



一种基于异或运算的(k,n)门限秘密共享算法

夏高,何成万

(武汉工程大学 计算机科学与工程学院,武汉 430205)

摘要: 现有典型的秘密共享算法多基于SHAMIR算法构建,涉及复杂的多项式计算,数据处理速度较慢。为提高计算效率,提出一种仅基于异或运算的秘密共享算法。根据 k, n 输入值计算待生成的线索个数,通过与随机二进制序列进行连续异或运算生成算法所需的全部线索,并借助排列组合将所有线索进行分组形成影子秘密。实验结果表明,该算法能够实现任意份额 (k, n) 门限秘密共享,相较于SHAMIR算法处理速度得到大幅提升,且不存在部分秘密信息泄露的安全隐患。

关键词: 异或运算;秘密共享;排列组合;信息安全;攻击者

开放科学(资源服务)标志码(OSID):



中文引用格式:夏高,何成万.一种基于异或运算的 (k, n) 门限秘密共享算法[J].计算机工程,2021,47(10):111-115,124.

英文引用格式:XIA G, HE C W. A (k, n) -threshold secret sharing algorithm based on XOR operation[J]. Computer Engineering, 2021, 47(10): 111-115, 124.

A (k, n) -Threshold Secret Sharing Algorithm Based on XOR Operation

XIA Gao, HE Chengwan

(School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430205, China)

[Abstract] Most of the existing secret sharing algorithms are constructed based on the SHAMIR algorithm, and involve complex polynomial calculations, which slows down data processing. In order to improve computational efficiency, a new secret sharing algorithm is proposed based on XOR operation. The number of clues to be generated is calculated according to the input values of k and n . Then all the clues required by the algorithm are generated by continuous XOR operation on a random binary sequence, and the clues are grouped in permutation and combinations to form a shadow secret. Experimental results show that the algorithm can realize arbitrary (k, n) -threshold secret sharing, and greatly improves the processing speed compared with the SHAMIR algorithm, while avoiding the security risk of partial secret information disclosure.

[Key words] XOR operation; secret sharing; permutation and combination; information safety; attacker

DOI: 10.19678/j.issn.1000-3428.0059258

0 概述

秘密共享体制可将一份秘密安全地交给多方保管,其在商业秘密保管、银行网络管理等注重信息安全的领域发挥重要作用。秘密共享又称 (k, n) 门限秘密共享。将一份秘密信息 S 在 n 个参与方中进行分配,每个参与方所保管的部分称为 S 的一个影子秘密。若掌握其中任意 k 个影子秘密,即可计算还原出 S ,但若只是掌握任意不足 k 个影子秘密,则无法获取 S 的任何内容, k 为恢复 S 所需影子的最小秘密个数。现有的秘密共享方案大部分基于SHAMIR算法发展而来,这类方法均涉及复杂多项式计算,在实

际应用中处理信息非常耗时。针对该问题,本文基于异或运算构建秘密共享算法,以期提高数据处理速度。

1 相关工作

1979年,SHAMIR^[1]提出了秘密共享的概念,其方案基于多项式插值进行,涉及大量多项式计算,方案较为复杂且计算量较大,速度较慢。此后多种基于异或运算的秘密共享算法被相继提出^[2-4],与SHAMIR算法相比显著提高了计算效率。

传统秘密共享面临的问题主要有分发者欺骗和恶意参与者攻击两类。分发者欺骗是指分发者在下发影子秘密时可能会给参与者无效的影子秘密,恶

基金项目:国家自然科学基金(61272115)。

作者简介:夏高(1995—),男,硕士研究生,主研方向为秘密共享算法;何成万,教授、博士。

收稿日期:2020-08-14 修回日期:2020-10-14 E-mail:xiagao666@outlook.com

意参与者攻击是指授权子集外的人伪装成授权子集的成员去骗取他们的影子秘密^[5]。为了解决这些问题,研究者相继提出可验证秘密共享以及可更新的秘密共享方案。何晓婷等^[6]基于SHAMIR算法提出一种集中式的认证方案,快速辨认出了非法参与者;刘鹏等^[7]提出基于异或运算的份额可更新(2,n)门限秘密共享算法,但该方案只能进行(2,n)门限秘密共享,不能实现任意份额的(k,n)秘密共享;谷婷等^[8]提出无可信中心可验证可更新的向量空间秘密共享。份额更新的基本思想是:在每一个阶段,对现存的秘密份额进行更新,每次更新后的影子秘密不与之前阶段的影子秘密有任何联系。这种做法可以有效解决多次部分秘密泄露导致原密码的泄露的问题。王彩芬等^[9]提出一种可验证的口令授权的多秘密分享方案,在秘密共享过程中可防止分发者欺骗和恶意参与者攻击。张艳硕等^[10]提出基于特征值的可验证特殊门限秘密共享方案,利用 n 阶矩阵的特征方程具有重根的特点,密钥分发者给每一个参与者分发2种不同的子密钥,这2种子密钥满足线性无关和对应的特征值相等。

近年来,抗泄露秘密共享^[11-12]成为一个热门的研究方向。随着侧信道攻击技术^[13-14]不断发展,传统秘密共享方案的安全性受到严重威胁,特别对于需长期维护的秘密,敌手可利用侧信道攻击技术获取一定数量的秘密、秘密份额或者方案相关信息^[15]。对此,AGGARWAL等^[16]提出一种强抗泄露的秘密共享方案,提出一种基于一般访问模型将非抗泄露的秘密共享方法转化为抗泄露的方法。

身份认证协议^[17-18]的目标是确认用户当前声称自己的身份信息是准确无误的。作为抵御攻击的第一道屏障,身份认证可以有效防止离线字典攻击和身份伪造攻击^[19-20]。生活中常见的输入密码登录网站、指纹验证解锁手机等场景均是身份认证协议的应用。

作为一种基础的加密协议,秘密共享协议可以用于身份认证。一种较直接的思路是,对某个密钥 S 进行(k,n)秘密共享分散,用户持有其中 $m(m < k)$ 个影子秘密。服务器持有剩余 $(k-m)$ 个影子秘密和密钥 S 。当用户发送验证请求后,服务器先尝试使用用户发送的影子秘密和自身存储的影子秘密对 S 进行还原。若能成功还原 S ,则验证成功,否则验证失败。

2 本文算法

2.1 异或运算的性质

异或(XOR)运算是一种二进制运算。假设对于一个二进制序列 L_1 ,将其与另一个等长的二进制序列 L_2 进行异或运算,记运算结果为 L_3 , $L_1 \text{ XOR } L_2 = L_3$ 。根据异或运算的可逆性,有 $L_3 \text{ XOR } L_2 = L_1$ 。将 L_1 视作待加密信息,将 L_2 视作加密钥,序列 L_3 为加密后的密文。在缺失 L_2 的情况下,解密者只掌握 L_3 的信息,无法根据 L_3 推测出秘密 L_1 。只有获取 L_2 和 L_3 后,解密者才可通过 $L_3 \text{ XOR } L_2 = L_1$ 来还原秘密 L_1 。

以(3,4)门限秘密共享为例介绍本文算法原理和具体过程。

2.2 (3,4)秘密共享算法

(3,4)秘密共享算法整体包含以下2个步骤:

1)将秘密 S 转化为6个二进制序列, S 能且只能由全部6个二进制序列一起通过计算还原,其中每个二进制序列称为秘密 S 的1条线索。

2)对这6条线索,按照一定规则进行分组存储,每组二进制序列形成 S 的1个影子秘密。只有掌握足够数量的影子秘密,使这些影子秘密包含着全部6条线索,原秘密 S 才能被还原。

在(3,4)秘密共享的情况下,将秘密 S 转化为线索的过程描述如下:

1)对于秘密 S ,将其转化为对应的二进制序列 G 。

2)对于 G ,随机生成1个等长的二进制序列,记为 K_1 。

3)对 G 和 K_1 进行异或运算,记运算结果为 G_1 , $G \text{ XOR } K_1 = G_1$ 。

4)随机生成1个与 G_1 等长的二进制序列 K_2 , $G_1 \text{ XOR } K_2 = G_2$ 。

5)随机生成1个与 G_2 等长的二进制序列 K_3 , $G_2 \text{ XOR } K_3 = G_3$ 。

6)随机生成1个与 G_3 等长的二进制序列 K_4 , $G_3 \text{ XOR } K_4 = G_4$ 。

7)随机生成1个与 G_4 等长的二进制序列 K_5 , $G_4 \text{ XOR } K_5 = G_5$ 。

8)保留 $K_1, K_2, K_3, K_4, K_5, G_5$ 这6个二进制序列为算法所需的全部线索,删除过程中所有其他二进制序列。线索获取完毕。

9)线索获取完毕后,按照一定规则将6个线索分为4组,分配结果将使得每组线索的集合成为原秘密 S 的一个影子秘密。记这4个影子秘密分别为 A, B, C, D 。在分组时,需遵循以下2条规则:

(1)对于 S 的任意一个线索,都要至少存储在2个不同的组中各1次。

(2)对于 A, B, C, D ,任取其中2组,都至少有1个线索是只存在于这2组中各1次的。

根据数学排列组合的知识, S 至少要有6条线索才能满足分组规则。分配结果并不具有唯一性。一种可行的分配结果如表1所示。分配完成后,当 A, B, C, D 中只缺失任意某1组时,剩余3组已携带全部6个线索, S 可被还原。若4组中缺失任意2组,必有1个线索缺失, S 无法还原,目标达到(3,4)秘密共享算法。

表1 分配结果

Table 1 Distribution result

| 影子秘密 | K_1 | K_2 | K_3 | K_4 | K_5 | G_6 |
|------|-------|-------|-------|-------|-------|-------|
| A | Y | Y | Y | — | — | — |
| B | Y | — | — | Y | Y | — |
| C | — | Y | — | Y | — | Y |
| D | — | — | Y | — | Y | Y |

假设掌握了 A 、 B 、 C 的信息,从 A 、 B 、 C 的影子秘密恢复出 S 的步骤如图1所示。

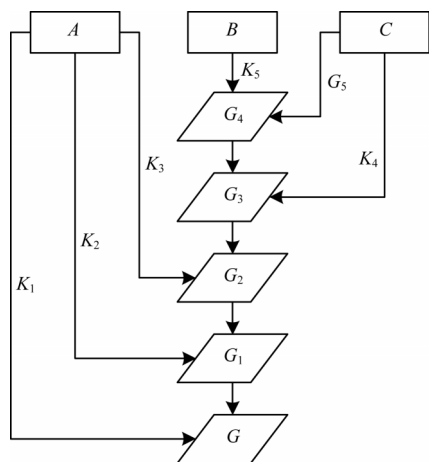


图1 基于异或运算的恢复过程

Fig.1 Restore process based on XOR operation

假设待加密信息 S 转化为二进制序列后为11010010,算法分散过程如图2所示。按照从上往下的顺序,随机二进制生成器生成一个与 G 等长的二进制 K_1 ,并与 G 进行异或运算,将运算结果记为 G_1 。在此基础上, G_1 再与随机生成的二进制序列 K_2 进行异或运算。依次异或,直至异或运算进行5次为止。

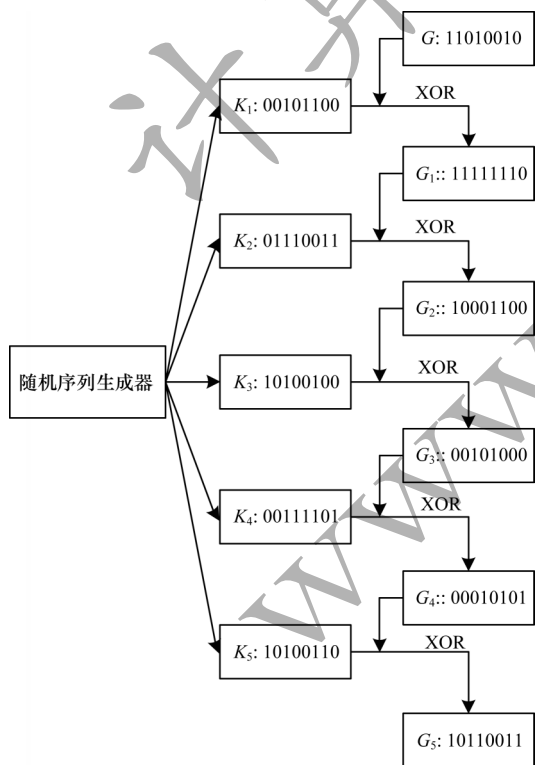


图2 秘密 S 的分散过程

Fig.2 Distribution process of secret S

在全部异或运算完成后,只保存 K_1 、 K_2 、 K_3 、 K_4 、 K_5 、 G_5 这6条线索,其余信息均不保存。对于线索的

分组,根据前述规则,某一个线索只能存入某2组中各1次,且每次影子秘密的组合不与之前任意1次组合重复。记4组线索集合为 A 、 B 、 C 、 D 。具体分组过程描述为:1)对于 K_1 ,分别存入 A 、 B 组;2)对于 K_2 ,分别存入 A 、 C 组;3)对于 K_3 ,分别存入 A 、 D 组;4)对于 K_4 ,分别存入 B 、 C 组;5)对于 K_5 ,分别存入 B 、 D 组;6)对于 G_5 ,分别存入 C 、 D 组。

假设已完全掌握了 A 、 B 、 C 的全部信息,还原 S 的步骤如图3所示。首先从 C 中取出 G_5 ,从 B 中取出 K_5 ,两者异或生成 G_4 。在此基础上,依次从 A 、 B 、 C 中取出线索 K_4 、 K_3 、 K_2 、 K_1 并依次与上一次异或运算的结果进行异或运算。当全部异或运算完成时,秘密信息 S 即被成功还原。

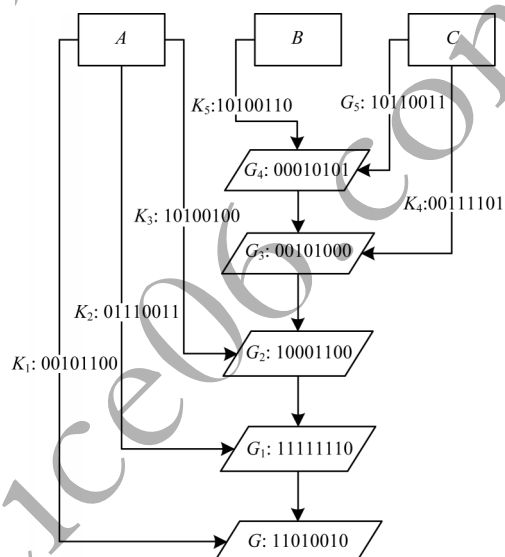


图3 秘密 S 的还原过程

Fig.3 Restore process of secret S

2.3 (k,n)秘密共享算法

基于2.2节的举例说明,本节提出一种基于异或运算的 (k,n) 秘密共享算法。记算法生成的影子秘密分别为 A_1, A_2, \dots, A_n ,全体影子秘密的集合为 U 。算法步骤描述如下:

算法1 (k,n)门限秘密共享算法

输入 (s, k, n, U)

输出 (A_1, A_2, \dots, A_n)

1. Convert S to binary sequence G_1
2. $J = C_n^{n-k+1} / J$ 为算法所需的线索个数
3. For $i \leftarrow 1$ to J do
4. Generate a new random sequence k_i
5. $G_{i+1} \leftarrow G_i \text{ XOR } k_i$
6. End
7. $k_j \leftarrow G_j$

8. $Co \leftarrow \text{GetAllCombinations}(n-k+1, U)$ /*GetAllCombination方法用于获得从全部影子秘密中挑出 $n-k+1$ 个影子秘密的全部结果的集合,返回值 Co 为影子秘密的组别的List*/

9. For $i \leftarrow 1$ to J do
10. For $q \leftarrow 1$ to $Co.Count$ do
11. $Co[q].add(k_i)$
12. End
13. End

将 S 转化为线索的方法并不唯一,本文采用图4所示方法进行分散。将 S 转换为二进制序列 G ,随机生成一个等长的二进制序列 K_1 , G 和 K_1 进行异或运算,将结果记为 G_2 。重复此步骤,每次生成的 K 与之前一次异或运算的结果进行异或运算。重复异或 $m-1$ 次,保留最后一次异或的结果 G_{m-1} 和全部的 K 。 S 能且只能由这 m 条线索还原,缺少任意一条线索都会导致 S 无法还原。

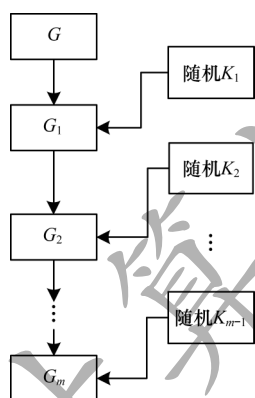


图4 基于异或运算的分散过程

Fig.4 Distribution process based on XOR operation

恢复 S 的过程是分散 S 的逆过程。在获取到任意 k 个影子秘密全部信息的前提下,按照分散的逆序,依次异或运算还原 G : $G_{m-1} \text{ XOR } k_{m-1} \text{ XOR } k_{m-2} \text{ XOR } \dots \text{ XOR } k_1 = G$ 。当获取的影子秘密个数不及 k 时,至少一个线索缺失, S 无法还原。

3 安全性证明

对本文算法进行安全建模,如图5所示,安全目标如下:

1)被攻破的服务器数量少于 k 时,原秘密信息仍维持保密。攻击者必须攻破至少 k 台服务器才能获得原秘密信息。

2)原秘密信息可被长期安全存储。一定时间后已经泄露的少量信息将失去价值。

以(2,3)秘密共享为例,假设攻击者试图获取秘密信息 S ,必须渗透2台存储影子秘密服务器,设法获取到每台服务器上的全部影子秘密的信息,理论上 S 才有泄露的可能性。由于单个影子秘密和原秘密信息 S 没有任何逻辑上的联系,即使1台服务器中的影子秘密的信息全部被窃取,理论上原秘密 S 仍

然维持保密状态,攻击者无法根据现有信息推测出任何原机密信息 S 的内容。

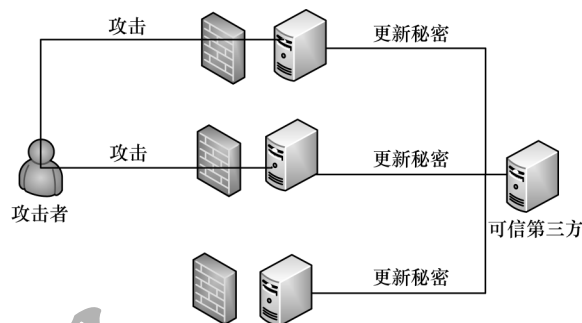


图5 安全模型

Fig.5 Safety model

算法的安全性依赖于可信第三方以及安全的信道。可信第三方可定期重新运行算法对现存影子秘密进行更新,由于算法基于随机二进制的生成而构建,每次更新后影子秘密与更新前影子秘密不存在联系,因此加大了攻击者获取 S 的难度,解决了多次部分影子秘密信息的泄露导致 S 泄露的问题,从而使秘密信息长时间存储的安全性得到保证。

4 实验与结果分析

Crypto++是基于C++语言实现的加密算法库,库中实现了SHAMIR秘密共享算法^[1]。由于Crypto++加密库和SHAMIR的秘密共享算法均被广泛使用,因此将本文算法与Crypto++所实现的SHAMIR算法进行对比,完成速度方面的性能评估。

在相同的实验环境下,通过对一个大小为10 MB的文件进行(3,4)秘密共享算法的处理,分别记录2种方法加密和还原过程的耗时。多次实验后,取平均耗时进行对比。实验环境设定为:操作系统为Windows 10 Professional Edition,处理器为Intel I7-4700HQ 2.40 GHz。本文算法和SHAMIR算法的处理耗时和还原耗时数据如表2所示。实验数据表明,无论加密或者恢复,本文算法处理效率都明显优于SHAMIR秘密共享算法。相较于SHAMIR算法最后生成的影子秘密体积10 MB,本文算法所储存的影子秘密体积更大,为30 MB,因此,本文算法对于运行内存和存储内存的要求较SHAMIR算法更高。

表2 2种算法的耗时对比

Table 2 Comparison of time consuming of two algorithms ms

| 秘密共享算法 | 处理耗时 | 还原耗时 |
|----------|--------|--------|
| 本文算法 | 817 | 221 |
| SHAMIR算法 | 90 187 | 93 984 |

通过以下3个方面对算法进行评估:1)是否能实现任意份额的 (k, n) 秘密共享;2)高效性;3)算法实现的难易程度。如果一个算法能够实现任意份额

的(k,n)秘密共享,则称该算法是通用的;如果一个算法的实现原理易于理解,生产中易于实现和维护,则认为该算法是易于实现的。

将本文算法与 SHAMIR 算法和 MATSUO 算法^[4]进行对比,如表3所示。由于都实现了任意份额的秘密共享,因此这3种算法都是通用的。由于 SHAMIR 算法涉及大量多项式计算,因此整体较为复杂,处理速度较慢,且难以实现。MATSUO 算法亦是基于异或运算构建,数据处理速度较快,但是当 k 和 n 的值改变时,其算法的实现原理也有较大改变,因此, MATSUO 算法整体上是难以实现的,并且该算法在部分影子秘密的信息泄露时,虽然不会导致原秘密信息整体泄露,但可能会导致部分秘密信息泄露。而本文算法不存在此类安全隐患,其不仅可实现任意(k,n)秘密共享算法,而是更易于实现和高效的。

表3 3种算法的评估结果

Table 3 Evaluation results of three algorithms

| 秘密共享算法 | 是否通用 | 处理速度/(MB·s ⁻¹) | 实现难易程度 |
|-----------|------|----------------------------|--------|
| 本文算法 | 是 | 10.3 | 容易 |
| SHAMIR 算法 | 是 | 0.1 | 困难 |
| MATSUO 算法 | 是 | 10.1 | 困难 |

5 结束语

本文提出一种基于异或运算的秘密共享算法。通过连续的异或运算生成算法所需的所有线索,并通过排列组合将这些线索进行合理分组。该算法不仅能够实现任意份额的秘密共享,而且还具有极快的运算速度。与 SHAMIR 算法相比,其对同一秘密信息分别多次加密解密的平均耗时大幅减少。但本文算法存在影子秘密体积过大的问题,单个影子秘密与原秘密的体积比值大于1。而在实际应用中,对运行算法的机器运行内存和对存储影子秘密的机器的存储空间要求较高。下一步将寻找更理想的将秘密分散为线索的方法,通过减小每个线索体积的方式缩小影子秘密的体积。

参考文献

- [1] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612-613.
- [2] FUJII Y. A fast (2,n)-threshold scheme and its application[C]//Proceedings of CSS'05. New York, USA: ACM Press, 2005: 631-636.
- [3] CHATTOPADHYAY A K, NAG A, SINGH J P, et al. A verifiable multi-secret image sharing scheme using XOR operation and Hash function[J]. Multimedia Tools and Applications, 2020(12): 1-5.
- [4] MATSUO M, MUTO K. (k,n)-threshold secret sharing scheme using exclusive OR[J]. Panasonic Technical Journal, 2013, 59(2): 29-34.
- [5] 荣辉桂,莫进侠,常炳国,等. 基于 Shamir 秘密共享的密钥分发与恢复算法[J]. 通信学报, 2015, 36(3): 60-69. RONG H G, MO J X, CHANG B G, et al. Key distribution and recovery algorithm based on Shamir's secret sharing[J]. Journal on Communications, 2015, 36(3): 60-69. (in Chinese)
- [6] 何晓婷,苗付友,方亮. 基于秘密共享的(t,m,n)-AS 组认证方案[J]. 计算机工程, 2018, 44(1): 154-159. HE X T, MIAO F Y, YOU L. (t,m,n)-AS group authentication scheme based on secret sharing[J]. Computer Engineering, 2018, 44(1): 154-159. (in Chinese)
- [7] 刘鹏,李飞. 基于异或运算的份额可更新(2,n)门限秘密共享方案[J]. 科学技术与工程, 2009, 9(14): 4186-4188. LIU P, LI F. New (2,n) threshold secret sharing scheme with share refreshing based on XOR operations[J]. Science Technology and Engineering, 2009, 9(14): 4186-4188. (in Chinese)
- [8] 谷婷,杜伟章. 无可信中心的可动态更新多秘密共享方案[J]. 计算机工程, 2016, 42(3): 148-155. GU T, DU W Z. Dynamically updatable multi-secret sharing scheme without trusted center [J]. Computer Engineering, 2016, 42(3): 148-155. (in Chinese)
- [9] 王彩芬,苏舜昌,杨小东. 可动态更新的口令授权多秘密共享方案[J]. 计算机工程与科学, 2019, 41(9): 1597-1602. WANG C F, SU S C, YANG X D. A dynamically updated password authorization multi-secret sharing scheme[J]. Computer Engineering, 2019, 41(9): 1597-1602. (in Chinese)
- [10] 张艳硕,李文敬,陈雷,等. 基于特征值的可验证特殊门限秘密共享方案[J]. 通信学报, 2018, 39(8): 169-175. ZHANG Y S, LI W J, CHEN L, et al. Verifiable special threshold secret sharing scheme based on eigenvalue [J]. Journal on Communications, 2018, 39(8): 169-175. (in Chinese)
- [11] SRINIVASAN A, VASUDEVAN P N. Leakage resilient secret sharing and applications[C]//Proceedings of 2019 Annual International Cryptology Conference. Berlin, Germany: Springer, 2019: 480-509.
- [12] KUMAR A, MEKA R, SAHAI A. Leakage-resilient secret sharing[C]//Proceedings of ECCV'18. New York, USA: ACM Press, 2018: 200.
- [13] NIELSEN J B, SIMKIN M. Lower bounds for leakage-resilient secret sharing[C]//Proceedings of 2020 Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin, Germany: Springer, 2020: 556-577.
- [14] BENHAMOUDA F, DEGWEKAR A, ISHAI Y, et al. On the local leakage resilience of linear secret sharing schemes [C]//Proceedings of 2018 Annual International Cryptology Conference. Berlin, Germany: Springer, 2018: 531-561.
- [15] 张明武,陈泌文,李发根,等. 强抗泄漏的无条件安全动态秘密共享方案[J]. 密码学报, 2016, 3(4): 361-373. ZHANG M W, CHEN B W, LI F G, et al. A strongly leakage-resilient and unconditionally secure dynamic secret-sharing scheme[J]. Journal of Cryptologic Research, 2016, 3(4): 361-373.

(上接第 115 页)

- [16] AGGARWAL D, DAMGÅRD I, NIELSEN J B, et al. Stronger leakage-resilient and non-malleable secret sharing schemes for general access structures[C]//Proceedings of 2019 Annual International Cryptology Conference. Berlin, Germany: Springer, 2019: 510-539.
- [17] 王晨宇,汪定,王菲菲,等. 面向多网关的无线传感器网络多因素认证协议[J]. 计算机学报,2020,43(4): 683-700.
WANG C Y, WANG D, WANG F F, et al. Multi-factor user authentication scheme for multi-gateway wireless sensor networks[J]. Chinese Journal of Computers, 2020, 43(4): 683-700. (in Chinese)
- [18] WANG D, WANG P. Two birds with one stone; two-factor authentication with security beyond conventional bound[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(4): 708-722.
- [19] WANG C Y, XU G A, SUN J. An enhanced three-factor user authentication scheme using elliptic curve cryptosystem for wireless sensor networks[J]. Sensors, 2017, 17(12): 2946.
- [20] XU G S, QIU S M, AHMAD H, et al. A multi-server two-factor authentication scheme with un-traceability using elliptic curve cryptography[J]. Sensors, 2018, 18(7): 2394.

编辑 金胡考