软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software, 2022,33(10):3891–3902 [doi: 10.13328/j.cnki.jos.006296]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

一般存取结构上抗内存泄露的多级秘密共享*

宋 云[1], 李志慧[2], 王文华[3]

[1](陕西师范大学 计算机科学学院, 陕西 西安 710119)

[2](陕西师范大学 数学与统计学院, 陕西 西安 710119)

[3](陕西师范大学 民族教育学院, 陕西 西安 710062)

通信作者: 李志慧, E-mail: lizhihui@snnu.edu.cn

摘 要: 在多级秘密共享方案中, 每级存取结构里的授权集中参与者可联合重构对应的秘密. 但在实际中, 腐化了非授权集的攻击者可通过内存攻击获取部分或全部其余参与者的份额信息, 从而非法得到部分甚至是全部的秘密信息. 面对这样的内存泄漏, 现有的多级秘密共享方案都不再安全. 基于此, 首先给出了抗内存泄漏的多级秘密共享对选择秘密攻击不可区分的形式化的计算安全模型. 然后, 利用物理不可克隆函数及模糊提取器的联合作用, 基于极小线性码构造了一个适用于一般存取结构的抗内存泄露的可验证多级秘密共享方案. 同时, 在内存攻击者存在的情况下, 证明方案在随机预言模型下是计算安全的. 最后, 将所提出方案与现有方案在性能和计算复杂度两方面进行了比较分析.

关键词: 多级秘密共享; 极小线性码; 物理不可克隆函数; 抗内存泄露; 可证明安全

中图法分类号: TP393

中文引用格式: 宋云, 李志慧, 王文华. 一般存取结构上抗内存泄露的多级秘密共享. 软件学报, 2022, 33(10): 3891–3902. http://www.jos.org.cn/1000-9825/6296.htm

英文引用格式: Song Y, Li ZH, Wang WH. Memory Leakage-resilient Multi-stage Secret Sharing Scheme with General Access Structures. Ruan Jian Xue Bao/Journal of Software, 2022, 33(10): 3891–3902 (in Chinese). http://www.jos.org.cn/1000-9825/6296.htm

Memory Leakage-resilient Multi-stage Secret Sharing Scheme with General Access Structures

SONG Yun[1], LI Zhi-Hui[2], WANG Wen-Hua[3]

[1](School of Computer Science, Shaanxi Normal University, Xi'an 710119, China)

[2](School of Mathematics and Statistics, Shaanxi Normal University, Xi'an 710119, China)

[3](School of Ethnic Education, Shaanxi Normal University, Xi'an 710062, China)

Abstract: In the multi-stage secret sharing scheme, the participants of authorized sets in each level of access structures can jointly reconstruct the corresponding secret. But in reality, adversaries who corrupted an unauthorized set can obtain some or even all of the share information of the uncorrupted participants through memory attacks, thereby illegally obtaining some or even all of the shared secrets. Facing with such memory leaks, the existing multi-stage secret sharing schemes are no longer secure. Based on this, this study firstly proposes a formal computational security model of indistinguishable ability against chosen secret attack for multi-stage secret sharing. Then, using the combination of the physical unclonable function and the fuzzy extractor, a verifiable memory leakage-resistant multi-stage secret sharing scheme for general access structures is constructed based on the minimal linear codes. Furthermore, in the presence of a memory attacker, it is proved that the scheme is computational secure in the random oracle model. Finally, the proposed scheme is compared with the existing schemes in terms of their properties and computational complexity.

* 基金项目: 国家自然科学基金(61602291, 61802241); 中国博士后科学基金(2018M633456); 陕西省自然科学基础研究计划(2019JQ-472); 中央高校基本科研业务费专项资金(GK202003093)

收稿时间: 2020-09-20; 修改时间: 2020-11-12; 采用时间: 2020-12-26; jos 在线出版时间: 2021-04-20

Key words: multi-stage secret sharing; minimal linear code; physical unclonable functions (PUFs); memory leakage-resilient; provable security

秘密共享是现代密码学领域的重要分支, 在信息安全存储、群组密钥协商、多方安全计算及面向组的分布式安全协议等方面均具有重要的应用价值. 秘密共享方法一般是在一组参与者集合 $P=\{P_1,P_2,\ldots,P_n\}$中设计一个份额生成算法和秘密重构算法, 其中授权的参与者子集可以重构秘密 s, 非授权的参与者集合不能得到秘密的任何信息, 这里, 授权集的集合称为存取结构$\Gamma\in 2^P$; 同时, 称 $B\in\Gamma$是极小授权子集, 若有 $A\subseteq B$ 且 $A\neq B$, 则 $A\notin\Gamma$. 在本文中约定, 授权子集均指极小授权子集. 1979 年, Shamir[1]和 Blakley[2]分别基于 Lagrange 插值法和射影几何理论独立提出(t,n)门限方案, 之后, 学者们开始研究不同性质的(t,n)门限方案[3−6], 但该类方案要求授权集中参与者的人数一致, 这个适用于参与者权力相当时, 所以在实际应用中有一定的局限. 因此, Ito 等人[7]提出了一般存取结构上的秘密共享方案, 并证明了对任何给定的存取结构均存在安全的秘密共享方案. 此后, 学者们基于线性码、图论及向量空间等理论开始研究一般存取结构上秘密共享方案的构造[8−10]及对应存取结构的确定方法.

然而, 很多秘密共享方案一次只能共享一个秘密, 在实际应用中, 我们需要共享多个秘密. 比如, 密钥分配机构需要为不同的保险柜分配不同的密钥, 此时有 3 种方法: 一种是为每一个密钥分别建立一个单秘密共享方案, 但每个参与方的子份额太多; 另一种称为一般多秘密共享方案[11−14], 该方案中的所有秘密在一个阶段同时被重构, 这类方法的使用和控制不够灵活、实用; 那么还有一种是多级秘密共享方案[15−19], 任何授权子集里的参与者在每一级只能恢复一个秘密, 这样不仅提高了效率, 而且非常实用. 因为在实际情况中, 不同秘密任务的执行(如签名和解密)可能具有不同程度的重要性或保密性, 例如下面的场景: 某公司财务部门重要的数据库共两个, 且会计工作人员是 A–E 共 5 人, 当 A、C、E 同时通过验证时, 数据库 I 才能进入; 而当 B 和 D 通过验证时, 数据库 II 才能进入. 这种实际需求是多级秘密共享才能实现的. 后两种方案统称为多秘密共享方案. 根据攻击者的计算能力, 多秘密共享安全性分为无条件(信息论)安全和计算安全. 在一个无条件安全的多秘密共享方案中[20,21], 攻击者拥有无限的计算资源; 而在一个计算安全的多秘密共享方案中, 攻击者在计算能力上受到安全参数的限制. 然而, 无条件安全的秘密共享方案中的每个参与者的份额必须至少等于秘密的长度[19]. 显然, 当秘密是一个大的隐私文件, 而在不安全信道上传输大的消息或存储大量数据时, 这种方法的效率是很低的. 为了克服无条件安全的多秘密共享方案的这一缺点, 近几年来, 很多学者开始研究计算安全的多秘密共享方案 [13,14,22−26], 但是上述方案虽然给出了相应的构造, 却没有提供所给方案的严格的计算安全证明. 直到 2013 年, Herranz 等人[27,28]在先前工作的基础上, 首次提出了多级秘密共享方案计算安全的形式化定义, 并构造了计算安全的(t,n)门限多级秘密共享方案. 随后, Mashhadi 等人[18,29]对多秘密共享方案种类进行了细分, 并给出其余类型方案的计算安全性的模型及证明.

另外, 现有的秘密共享安全性分析均基于腐化了非授权集的攻击者无法再获得其他未腐化参与者的内存信息这一前提. 而实际中, 参与者的长期秘密份额通常保管在非易失性的内存中, 敌手可以通过内存攻击得到非易失性的内存里的秘密信息, 此时, 秘密共享安全性分析的假设便不再成立. 因此, 面对存储信息的泄露, 现有的绝大多数的多秘密共享方案已不再安全. 针对上述问题, 文献[30]首次提出了抗内存泄露的一般多秘密共享方案的定义, 并给出了具体的构造, 而抗内存泄露的多级秘密共享方案的构造还没有得到研究; 另外, 该方案[30]仅针对门限存取结构, 且未给出抗内存泄露多秘密共享方案形式化的计算安全模型及其安全证明.

基于以上讨论, 本文首先在内存泄露存在的情况下, 定义多级秘密共享的基于选择秘密攻击的不可区分实验的安全模型. 然后, 利用物理不可克隆函数以及模糊提取器提取相同的随机均匀分布的字符串(即秘密份额), 而不是将长期秘密份额保存在非易失性的内存里这一思想, 基于极小线性码构造了一个完全抗内存泄露的、多用的、可验证的可证安全多级秘密共享方案, 并证明了该方案是随机预言模型下计算安全的. 方案中每个参与者只需维护一个秘密份额, 就能实现不同的存取结构重构不同的秘密; 且相较于以往的(t,n)门限秘

密共享方案, 由于该方案是基于极小线性码的对偶码上多级秘密共享构建的, 故每级的存取结构具有更丰富的授权子集, 且其存取结构较一般线性码上的更易确定, 因此更具实用性.

1 预备知识

1.1 极小线性码的对偶码上的多级秘密共享

一个[$n,k;q$]线性码 C 是指 F_q^n 的一个 k 维线性子空间, q 是一个大素数. 设矩阵 $\boldsymbol{G}=(\boldsymbol{g}_1,\boldsymbol{g}_2,\ldots,\boldsymbol{g}_n)_{k\times n}$, 其中, $\boldsymbol{g}_i$ 是 $\boldsymbol{G}$ 的一个列向量, $1\leqslant i\leqslant n$, 即由 $\boldsymbol{G}$ 的列向量生成线性子空间 C.

定义 1[31]. 设向量 $c=(c_1,c_2,...,c_n)\in F_q^n$, 指标集$\{1\leqslant i\leqslant n|c_i\neq 0\}$称为向量 $\boldsymbol{c}$ 的支撑. 如果码字 $\boldsymbol{c}_2$ 的支撑包含码字 $\boldsymbol{c}_1$ 的支撑, 则称码字 $\boldsymbol{c}_2$ 覆盖码字 $\boldsymbol{c}_1$.

定义 2[18]. 如果码 C 的一个码字 $\boldsymbol{c}$ 的第 j 个分量为 1, 并且不覆盖其他第 j 个分量为 1 的码字, 则称这样的码字为 j-极小码字.

定义 3[18]. 如果线性码 C 的生成矩阵中不含零列, 且码 C 的每个非零码字均是只能覆盖其倍数而不能覆盖其他码字, 则称码 C 为极小线性码.

在一个以 G 为生成矩阵的[$n,k;q$]线性码 C 上的秘密共享方案中, 秘密是 F_q 中的一个元素, 秘密分发者想要在 n 个参与者 $P_1,P_2,\ldots,P_n$ 中根据存取结构$\Gamma_1,\ldots,\Gamma_n$ 分别共享 n 个秘密 $s_1,\ldots,s_n$. 为了得到与秘密 $s_1,s_2,\ldots,s_n$ 有关的每个参与者的份额, 分发者随机选取向量 $\boldsymbol{u}_j=(u_{j1},u_{j2},...,u_{jk})\in F_q^k$, 使得 $s_j=\boldsymbol{u}_j\boldsymbol{g}_j$, 其中, $1\leqslant j\leqslant n$. 对于每一个 j, 易证存在 q^{k-1} 个这样的向量 $\boldsymbol{u}_j\in F_q^k$. 而后, 分发者将 $\boldsymbol{u}_j$ 作为信息向量并计算对应的码字 $\boldsymbol{t}_j=(t_{j1},t_{j2},\ldots,t_{jn})=\boldsymbol{u}_j\boldsymbol{G}$, 之后, 他将 t_{ji} 依次分发给 P_i 作为他们的秘密份额($1\leqslant i\leqslant n,i\neq j$).

容易看出: 对于某个 $1\leqslant j\leqslant n$, 如果 $\boldsymbol{g}_j$ 是 $\boldsymbol{g}_{i_1},...,\boldsymbol{g}_{i_\eta}$ 的线性组合, 即 $\boldsymbol{g}_j=\sum_{\gamma=1}^{\eta}x_\gamma\boldsymbol{g}_{i_\gamma}$, 则秘密 s_j 的重构可以通过计算 $s_j=\boldsymbol{u}_j\boldsymbol{g}_j=\sum_{\gamma=1}^{\eta}x_\gamma\boldsymbol{u}_j\boldsymbol{g}_{i_\gamma}=\sum_{\gamma=1}^{\eta}x_\gamma t_{ji_\gamma}$, 其中, $1\leqslant j\leqslant n$, $1\leqslant i_1<i_2<\ldots<i_n\leqslant n$, $1\leqslant\eta\leqslant n-1$, $i_\gamma\neq j$, 系数 $x_\gamma\in F_q$ 可由集合 B 中的参与者公开计算[18]. 故可知, $B=\{P_{i_1},...,P_{i_\eta}\}$ 是基于线性码上的秘密共享的存取结构Γ_j 的一个授权子集. 由此, 我们可以定义 n 元存取结构$\Gamma_1,\ldots,\Gamma_n$ 如下:

$$\Gamma_j=\{A|P_i\in A\Leftrightarrow C^{\perp}\text{码中的一个 }j\text{-极小码字的第 }i\text{ 个分量非零},1\leqslant i\leqslant n,i\neq j\} \tag{1}$$

根据上述讨论, $C^{\perp}$码中的 j-极小码字的集合于Γ_j是一一对应的. 因此, 确定 $C^{\perp}$码中的 j-极小码字的集合是非常重要的. 根据定义 3, 比起一般的线性码, 这对于极小线性码 $C^{\perp}$是容易做到的. 故本文考虑的是: 利用极小线性码的对偶码上的多级秘密共享, 来构建实现一般存取结构的抗内存泄漏的多级秘密共享方案.

1.2 物理不可克隆函数

物理不可克隆函数(physically unclonable functions, PUFs)是由 Pappu 等人于 2002 年首次提出的[32], 该噪音函数是由物理系统实现的. 当被称为外部激励 c 挑战时, PUFs 会产生一个唯一确定的响应 r=PUF(c), 该响应取决于输入 c 及 PUFs 本身的物理结构特性. PUFs 的输入/输出对(c,r)称为 PUFs 的激励/响应对. 以下是本文会用到的 PUFs 的较好性质.

- 噪声有界性. 对一个激励 $c\in\{0,1\}^{\lambda'}$, 询问同一个 PUF 两次会输出不同的响应 r_1、r_2. 但是, r_1 和 r_2 间的汉明距离小于一个固定的值 d, 即 $dis_{ham}(r_1,r_2)<d$;
- 不可克隆性. 给定一个 PUF, 对任意的激励 $c\in\{0,1\}^{\lambda'}$, 构造另一个 PUF′≠PUF 且 PUF(c)和 PUF′(c)的汉明距离小于 d 是计算不可行的;
- 不可预测性. 假设一个 PUF 已接受过多项式次数的激励 c 的挑战, 但对于一个新的激励, PUF 的响应仍具有很大的熵. 给定一个随机变量 Y, 我们定义 X 的平均条件最小熵为

$$\tilde{H}_\infty(X|Y)=-\log_2(E_{y\leftarrow Y}[\max_x\Pr[X=x|Y=y]]).$$

正式地，如果对于一个激励的集合 $C=(c_1,c_2,\ldots,c_l)$和一个新的激励 $c\notin C$ 满足 $dis_{ham}(c,c_k)\geqslant d'$，有平均条件最小熵 $\tilde{H}_\infty(\mathrm{PUF}(c)|\mathrm{PUF}(C))\geqslant m'$，其中，$\mathrm{PUF}(C)$表示一列随机变量 $\mathrm{PUF}(c_1),\mathrm{PUF}(c_2),\ldots,\mathrm{PUF}(c_l)$，则称该 $\mathrm{PUF}:\{0,1\}^{\lambda'}\to\{0,1\}^{n'}$是 m'-不可预测的.

我们用(λ',n',d,d',m')-PUF 表示一个不可克隆函数，其中，λ'和 n'分别表示输入和输出比特数，d 是噪音界限，d'是新的激励 $c\notin C$ 与集合 C 中的激励间的汉明距离的下界，响应的平均条件最小熵不小于 m'比特.

1.3 模糊提取器

在本文中，我们利用 PUFs 替代存储在非易失性内存中的长期密钥. 但是显然，PUF 是一个噪音函数，即同一激励会产生不同的响应，从而无法直接应用于密钥的产生. 为此，我们使用了 Dodis 等人[33]提出的模糊提取器(fuzzy extractors, FE). 模糊提取器可以从非均匀的输入信息ω中提取随机串 k，并能够通过帮助信息，从与ω接近的ω'中再次提取出相同的随机串 k.

定义 4(模糊提取器)[33]. 令 n'、l、d、m'是正整数. 一个(n',d,m')-模糊提取器包含一对算法(Gen,Rep).

- $\mathrm{Gen}:\{0,1\}^{n'}\to\{0,1\}^{l}\times\{0,1\}^{*}$. 这是概率性的密钥产生算法. 当输入一个 n'比特串ω时，则输出一个 l 比特均匀密钥随机串 k 和一个帮助信息 hd;
- $\mathrm{Rep}:\{0,1\}^{n'}\times\{0,1\}^{*}\to\{0,1\}^{l}$. 这是确定性的密钥重构算法. 当输入一个 n'比特串ω'和一个帮助信息时，该算法会输出一个 l 比特密钥随机串 k'.

模糊提取器的正确性确保当ω与ω'间的 Hamming 距离满足 $dis_{ham}(\omega,\omega')<d$ 时，则有 $k'=k$.

模糊提取器的安全性保证：当输入的比特串ω含有至少 m'比特的最小熵时，即使给定帮助信息 hd，所输出的随机串 k 与随机分布 U_l 也仍是计算不可区分的.

物理不可克隆函数与模糊提取器的联合作用即可产生密钥，同一个 PUF 作用于同一个激励时会产生不同的响应，但只要其差异足够小，便可利用模糊提取器进行处理，之后，还是能得到唯一确定的均匀随机密钥(如图 1 所示).

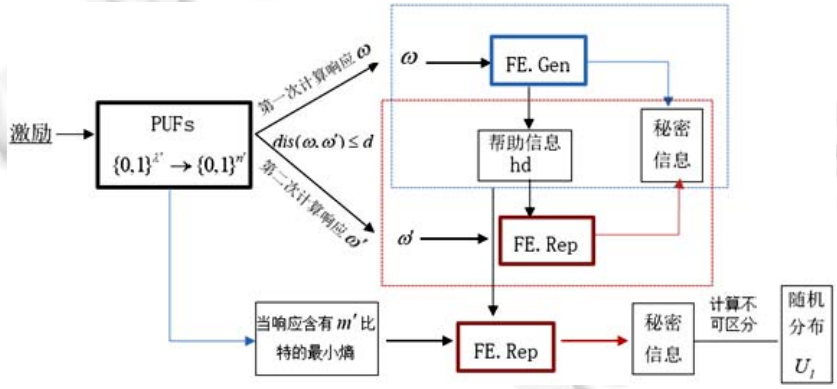


图 1 物理不可克隆函数与模糊提取器的结合机理

2 抗内存泄露的多级秘密共享的计算安全模型

在本节中，我们将详细描述一个秘密共享方案的内存泄露模型[34]，并正式提出抗内存泄露多级秘密共享的安全模型.

2.1 泄露模型

在现有的密码方案中，绝大部分方案的长期秘密信息均存储在非易失性存储器中. 因此，我们考虑如下的内存攻击者，该攻击者不仅能够腐化非授权集合中的参与者，而且还能获取其余未被腐化参与者的秘密信息的泄露信息.

2.2 抗内存泄露的计算安全模型

在一个多级秘密共享方案中, 分发者想要在 n 个参与者中根据 m 个存取结构$\Gamma_1,\ldots,\Gamma_m$分别共享 m 个秘密 $s_1,\ldots,s_m$. 下面, 我们将提出抗内存泄露的多级秘密共享的定义及基于选择秘密攻击下不可区分实验的计算安全模型.

2.2.1 抗内存泄露的多级秘密共享

一个多级秘密共享是一个四元组: Ω=(Stp,Dist,Sub,Rec), 包括 4 个算法.

- 初始化算法 Stp 是以安全参数λ、参与者集合 P 和 m 个存取结构$\Gamma_1,\ldots,\Gamma_m$为输入, i=2,…,m. 输出公共参数: $pms\leftarrow \mathrm{Stp}(1^{\lambda},P,\Gamma_1,\ldots,\Gamma_m)$;
- 分发算法 *Dist* 是以 *pms*、在参与者集合中共享的秘密 $s_1,\ldots,s_m$ 为输入, 输出秘密份额的集合 $\{sh_i\}_{P_i\in P}$ 以及一些公共的输出 out_{pub}, 即 $(out_{pub},\{sh_i\}_{P_i\in P})\leftarrow \mathrm{Dist}(pms,s_1,...,s_m)$;
- 伪份额生成算法 Sub 是以秘密份额的集合 $\{sh_i\}_{P_i\in P}$ 为输入, 依据所要恢复的不同秘密输出相应的伪份额, 即 $sub_{ij}\leftarrow \mathrm{Sub}(sh_i,j)$. 该算法主要对分发算法所输出的秘密份额不针对某个秘密的情形, 可隐藏于分发算法中;
- 重构算法 *Rec* 以 *pms*、out_{pub}、指标 $j\in\{1,\ldots,m\}$及某些子集 $A\subset P$ 中的参与者的份额 $\{sh_i\}_{P_i\in A}$ 为输入, 输出第 j 个秘密的可能取值, 即 $s'_j=\mathrm{Rec}(pms,out_{pub},j,\{sub_{ij}\}_{P_i\in A})$.

一个多级秘密共享Ω=(Stp,Dist,Sub,Rec)是一个实现存取结构$\Gamma_1,\ldots,\Gamma_m$ 的θ-抗内存泄露秘密共享, 如果满足下面两个条件.

(1) 正确性要求: 对任意一个 $j\in\{1,\ldots,m\}$及任意授权集 $A\in\Gamma_j$ 中的所有参与者 P_i 合作可正确恢复秘密 s_j, 即 $\Pr[\mathrm{Rec}(pms,out_{pub},j,\{sub_{ij}\}_{P_i\in A})=s_j]>1-negl(\lambda)$;

(2) 安全性要求: 对某些 $j(1\leqslant j\leqslant m)$, 腐化了非授权集 $B\notin\Gamma_j$的任何多项式时间的θ-非易失性内存攻击者 $\mathcal{A}'$均无法以不可忽略的概率成功获得第 j 个秘密 s_j 的任何信息.

2.2.2 计算安全模型

抗内存泄露的多级秘密共享Ω=(Stp,Dist,Rec)的计算安全性的模型是利用一个在多项式时间的内存攻击者$\mathcal{A}$和一个挑战者之间的游戏$\mathcal{G}$来刻画的, 具体如下.

- 初始化

攻击者$\mathcal{A}$选择并公布参与者的集合以及存取结构$\Gamma_1,\ldots,\Gamma_m$.

- 建立

挑战者运行参数建立过程 $pms\leftarrow \mathrm{Stp}(1^{\lambda},P,\{\Gamma_j\}_{1\leqslant j\leqslant m})$, 而后将 *pms* 发送给攻击者$\mathcal{A}$. 攻击者$\mathcal{A}$选择一个被挑战的参与者集合 $\tilde{B}\subset P$. 同时, 攻击者可以询问预言机$\mathcal{O}$, 该预言机输入的是$|C|$个自适应选择的多项式大小的泄露函数 $L_{i_v}(\cdot)$, 输出的是泄露份额 $L_{i_v}(S_{i_v})$, 其中, $1\leqslant i_v\leqslant n$, $1\leqslant v\leqslant|C|$, $C=\{P_{i_1},...,P_{i_{|C|}}\}=P/\tilde{B}$, S_{i_v} 是参与者 P_{i_v} 存储在其非易失性内存中的秘密信息. 攻击者$\mathcal{A}$提交两个长度相同的秘密 $\boldsymbol{s}^0=(s_1^0,...,s_m^0)\neq(s_1^1,...,s_m^1)=\boldsymbol{s}^1$, 且要求如果 $\tilde{B}\in\Gamma_j$, 则对任意 $j\in\{1,\ldots,m\}$, 有 $s_j^0=s_j^1$.

- 预备阶段

攻击者发出询问请求, 以获取被挑战参与者集合里参与者的份额.

- 挑战

挑战者随机选取 $b\in\{0,1\}$并运行分发算法 $(out_{pub},\{sh_i\}_{P_i\in P})\leftarrow \mathrm{Dist}(pms,\boldsymbol{s}^b)$, 最后将 $(out_{pub},\{sh_i\}_{P_i\in\tilde{B}})$ 发送给攻击者$\mathcal{A}$. 此时, 攻击者可以继续询问预言机$\mathcal{O}$.

- 猜测

攻击者$\mathcal{A}$输出对 b 的猜测值 b'.

定义攻击者$\mathcal{A}$打破抗泄露多秘密共享方案Ω的优势为

$$Adv_{\mathcal{A}_{\Omega}}(\lambda)=\left|\Pr[b'=b]-\frac{1}{2}\right|.$$

如果对任何多项式时间的攻击者$\mathcal{A}$, $Adv_{\mathcal{A}_{\Omega}}(\lambda)$是一个关于$\lambda$可忽略的函数, 那么就称该抗内存泄露的多级秘密共享方案Ω=(Stp,Dist,Rec)是计算安全的.

3 完全抗内存泄漏的可验证多级秘密共享方案

如果在完全内存攻击者存在的情况下, 多秘密共享方案是安全的, 则我们称这种方案是完全抗内存泄露的多秘密共享方案[30]. 为了设计一个这样的方案, 我们不应将长期秘密信息存储在每个参与者的非易失性存储器中. 在本节中, 我们利用 PUFs 和模糊提取器来生成方案的秘密份额, 这样可以防止完全内存攻击者通过查询泄露预言机来获取参与者的秘密信息.

3.1 初始化阶段: $\boldsymbol{\Omega_1}$.Stp(1^{λ},$\boldsymbol{P}$,$\boldsymbol{\Gamma_1}$,…,$\boldsymbol{\Gamma_n}$)

令 $P=\{P_1,\ldots,P_n\}$为 n 个参与者的集合, $q>n$ 是一个大素数, 且长度至少是λ比特. 设 C 是一个生成矩阵为 $G=(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n)_{k\times n}$的$[n,k;q]$线性码, C 的对偶码是极小的, 矩阵 $\boldsymbol{G}$ 中的列向量 $\boldsymbol{g}_j$ 是其余列向量的线性组合, 其中, $1\leqslant j\leqslant n$. 每个参与者均拥有一个不可克隆函数$(\log q,\log q,d,d',m')$-PUF_i: $\{0,1\}^{\log q}\to\{0,1\}^{\log q}$, 该函数基于物理系统来实现单向输出, 每个 PUF_i 能够根据激励信号与自身随机性确定唯一的一个响应. 诚实的分发者 D 想要根据存取结构$\Gamma_1,\Gamma_2,\ldots,\Gamma_n$在参与者集合 P 中共享 n 个秘密 $s_1,s_2,\ldots,s_n$. 令 x_i 为每个参与者持有的公开的身份信息, 其满足 $dis_{ham}(x_i,x_j)>d'$(不妨设 $d'=1$). 设 FE=(FE.Gen,FE.Rep)表示一个$(\log q,d,m')$模糊提取器. 一个哈希函数 $H_1: N\times F_q^*\to F_q$ 及一个哈希函数 H_2:$\{0,1\}^q\to\{0,1\}^{\mu(\lambda)}$. 设公开参数为 $pms=(q,x_i,P,G,H_1,H_2,\Gamma_1,\ldots,\Gamma_n)$.

3.2 份额生成阶段: $\boldsymbol{\Omega_1}$.Dist($\boldsymbol{pms}$,$\boldsymbol{s_1}$,…,$\boldsymbol{s_n}$)

每个参与者 P_i 计算 $r_i=\mathrm{PUF}_i(x_i)$, $(sh_i,hd_i)\leftarrow\mathrm{FE.Gen}(r_i)$, 并且将 sh_i 秘密发送给分发者 D, 而后, P_i 在其内存中保存帮助信息 hd_i, $1\leqslant i\leqslant n$. 在得到所有的份额$(sh_1,sh_2,\ldots,sh_n)$之后, D 按照以下步骤在 n 个参与者中共享 n 个秘密 $(s_1,s_2,\ldots,s_n)\in F_q^n$.

(1) 随机选择一个向量 $\boldsymbol{u}_j=(u_{j1},u_{j2},\ldots,u_{jk})\in F_q^k$ 且其满足 $\boldsymbol{u}_j\boldsymbol{g}_j=s_j$, $1\leqslant j\leqslant n$;

(2) 计算 $h_{ij}=H_1(j,sh_i)$及 $r_{ij}=\boldsymbol{u}_j\boldsymbol{g}_i-h_{ij} \bmod q$, 其中, $i\neq j$;

(3) 计算并公布 $l_{ij}=H_2(H_1(j,sh_i))$, 其中, $i\neq j$;

(4) 公开输出 $out_{pub}=\{(i,j,r_{ij}),l_{ij}\}_{P_i\in P,\,j\in\{1,\ldots,n\},\,i\neq j}$.

3.3 秘密重构阶段: $\boldsymbol{\Omega_1}$.Rec($\boldsymbol{pms}$,$\boldsymbol{out_{pub}}$,$\boldsymbol{j}$,$\{\boldsymbol{sub_{ij}}\}_{\boldsymbol{P_i\in A}}$)

假设授权集 $A=\{P_{i_1},P_{i_2},\ldots,P_{i_c}\}\in\Gamma_j$ 中的参与者想要一起重构秘密 s_j.

(1) 每一个参与者 $P_{i_1},P_{i_2},\ldots,P_{i_c}$ 计算 $r'_{i_v}=\mathrm{PUF}_{i_v}(x_{i_v}),(sh'_{i_v})\leftarrow\mathrm{FE.Rep}(r'_{i_v},hd'_{i_v})$, 而后, 每个参与者计算并将各自的伪份额 $H_1(j,sh'_{i_v})$ 秘密发送给指定的秘密生成者 DC(DC 可以是 A 中的某个参与者, 也可以是其他某个人), $1\leqslant v\leqslant c$.

- 验证阶段: DC 在收到 A 中参与者 P_{i_v} 发送的伪份额后, 再从公告牌上下载 $l_{i_v,j}$, 并通过计算 $l_{i_v,j}=H_2(H_1(j,sh'_{i_v}))$ 是否成立来验证 P_{i_v} 是否存在欺诈行为.

(2) 若无欺诈, DC 从公告牌上下载 $\{(i_v,j,r_{i_v,j})\}_{P_{i_v}\in A}$ 并计算 $t'_{ji_v}=r_{i_v,j}+H_1(j,sh'_{i_v}) \bmod q$. 而后, 再从公告牌下载线性码 C 的生成矩阵 $G=(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n)_{k\times n}$, 计算满足 $\boldsymbol{g}_j=\sum_{\substack{v=1\\P_{i_v}\in A}}^{c}x_v\boldsymbol{g}_{i_v}$ 的系数.

(3) 根据线性码上秘密共享方案中秘密重构算法, DC 计算 $s'_j=\sum_{\substack{v=1\\P_{i_v}\in A}}^{c}x_vt'_{ji_v}$.

4 方案分析

4.1 存取结构

由于该方案最终重构秘密是基于极小线性码的对偶码上的多级秘密共享, 因此方案的存取结构满足以下定理.

定理 1. 设 C 是一个生成矩阵为 $G=(\boldsymbol{g}_1,\ldots,\boldsymbol{g}_n)_{k\times n}$ 的$[n,k;q]$极小线性码, 则基于 $C^{\perp}$上的多级秘密共享的存取结构是 n 元的$\Gamma_1,\Gamma_2,\ldots,\Gamma_n$, 且每个$\Gamma_j$中有 q^{k-1} 个极小授权子集, $1\leqslant j\leqslant n$.

证明: 设 C 的生成矩阵为

$$G=\begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1,n-1} & g_{1,n} \\ g_{21} & g_{22} & \cdots & g_{2,n-1} & g_{2,n} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ g_{k1} & g_{k2} & \cdots & g_{k,n-1} & g_{k,n} \end{bmatrix},$$

且根据定义 3, 生成矩阵的每个列向量均为非零向量, 即 $\boldsymbol{g}_j\neq\boldsymbol{0}$. 又因为线性码 C 中的每个码字均是矩阵 G 的行向量的线性组合, 故 C 中的码字 $\boldsymbol{c}=(c_1,c_2,...,c_n)\in F_q^n$ 的第 j 个分量 $c_j=a_1g_{1,j}+a_2g_{2,j}+\ldots+a_kg_{k,j}$. 当 $\boldsymbol{a}=(a_1,a_2,\ldots,a_k)$ 遍历 F_q^k 时, C 中有 q^{k-1} 个码字的第 j 个分量使得$\boldsymbol{\alpha}\cdot\boldsymbol{g}_j=0$ 成立. 那么, 极小线性码 C 中共有 q^k-q^{k-1} 个码字的第 j 个分量是非零的. 又因为根据定义 3, 极小线性码 C 中每个非零码字都是极小向量, 所以, 对于每个 $1\leqslant j\leqslant n$, C 中的 j-极小码字共有$(q^k-q^{k-1})/(q-1)=q^{k-1}$, 故 $C^{\perp}$上的多级秘密共享的 n 元存取结构中的每个Γ_j里有 q^{k-1} 个极小授权子集, 其中, $1\leqslant j\leqslant n$. □

例 1: 由文献[35]可知, 线性码 $\overline{C}=\{Tr_{3^4/3}(\beta),Tr_{3^4/3}(\beta\theta),...,Tr_{3^4/3}(\beta\theta^{19})\mid\beta\in F_{3^4},\theta^{20}\in F_3\}$ 是一个[20,4;3]极小线性码, 其中, Tr 是域 F_{3^4} 到 F_3 的迹函数. 根据迹函数的定义[36]及公式(1), 可以得出基于码 $\overline{C}^{\perp}$ 上的参与者人数共 20 的多级秘密共享方案的多级存取结构Γ_j (1≤j≤20). 下面以可重构秘密 s_1 的存取结构Γ_1为例, 给出其中全部的 27 个授权子集.

{2,3,4,5,6,9,11,12,13,15,17,18,19,20},
{2,3,4,5,8,10,11,12,14,16,17,18,19,20},
{5,6,8,9,10,13,14,15,16,18,19},
{2,3,6,8,9,10,12,14,15,16,17,18,19,20}
{3,4,5,8,9,10,11,13,14,16,20},
{3,4,5,7,9,10,11,12,13,14,15,16,17,18},
{2,4,6,7,8,9,10,11,12,13,14,15,18,20},
{4,5,6,7,9,10,12,16,17,19,20},
{3,7,8,10,11,12,15,16,17,18,20},
{2,3,5,6,8,12,13,15,16,17,20},
{2,4,5,7,11,12,14,15,16,19,20},
{2,3,4,5,6,7,8,9,12,14,15,16,18,20},
{3,4,6,10,11,13,14,15,18,19,20},
{2,3,4,5,6,7,8,11,13,14,15,17,19,20}.
{2,3,4,5,6,7,10,12,13,14,16,18,19,20},
{2,4,8,9,11,12,13,16,17,18,19},
{2,3,4,6,7,9,13,14,16,17,18},
{2,3,4,7,9,10,11,13,15,16,17,18,19,20},
{2,5,7,8,9,11,13,14,15,16,17,18,19,20},
{2,4,5,6,9,10,11,12,14,15,17},
{4,6,7,8,10,12,13,14,15,16,17,18,19,20},
{2,3,6,7,8,9,11,12,14,18,19},
{2,3,5,7,8,9,10,11,12,13,14,15,16,19},
{2,5,6,7,8,10,11,13,17,18,20},
{3,5,6,7,8,9,10,11,12,13,14,17,19,20},
{3,4,5,6,7,8,9,10,11,12,15,17,18,19},
{2,3,4,5,6,7,8,9,10,13,15,16,17,19}.

4.2 正确性

定理 2. 对于 $1\leqslant j\leqslant n$, 方案Ω_1的每级存取结构Γ_j中的任一授权集里的所有参与者合作可正确重构出其对应的秘密 s_j, 即 $s'_j=s_j$.

证明: 令 $\{P_{i_1},P_{i_2},...,P_{i_c}\}$ 是存取结构Γ_j 的一个授权集. 根据不可克隆函数 $(\log q,\log q,d,d',m')$-PUF_{i_v} 的噪音上界的性质, $dis_{ham}(r_{i_v},r'_{i_v})\leqslant d$, 其中, $1\leqslant v\leqslant c$. 又根据$(\log q,d,m')$模糊提取器的正确性要求, 可得 $sh_{i_v}=sh'_{i_v}$,

从而 $t'_{ji_v}=\boldsymbol{u}_j\boldsymbol{g}_{i_v}$，故有 $s'_j=\sum_{\substack{v=1\\P_{i_v}\in A}}^{c}x_v\boldsymbol{u}_j\boldsymbol{g}_{i_v}=\sum_{\substack{v=1\\P_{i_v}\in A}}^{c}\boldsymbol{u}_jx_v\boldsymbol{g}_{i_v}=\boldsymbol{u}_j\boldsymbol{g}_j=s_j$.

因此，存取结构Γ_j的授权集中的参与者可以利用各自的伪份额 $H(j,sh'_{i_v})$ 正确恢复出秘密 s_j，即 $s'_j=s_j$. □

4.3 安全性

定理 3. 在随机预言模型下，方案Ω_1是一个选择秘密攻击下不可区分的计算安全的完全抗内存泄漏的多级秘密共享方案.

证明：令$\mathcal{A}_1$是计算安全的抗内存泄露的多级秘密共享方案Ω_1的选择秘密攻击的不可区分实验的攻击者，令 CH_1是该安全游戏的挑战者. 具体证明如下.

- 初始化

$\mathcal{A}_1$通过选择并公布参与者集合 $P=\{P_1,P_2,\ldots,P_n\}$及 n 个存取结构$\{\Gamma_1,\Gamma_2,\ldots,\Gamma_n\}$发起游戏$\mathcal{G}$.

- 建立

挑战者 CH_1运行参数建立过程 $pms\leftarrow Stp(1^\lambda,P,\Gamma_1,\ldots,\Gamma_n)$，最后将 pms 发送给攻击者$\mathcal{A}_1$. 之后，$\mathcal{A}_1$公布一个被挑战的参与者集合 $\tilde{B}\subset P$，其中，$|\tilde{B}|=d$，不失一般性，不妨设 $\tilde{B}=\{P_{i_1},P_{i_2},...,P_{i_d}\}$.

令 $J^*=\{j\in\{1,...,n\}$且满足$s_j^0=s_j^1\}$，容易看出：如果 $\tilde{B}\in\Gamma_j$，则有$j\in J^*$. 令 $C=P/\tilde{B}=\{P_{j_1},P_{j_2},...,P_{j_{n-d}}\}$，由方案可知，每个参与者在其非易失性的内存里都会存储一个帮助信息 hd_i，且攻击者$\mathcal{A}_1$能够计算 $sh_{i_1},...,sh_{i_d}$. 由于攻击者$\mathcal{A}_1$可以进行完全内存攻击，因此他可以通过询问预言机$\mathcal{O}$获得 $(hd_{j_1},...,hd_{j_{n-d}})$. 但是根据物理不可克隆函数 PUF 的不可预测性以及模糊提取器 FE 的安全性，攻击者$\mathcal{A}_1$不能以不可忽略的区分真正的份额 sh_{i_κ} 和均匀随机分布 U_l(设份额长度是 l)，其中，$1\leqslant\kappa\leqslant n-d$. 最后，攻击者$\mathcal{A}_1$输出两个不同的多秘密 $\boldsymbol{s}^0=(s_1^0,...,s_n^0)\neq(s_1^1,...,s_n^1)=\boldsymbol{s}^1$，且满足 $|s_j^0|=|s_j^1|$，$j=1,\ldots,n$.

- 预备阶段

如果对于 $j\in\{1,\ldots,n\}$，$j\notin J^*$，且 $x\in\{sh_i\}_{P_i\notin\tilde{B}}$，攻击者$\mathcal{A}_1$向随机预言模型 H_1 询问(j,x)对应的哈希函数值，则中止游戏；否则，随机选择 F_q 中的一个元素 h_1 回答询问，并将此关系 $H_1(j,x)=h_1$ 存储于哈希表 1 中，然后将 h_1 发送给$\mathcal{A}_1$. 如果哈希表 1 中已经存在$\mathcal{A}_1$的哈希查询 $h'_1(j,x)$，则将存储的值 h_1 直接发送回给$\mathcal{A}_1$. 同理，攻击者$\mathcal{A}_1$向随机预言模型 H_2 询问 h'_1 对应的哈希函数值，随机选择 Z_q 中的一个元素 h_2 回答询问，并将此关系 $H_2(h'_1)=h_2$ 存储于哈希表 2 中，然后将 h_2 发送给$\mathcal{A}_1$. 如果哈希表 2 中已经存在$\mathcal{A}_1$的哈希询问的哈希值，则将存储的值 h_2 直接发送回给$\mathcal{A}_1$.

- 挑战

$\mathcal{A}_1$ 向游戏$\mathcal{G}$的挑战者 CH_1 提交两个向量组 $\boldsymbol{s}^0=(s_1^0,...,s_n^0)$ 及 $\boldsymbol{s}^1=(s_1^1,...,s_n^1)$，其中，对于所有的 $j\in J^*$，有 $s_j^{(0)}=s_j^{(1)}$. 对所有的 $j\in J^*$，CH_1随机选择向量 $\boldsymbol{u}_j=(u_1,...,u_k)\in F_q^k$，使得 $\boldsymbol{u}_j\boldsymbol{g}_j=s_j^{(0)}$. 之后，$CH_1$依据预备阶段查询哈希表 1、表 2 的过程，可计算 $h_{ij}=H_1(j,sh_i)$、$l_{ij}=H_2(H_1(j,sh_i))$以及 $r_{ij}=\boldsymbol{u}_j\boldsymbol{g}_i-h_{ij}\bmod q$，其中，$j\in J^*$, $P_i\in P$, $i\neq j$.

对于其余的$j\in\{1,\ldots,n\}$，$j\notin J^*$, CH_1随机选择 $\boldsymbol{u}_j^{(0)},\boldsymbol{u}_j^{(1)}\in F_q^k$，使得 $\boldsymbol{u}_j^{(0)}\boldsymbol{g}_j=s_j^{(0)}$, $\boldsymbol{u}_j^{(1)}\boldsymbol{g}_j=s_j^{(1)}$，且对所有的 $P_i\in\tilde{B}$ 及 $i\neq j$，有 $\boldsymbol{u}_j^{(0)}\boldsymbol{g}_i=\boldsymbol{u}_j^{(1)}\boldsymbol{g}_i$. 对满足 $P_i\in\tilde{B}$ 的角标 i, CH_1依据预备阶段查询哈希表 1、表 2 的过程，计算 $h_{ij}=H_1(j,sh_i)$、$l_{ij}=H_2(H_1(j,sh_i))$以及 $r_{ij}=\boldsymbol{u}_j^{(0)}\boldsymbol{g}_i-h_{ij}\bmod q$；对 $P_i\notin\tilde{B}$ 的角标 i, CH_1随机选择 $r_{ij}\in F_q$，上述出现的 i 和 j 角标均有 $i\neq j$. CH_1 公开输出 $Out_{put}=\{(i,j,r_{ij})_{P_i\in P,j\in\{1,...,n\},i\neq j},(l_{ij})_{P_i\in P,j\in J^*,i\neq j},(l_{ij})_{P_i\in B,j\notin J^*,i\neq j}\}$ 给攻击者$\mathcal{A}_1$. 对 $j\in\{1,\ldots,n\}$，$j\notin J^*$，及满足 $P_i\notin\tilde{B}$ 的角标 i, CH_1定义 $h_{ij}^{(0)}=r_{ij}-\boldsymbol{u}_j^{(0)}\boldsymbol{g}_i\bmod q$ 与 $h_{ij}^{(1)}=r_{ij}-\boldsymbol{u}_j^{(1)}\boldsymbol{g}_i\bmod q$，其中，$i\neq j$. CH_1选择一个随机比特$\beta\in\{0,1\}$，并将 $H_1(j,sh_i)=h_{ij}^{(\beta)}$ 以及 $l_{ij}=H_2(h_{ij}^{(\beta)})$ 分别包含在哈希表 1、表 2 中，其中，$j\in\{1,\ldots,n\}$, $j\notin J^*$, $i\neq j$ 且 $P_i\notin\tilde{B}$. 这样，CH_1完美地模拟了秘密 $\boldsymbol{s}^{(\beta)}=(s_1^{(\beta)},...,s_m^{(\beta)})$ 份额分发的过程.

- 猜测

对于 $j\in\{1,\ldots,n\}$, $j\notin J^*$ 且 $P_i\notin\tilde{B}$ 所对应的 i 和 j, 只要 $\mathcal{A}_1$ 不进行哈希询问 $H_1(j,sh_i)$, 得到的信息与其分享秘密 $s^{(1-\beta)}$ 所得到的是一样的, 其中, $i\neq j$.

所以, 为了计算 $\mathcal{A}_1$ 猜出所要共享的秘密的概率, 我们需根据 $\mathcal{A}_1$ 是否进行哈希询问 $H_1(j,sh_i)$ 来分两种情况讨论, 其中, $j\in\{1,\ldots,n\}$, $j\notin J^*$, $i\neq j$ 且 $P_i\notin\tilde{B}$. 如果其进行了哈希查询, 且设这种情况发生的概率为 δ, 那么这是 $\mathcal{A}_1$ 遇到的最好的情况, 因为此时他总能猜出正确的秘密. 另一方面, 如果 $\mathcal{A}_1$ 不进行这样的哈希查询, 设此情况发生的概率为 $1-\delta$, 则 $\mathcal{A}_1$ 能够正确猜测出秘密的概率恰好是 1/2. 由以上分析可知, $\mathcal{A}_1$ 能够猜出正确秘密的概率最多为 $\delta+1/2(1-\delta)$. 因此, $\mathcal{A}_1$ 的优势是

$$Adv_{\mathcal{A}_\Omega}(\lambda)=\left|\Pr[b'=b]-\frac{1}{2}\right|\leqslant\left|\delta(\lambda)+1/2(1-\delta(\lambda))-\frac{1}{2}\right|=\frac{1}{2}\delta(\lambda).$$

令 E_1 是 $\mathcal{A}_1$ 进行哈希询问 $H_1(j,sh_i)$ 事件, 其中, $j\in\{1,\ldots,n\}$, $j\notin J^*$, $i\neq j$ 且 $P_i\notin\tilde{B}$. 不妨设 Q_{H_1} 是 $\mathcal{A}_1$ 进行哈希 H_1 询问的次数, 则有:

$$\Pr[E_1]=1-\left(1-\frac{n-d}{q-d}\right)\left(1-\frac{n-d}{q-d-1}\right)\ldots\left(1-\frac{n-d}{q-d-Q_{H_1}+1}\right)<1-\left(1-\frac{n-d}{q}\right)^{Q_{H_1}}<\frac{Q_{H_1}(n-d)}{q}\leqslant\frac{nQ_{H_1}}{q}.$$

由于 $q>2^\lambda$, 故 $\Pr[E_1]<\dfrac{nQ_{H_1}}{2^\lambda}$.

令 E_2 是从公开的 l_{ij} 中猜测 $H_1(j,sh_i)$ 值的事件, 其中, $j\in\{1,\ldots,n\}$, $j\notin J^*$ 且 $P_i\notin\tilde{B}$. 由于 $H_2:\{0,1\}^q\to\{0,1\}^{\mu(\lambda)}$ 且令 Q_{H_2} 是 $\mathcal{A}_1$ 进行哈希 H_2 询问的次数, 则有 $\Pr[E_2]=1-\left(1-\dfrac{1}{2^{\mu(\lambda)}}\right)^{Q_{H_2}}\leqslant\dfrac{Q_{H_2}}{2^{\mu(\lambda)}}$.

综上, $\delta(\lambda)=\Pr[E_1\cup E_2]\leqslant\Pr[E_1]+\Pr[E_2]\leqslant\dfrac{nQ_{H_1}}{2^\lambda}+\dfrac{Q_{H_2}}{2^{\mu(\lambda)}}$, 故 $Adv_{\mathcal{A}_\Omega}(\lambda)=\left|\Pr[b'=b]-\dfrac{1}{2}\right|\leqslant\dfrac{nQ_{H_1}}{2^{\lambda+1}}+\dfrac{Q_{H_2}}{2^{\mu(\lambda)+1}}$.

由此可知, 只要 q 足够大, 本文所提出的抗内存泄露的多秘密共享方案在随机预言模型下是计算性可证明安全的. □

注: 在定理 3 中, 因为参与者人数通常比哈希询问次数小, 所以 $\delta(\lambda)$ 的上界的分子主要项可取 $Q_{H_1}\cdot Q_{H_2}$. 密码学中, 哈希查询的数量通常估计为 $Q_H\leqslant 2^{60}$, 因此, 当 $2^\lambda\geqslant 2^{200}$ 及 $2^{\mu(\lambda)}\geqslant 2^{200}$ 时, 本文提出的抗泄露多秘密共享方案则满足 80 位的安全级别, 即 $Adv_{\mathcal{A}_\Omega}(\lambda)<2^{-80}$.

4.4 性能分析

本节将所提出的多级秘密共享方案与文献[15,17,23,27,30]中的方案在性能方面进行了比较, 结果见表 1, 其中, SM 表示标准模型, ROM 表示随机预言模型.

表 1 多秘密共享方案的性能比较

性能	文献[17]方案	文献[23]方案	文献[15]方案 2	文献[27]方案	文献[30]方案 2	本文方案
多秘密	是	是	是	是	是	是
一般化的存取结构	是	是	是	否	否	是
可验证性	是	无	是	否	是	是
插值运算	无	无	无	有	有	有
模指数运算	有	无	无	无	有	无
安全模型	SM	–	–	SM	–	ROM
秘密份额多用	是	否	否	否	是	是
选择秘密攻击是不可区分的	是	–	–	是	–	是
抗内存泄漏的多秘密	否	否	否	否	是	是

下面我们将具体讨论本方案的一些重要性质.

(1) 抗泄露. 本方案是完全抗内存泄露的, 且在随机预言模型下, 方案对选择秘密攻击是不可区分的;

(2) 多用性. 因每个参与者计算伪份额 $H_1(j,sh_i)$, 而后利用伪份额恢复秘密, 所以在我们的方案中, 即使是在验证和重构阶段之后, 也永远不会暴露每个参与者真实的份额. 由此可知, 本文所提方案具有

多用性;

(3) 防欺诈. 由于秘密生成者 DC 在验证阶段会检验参与者份额的真实性, 因此, 每个参与者的欺骗行为都不可能成功;

(4) 现有的大部分多秘密共享的存取结构是门限策略, 这种情况适用于参与者权力都相同的情况, 具有一定的局限性. 本方案突破了门限策略带来的约束, 实现了基于线性码上一般的存取结构.

另外, 在方案的计算复杂度方面, 我们将所提方案与文献[15,17,23]中一般存取结构上的多秘密方案进行了比较, 结果见表 2. 在本方案中, 初始化阶段共运行物理不可克隆函数 n 次; 分发阶段中, 分发者计算 nk 次乘法, $2n(n-1)$次哈希函数运算以及 $n(n-1)$次加法运算; 验证阶段, 秘密生成者 DC 进行了 1 次哈希函数运算; 重构阶段中, 设$|A|$是恢复秘密的授权子集中参与者人数, DC 需要计算$|A|$次哈希函数、$|A|$次加法运算以及$|A|$次物理不可克隆函数及 1 次逆运算. 容易看出, 本文方案实现了用较低的复杂性达到完全的抗内存泄露的特性. 表 2 中, T_H 是哈希函数运算 1 次的时间, T_{owf} 是单向函数运算 1 次的时间, T_f 是物理不可克隆函数运算 1 次的时间, T_{ep} 是 1 次多线性映射运算的时间, T_m 是 1 次乘法运算的时间, T_a 是 1 次加法运算的时间, T_e 是 1 次模指数运算的时间, T_i 是 1 次逆运算的时间; n 是参与者人数, m 是要重构的秘密的数目, $|A|$是恢复秘密的授权子集中参与者的人数, l 是 MSP 中矩阵 A 的列数, k 是线性码生成矩阵的行数.

表 2 一般存取结构上多秘密共享方案计算复杂度对比

性能	文献[15]方案 2	文献[17]方案	文献[23]方案	本文方案
初始化	nT_e	nT_{owf}	–	nT_f
分发阶段	$nlT_m+n(l-1)T_a+lT_{ep}$	$nlT_m+n(1+n)T_e$	nlT_m	$(n^2-n)(T_a+2T_H)+nkT_m$
实验阶段	$(2l+1)T_{ep}$	$l(T_e+T_m)$	–	T_H
重构阶段	$\|A\|(nl+1)T_m+(\|A\|n(l-1)+2\|A\|-1)T_a+2T_{ep}$	$\|A\|T_e+2T_i$	$n\|A\|T_m$	$\|A\|(T_H+T_a+T_f)+T_i$

5 结 论

本文首次定义了抗内存泄漏的多级秘密共享的形式化的计算安全模型. 然后, 利用物理不可克隆函数以及模糊提取器联合产生密钥的原理, 构造了一个完全抗内存泄露的多级秘密共享方案. 该方案具有 4 个重要特性: (1) 方案是基于极小线性码的对偶码上多级秘密共享体系上建立的, 故其存取结构容易确定且较门限的方案更具一般性; (2) 最终的秘密是在参与者利用哈希函数生成的伪份额的帮助下重构的, 故其具有多用性; (3) 在秘密重构阶段, 能够验证参与者是否有欺诈行为; (4) 方案是计算可证明安全的. 因此, 本文的方案拓宽了多秘密共享的应用范围, 具有更好的应用前景.

References:

[1] Shamir A. How to share a secret. Communications of the ACM, 1979, 22: 612−613.

[2] Blakley GR. Safeguarding cryptographic keys. Proc. of AFIPS NCC, 1979, 48: 313−317.

[3] Zhang YS, Li WJ, Chen L, Bi W, Yang T. Verifiable special threshold secret sharing scheme based on eigenvalue. Journal on Communications, 2018, 39(8): 169−175 (in Chinese with English abstract).

[4] Tan ZH, Yang GM, Wang XW, Cheng W, Ning JY. Multidimensional spherical threshold secret sharing scheme for cloud storage. Ruan Jian Xue Bao/Journal of Software, 2016, 27(11): 2912−2928 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4943.htm [doi: 10.13328/j.cnki.jos.004943]

[5] Meng KJ, Miao FY, Huang WC, *et al*. Threshold changeable secret sharing with secure secret reconstruction. Information Processing Letters, 2020, 157: 105928.

[6] Liu H, Li XH, Tian YL, *et al*. Rational fair secret sharing scheme. Chinese Journal of Computers, 2020, 43(8): 1517−1533 (in Chinese with English abstract).

[7] Ito M, Saito A, Nishizeki T. Secret sharing schemes realizing general access structures. In: Proc. of the IEEE Global Telecommunications Conf. 1987. 99−102.

[8] Massey JL. Some applications of coding theory in cryptography. In: Proc. of the Cryptography and Coding IV. Formara Ltd., 1995. 33−47.

[9] Stinson DR. An explication of secret sharing schemes. Designs Codes & Cryptography, 1992, 2(4): 357−390.

[10] Brickell EF. Some ideal secret sharing schemes. Journal of Combinatorial Mathematics & Combinatorial Computing, 1989, 9(6): 105−113.

[11] Lin C, Hu H, Chang CC, *et al*. A publicly verifiable multi-secret sharing scheme with outsourcing secret reconstruction. In: Proc. of the IEEE Access. 2018. 1.

[12] Kabirirad S, Eslami Z. Improvement of (*n*,*n*)-multi-secret image sharing schemes based on boolean operations. Journal of Information Security and Applications, 2019, 47: 16–27.

[13] Zhang BH, Tang YS. On the construction and analysis of verifiable multi-secret sharing based on non-homogeneous linear recursion. Journal of Information Science and Engineering, 2018, 34(3): 749−763.

[14] Miao F, Wang L, Ji Y, *et al*. GOMSS: A simple group oriented (*t*,*m*,*n*) multi-secret sharing scheme. Chinese Journal of Electronics, 2017, 26(3): 557−563.

[15] Dehkordi MH, Oraei H. How to construct a verifiable multi-secret sharing scheme based on graded encoding schemes. IET Information Security, 2019, 13(4): 343−351.

[16] Li J, Wang X, Huang Z, *et al*. Multi-level muti-secret sharing scheme for decentralized e-voting in cloud computing. Journal of Parallel and Distributed Computing, 2019, 130: 91–97.

[17] Mashhadi S, Dehkordi MH, Kiamari N. Provably secure verifiable multi-stage secret sharing scheme based on monotone span program. IET Information Security, 2017, 11(6): 326−331.

[18] Song Y, Li ZH, Li YM, *et al*. A new multi-use multi-secret sharing scheme based on the duals of minimal linear codes. Security and Communication Networks, 2015, 8(2): 202−211.

[19] Basit A, Chanakya P, Venkaiah VC, *et al*. New multi-secret sharing scheme based on super increasing sequence for level-ordered access structure. Int'l Journal of Communication Networks and Distributed Systems, 2020, 24(1): 1.

[20] Zhang J, Zhang F. Information-theoretical secure verifiable secret sharing with vector space access structures over bilinear groups and its applications. Future Generation Computer Systems, 2015, 52: 109−115.

[21] Harn L. Unconditionally secure verifiable secret sharing scheme. Advances in Information Sciences & Service Sciences, 2012, 4(17): 514−518.

[22] Krawczyk H. Secret sharing made short. In: Proc. of the Crypto'93. LNCS 773, Springer, 1993. 136−146.

[23] Hsu CF, Harn L, Cui G. An ideal multi-secret sharing scheme based on connectivity of graphs. Wireless Personal Communications, 2014, 77(1): 383−394.

[24] Dehkordi MH, Mashhadi S, Oraei H. A proactive multi stage secret sharing scheme for any given access structure. Wireless Personal Communications, 2019, 104: 491−503.

[25] Lin CL, Yan XF, Niu QW, *et al*. Cheating immune multi-secret sharing without predefined order of secrets. Journal of the Chinese Institute of Engineers, 2019, 42(1): 15−19.

[26] Zhang T, Ke X, Liu Y. (*t*,*n*) multi-secret sharing scheme extended from Harn-Hsu's scheme. Eurasip Journal on Wireless Communications and Networking, 2018(1): 1−4.

[27] Herranz J, Ruiz A, Saez G. Sharing many secrets with computational provable security. Information Processing Letters, 2013, 113(14−16): 572−579.

[28] Herranz J, Ruiz A, Saez G. New results and applications for multi-secret sharing schemes. Designs, Codes and Cryptography, 2014, 73(3): 841−864.

[29] Mashhadi S. A CSA-secure multi-secret sharing scheme in the standard model. Journal of Applied Security Research, 2020, 15(1): 84−95.

[30] Dai SG, Wei JF, Zhang FG. Memory leakage-resilient secret sharing schemes. Science China (Information Sciences), 2015, 58(11): 1−9.

[31] Ding C, Yuan J. Covering and secret sharing with linear codes. In: Proc. of the Discrete Mathematics and Theoretical Computer Science. LNCS 2731, Berlin: Springer, 2003. 11−25.

[32] Pappu R, Recht B, Taylor J, *et al*. Physical one-way functions. Science, 2002, 297(5589): 2026–2030.

[33] Dodis Y, Ostrovsky R, Reuzin L, *et al*. Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. SIAM Journal of Compute, 2008, 38: 97–139.

[34] Armknecht F, Maes R, Sadeghi AR, *et al*. Memory leakage-resilient encryption based on physically unclonable functions. In: Proc. of the 15th Int'l Conf. on the Theory and Application of Cryptology and Information Security. Tokyo, 2009. 685–702.

[35] Vega G, Wolfmann J. New classes of 2-weight cyclic codes. Designs Codes & Cryptography, 2007, 42(3): 327–334.

[36] Lidl R, Niederreiter H. Finite Fields, Encyclopedia of Mathematics and Its Applications. Vol.20. 2nd ed., Cambridge University Press, 1997.

附中文参考文献:

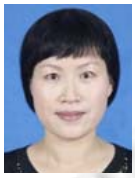
[3] 张艳硕, 李文敬, 陈雷, 毕伟, 杨涛. 基于特征值的可验证特殊门限秘密共享方案. 通信学报, 2018, 39(8): 169–175. [doi: 10.11959/j.issn.1000-436x.2018143]

[4] 谭振华, 杨广明, 王兴伟, 程维, 宁婧宇. 面向云存储的多维球面门限秘密共享方案. 软件学报, 2016, 27(11): 2912–2928. http://www.jos.org.cn/1000-9825/4943.htm [doi: 10.13328/j.cnki.jos.004943]

[6] 刘海, 李兴华, 田有亮, 雒彬, 马建峰, 彭长根. 理性公平的秘密共享方案. 计算机学报, 2020, 43(8): 1517–1533.

宋云(1987－), 女, 博士, 副教授, CCF 专业会员, 主要研究领域为密码学, 信息安全.

王文华(1987－), 女, 博士, 副教授, 主要研究领域为算子代数, 量子信息.



李志慧(1966－), 女, 博士, 教授, 博士生导师, 主要研究领域为密码编码信息安全.