计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2020,29(5):29−35 [doi: 10.15888/j.cnki.csa.007420]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

基于多项式插值的门限函数秘密分享方案①

罗景龙, 林昌露, 李朝珍, 张　剑

(福建师范大学 数学与信息学院, 福州 350117)
(福建师范大学 福建省网络安全与密码技术重点实验室, 福州 350007)
通讯作者: 林昌露, E-mail: cllin@fjnu.edu.cn

摘　要: 针对现存的函数秘密分享方案在重构的过程中需要所有的参与者全部参与, 不能灵活地适用于现实场景的问题, 本文运用多项式技术构造了含有门限的函数秘密分享方案. 按照函数秘密分享的安全模型证明了新构造的方案具有信息论意义下的安全性. 此外本文分析了 Yuan 等学者提出的函数秘密分享方案, 阐述了其方案不满足函数秘密分享方案安全性的原因. 最后将本文构造的方案与现有的函数秘密分享方案进行了比较, 发现其具有更高级别的安全性和更高的效率.

关键词: 秘密分享; 函数秘密分享; 私密信息恢复; 多项式插值; 安全性分析

引用格式: 罗景龙,林昌露,李朝珍,张剑.基于多项式插值的门限函数秘密分享方案.计算机系统应用,2020,29(5):29–35. http://www.c-s-a.org.cn/1003-3254/7420.html

Threshold Function Secret Sharing Scheme Based on Polynomial Interpolation

LUO Jing-Long, LIN Chang-Lu, LI Chao-Zhen, ZHANG Jian

(College of Mathematics and informatics, Fujian Normal University, Fuzhou 350117, China)
(Fujian Provincial Key Lab of Network Security & Cryptology, Fujian Normal University, Fuzhou 350007, China)

Abstract: Since the existing function secret sharing schemes require all participants to join in the reconstruction phase. Therefore, it cannot be flexibly applied to real-world scenarios. A function secret sharing scheme with thresholds is constructed in this study using polynomial techniques. According to the security model of function secret sharing, we proved that the proposed scheme has security in the sense of information theory. In addition, this study analyzes the function secret sharing scheme proposed by Yuan et al., and expounds the reason why their scheme does not satisfy the security of function secret sharing. Finally, a comprehensive comparison between the newly constructed scheme and the existing function secret sharing scheme is found. We note that the newly constructed scheme has higher level of security and higher efficiency through the comprehensive comparison.

Key words: secret sharing; function secret sharing; private information retrieval; polynomial interpolation; security analysis

1979 年, Shamir[1] 和 Blakley[2]分别独立提出了秘密分享 (Secret Sharing, SS) 概念并基于不同数学工具构造了相应方案. 秘密分享一般包含一个发送者和多个参与者, 在秘密分发阶段发送者将秘密值分成多个子秘密, 并将每个子秘密安全地发送给对应的参与者. 在重构阶段满足条件的参与者集合中的参与者合作重构出秘密值, 不满足条件的参与者集合中的参与者则不能得到关于秘密值的任何信息. 作为重要的密码技术, 秘密分享自提出以来一直受到研究者的持续关注.

函数秘密分享方案 (Function Secret Sharing, FSS)

① 基金项目: 国家自然科学基金 (U1705264, 61572132); 福建省自然科学基金 (2019J01275)
Foundation item: National Natural Science Foundation of China (U1705264, 61572132); Natural Science Foundation of Fujian Province, China (2019J01275)
收稿时间: 2019-09-28; 修改时间: 2019-10-29, 2019-11-15, 2019-11-22; 采用时间: 2019-11-29; csa 在线出版时间: 2020-05-07

与传统的秘密分享方案[1,2]最主要的不同在于它分享的秘密为函数而不是具体数值. 一个n方的 FSS 方案可以简单地描述如下, 在分发阶段发送者将秘密函数$f: D_f \to R_f$, 其中D_f, R_f分别表示秘密函数的定义域和值域, 可加地分为n个子函数$f_1, \cdots, f_n$, 并将子函数安全地发送给相应的参与者. 其正确性要求秘密函数可以通过计算$f = \sum_{i=1}^{n} f_i$被正确地重构, 安全性要求子函数集合$\{f_1, \cdots, f_n\}$的任何真子集完全掩盖了秘密函数的全部信息. 在 FSS 方案中, 子函数的长度与该 FSS 方案的通信复杂度直接相关, FSS 方案的通信复杂度为所有需要传输的子函数的长度之和. FSS 在提高多服务器环境下的私密信息存取的效率, 如: 私密信息恢复[3], 私密信息存储[4]等方面有着重要的应用.

针对于点函数$f_{a,b}(x): \{0,1\}^l \to \{0,1\}^m$, 其中$\{0,1\}^l$, $\{0,1\}^m$分别表示点函数的定义域和值域, 任意的$x_0 \in \{0,1\}^n$, 若$x_0 = a$有$f_{a,b}(x_0) = b$, 否则有$f_{a,b}(x_0) = 0$. Gilboa 等[5]基于伪随机生成器构造了子函数长度为$O(\lambda l^{\log_2^3})$的两方 FSS 方案, 其中λ为伪随机生成器的种子长度, 并将构造的两方 FSS 方案应用到了提高两服务器的私密信息检索协议的效率中, 得到了通信复杂度为$O(\lambda l^{\log_2^3})$的两方私密信息检索协议. 之后 Boyle 等[6]利用二叉树技术降低了文献[5]中两方 FSS 方案的通信复杂度, 构造了子函数长度为$O(\lambda l)$的两方 FSS 方案. 此外 Boyle 等[6]构造了通信复杂度为$O(\lambda 2^{l+n-1/2})$的$n(n \geq 3)$方的 FSS 方案. 在此基础上 Boyle 等[6]提出了函数秘密分享概念并对其进行了系统的介绍, 指出 FSS 与密码学中的其他概念, 例如: 同态秘密分享 (Homomorphic Secret Sharing, HSS)[7], 完全同态加密 (Fully Homomorphic Encryption, FHE)[8]等其他密码学概念之间存在着密切的联系. 在文献[9]中 Boyle 等使用代数中的张量操作简化了文献[6]中的两方的 FSS 方案的构造并将其通信复杂度降低了 4 倍.

文献[5,6,9]中的 FSS 方案在重构阶段要求所有的参与者均参与重构. 而在实际生活中往往存在参与者因为自身原因不能参与重构的场景, 例如某个参与者在方案运行的过程中掉线, 因此导致整个 FSS 方案无法正常运行. 除此之外他们的 FSS 方案都是基于伪随机生成器构造的, 因此其安全性基于密码学中单向函数存在性假设[10], 这说明了他们的 FSS 方案均为计算意义下安全的 FSS 方案, 即只可以抵抗计算能力有限的敌手.

为了使 FSS 方案可以更加灵活地应用于现实场景, 以及具有更高级别的安全性, 本文采用多项式技术构造了信息论意义下安全的门限函数秘密分享 (Threshold Function Secret Sharing, TFSS) 方案, 在重构过程中可以容忍部分参与者不参与重构, 因此可以更加灵活地应用到实际场景中. 对构造的方案按照 FSS 方案所定义的 t-安全性 (即任意 t 个参与者联合不能得到秘密函数任何信息) 进行严格的安全性证明, 证明结果表明本文构造的 TFSS 方案满足信息论意义下的 t-安全性, 即可以抵抗具有无限计算能力的敌手, 因此相对于文献[5,6,9]中的 FSS 方案本文构造的 TFSS 方案具有更高级别的安全性. 除此之外当分享的秘密函数为点函数$f_{a,b}(x): \{0,1\}^l \to \{0,1\}^m$时, 本文构造的 TFSS 方案的通信复杂度为$O(l)$. 低于文献[5,6,9]中 FSS 方案的通信复杂度. 另外我们注意到 Yuan 等[11]运用多项式插值技术构造了含有门限的 FSS 方案, 但是 Yuan 等并没有对其方案按照 FSS 方案所定义的 t-安全性进行严格的安全性证明. 经过分析发现, 在他们的方案中任意一个参与者可以通过自己收到的子函数得到秘密函数的部分信息, 进而任意两个参与者联合就可以得到整个秘密函数, 因此其方案不能满足 t-安全性. 本文对他们构造不满足 t-安全性的原因进行了具体分析和阐述.

1 相关概念

本节将给出本文所用到的一些概念和定义, 包括点函数的定义, Shamir 门限秘密分享方案, 函数秘密分享方案的定义及其安全模型.

定义 1(点函数). 设$\{0,1\}^l$为定义域, $\{0,1\}^m$为值域, 其中l, m为正整数, 则对任意的$a \in \{0,1\}^l$, $b \in \{0,1\}^m$, 点函数$f_{a,b}(x): \{0,1\}^l \to \{0,1\}^m$的定义如下: 对任意的$x \in \{0,1\}^l$有, $f_{a,b}(x) = \begin{cases} 0, x \neq a \\ b, x = a \end{cases}$.

定义 2 (Shamir 门限秘密分享方案[1]). 令GF_q为 q 元有限域, D 为发送者, $P = \{P_1, \cdots, P_n\}$为 n 个参与者组成的集合且$q > n$, $z_1, \cdots, z_n$为GF_q上 n 个两两不同的非零公开值. 若$s \in GF_q$为秘密值, r 为门限值, 则(r,n)-Shamir 门限秘密分享方案包含以下两个阶段:

1) 分发阶段: 发送者 D 从有限域GF_q中随机均匀地选取$r-1$个值$a_1, \cdots, a_{r-1}$, 生成$r-1$次多项式$f(x) = s+$

$a_1x+\cdots+a_{r-1}x^{r-1}$, 发送者 D 计算$s_i=f(z_i)(i=1,\cdots,n)$, 并将s_i安全地发送给每位参与者P_i.

2) 重构阶段:对任意 r 个参与者$\{P_{i_1},\cdots,P_{i_r}\}\subseteq\{P_1,\cdots,P_n\}$, 他们利用公开值$z_1,\cdots,z_n$计算值插值系数$c_{i_h}=\prod_{h'=1,h'\neq h}^{r}\frac{-z_{i_{h'}}}{z_{i_h}-z_{i_{h'}}}$, 并通过多项式插值公式重构出秘密$s=\sum_{h=1}^{r}c_{i_h}\cdot s_{i_h}$.

定义 3 (函数秘密分享[5]). 设 1^λ 为安全参数, D_f为定义域, R_f为值域, 秘密函数为$f(x):D_f\to R_f$, n 为参与者个数, r 为重构门限值, 则 t-安全($t<r<n$)的 FSS 方案定义为以下 3 个算法$(Gen,Eval,Dec)$, 其中Gen为子函数生成算法, $Eval$为子函数计算算法, Dec为输出解码器, 具体如下:

(1) $Gen(1^\lambda,f)\to(k_1,\cdots,k_n)$: 输入安全参数和秘密函数, 输出 n 个子函数$k_1,\cdots,k_n$.

(2) $Eval(i,k_i,x_0)\to y_i$: 任意的$x_0\in D_f$, 参与者$P_i(i=1,\cdots,n)$输入(i,k_i,x_0), 输出值y_i.

(3) $Dec(y_{i_1},\cdots,y_{i_r})\to f(x_0)$: 输入 $(y_{i_1},\cdots,y_{i_r})$, 输出秘密函数在点x_0处的函数值$f(x_0)$.

上述方案满足以下正确性和 t-安全性:

正确性: 若上述 FSS 方案中的 3 个算法 (Gen, $Eval$, Dec) 都能顺利且正确的执行, 则任意 r 个参与者可以重构出秘密函数$f(x):D_f\to R_f$在x_0处的函数值$f(x_0)$.

t-安全: 若存在受贿参与者集合$T\subseteq\{P_1,\cdots,P_n\}$且$|T|=t$, 则不可区分性实验$\prod$可描述如下:

1) 敌手$\mathcal{A}$输入安全参数 1^λ 输出 (f_0,f_1), 满足$D_{f_0}=D_{f_1}$, 并将产生的 (f_0,f_1) 发送给挑战者C.

2) 挑战者C收到 (f_0,f_1) 后, 随机地选取挑战值$c\in\{0,1\}$, 执行子函数生成算法输入f_c, 输出$(k_1^c,\cdots,k_n^c)$, 并将$\{k_i^c\}_{i\in T}$发送给$\mathcal{A}$.

3) 敌手$\mathcal{A}$收到$\{k_i^c\}_{i\in T}$后根据自己所掌握的信息给出关于挑战值c的猜测$\hat{c}$.

用$Adv(1^\lambda,A,T)=Pr[\hat{c}=c]-\frac{1}{2}$表示敌手$\mathcal{A}$在上述实验$\prod$中猜对挑战值$c$的概率与$\mathcal{A}$在不知道任何信息下随机猜测猜对 c 的概率的之差.

对于任意概率多项式时间敌手$\mathcal{A}$, 若存在关于λ的可忽略函数$u(\lambda)$[12]使得$Adv(1^\lambda,A,T)=Pr[\hat{c}=c]-\frac{1}{2}\le u(\lambda)$, 则我们称$(Gen,Eval,Dec)$为计算意义下 t-安全的 FSS 方案.

对任意拥有无限计算能力的敌手$\mathcal{A}$, 若$Adv(1^\lambda,A,T)=Pr[\hat{c}=c]-\frac{1}{2}=0$, 则我们称$(Gen,Eval,Dec)$为信息论意义下$t$-安全的 FSS 方案.

通信复杂度: 用Ψ表示一个 FSS 方案的通信复杂度, $\Psi=\sum_{i=1}^{n}|k_i|+\sum_{h=1}^{r}|y_{i_h}|$, 其中$|k_i|,|y_{i_h}|$分别表示$k_i,y_{i_h}$转化为二进制比特串的长度.

2 门限 FSS 方案 TFSS(*Gen*, *Eval*, *Dec*)

本节采用了多项式技术构造了 TFSS 方案, 按照 FSS 定义的 t-安全的安全模型证明了本文构造的 TFSS 方案具有信息论意义下的 t-安全性, 并对方案的通信复杂度进行了具体的分析.

2.1 TFSS(*Gen*, *Eval*, *Dec*)

若秘密函数为点函数$f_{a,b}(x):\{0,1\}^l\to F_q$, n 为参与者个数, $r=2lt+1(r<n)$为重构门限值, 则门限函数秘密分享方案 TFSS$(Gen,Eval,Dec)$具体如下:

(1) $Gen(1^\lambda,f_{a,b})\to(k_1,\cdots,k_n)$

① 将$a\in\{0,1\}^l$转换为二进制进行表示, 则 $a=(a_1,\cdots,a_l)$, 其中$a_j\in\{0,1\}(j=1,\cdots,l)$.

② 令$b=\prod_{j=1}^{l}b_j$, 其中$b_j\in GF_q$.

③ 发送者 D 从GF_q中随机均匀的选取lt个值$r_{j,1},\cdots,r_{j,t}(j=1,\cdots,l)$, 并产生$2l$个关于 z 的$2t$次多项式

$$\begin{cases}g_j(z)=(a_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t)\cdot(b_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t)\\ \hat{g}_j(z)=(1-(a_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t))\cdot(b_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t)\end{cases}$$

④D 使用公开值$z_1,\cdots,z_n$计算$g_{j,i}=g_j(z_i),\hat{g}_{j,i}=\hat{g}_j(z_i)$, 并生成$k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})$.

⑤ 输出$(k_1,\cdots,k_n)$.

(2) $Eval(i,k_i,x_0)\to y_i$

① 对于任意的$x_0\in\{0,1\}^l$, 将x_0用二进制表示为$x_0=(x_1^0,\cdots,x_l^0)$.

② P_i计算$y_i=\prod_{j=1}^{l}(x_j^0\cdot g_{j,i}+(1-x_j^0)(1-\hat{g}_{j,i}))$.

③ 输出y_i.

(3) $Dec(y_{i_1},\cdots,y_{i_r})\to y$

① 参与者$P_{i_h}(h=1,\cdots,r)$通过公开值$z_1,\cdots,z_n$计算$c_{i_h}=\prod_{h'=1,h'\neq r}^{r}\frac{-z_{i_{h'}}}{z_{i_h}-z_{i_{h'}}}$.

② 输出$y=\sum_{h=1}^{r}c_{i_h}\cdot y_{i_h}$.

正确性: 在 TFSS 方案中子函数$k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})(i=1,\cdots,n)$, 其中,

$$\begin{cases} g_j(z)=(a_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t)\cdot(b_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t) \\ \hat{g}_j(z)=(1-(a_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t))\cdot(b_j+r_{j,1}\cdot z+\cdots+r_{j,t}\cdot z^t) \end{cases}$$

为关于 z 的 $2t$ 次多项式, 且$\begin{cases} g_j(0)=a_j\cdot b_j \\ \hat{g}_j(0)=(1-a_j)\cdot b_j \end{cases}$. 在子函数计算算法中每个参与者$P_i(i=1,\cdots,n)$计算$y_i=Eval(i,k_i,x_0)=\prod_{j=1}^{l}(x_j^0\cdot g_{j,i}+(1-x_j^0)(1-\hat{g}_{j,i}))$.因为$(z_i,y_i)$为关于 z 的 $2lt$ 次多项式$f(z)=\prod_{j=1}^{l}(x_j^0\cdot g_j(z)+(1-x_j^0)(1-\hat{g}_j(z)))$上的一点, 所以任意$r=2lt+1$个参与者$P_{i_h}(h=1,\cdots,r)$可通过子函数$k_{i_h}$和公开值$z_1,\cdots,z_n$计算得到的$(z_{i_1},y_{i_1}),\cdots,(z_{i_r},y_{i_r})$为多项式$f(z)$上的$2lt+1$个点. 此时参与者$\{P_{i_1},\cdots,P_{i_r}\}$可以通过多项式插值公式计算:

$$\begin{aligned} y=f(0)&=\sum_{h=1}^{r}c_{i_h}\cdot y_{i_h}=\prod_{j=1}^{l}x_j^0\cdot g_j(0)+(1-x_j^0)\cdot\hat{g}_j(0) \\ &=\prod_{j=1}^{l}x_j^0\cdot a_j\cdot b_j+(1-x_j^0)\cdot(1-a_j)\cdot b_j, \end{aligned}$$

其中, $c_{i_h}=\prod_{h'=1,h'\neq r}^{r}\frac{-z_{i_{h'}}}{z_{i_h}-z_{i_{h'}}}$. 对任意的$a,x_0\in\{0,1\}^l$, 有$a=(a_1,\cdots,a_l),x_0=(x_1^0,\cdots,x_l^0)$, 若$x\neq a$, 则存在$j\in\{1,\cdots,l\}$使得$x_j^0\neq a_j$, 因此$y=f(0)=\prod_{j=1}^{l}x_j^0\cdot a_j\cdot b_j+(1-x_j^0)\cdot(1-a_j)\cdot b_j=0$.若$x_0=a$, 对任意的$j\in\{0,1\}^l$, 有$x_j^0=a_j$, 则$y=f(0)=\prod_{j=1}^{l}x_j^0\cdot a_i\cdot b_i+(1-x_j^0)\cdot(1-a_i)\cdot b_i=\prod_{i=1}^{n}b_i=b$.

因此$y=\begin{cases} 0,x\neq a \\ b,x=a \end{cases}=f_{a,b}(x_0)$. 因为只需要$r(r=2lt+1,r<n)$个参与者就可以重构出秘密函数$f_{a,b}$在$x^0$点处的函数值, 可以容忍$n-r$个参与者不参与重构.

2.2 TFSS 方案的安全性分析

定理 1. 若GF_q为q元有限域, 1^λ 为安全参数, 则 TFSS$(Gen,Eval,Dec)$是关于秘密函数$f_{a,b}(x):\{0,1\}^l\to\{0,1\}^m$的信息论意义下 t-安全的 FSS 方案.

证明: 假设$\mathcal{A}$为任意具有无限计算能力的敌手则关于敌手$\mathcal{A}$存在受贿参与者集合$T\subseteq\{P_1,\cdots,P_n\}$, 且$|T|=t$(为了简便起见我们不妨取$T=\{P_1,\cdots,P_t\}$) 的不可区分性实验$\prod$, 描述如下:

(1) 敌手$\mathcal{A}$输入安全参数 1^λ 输出$(f_{a^0,b^0},f_{a^1,b^1})$, 满足$a^0,a^1\in\{0,1\}^l,b^0,b^1\in\{0,1\}^m$, 并将$(f_{a^0,b^0},f_{a^1,b^1})$发送给挑战者$C$.

(2) 挑战者C收到$(f_{a^0,b^0},f_{a^1,b^1})$后, 随机地产生挑战值$c\in\{0,1\}$, 执行子函数生成算法输入$f_{a^c,b^c}$, 输出$(k_1^c,\cdots,k_n^c)$, 并将$\{k_i^c\}_{i\in T}$发送给敌手$\mathcal{A}$.

(3) 敌手$\mathcal{A}$收到$\{k_i^c\}_{i\in T}$后, 根据自己所掌握的信息给出一个关于挑战值c的猜测$\hat{c}$.

在上述实验$\prod$中, 敌手$\mathcal{A}$从受贿参与者集合$T=\{P_1,\cdots,P_t\}$中收到了t个子函数$\{k_i^c\}_{i\in T}$, 其中子函数$k_u^c=(g_{1,u}^c,\cdots,g_{l,u}^c,\hat{g}_{1,u}^c,\cdots,\hat{g}_{l,u}^c)(u=1,\cdots,t)$.

$$\begin{cases} g_{j,u}^c=(a_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t)\cdot(b_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t) \\ \hat{g}_{j,u}^c=(1-(a_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t))\cdot(b_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t) \end{cases}$$

此时敌手$\mathcal{A}$可以通过子函数k_u^c计算$w_{j,u}=g_{j,u}^c+\hat{g}_{j,u}^c=b_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t,j=1,\cdots,l,u=1,\cdots,t$, 满足以下等式关系:

$$\underbrace{\begin{pmatrix} b_1^c & b_1^c & \cdots & b_1^c \\ b_2^c & b_2^c & \cdots & b_2^c \\ \vdots & \vdots & & \vdots \\ b_l^c & b_l^c & \cdots & b_l^c \end{pmatrix}}_{B}+\underbrace{\begin{pmatrix} r_{1,1} & r_{1,2} & \cdots & r_{1,t} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,t} \\ \vdots & \vdots & & \vdots \\ r_{l,1} & r_{l,2} & \cdots & r_{l,t} \end{pmatrix}}_{R}\underbrace{\begin{pmatrix} z_1 & z_2 & \cdots & z_t \\ {z_1}^2 & {z_2}^2 & \cdots & {z_t}^2 \\ \vdots & \vdots & & \vdots \\ {z_1}^t & {z_2}^t & \cdots & {z_t}^t \end{pmatrix}}_{Z}=\underbrace{\begin{pmatrix} w_{1,1} & w_{1,2} & \cdots & w_{1,t} \\ w_{2,1} & w_{2,2} & \cdots & w_{2,t} \\ \vdots & \vdots & & \vdots \\ w_{l,1} & w_{l,2} & \cdots & w_{l,t} \end{pmatrix}}_{W}$$

因为 Z 为非退化矩阵, R 为一个随机矩阵, 所以对于敌手$\mathcal{A}$来说矩阵 W 为一个随机矩阵, 则敌手$\mathcal{A}$不能得到$b_j^c(j=1,\cdots,l)$的任何信息. 又因为:

$$\begin{cases} g_{j,u}^c=(a_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t)\cdot(b_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t) \\ \hat{g}_{j,u}^c=(1-(a_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t))\cdot(b_j^c+r_{j,1}\cdot z_u+\cdots+r_{j,t}\cdot z_u^t) \end{cases}$$

则敌手$\mathcal{A}$不能得到$a_j^c,b_j^c,(c\in\{0,1\})$的任何信息, 进而敌手$\mathcal{A}$不能得到关于挑战值$c$的任何信息. 所以$Adv(1^\lambda,A,T)=Pr[\hat{c}=c]-\frac{1}{2}=0$. 则定理 1 得证.

2.3 TFSS 方案的通信复杂度分析

在方案 TFSS 中$k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})(i=1,\cdots,n)$, 其$g_{j,i}=g_j(z_i),\hat{g}_{j,i}=\hat{g}_j(z_i)\in F_q$, 因此$|k_i|=2l\cdot\log_2^q$. 因为 $y_i\in F_q$, 则$|y_i|=\log_2^q$. 因此$\Psi=\sum_{i=1}^{n}|k_i|+\sum_{h=1}^{r}|y_{i_h}|=(2nl+h)\cdot\log_2^q$. 因为$(n,r,\log_2^q)$为常数, 则$\Psi=O(l)$.

3 文献[11]的方案及其分析

本节对文献[11]构造的方案进行了具体的描述, 并对其方案的不满足方案所定义的 t-安全性的原因进行了具体的分析.

3.1 文献[11]的方案

为了解决现有 FSS 方案不具有门限的问题, 文献[11]给出了秘密函数为 $f_{a,b}(x):\{0,1\}^l\to GF_q$, n为参与者的个数, $t(t=l+1,t<n)$为重构门限的 FSS 方案在其方案中发送者 D 将$a\in\{0,1\}^l$转换为二进制表示, 则$a=(a_1,\cdots,a_l)$, 令$b=\prod_{j=1}^{l}b_j$, 其中$b_j\in GF_q$. 发送者 D 从GF_q中随机均匀地选取l个值$r_1,\cdots,r_l$, 生成$2l$个关于 z 的一次多项式$\begin{cases}g_j(z)=(r_j\cdot z+a_j)\cdot b_j\\ \hat{g}_j(z)=(1-(r_j\cdot z+a_j))\cdot b_j\end{cases}$. 之后 D 使用公开值$z_1,\cdots,z_n$计算$g_{j,i}=g_j(z_i),\hat{g}_{j,i}=\hat{g}_j(z_i)(i=1,\cdots,n)$, 并生成$k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})$. 之后 D 将生成的k_i发送给参与者P_i. 参与者P_i收到$k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})$后, 对于任意的$x_0\in\{0,1\}^l$, 将$x_0$用二进制表示为$x_0=(x_1^0,\cdots,x_l^0)$. 并计算$y_i=\prod_{j=1}^{l}(x_j^0\cdot g_{j,i}+(1-x_j^0)(1-\hat{g}_{j,i}))$. 在重构阶段任意 t 个参与者$P_{i_u}(u=1,\cdots,t)$通过$y=\sum_{u=1}^{t}c_{i_u}\cdot y_{i_u}$计算出秘密函数$f_{a,b}$在$x_0$点处的函数值, 其中, $c_{i_u}=\prod_{u'=1,u'\neq u}^{t}\frac{-z_{i_{u'}}}{z_{i_u}-z_{i_{u'}}}$.

正确性: 在文献[11]方案中存在关于 z 的 l 次多项式$f(z)=\prod_{j=1}^{l}(x_j^0\cdot g_j(z)+(1-x_j^0)(1-\hat{g}_j(z)))$,其常数项$f(0)=\prod_{j=1}^{l}(x_j^0\cdot a_j\cdot b_j+(1-x_j^0)(1-a_j\cdot b_j))$,对于任意的$x_0\in\{0,1\}^l$, 若$x_0\neq a$, 则存在$x_j^0\neq a_j$, 因此有$f(0)=0$. 若$x_0=a$, 则对任意的$(j=1,\cdots,l)$都有$x_j^0=a_j$, 因此有$f(0)=\prod_{j}^{l}b_j=b$. 所以$f(0)=\begin{cases}0,x_0\neq a\\ b,x_0=a\end{cases}$, 因此关于$z$的$l$次多项式$f(z)$的常数项$f(0)=f_{a,b}(x_0)$. 而每个参与者$P_i$通过$k_i(i=1,\cdots,n)$计算的值$y_i=\prod_{j=1}^{l}(x_j^0\cdot g_{j,i}+(1-x_j^0)(1-\hat{g}_{j,i}))=f(z_j)$. 因此任意$t(t=l+1,t<n)$个参与者可以通过多项式插值计算出$f(0)$, 进而重构出秘密函数$f_{a,b}$在点$x_0$处的函数值.

通过分析发现文献[11]的方案在保证方案正确重构的前提下, 其方案可以容忍$n-t$个参与者不参与重, 因此其方案可以更加灵活地应用到现实场景.

3.2 安全性分析

本小节对文献[11]方案的安全性进行分析. 详细分析了每个参与者如何通过自己的子函数来计算得到秘密函数$f_{a,b}$中 b 的值, 以及任意两个参与者联合如何计算得到整个秘密函数$f_{a,b}$, 具体过程如下:

1) 参与者$P_i(i=1,\cdots,n)$, 通过k_i计算秘密函数$f_{a,b}$中 b 的值.

在文献[11]的方案中 $k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})$, 其中 $g_{j,i}=g_j(z_i)=(r_j\cdot z_i+a_j)\cdot b_j$, $\hat{g}_{j,i}=\hat{g}_j(z_i)=(1-(r_j\cdot z_i+a_j))\cdot b_j$,任意的参与者$P_i$计算$b_j=g_{j,i}+\hat{g}_{j,i}(j=1,\cdots,l)$, 从而得到$b=\prod_{j=1}^{l}b_j$.

2) 任意两个参与者 (不妨设为P_1,P_2) 联合通过k_1,k_2计算出秘密函数$f_{a,b}$.

参与者P_1收到子函数$k_1=(g_{1,1},\cdots,g_{l,1};\hat{g}_{1,1},\cdots,\hat{g}_{l,1})$, 参与者$P_2$收到子函数$k_2=(g_{1,2},\cdots,g_{l,2};\hat{g}_{1,2},\cdots,\hat{g}_{l,2})$, 由 1) 中的分析可知, 参与者$P_1$, P_2分别通过子函数k_1, k_2计算出$b_j(j=1,\cdots,l)$. 此时参与者P_1利用计算得到的b_j, 子函数k_1, k_2和公开值z_1, z_2计算$r_j=\frac{g_{j,2}-g_{j,1}}{b_j(z_2-z_1)}$, 同样参与者$P_2$利用计算得到的$b_j$, 子函数$k_1$, k_2和公开值z_1, z_2计算$r_j=\frac{g_{j,2}-g_{j,1}}{b_j(z_2-z_1)}$, 随后$P_1$计算$a_j=\frac{g_{j,1}}{b_j}-r_jz_1$. P_2计算$a_j=\frac{g_{j,2}}{b_j}-r_jz_2$. 此时参与者$P_1$, P_2可以分别独自计算出$a=(a_1,\cdots,a_l)$.

基于上述分析可知, 在文献[11]方案中任意两个参与者联合可以通过子函数和公开值计算出a, b的值, 从而得到秘密函数$f_{a,b}$且任意参与者可以通过子函数计算得到秘密函数$f_{a,b}$中 b 的值. 所以其方案不能抵抗 t 个参与者的联合, 因此不满足 FSS 方案中所定义的 t-安全性.

3.3 通信复杂度分析

在文献[11]的方案中$k_i=(g_{1,i},\cdots,g_{l,i};\hat{g}_{1,i},\cdots,\hat{g}_{l,i})(i=1,\cdots,n)$, 其$g_{j,i}=g_j(z_i),\hat{g}_{j,i}=\hat{g}_j(z_i)\in GF_q$, 因此$|k_i|=2l\cdot\log_2^q$. 因为 $y_i\in F_q$, 则$|y_i|=\log_2^q$. 因此$\Psi=\sum_{i=1}^{n}|k_i|+\sum_{u=1}^{t}|y_{i_u}|=(2nl+t)\cdot\log_2^q$. 因为$(n,t,\log_2^q)$为常数, 则$\Psi=O(l)$.

4 TFSS 方案与现有 FSS 方案的比较

本节我们将本文构造的基于多项式插值的门限函数秘密分享 TFSS 方案与现存的文献[5,6,9,11]中的

FSS 方案在方案所基于的工具, 有无门限值特性, 方案的安全性的级别以及方案的通信复杂度 4 个方面进行全面的比较. 为了比较的方便, 假设所有 FSS 方案分享的秘密函数均为点函数$f_{a,b}(x):\{0,1\}^l \to GF_q$, λ表示伪随机生成器种子的长度, t 表示重构门限值, n 表示参与者的个数. 具体比较结果见表 1.

表 1 TFSS 方案与现有 FSS 方案的比较

方案	基于的工具	(门限值, 参与者个数)	安全性	通信复杂度
文献[5]	伪随机生成器	(2,2)	计算意义下 t-安全	$O(\lambda l^{\log_2^3})$
文献[6]	伪随机生成器	(2,2)	计算意义下 t-安全	$O(\lambda l)$
文献[6]	伪随机生成器	(n,n)	计算意义下 t-安全	$O(\lambda 2^{(l+n-1)/2})$
文献[9]	伪随机生成器	(2,2)	计算意义下 t-安全	$O(\lambda l)$
文献[11]	多项式	(t,n)	无安全性	$O(l)$
TFSS	多项式	(r,n)	信息论意义下 t-安全	$O(l)$

经过比较发现本文构造的 TFSS 方案相对于文献[5,6,9]中构造的 FSS 方案具有额外的门限特性, 即在重构的过程中可以容忍参$(n-r)$个参与者不参与, 因此可以更加灵活地应用于现实场景, 且在安全性级别上由 2.2 节中对 TFSS 方案的安全性证明可得其为信息论意义下t-安全的, 而文献[5,6,9]中构造的 FSS 方案构造均基于伪随机生成器, 所以其方案的安全性基于密码学中单向函数的存在性假设, 进而为计算意义下 t-安全的. 因此 TFSS 方案相对于文献[5,6,9]中构造的 FSS 方案具有更高级别的安全性. 此外在分享的秘密函数均为$f_{a,b}(x):\{0,1\}^l \to GF_q$的前提下, 由 2.3 节对 TFSS 方案的通信复杂的分析可得, TFSS 方案的通信复杂度为$O(l)$, 低于文献[5,6,9]中构造的 FSS 方案的通信复杂度. 在与文献[11]中构造的 FSS 方案对比中可以发现, 虽然 TFSS 方案与文献[11]中构造的 FSS 方案均具有门限的特性, 且具有相同级别的通信复杂度. 但在安全性上经过 3.3 节对文献[11]中构造的 FSS 方案的安全性分析可得, 其方案不具有 FSS 方案定义的 t-安全性, 而本文构造的 TFSS 方案具有信息论意义下 t-安全性.

5 结语

本文针对现有的函数秘密分享方在重构阶段需要所有参与者参与不能灵活的适用于现实场景的问题, 采用多项式技术构造了门限函数秘密分享方案. 并按照函数秘密分享方案定义的安全模型证明了新构造的门限函数秘密分享方案为信息论意义下安全的. 并对文献[11]构造了门限函数秘密分享方案进行的分析, 指出了其方案存在安全性漏洞. 最后本文将新构造的门限函数秘密分享方案与现有的函数秘密分享方案进行了比较, 发现其具有更高级别的安全性和更高的效率. 但事实上, 本文构造的门限函数秘密分享方案的门限值 r 是受限的, 要求$r=2lt+1$. 因此在未来是否能构造出门限值自由的函数秘密分享方案是一个值得继续思考的问题.

参考文献

1 Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612–613. [doi: 10.1145/359168.359176]

2 Blakley GR. Safeguarding cryptographic keys. Proceedings of the AFIPS 1979 National Computer Conference. Montvale, NJ, USA. 1979. 313–317.

3 Chor B, Kushilevitz E, Goldreich O, *et al*. Private information retrieval. Journal of the ACM, 1998, 45(6): 965–981. [doi: 10.1145/293347.293350]

4 Ostrovsky R, Shoup V. Private information storage (extended abstract). Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing. New York, NY, USA. 1997. 294–303. [doi: 10.1145/258533.258606]

5 Gilboa N, Ishai Y. Distributed point functions and their applications. Proceedings of the 33rd Annual International Conference on the Theory and Applications of Cryptographic Techniques. Copenhagen, Denmark. 2014. 640–658. [doi: 10.1007/978-3-642-55220-5_35]

6 Boyle E, Gilboa N, Ishai Y. Function secret sharing. Proceedings of the 34th Annual International Conference on the Theory and Applications of Cryptographic Techniques. Sofia, Bulgaria. 2015. 337–367. [doi: 10.1007/978-3-662-46803-6_12]

7 Boyle E, Couteau G, Gilboa N, *et al*. Homomorphic secret sharing: Optimizations and applications. Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, TX, USA. 2017. 2105–2122. [doi: 10.1145/3133956.3134107]

8 Plantard T, Susilo W, Zhang ZF. Fully homomorphic encryption using hidden ideal lattice. IEEE Transactions on Information Forensics and Security, 2013, 8(12): 2127–2137. [doi: 10.1109/TIFS.2013.2287732]

9 Boyle E, Gilboa N, Ishai Y. Function secret sharing: Improvements and extensions. Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria. 2016. 1292–1303. [doi: 10.1145/2976749.2978429]

10 Håstad J, Impagliazzo R, Levin LA, *et al*. A pseudorandom generator from any one-way function. SIAM Journal on Computing, 1999, 28(4): 1364–1396. [doi: 10.1137/s0097539793244708]

11 Yuan DZ, He MX, Zeng SK, *et al*. (*t*, *p*)-threshold point function secret sharing scheme based on polynomial interpolation and its application. Proceedings of 2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing. Shanghai, China. 2016. 269–275. [doi: 10.1145/2996890.3007871.]

12 Bellare M. A note on negligible functions. Journal of Cryptology, 2002, 15(4): 271–284. [doi: 10.1007/s00145-002-0116-x]