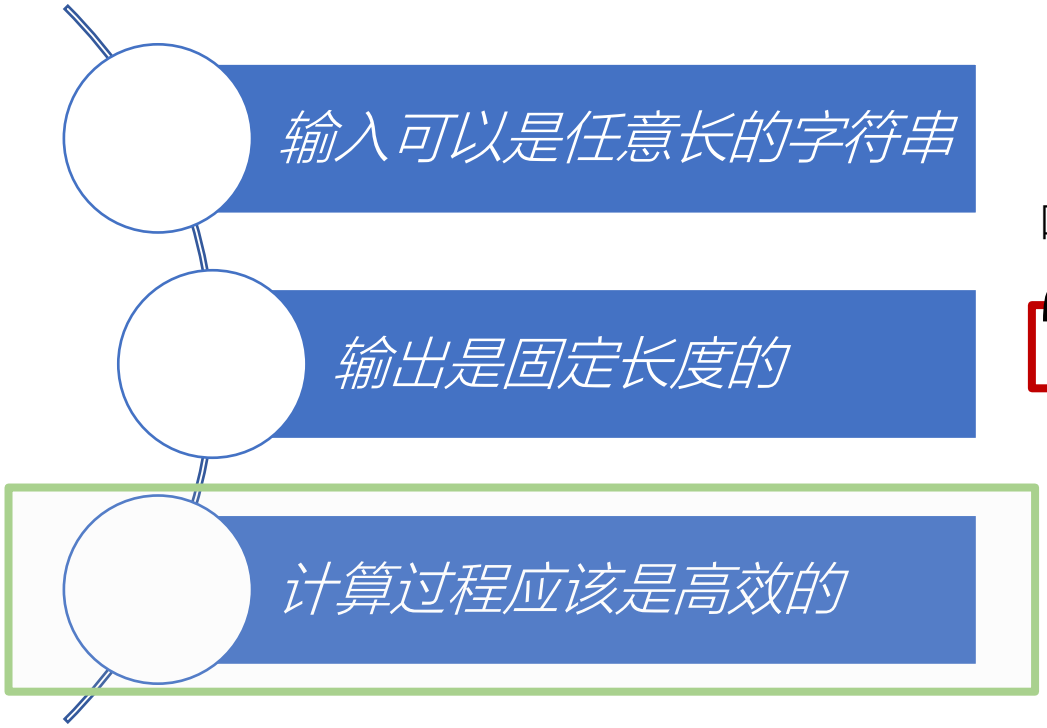
SHA256(“素图科技”)=

输入信息

```
00110101011100000101101001011101
01111001000111010101011011100010
11101011000110110111110001110000
00010010100100110111011011110001
10010010010010110011100110001111
11000101101110100100011010011110
00011001000010111000001001110000
01001001010010110000111100001110
```

哈希值

哈希计算——特点



哈希计算

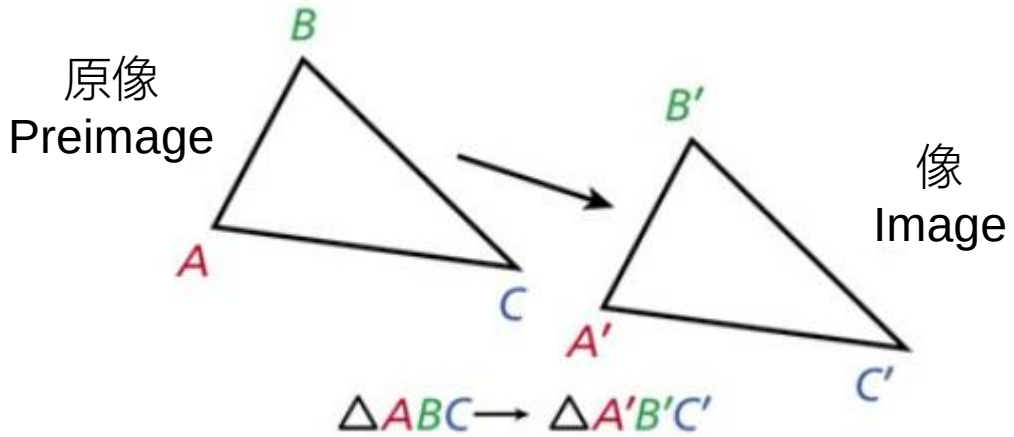
SHA256(“素图科技”)=

输入信息

```
00110101011100000101101001011101
01111001000111010101011011100010
11101011000110110111110001110000
00010010100100110111011011110001
10010010010010110011100110001111
11000101101110100100011010011110
00011001000010111000001001110000
01001001010010110000111100001110
```

哈希值

基本概念



哈希计算

SHA256(“素图科技”)=

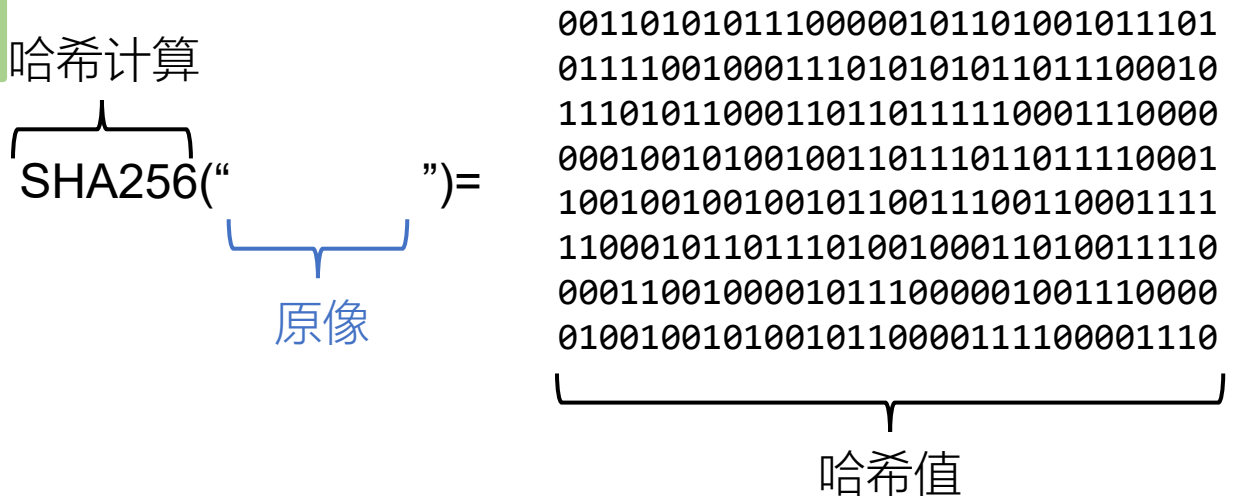
原像

```
00110101011100000101101001011101
01111001000111010101011011100010
11101011000110110111110001110000
00010010100100110111011011110001
10010010010010110011100110001111
11000101101110100100011010011110
00011001000010111000001001110000
01001001010010110000111100001110
```

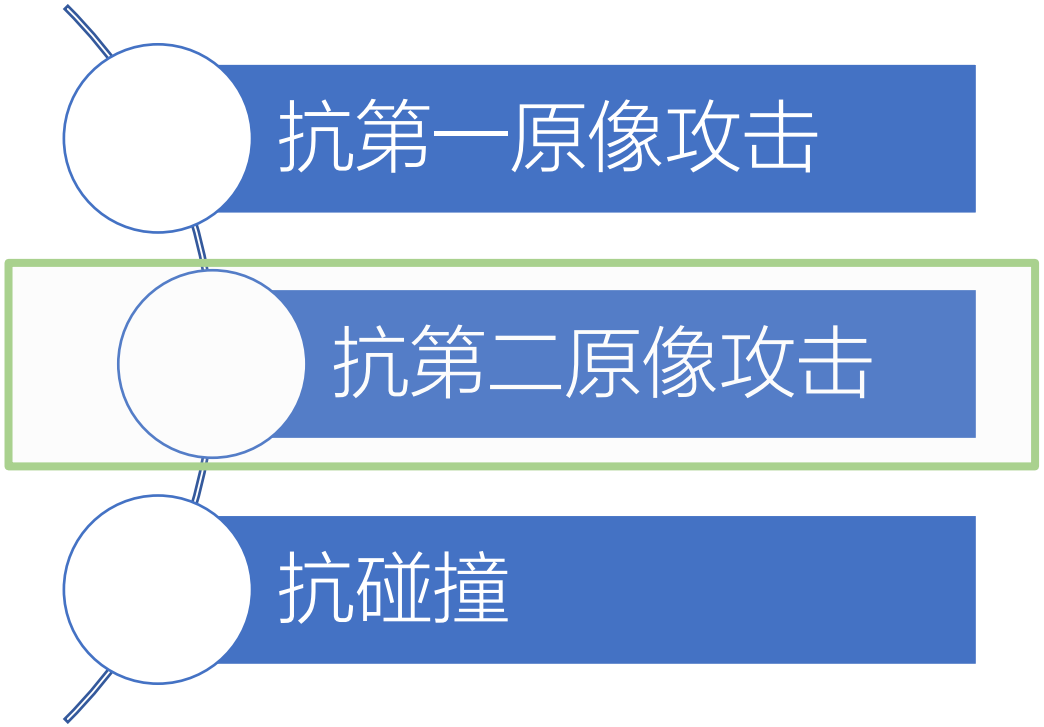
哈希值（像）

哈希计算——安全特性

- 抗第一原像攻击
- 抗第二原像攻击
- 抗碰撞



哈希计算——安全特性



第二原像

SHA256(“ ”)=

≠

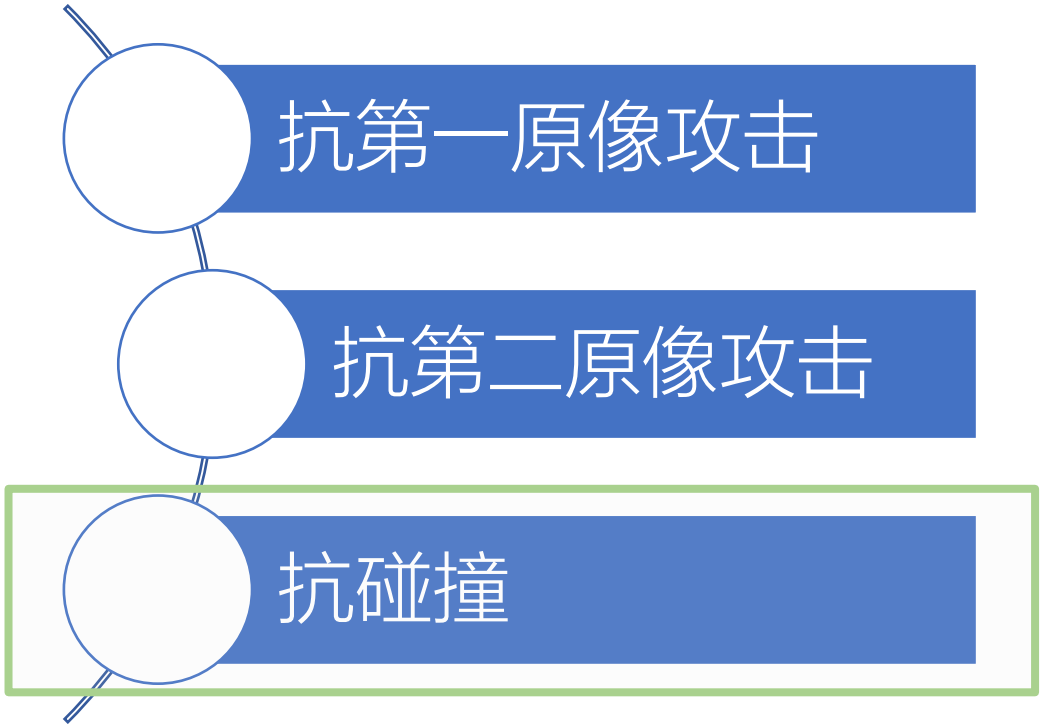
SHA256(“素图科技”)=

原像

```
01001001001100011101001001101011
11011110111111010011001110111110
11011110100101101011010111001010
00100110110100100010111000011011
10101000000111010110110010111110
00110100111010110101101110110110
00011110010001011100110001000100
00011101010010011001000111011100
```

哈希值

哈希计算——安全特性



2^{256}种可能性

```
0100100100110001110100100110101 1
1101111011111101001100111011111 0
1101111010010110101101011100101 0
0010011011010010001011100001101 1
1010100000011101011011001011111 0
0011010011101011010111011011011 0
0001111001000101110011000100010 0
0001110101001001100100011101110 0
```

可能性够多吗?

2^{256}种可能性

00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性
00000000000000000000000000000000	2^{32}种可能性	≈43亿种可能性

256=32x8
2^{32} =4,294,967,296（约43亿）
2^{256} =115,792,089,237,316,195,423,570,985,008,687,907,853,269,984,665,640,564,039,457,584,007,913,129,639,936 (78位)

$$\frac{43亿种可能 \times 43亿种可能 \times 43亿种可能 \times 43亿种可能 \times 43亿种可能 \times 43亿种可能 \times 43亿种可能}{\text{₿}}$$

比特币网络拥有现在全球计算SHA256哈希最强的能力
网络计算能力最高达到123E=123*1024*1024*1024G（G=10亿）
123E≈ 43亿×43亿

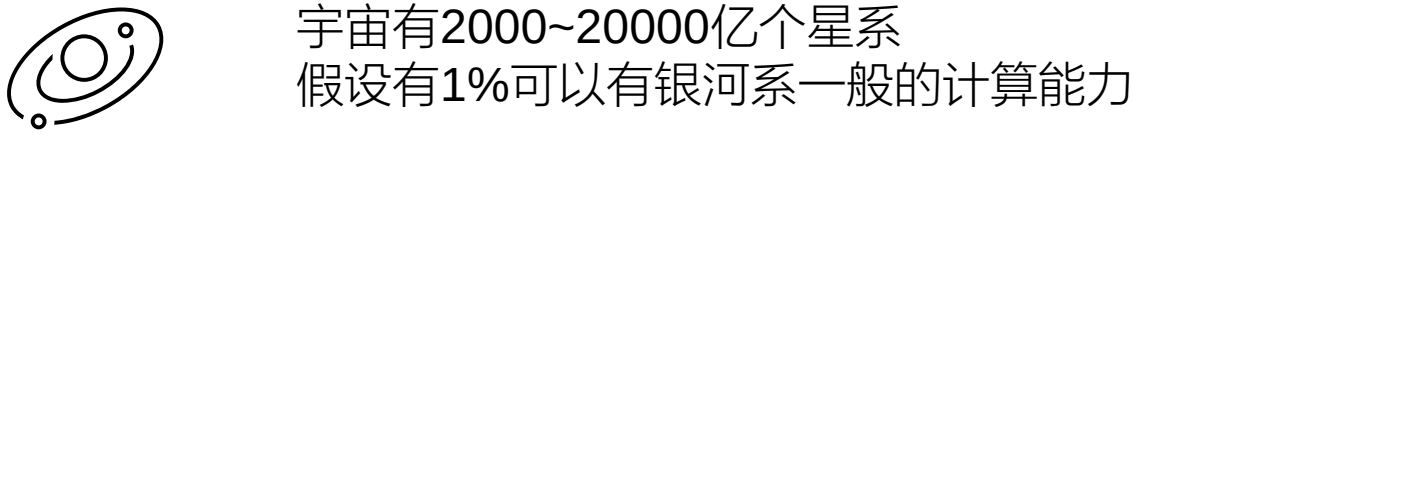
43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能

地球上有76亿人口
假设大部分人都各自拥有一个等同一个比特币网络的算力

43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能

银河系有1000~4000亿颗恒星
假设有1%可以有地球一般的计算能力

43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能



宇宙有2000~20000亿个星系
假设有1%可以有银河系一般的计算能力

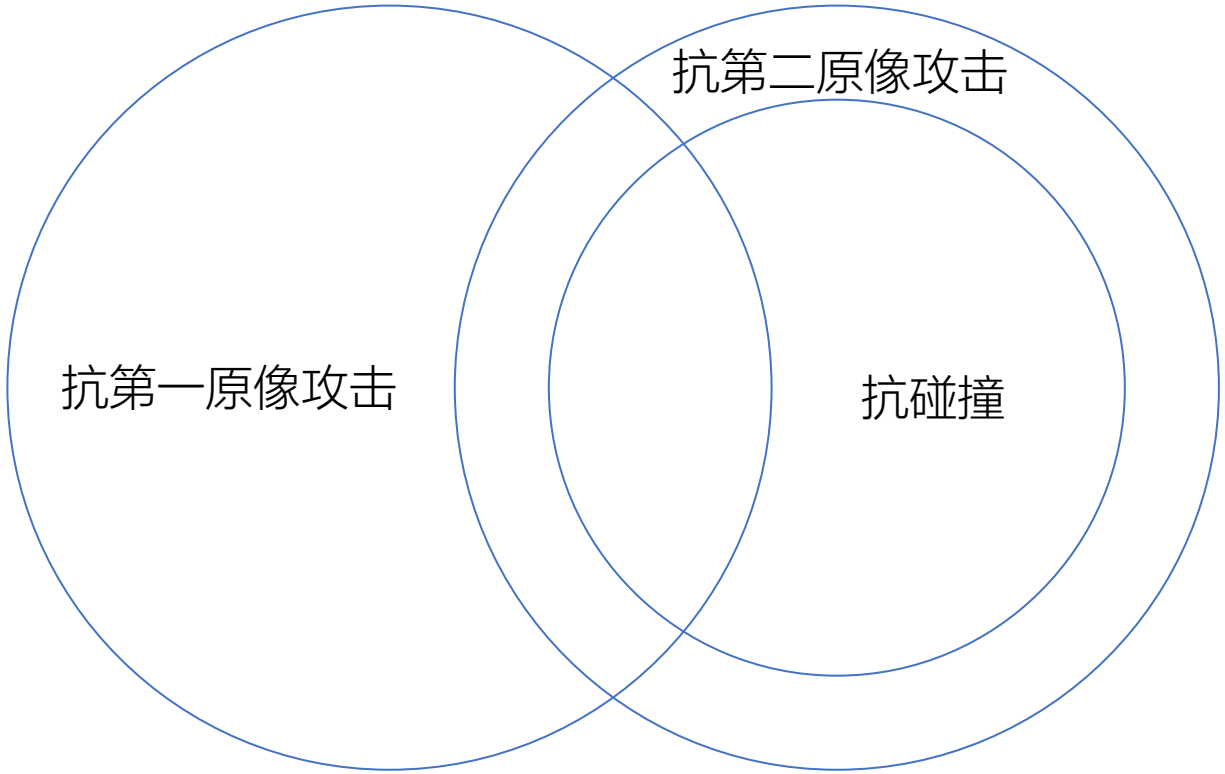
43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能

43亿秒≈136年
43亿×136年≈6000亿年≈42个宇宙年龄

43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能 x 43亿种可能

依然只有1/43亿的可能性会重复

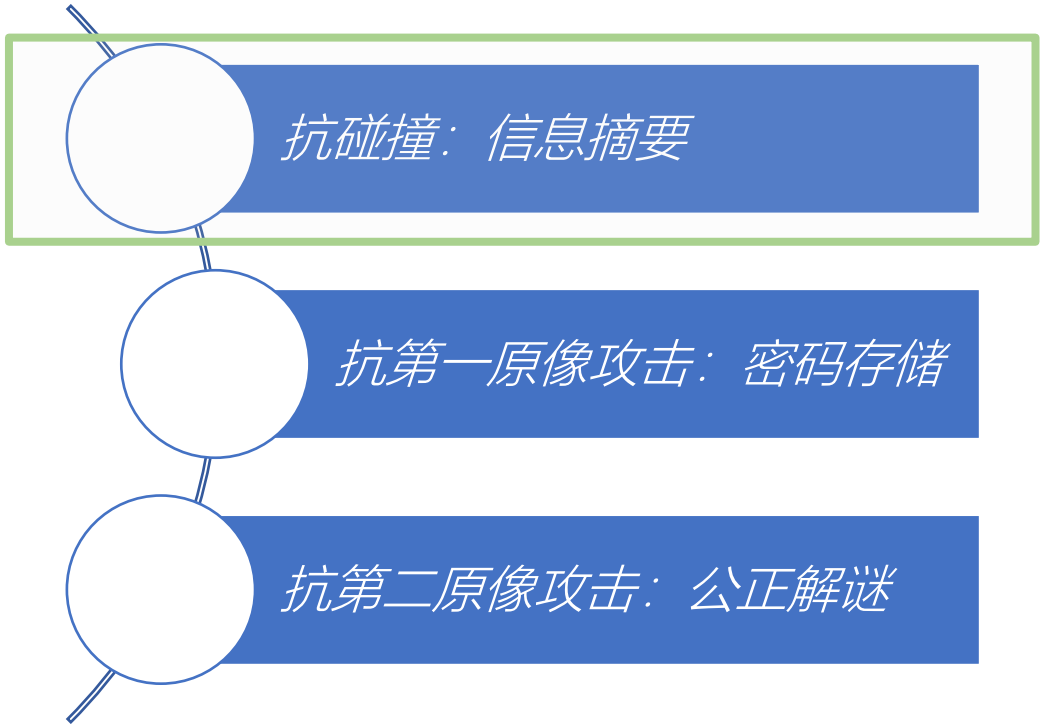
安全特性之间的关系



哈希是什么?

哈希

哈希计算——应用

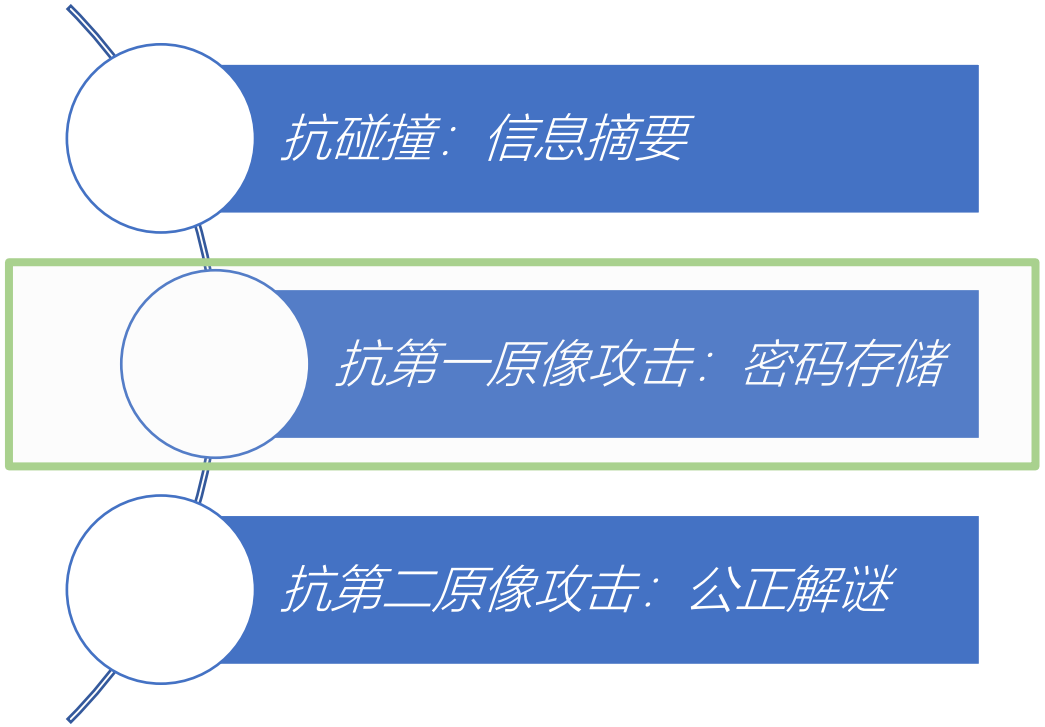


当两条信息的哈希值相同时

便可以认为两条信息是相同的

哈希值比原信息短小，更加方便

哈希计算——应用

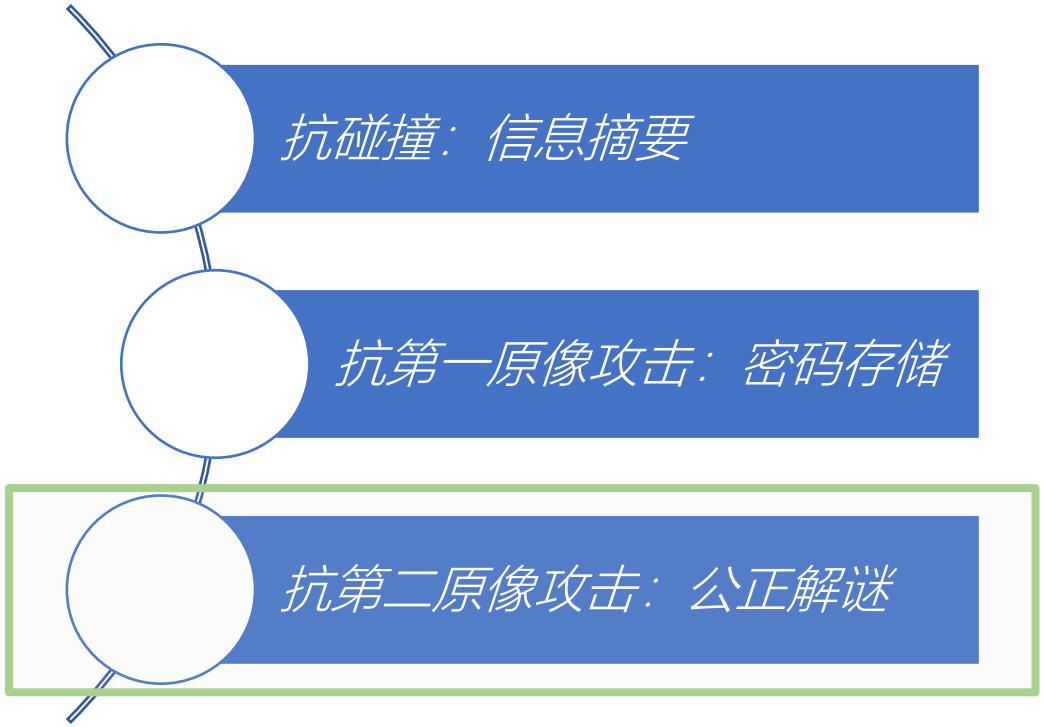


服务器上只存储密码哈希值

登陆过程仅对比哈希值

仅有用户知道密码原文

哈希计算——应用



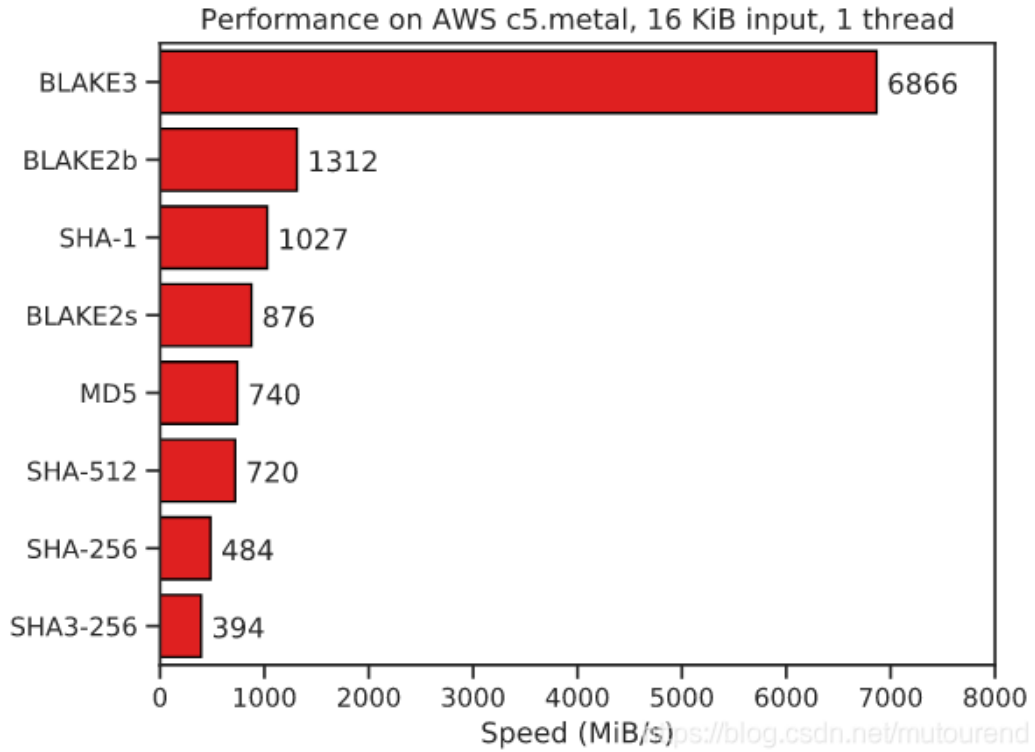
先公布答案的哈希值

到时间后公布原始答案

不公布答案也不可能改答案

哈希函数

主流哈希函数的效率



为什么要设计新的哈希函数?

- 设计低复杂度的哈希函数
- Merkle树中需要能接受两个哈希长度的哈希函数

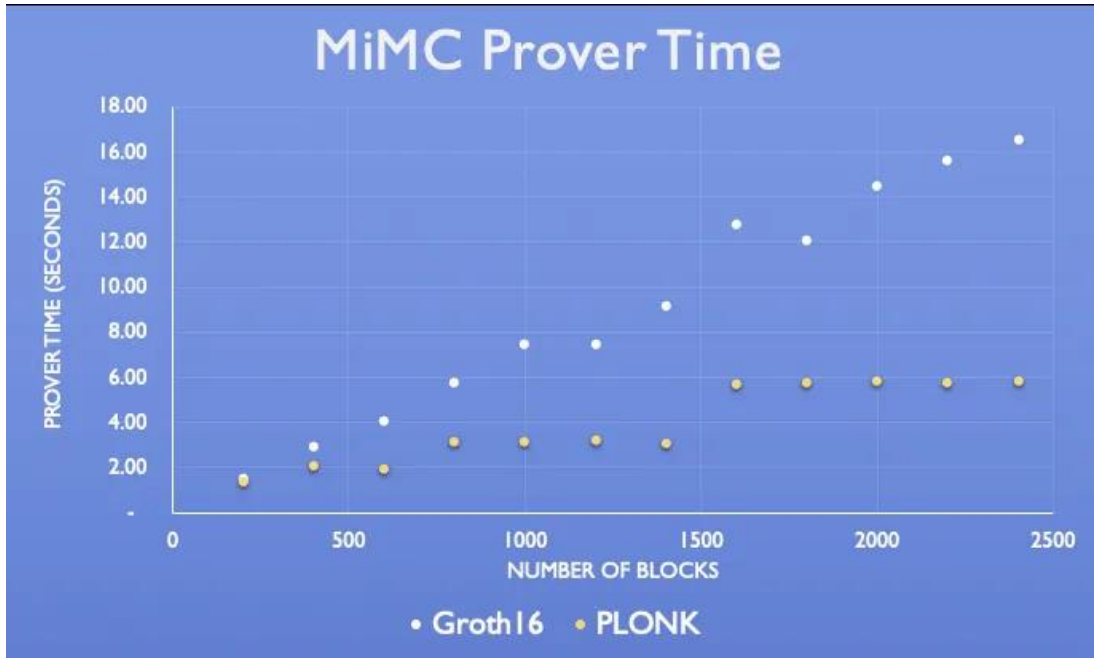
STARK Friendly Hash

Family	Name	$\mathbb{F}$	t / invocation	w	d	c / 10^5 invocations
Standard	$\mathsf{SHA2}_{128}$	64-bit prime	1000	20	2	5.6×10^{10}
		$\mathrm{GF}(2^{64})$	3762	56	11	7.2×10^{11}
	$\mathsf{SHA3}_{128}$	$\mathrm{GF}(2^{64})$	1536	25	2	1.1×10^{11}
AES-DM	AES_{64}	$\mathrm{GF}(2^{64})$	48	62	8	7.5×10^{9}
	$\mathsf{Rijndael}_{80}$	$\mathrm{GF}(2^{64})$	58	68	8	9.9×10^{9}
Algebraic Sponge	$\mathsf{Pedersen}_{128}$	256-bit prime	128	16	2	8.7×10^{10}
	MiMC_{126}	253-bit prime	320	2	3	6.4×10^{10}
	GMiMC_{122}	61-bit prime	101	1	3	9.4×10^{8}
	$\mathsf{Starkad}_{126}$	$\mathrm{GF}(2^{63})$	10	14	3	3.4×10^{8}
	$\mathsf{Poseidon}_{122}$	61-bit prime	8	17	3	3.1×10^{8}
	Rescue_{122}	61-bit prime	10	12	3	3×10^{8}
	Vision_{126}	$\mathrm{GF}(2^{63})$	20	12	6	7.5×10^{8}
			40	12	4	1.4×10^{9}

STARK Friendly Hash – Survey and Recommendation

哈希：PLONK vs Groth16

	PLONK	Groth16
MiMC Prover Time	5.6s	16.5s
SHA-256 Prover Time	6.6s	1.4s
Verifier Gas Cost	223k	203k
Proof Size	0.51kb	0.13kB



https://medium.com/aztec-protocol/plonk-benchmarks-2-5x-faster-than-groth16-on-mimc-9e1009f96dfe

EVM上哈希算法Gas对比

Merkle Accumulator Hash Function Comparison (Ranked Best to Worst)

ETH Gas Costs

1. Keccak256
2. SHA256
3. Poseidon T3 (Binary)
4. Poseidon T6 (Quinary)
5. MiMC Sponge

ZK Circuit Constraints

1. Poseidon T6 (Quinary)
2. Poseidon T3 (Binary)
3. MiMC Sponge
4. Keccak256
5. SHA256

Poseidon T3 (Binary) offers the best tradeoff between ETH Gas Costs and ZK Circuit Constraints for Railgun

RAILGUN_

Hash function	Gas cost to hash 2 values
MiMC	59840
Poseidon	49858
SHA256	23179

EVM中启用预编译的Keccak：Gas只要700

https://twitter.com/RAILGUN_Project/status/1363686166734675968?s=20
https://ethresear.ch/t/gas-and-circuit-constraint-benchmarks-of-binary-and-quinary-incremental-merkle-trees-using-the-poseidon-hash-function/7446
https://github.com/crytic/evm-opcodes

选择哈希函数

- 选择链上计算及电路计算都高效的哈希函数
- 不同的哈希函数的输入参数不一样，需要根据使用的哈希函数进行代码逻辑构造