

DOI: 10.11830/ISSN.1000-5013.202303012



# 鼓励合作秘密分享方案的概念与构建

程小刚<sup>1,2</sup>, 郭韧<sup>3</sup>, 卢正添<sup>1,2</sup>, 周长利<sup>1,2</sup>, 陈永红<sup>1,2</sup>

- (1. 华侨大学 计算机科学与技术学院, 福建 厦门 361021;  
2. 华侨大学 厦门市数据安全与区块链技术重点实验室, 福建 厦门 361021;  
3. 华侨大学 工商管理学院, 福建 泉州 362021)

**摘要:** 提出一种新的鼓励合作秘密分享方案的概念,即参与秘密重建的成员越多,则重建过程越简单、计算量越小;若有少数成员缺席重建秘密过程,则秘密重建仍然是可能的,只是计算量有所增加,即重建计算工作量随缺席成员的个数指数级增加,而成功概率指数级降低。基于区块链中的工作量证明(PoW)和哈希函数碰撞方法,构建一个具体可行的方案。通过随机预言模型(ROM)证明了所提方案的安全性。

**关键词:** 秘密分享; 哈希函数; 区块链; 比特币; 计算复杂度

中图分类号: TN 918

文献标志码: A

文章编号: 1000-5013(2023)04-0518-08

## Concept and Construction of Collaboration-Encouraging Secret Sharing Scheme

CHENG Xiaogang<sup>1,2</sup>, GUO Ren<sup>3</sup>, LU Zhengtian<sup>1,2</sup>,  
ZHOU Changli<sup>1,2</sup>, CHEN Yonghong<sup>1,2</sup>

- (1. College of Computer Science and Technology, Huaqiao University, Xiamen 361021, China;  
2. Xiamen Key Laboratory of Data Security and Blockchain Technology, Huaqiao University, Xiamen 361021, China;  
3. College of Business Administration, Huaqiao University, Quanzhou 362021, China)

**Abstract:** A new concept of collaboration-encouraging secret sharing scheme is proposed, it shows that more members participate in secret reconstruction, the simpler the reconstruction process and the smaller the computation. If a few members are absent from the secret reconstruction process, secret reconstruction is still possible, but the amount of computation increases, that is, the amount of reconstruction computation increases exponentially with the decreasing of absent members, while the success probability decreases exponentially. Based on the proof-of-work (PoW) and Hash function collision method in the blockchain, a concrete and feasible scheme is constructed. The security of the proposed scheme is proved through the random oracle model (ROM).

**Keywords:** secret sharing; Hash function; blockchain; Bitcoin; computation complexity

秘密分享<sup>[1-2]</sup>是指秘密分发者(dealer)可以把一个秘密值分成多份,分别传输给不同的参与方,全部或部分满足预先设定好条件的参与方集合(访问结构)就能恢复秘密,而不满足条件的参与方集合则不能恢复秘密。如果不能恢复,甚至不能获得任何关于秘密的信息,则称为完美秘密分享方案。

常用的 $(t, n)$ 秘密分享是指 $n$ 个参与方中如有 $t$ 方以上参与秘密重建,则可以重建秘密;若参与重建秘密的人数不足 $t$ 个,则不能获得关于秘密的任何信息。基于多项式的拉格朗日差值公式,Shamir<sup>[1]</sup>

收稿日期: 2023-03-12

通信作者: 程小刚(1973-),男,讲师,博士,主要从事信息安全、应用密码学的研究。E-mail: cxg@hqu.edu.cn.

基金项目: 福建省社会科学基金资助项目(FJ2021B163, FJ2020B044)

构建了第一个且最重要的 $(t, n)$ 完美秘密分享方案. 其他的实现方式还有基于几何的秘密分享方案<sup>[2]</sup>、基于中国剩余定理的 $(t, n)$ 秘密分享方案<sup>[3]</sup>等. 秘密分享方案有很多变种, 如支持不同的访问结构<sup>[4-5]</sup>、可验证秘密分享(VSS)<sup>[6]</sup>、对量子比特进行分享的量子秘密分享(QSS)方案<sup>[7]</sup>、支持越多人参与可减少通讯复杂度的 QSS 方案<sup>[8]</sup>、混合式秘密分享<sup>[9]</sup>、阈值可调秘密分享方案<sup>[10]</sup>、理性秘密分享(RSS)<sup>[11-12]</sup>、量子理性秘密分享<sup>[13-14]</sup>、主动安全秘密分享(PSS)<sup>[15-16]</sup>等.

在区块链技术<sup>[17-18]</sup>中, 多参与方以分布式方式共同维护一个记录账本, 在没有一个中心可信方的情况下, 确保记录的唯一性, 其典型应用是为比特币(Bitcoin)<sup>[18]</sup>维护一个交易账本, 用计算难度大的数学难题的解决, 证明某个交易的合法性. 常用的数学难题就是哈希(Hash)函数的碰撞. 近年来, 区块链技术在隐私保护、智能家居、共识协议、智能合约等方面都有广泛的研究和应用<sup>[19-27]</sup>.

Shamir<sup>[1]</sup>的秘密分享方案特性是只要参与重建秘密的人数大于阈值 $t$ 即可, 更多的人参与不能带来任何益处, 甚至可能使重建工作量更大, 因为重建 $t-1$ 次多项式只要 $t$ 个点即可. 基于此, 本文提出一种新的鼓励合作秘密分享方案.

## 1 初步知识

**定义 1** 鼓励合作多方秘密分享方案如下.

秘密分发者 Dealer 可将秘密分成 $n$ 份, 分别分给 $n$ 个参与方; 若有 $n$ 方都参与秘密重建过程, 则可高效恢复秘密并验证秘密的正确性; 若有 $1$ 方缺席秘密重建过程, 如果想重建秘密, 则要花费较多的计算量且成功概率稍低; 一般地, 如果有 $m$ 方缺席重建过程, 若要重建秘密, 花费的计算工作量随 $m$ 指数级增长或成功概率随 $m$ 指数级降低.

区块链中, 共识机制是在分布式环境下多方能达成一致意见的机制, 在 Bitcoin 的应用中是让各方都同意和维护一个唯一的交易账本, 从而防止货币的重复使用问题. 基于工作量证明(PoW)的共识机制是通过计算能力来保证达成共识, 即只要网络中诚实的参与者(矿工)计算力大于攻击者, 就可以保证交易账本的唯一性, 保证 Bitcoin 的正常运行. 计算力主要是通过 Hash 函数(如 SHA256)的碰撞实现, 如要求找到一个随机数, 使得交易数据与此随机数的哈希结果前面若干位必须为 0. 在基于区块链的加密货币 Bitcoin 中, 挖矿的目的就是找到 Hash 值符合一定条件(如前若干位为 0)的原像作为工作量证明, 奖励系统分配给发现 Hash 碰撞的矿工若干比特币进行激励, 从而保证系统的正常运行.

借鉴 Bitcoin, 把符合条件的 Hash 函数值作为秘密, 参与人的秘密为原像, 全部都参与, 则有原像, 可简单构建出作为秘密的 Hash 值, 而缺 $1$ 方或多方的话, 则原像不完整, 必须对缺少的部分原像进行猜测, 才能构建秘密. 显然, 缺少的秘密越多, 则要随机猜测的比特数越多, 工作量也就越大.

## 2 鼓励合作秘密分享方案的构建

### 2.1 初始构建

秘密分享的步骤如下.

1) 若有 $n$ 方参与秘密分享, 那么, Dealer 就要为每一方生成 $m$ 比特的随机信息, 即 $x_1, x_2, \dots, x_n$ , 每个 $x_i$ 的长度都为 $m$ 比特.

2) Dealer 对这些数据进行 Hash 运算,  $h(x_1, x_2, \dots, x_n)$ . 那么, 得到的 Hash 值的指定部分比特(如后 50% 比特)就是要共享的秘密, 而其他部分(如前 50% 比特)是作为验证的比特, 即可验证重构出来的秘密是否正确; 此时, 共享的秘密显然是随机值, 若希望共享一预先给定的值, 则可以将此随机值作为对称加密方案(如 AES)的密钥来加密给定值, 重建秘密时只需重建出共享的随机值(即密钥), 再进行解密即可.

3) Dealer 把 $x_1, x_2, \dots, x_n$ 分别传送给参与秘密分享的 $n$ 方作为他们的秘密值, 同时传送作为验证的那部分(如前 50% 比特)Hash 值比特.

重建秘密的步骤如下.

1) 如果参与秘密分享的 $n$ 方都参与重建, 那么, 重构过程非常简单, 只要计算 Hash 值, 即

$$h' = h(y_1, y_2, \cdots, y_n).$$

2) 验证. 把  $h'$  中的指定部分(如前 50% 比特)同公开的验证比特进行比较, 如果相同, 那么剩下的部分就是重建出来的秘密; 如果不同, 就说明成员中有一部分是恶意的, 即没有贡献出真正的秘密, 或者不是参与方之一.

3) 缺 1 方. 如果缺 1 方, 不失一般性地, 假设缺席的是第  $n$  个参与方(即  $x_n$ ), 则前  $n-1$  个参与方也可尝试重构秘密, 即对每个可能的  $x_n$  值计算 Hash 值, 如果某个 Hash 值的指定部分同公开验证比特相同, 那么, 高概率剩下的部分就是秘密, 计算代价为  $O(2^m)$ , 其中,  $m$  为 1 个参与方从 Dealer 处获得的分享(Share)的比特长度.

4) 缺 2 方、多方. 如果缺席是 2 方, 那么, 计算代价为  $O(2^{2 \times m})$ ; 一般地, 若缺席  $r$  方, 那么, 计算代价为  $O(2^{r \times m})$ , 计算代价随着缺席的人数指数级上升; 若  $r \times m > l$  (即缺失的比特数大于 Hash 函数的输出长度  $l$ ), 那么, 会有原像冲突, 即 1 个 Hash 值可能对应多个原像, 则平均每个 Hash 值有  $2^{r \times m - l}$  个原像, 猜测成功的概率为  $(1/2)^{r \times m - l}$ .

上述分析其实不够精确, 由于敌手可以直接盲目猜测作为秘密的那部分 Hash 比特, 那么, 计算代价就是固定的  $O(1)$ , 成功的概率为  $1/2^{\text{秘密长度}}$ . 因此, 可以通过加长秘密的长度来增强安全性.

2.2 提高安全性的构建

为避免上述的安全性问题, 可考虑进行多次 Hash, 把多个 Hash 函数输出值作为秘密, 即在一次  $h_j(x_1, x_2, \cdots, x_n, 0)$  的基础上, 延长秘密包含

$$h_j(x_1, x_2, \cdots, x_n, 1), h_j(x_1, x_2, \cdots, x_n, 2), \cdots, h_j(x_1, x_2, \cdots, x_n, k),$$

即加长了秘密的长度, 可把  $h_j(x_1, x_2, \cdots, x_n, 0)$  的部分或全部比特作为验证比特, 后面的 Hash 值作为秘密值. 此时, 可根据需要设定  $k$  的值,  $k$  值越大, 秘密越长. 由于秘密的长度较大, 直接盲目猜测的成功概率就大大降低, 甚至可忽略不计.

2.3 一个简单的示例

选取一个简单的 Hash 函数(实际应用中, 应该选取更安全的 Hash 函数, 如 SHA256)如下

$$h(x_1, x_2, \cdots, x_n) = \sum_{i=1}^n a_i \cdot x_i \bmod N.$$

上式中:  $N$  为一素数; 系数  $(a_1, a_2, \cdots, a_n) \in \{0, 1, \cdots, N-1\}$  为随机数.

以 3 方为例, 令  $N=257, (a_1, a_2, a_3)=(17, 123, 25)$ , 则 Hash 函数为

$$h(x_1, x_2, x_3) = 17x_1 + 123x_2 + 25x_3 \bmod 257,$$

则容易得到  $h(1, 2, 3)=81, h(7, 8, 9)=43$ .

基于此 Hash 函数可设计秘密分享方案如下.

随机选择  $x_1=59, x_2=98, x_3=213$ , 计算 Hash 值为

$$h(59, 98, 213) = 17 \cdot 59 + 123 \cdot 98 + 25 \cdot 213 \bmod 257 = 135 = (1000\ 0111)_2.$$

可设置要分享的秘密为后四位 0111, 而前四位 1000 作为验证比特.

显然, 当 3 方都参与秘密重建时, 只需 3 方各自提供自己的部分, 进行一次 Hash 运算即可, 并可用得到的 Hash 值前四位同 1000 比较, 若相同, 则剩下四位即为重建的秘密 0111; 若不同, 则重建失败, 说明有用户提供了假的秘密部分.

若只用 2 方参与秘密重建, 设参与方为  $x_1=59, x_2=98$ , 此 2 方也可尝试重建秘密, 即对  $x_3=\{0, 1, 2, \cdots, 256\}$  的每个值尝试计算 Hash 值, 前四位为 1000 的 Hash 值区间为 128~143. Hash 值满足前四位条件的原像与秘密值, 如表 1 所示.

由表 1 可知: 16 个 Hash 值都可能是秘密, 因为其值的前四位都是 1000, 假如任选其中之一作为秘密, 显然成功的概率为  $1/16$ .

符合条件的原像个数与成功概率, 如表 2 所示. 类似地, 假设只有 1 方  $x_1=59$  想重建秘密, 那么, 可对  $x_2, x_3$  的每个可能值进行 Hash 运算, 可知符合前四位条件的 Hash 值有 4 112, 与理论估计值 4 096 非常接近, 任选其中一组, 重建成功概率为  $\frac{1}{4\ 112}$ ; 若假设一个参与方都没有, 成功的概率为  $\frac{1}{1\ 056\ 784}$ ,

与理论估计值 1 048 576 也非常接近,说明此 Hash 函数的输出随机性较好,即实现了计算量和成功概率随缺失方数指数级增加和降低.

表 1 Hash 值满足前四位条件的原像与秘密值

Tab. 1 Original images and secret values with Hash values satisfying first four conditions

$x_3$	Hash 值	二进制展开	秘密(后四位)	$x_3$	Hash 值	二进制展开	秘密(后四位)
18	143	1000 1111	1111	141	134	1000 0110	0110
28	136	1000 1000	1000	172	138	1000 1010	1010
38	129	1000 0001	0001	182	131	1000 0011	0011
59	140	1000 1100	1100	203	142	1000 1110	1110
69	133	1000 0101	0101	213	135	1000 0111	0111
100	137	1000 1001	1001	223	128	1000 0000	0000
110	130	1000 0010	0010	244	139	1000 1011	1011
131	141	1000 1101	1101	254	132	1000 0100	0100

表 2 符合条件的原像个数与成功概率

Tab. 2 Number of eligible original images and success probability

缺失方数	计算量	符合条件的原像个数		成功概率
		理论估计值	实际个数	
不缺	1	1	1	1
缺 1 方	$2^8$	$2^8/2^4=16$	16	1/16
缺 2 方	$2^{16}$	$2^{16}/2^4=4\ 096$	4 112	1/4 112
缺 3 方	$2^{24}$	$2^{24}/2^4=1\ 048\ 576$	1 056 784	1/1 056 784

但上述的分析有问题,即如果敌手直接忽略各方的 Share,直接瞎猜后四位,成功的概率也是 1/16,即此时多参与方并不能真正带来重建工作量的减少或成功概率的提高. 需要进行安全性的加强,即延长秘密的长度,降低盲目猜测的概率,促使敌手进行大量计算.

此时,秘密不只是上述 Hash 值的后四位,还包括  $h_2(x_1,x_2,x_3,1)$ ,需要可继续加长秘密,包含  $h_2(x_1,x_2,x_3,2),h_2(x_1,x_2,x_3,3)$  等).  $h_2$  是另一个 Hash 函数,定义为

$$h_2(x_1,x_2,x_3,i)=17x_1+123x_2+25x_3+81i\text{mod }257.$$

显然, $h_2(x_1,x_2,x_3,0)$  跟上述的 Hash 函数是一样的.

同上,取  $x_1=59,x_2=98,x_3=213\text{mod }257$ ,计算 Hash 值为

$$h_2(x_1,x_2,x_3,0)=135,\quad h_2(x_1,x_2,x_3,1)=216.$$

用  $h_2(x_1,x_2,x_3,0)$  即 135 的前 50% 比特 1000 作为验证比特,而后四位和  $h_2(x_1,x_2,x_3,1)=216=1101\ 1000$  作为秘密,此时秘密长度为 12 位;若敌手忽略  $x_1,x_2$  直接盲目猜测,那么,成功概率为  $1/2^{12}$ ,此时,敌手的理性选择是利用  $x_1,x_2$  对所有可能的  $x_3$  进行 Hash 运算尝试,运算量为  $2^8$ ,成功概率是 1/16. 二次 Hash 延长秘密长度,如表 3 所示. 表 3 中任何一行都是可能的秘密.

表 3 二次 Hash 延长秘密长度

Tab. 3 Second Hash extends secret length

$x_3$	$h_2$ ( $x_1,x_2,x_3,0$ )	二进制 展开	秘密 (后四位)	秘密 $h_2$ ( $x_1,x_2,x_3,1$ )	$x_3$	$h_2$ ( $x_1,x_2,x_3,0$ )	二进制 展开	秘密 (后四位)	秘密 $h_2$ ( $x_1,x_2,x_3,1$ )
18	143	1000 1111	1111	224	141	134	1000 0110	0110	215
28	136	1000 1000	1000	217	172	138	1000 1010	1010	219
38	129	1000 0001	0001	210	182	131	1000 0011	0011	212
59	140	1000 1100	1100	221	203	142	1000 1110	1110	223
69	133	1000 0101	0101	214	213	135	1000 0111	0111	216
100	137	1000 1001	1001	218	223	128	1000 0000	0000	209
110	130	1000 0010	0010	211	244	139	1000 1011	1011	220
131	141	1000 1101	1101	222	254	132	1000 0100	0100	213

由表 3 可知: $h_2(x_1,x_2,x_3,1)$  与  $h_2(x_1,x_2,x_3,0)$  有简单的线性关系,即敌手可在盲目猜测  $h_2(x_1,x_2,x_3,0)$  后加上 81 得到后八位,成功概率不变;然而,此处用的 Hash 函数很简单(线性运算),实际中

的 Hash 函数(如 SHA256)采用的是比较复杂的非线性运算,即敌手盲目猜测  $h_2(x_1, x_2, x_3, 0)$  后,不会得到关于  $h_2(x_1, x_2, x_3, 1)$  的任何信息,只能先猜测原像  $x_1, x_2, x_3$  的值,再计算  $h_2(x_1, x_2, x_3, 0)$  和  $h_2(x_1, x_2, x_3, 1)$ ,直接猜测 Hash 值的成功率较低.

假设缺 2 方,直接盲目猜测的成功概率为  $1/2^{12}$ ,此时秘密的长度为 12 位,对缺的 2 方所有值进行 Hash 计算工作量为  $1/2^{16}$ ,可得到符合条件的 Hash 值约为  $2^{16}/2^4$  (假设每个 Hash 值都是随机),即成功概率为  $1/2^{12}$ ,同盲目猜测概率相当;采取类似方法继续扩展秘密长度,把  $h_2(x_1, x_2, x_3, 2)$  也作为秘密,秘密长度为  $12+8=20$ ,盲目猜测的概率为  $1/2^{20}$ ,此时,敌手的理性选择为对所缺的 2 方所有可能值进行尝试.在实际应用中,可根据具体情况调整秘密的长度,以确保重建的工作量满足设计要求,利用包含  $h_2(x_1, x_2, x_3, i)$  技术增加秘密长度.

3 安全性与参数分析

上述方案的安全性主要是基于 Hash 函数的安全性,同 Bitcoin 中的工作量证明 PoW 类似,持有原像,则可直接通过 Hash 运算得到 Hash 值,否则,只能随机尝试.只要 Hash 函数是安全的,文中方案就是安全的,下面基于随机预言模型(ROM)证明文中方案的安全性.

**定理 1** 基于随机预言模型,鼓励合作秘密分享方案满足安全性性质,即计算量和成功概率随缺失方数分别指数级增长或指数级下降.

证明:设参与秘密重建的人数少 1 人,即缺失的 Hash 原像位数为  $m$ ,对缺失的  $m$  位的每一种可能进行尝试,计算量为  $2^m$ ,目标是找到 Hash 函数值的前 128 位(假设用于验证的比特位数为 128 位)满足给定值的缺失的  $m$  位原像.注意 ROM 下 Hash 有 2 个性质.

**性质 1** 在 ROM 中,获得 Hash 值的唯一方法是查询预言器 Oracle,查询 1 次就获得 1 个对应的 Hash 值,Oracle 内部如何运作是黑箱,不能被了解.

**性质 2** 每次查询,Oracle 返回的 Hash 值都是独立和随机的(要满足对同一个原像的查询返回的 Hash 值一致的条件).

在这 2 个性质下,不能快速求逆,因为其黑箱运作不能逆向工程破解,要找到符合条件的原像只能做随机尝试,即

- 1) 当缺失的原像位数  $m < 128$  时,重建的成功概率较高,而计算量随  $m$  指数级增加,即  $2^m$ ;
- 2) 当缺失的原像位数  $m > 128$  时,重建成功的概率随  $m$  指数级降低,约为  $1/2^{m-128}$ ,因为缺失的原像位数为  $m(m > 128)$  时,对每个可能的原像(有  $2^m$  个)求 Hash 值,会得到  $2^m$  个 Hash 函数值(每个长度为 128),而所有可能的 Hash 函数值只有  $2^{128}$  个,在 Hash 函数是随机的假设下,每个 Hash 值对应的原像数量平均为  $2^m/2^{128} = 2^{m-128}$  个,其中,只有 1 个是真正的秘密,所以,成功的概率为  $1/2^{m-128}$ .

设每个人的秘密 Share 为  $m$  位、 $n$  个参与人,Hash 函数的输入消息为  $n \times m$  位,若采用 SHA256 哈希函数,则一个 Hash 输出为 256 位,平均每个输出对应的原像个数为

$$2^{n \times m} / 2^{256} = 2^{n \times m - 256}.$$

假设把  $h_j(x_1, x_2, \dots, x_n, 0)$  的前 50% 比特即 128 位作为验证比特,则后 50% 比特为秘密比特,如果秘密长度不足,还可扩展秘密包括  $h_j(x_1, x_2, \dots, x_n, 1), h_j(x_1, x_2, \dots, x_n, 2), \dots, h_j(x_1, x_2, \dots, x_n, k)$ ,此时,秘密长度为  $128 + k \times 256$  比特.

- 1) 当每方的分享长度  $m = 64$  时,则缺 1 方时需要尝试的次数为  $2^{64}$ ,计算量较大,假设每个 Hash 值都是随机的,那么,1 个随机 256 位 Hash 值有原像的概率为  $2^{64}/2^{256}$ ,即只要尝试缺失那方的每一个值,成功重建秘密的概率很高;失败只可能是符合 128 位验证比特的 Hash 值有多个,产生了冲突,即重建出的秘密有多个,不能分辨哪一个是正确的,只能随机选择.此时,总共  $2^{64}$  个 Hash 值,每个 Hash 值符合条件(即前 128 位符合给定的验证比特)的概率为  $1/2^{128}$ ,则可能产生冲突(重建失败)的概率较小,为  $1 - \left(1 - \frac{1}{2^{128}}\right)^{2^{64}}$ .

精确来说,当有  $k$  个 Hash 值冲突,这  $k$  个值中只有一个是正确的,即错误的概率为  $(k-1)/k$ ,而恰有  $k$  个 Hash 值冲突的概率  $C_n^k$  为

$$C_n^k = \left(\frac{1}{2^{128}}\right)^k \left(1 - \frac{1}{2^{128}}\right)^{n-k}.$$

上式中： $n=2^{64}$ .

所以，重建失败的概率  $P_{\text{Fail}}$  为

$$P_{\text{Fail}} = \sum_{k=1}^{2^{64}} \frac{k}{k+1} C_n^k \left(\frac{1}{2^{128}}\right)^k \left(1 - \frac{1}{2^{128}}\right)^{n-k}.$$

当  $k=0$  时是没有冲突的，只有一个值符合条件的情况，即

$$P_{\text{Fail}} < \sum_{k=1}^{2^{64}} C_n^k \left(\frac{1}{2^{128}}\right)^k \left(1 - \frac{1}{2^{128}}\right)^{n-k} = \sum_{k=0}^{2^{64}} C_n^k \left(\frac{1}{2^{128}}\right)^k \left(1 - \frac{1}{2^{128}}\right)^{n-k} - \left(1 - \frac{1}{2^{128}}\right)^n = 1 - \left(1 - \frac{1}{2^{128}}\right)^{2^{64}} < 1 - \left(1 - \frac{1}{2^{64}}\right) = \frac{1}{2^{64}}.$$

当  $0 < p < 1$ ， $n$  是大于 1 的正整数时，有  $(1-p)n > 1-np$ ，可用数学归纳法证明：当  $n=2$  时， $(1-p)^2 = 1-2p+p^2 > 1-2p$  显然成立，若  $n=k$  成立，则  $(1-p)^{k+1} > (1-kp)(1-p) = 1-(k+1)p+kp^2 > 1-(k+1)p$  成立。此时，产生冲突（重建失败）的概率很小，成功的概率很高。

而缺 2 方时计算量为  $2^{128}$ ，类似上述计算，成功是没有冲突发生（符合前 128 位条件的 Hash 值唯一），概率较低。则成功概率为  $P_{\text{Success}} \approx \left(1 - \frac{1}{2^{128}}\right)^{2^{128}} \approx \frac{1}{e} \approx 0.367\ 88$ ，这是因为  $\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e}$ 。

缺 3 方时，计算量为  $2^{192}$ ，平均一个符合前 128 位条件 Hash 值有  $1/2^{64}$  个原像，即成功的概率约为  $1/2^{64}$ ，而敌手的另一种选择是直接盲目猜测 128 为秘密值，成功概率很低，为  $1/2^{128}$ ，但计算开销很小，为  $O(1)$ 。

缺 4 方时，计算量为  $2^{256}$ ，成功概率约为  $1/2^{128}$ ，原秘密长度 128 位就不够了，若保持 128 位秘密长度，则敌手可以选择直接盲目猜测，成功概率为  $1/2^{128}$ ，与上述经过大量计算后成功的概率一致，所以，理性的敌手会选择直接盲目猜测，此时，要求秘密长度  $S$  远远大于 128，即  $|S| \gg 128$ ；若经过计算后成功概率为  $p$ ，那么要求秘密长度远远大于  $\lg(1/p)$ ，否则，敌手可直接盲目猜测，这违背了文中方案的设计初衷。增加秘密的长度可用上述方法，即加上  $h_j(x_1, x_2, \dots, x_n, k) (k=1, 2, 3, \dots)$  作为秘密。

2) 当  $m=128$  时，缺 1 方计算量为  $2^{128}$  且成功概率较低，而缺 2 方时计算量为  $2^{256}$ ，此时，要求秘密长度远大于 128 位。

3) 当  $m=256$  时，缺 1 方计算量为  $2^{256}$ ，要求秘密长度远大于 128 位。

4) 当  $m > 256$  时，缺 1 方计算量为  $2^m$ ，要求秘密长度远大于  $(m-128)$  位。

综上，不同的每方分享长度下，计算量、成功概率、缺失方数和缺失位数的关系，如表 4 所示。

表 4 计算量、成功概率、缺失方数和缺失位数的关系

Tab. 4 Relationship among computation, success probability, and number of missing squares and missing digits					
缺失方数	参数	$m=64$	$m=128$	$m=256$	$m>256$
缺 1 方	计算量	$2^{64}$	$2^{128}$	$2^{256}$	$2^m$
	成功概率	很高	较低	$1/2^{128}$	$1/2^{m-128}$
	秘密长度	—	—	$ S  \gg 128$	$ S  \gg m-128$
缺 2 方	计算量	$2^{128}$	$2^{256}$	$2^{512}$	$2^{2 \times m}$
	成功概率	较低	$1/2^{128}$	$1/2^{384}$	$1/2^{2 \times m-128}$
	秘密长度	—	$ S  \gg 128$	$ S  \gg 384$	$ S  \gg 2 \times m-128$
缺 3 方	计算量	$2^{192}$	$2^{384}$	$2^{768}$	$2^{3 \times m}$
	成功概率	$1/2^{64}$	$1/2^{256}$	$1/2^{640}$	$1/2^{3 \times m-128}$
	秘密长度	$ S  \gg 64$	$ S  \gg 256$	$ S  \gg 640$	$ S  \gg 3 \times m-128$
缺 4 方	计算量	$2^{256}$	$2^{512}$	$2^{1\ 024}$	$2^{4 \times m}$
	成功概率	$1/2^{128}$	$1/2^{384}$	$1/2^{896}$	$1/2^{4 \times m-128}$
	秘密长度	$ S  \gg 128$	$ S  \gg 384$	$ S  \gg 896$	$ S  \gg 4 \times m-128$

当缺失的原像位数  $m > 128$  时，找到一个符合条件的 Hash 原像需要花费的计算量约为  $1/2^{128}$ ，而表 4 中列出的工作量  $1/2^m$  为找到全部符合条件的原像的总工作量，即尝试缺失  $m$  位的每一种可能性。

显然,若敌手直接盲目猜测,计算量为  $O(1)$ ,成功概率与秘密长度有关,即

1) 当验证与秘密总长度为 256 位( $h_j(x_1, x_2, \dots, x_n, 0)$ ),验证比特为 128,秘密也为 128 位,则盲目猜测成功概率为  $1/2^{128}$ .

2) 当验证与秘密总长度为 512 位( $h_j(x_1, x_2, \dots, x_n, 0)$ 和  $h_j(x_1, x_2, \dots, x_n, 1)$ ),秘密为 384 位,盲目猜测成功概率为  $1/2^{384}$ .

3) 当验证与秘密总长度为 768 位( $h_j(x_1, x_2, \dots, x_n, 0)$ ,  $h_j(x_1, x_2, \dots, x_n, 1)$ ,  $h_j(x_1, x_2, \dots, x_n, 2)$ ),秘密为 640 位,盲目猜测成功概率为  $1/2^{640}$ .

## 4 结 论

通常的( $t, n$ )门限秘密分享方案中,只要参与重建秘密的人数超过阈值  $t$ ,就可以重建出秘密,超过阈值后,更多人的参与不会带来任何好处,甚至可能使计算量更大.提出一种新的鼓励合作的秘密分享方案,即参与秘密重建的成员越多,则重建过程越简单、计算量越小;若有少数成员缺席重建秘密过程,则秘密重建仍然是可能的,只是计算量有所增加,即重建计算工作量随缺席成员的个数指数级增加,而成功概率指数级降低.

未来准备进一步扩展的方向有以下 4 点.

1) 结合( $t, n$ )秘密分享方案,设置一个最低的阈值  $t$ ,参与人数大于等于  $t$  时,重建秘密才是可能的,否则,重建秘密是不可能的.

2) 扩展至量子信息领域,设计实现符合文中秘密分享概念的量子秘密分享方案,即参与人数越多,重建量子秘密的计算量越小.

3) 结合能节约通讯复杂度的秘密分享概念,即设计实现参与人数越多时,不仅计算量越小,而且所需的通讯复杂度也越低.同时,结合量子纠缠态探索进一步节约重建秘密时所需的通讯复杂度.

4) 设计实现重建工作量日增的秘密分享方案,以保证长期秘密分享方案的安全性.

## 参考文献:

[1] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613. DOI:10.1145/359168.359176.

[2] BLAKLEY G R. Safeguarding cryptographic keys[C]// Proceedings of Managing Requirements Knowledge, International Workshop on IEEE Computer Society. New York: IEEE Press, 1979: 313-318. DOI:10.1109/MARK.1979.8817296.

[3] ASMUTH C A, BLOOM J. A modular approach to key safeguarding[J]. IEEE Transactions on Information Theory, 1983, 29(2): 208-210. DOI:10.1109/TIT.1983.1056651.

[4] HARN L, HSU C, ZHANG Mingwu, *et al.* Realizing secret sharing with general access structure[J]. Information Sciences, 2016, 367/368: 209-220. DOI:10.1016/j.ins.2016.06.006.

[5] JIA Xingxing, GUO Yusheng, LUO Xiangyang, *et al.* A perfect secret sharing scheme for general access structures [J]. Information Sciences, 2022, 595: 54-69. DOI:10.1016/j.ins.2022.02.016.

[6] CHOR B, GOLDWASSER S, MICALI S, *et al.* Verifiable secret sharing and achieving simultaneity in the presence of faults[C]// Proceedings of the 26th IEEE Symposium on the Foundations of Computer Science. Portland: IEEE Press, 1985: 383-395. DOI:10.1109/SFCS.1985.64.

[7] HILLERY M, BUŽEK V, BERTHIAUME A. Quantum secret sharing[J]. Physical Review A, 1999, 59(3): 1829-1834. DOI:10.1103/PhysRevA.59.1829.

[8] SENTHOOR K, SARVEPALLI P K. Communication efficient quantum secret sharing[J]. Physical Review A, 2019, 100(5): 052313. DOI:10.1103/PhysRevA.100.052313.

[9] LIPINSKA V, MURTA G, RIBEIRO J, *et al.* Verifiable hybrid secret sharing with few qubits[J]. Physical Review A, 2020, 101(3): 032332. DOI:10.1103/PhysRevA.101.032332.

[10] HARN L, HSU C, ZHE Xia. A novel threshold changeable secret sharing scheme[J]. Frontiers of Computer Science, 2022, 16: 161807. DOI:10.1007/s11704-020-0300-x.

- [11] HALPERN J Y, TEAGUE V. Rational secret sharing and multiparty computation; Extended abstract[C]// Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing, New York: ACM, 2004: 623-632. DOI:10.1145/1007352.1007447.
- [12] GORDON S D, KATZ J. Rational secret sharing, revisited[C]// International Conference on Security and Cryptography for Networks, Berlin: Springer, 2006: 229-241. DOI:10.1007/11832072\_16.
- [13] MAITEA A, DE S J, PAUL G, *et al.* Proposal for quantum rational secret sharing[J]. Physical Review A, 2015, 92(2): 022305. DOI:10.1103/PhysRevA.92.022305.
- [14] QIN Huawang, TANG W K S, TSO R. Rational quantum secret sharing[J]. Scientific Reports, 2018, 8: 11115. DOI:10.1038/s41598-018-29051-z.
- [15] HERZBERG A, JARECKI S, KRAWCZYK H, *et al.* Proactive secret sharing or: How to cope with perpetual leakage[C]// Proceedings of the 15th Annual International Cryptology Conference on Advances in Cryptology, Berlin: Springer, 1995: 339-352. DOI:10.1007/3-540-44750-4\_27.
- [16] DEHKORDI M H, MASHHADI S, ORAEI H. A proactive multi stage secret sharing scheme for any given access structure[J]. Wireless Personal Communications, 2019, 104(1): 491-503. DOI:10.1007/s11277-018-6032-7.
- [17] 邵奇峰, 金澈清, 张召, 等. 区块链技术: 架构及进展[J]. 计算机学报, 2018, 41(5): 969-988. DOI:10.11897/SP. J. 1016. 2018. 00969.
- [18] 秦波, 李昌豪, 伍前红, 等. 比特币与法定数字货币[J]. 密码学报, 2017, 4(2): 176-186. DOI:10.13868/j. cnki. jcr. 000172.
- [19] 祝烈煌, 高峰, 沈蒙, 等. 区块链隐私保护研究综述[J]. 计算机研究与发展, 2017, 54(10): 2170-2186. DOI: 10.7544/issn1000-1239. 2017. 20170471.
- [20] 刘敦迪, 杜学绘, 王娜, 等. 区块链技术及其在信息安全领域的研究进展[J]. 软件学报, 2018, 29(7): 2092-2115. DOI:10.13328/j. cnki. jos. 005589.
- [21] 曾诗钦, 霍如, 黄韬, 等. 区块链技术研究综述: 原理、进展与应用[J]. 通信学报, 2020, 41(1): 134-151. DOI:10.11959/j. issn. 1000-436x. 2020027.
- [22] 刘明达, 陈左宁, 拾以娟, 等. 区块链在数据安全领域的研究进展[J]. 计算机学报, 2021, 44(1): 1-27. DOI:10.11897/SP. J. 1016. 2021. 00001.
- [23] 夏清, 窦文生, 郭凯文, 等. 区块链共识协议综述[J]. 软件学报, 2021, 32(2): 277-299. DOI:10.13328/j. cnki. jos. 006150.
- [24] 徐格, 凌思通, 李琦, 等. 基于区块链的网络安全体系结构与关键技术研究进展[J]. 计算机学报, 2021, 44(1): 55-83. DOI:10.11897/SP. J. 1016. 2021. 00055.
- [25] 张利华, 张赣哲, 曹宇, 等. 基于区块链的智能家居认证与访问控制方案[J]. 计算机应用研究, 2022, 39(3): 863-867, 873. DOI:10.19734/j. issn. 1001-3695. 2021. 08. 0321.
- [26] 卫宏儒, 李思月, 郭涌浩. 基于智能合约的秘密重建协议[J]. 计算机科学, 2022, 49(6A): 469-473. DOI:10.11896/ jsj. 210700033.
- [27] 张亮, 刘百祥. 区块链与秘密分享融合技术综述[J]. 计算机工程, 2022, 48(8): 1-11. DOI:10.19678/j. issn. 1000-3428. 0064102.

(责任编辑: 黄晓楠 英文审校: 吴逢铁)