文章编号:1005-0523(2010)04-0063-05

可验证的动态多秘密共享

赵丽萍,汤文亮

(华东交通大学 软件学院,江西 南昌 330013)

摘要:提出了一种新的可验证的动态门限多秘密共享方案。该方案的安全性基于 Shamir 的秘密共享体制和椭圆曲线加密算法的安全性以及椭圆曲线离散对数问题的求解困难性。共享秘密可以周期性的改变,秘密分发者周期性的改变公告栏上的信息以增强系统的健壮性。对于不同的共享秘密,秘密分发者可以动态调整该秘密的门限值。此外,方案能有效检测和识别参与者的欺骗行为,参与者也可以验证其接受到的信息,且无需改变私有信息在任何时候都可以重构秘密。由于公告栏上的信息是定期更新的,所以不会影响新秘密的共享。

关 键 词:门限;椭圆曲线;秘密共享;多秘密

中图分类号:TP393.08　　**文献标识码**:A

秘密共享[1]是信息安全和数据保密的重要手段。1979 年 Shamir[2]和 Blakey[3]分别独立地提出了门限秘密共享体制,使授权的合格子集能容易地恢复共享秘密,而非合格子集得不到关于共享秘密的任何信息。由于有着广泛的实际应用前景,许多学者投入了大量精力对其本身和相关问题进行深入的研究,并获得了一批研究成果。在 Shamir 的秘密共享方案中存在两个问题:庄家(秘密的分发者)的诚实性和成员的诚实性。对这两个问题的研究,就形成了可验证的秘密共享方案[4-5]。随后的大多数方案在设计时都基于 2 个假设:(1) 在秘密被重构之前所有的参与者和秘密都不改变。(2) 秘密分发者和各参与者之间需要建立一条安全信道。但这 2 个假设会影响秘密共享方案的实际应用。文献[6]提出了一种检测方案,一次计算即可知道该授权子集中是否有不诚实的用户。如果没有欺诈者,即可得到正确秘密;如果存在欺诈者,则需要对各个用户进行验证。当参与者中没有欺诈者存在时,验证所需的计算量将大大减少。文献[7]提出了一个动态的秘密共享方案,在该方案中使用了 RSA 密码体制,这使得方案不需要建立安全信道并且共享秘密和参与者都可以动态地发生变化,但它只适用于单秘密共享,存在一定的局限性。

本文基于椭圆曲线加密体制和 Shamir 的秘密共享体制,提出了一种新的多秘密共享方案。该方案具有如下特点:(1) 可进行多秘密的共享;(2) 对于不同的共享秘密,秘密分发者可动态调整其门限值;(3) 可防止秘密分发者与参与者的相互欺诈。

1 方案描述

方案由系统初始化、秘密的分发、秘密的重构、秘密的更新 4 个阶段构成。

1.1 系统初始化

系统中包含一个秘密分发者和 n 个参与者。用 U 表示由 n 个参与者构成的集合,并且 U 由集合构成,即

$$U=\bigcup_{i=0}^{m}U_i,\text{且 } U_i\cap \mathrm{U}_j=\Phi,\ i\neq \mathrm{j}$$

收稿日期:2010-03-16

基金项目:江西省南昌市科技课题经费资助项目(洪财企[2008]68 号)

作者简介:赵丽萍(1981—),女,硕士研究生,助教,主要研究方向为软件工程、信息安全。

设 $E(F_q)$ 是一条椭圆曲线，G_1 是该曲线上的一个 α 阶子群，其中 α 为质数。G 为 α 阶循环子群，$G_2 = G - \{0\}$。令 $e: G_1 \times G_1 \rightarrow G_2$ 上的双线性映射，秘密分发者选择 G_1 的一个生成子 g，同时选择加密哈希函数 $h: G_1 \rightarrow Z_\alpha^*$，将 $< \alpha, G_1, G_2, e, g, h >$ 发布在公告栏，同时公布参与者的数目。秘密分发者随机秘密选取 $2k-1$ 个随机数 $a_0, a_1, \cdots, a_{2k-2}$，构造多项式[8]：$P(x) = \sum_{i=0}^{2k-2} a_i x^i$。秘密分发者计算 $g^{a_i}, 0 \leqslant i \leqslant 2k-2$，并将 g^{a_i} 公开。

每个参与者 $U_i(i = 0,1,2,\cdots,n-1)$ 下载公告栏上的信息，随机选取秘密整数 $s_i \in Z_p^*$，计算公钥 $U_i = s_i g$，并将 U_i 发回给秘密发送者，以保证不同的参与者使用不同的密钥。秘密分发者将 $U_i(i = 1,2,\cdots,n-1)$ 公布在公告栏。

1.2 秘密分发

(1) 秘密分发者随机选择 $s \in Z_\alpha^*$，对所有的 $U_i(i = 0,1,\cdots,n-1)$，使用哈希函数 h，构造一个 $n * t$ 的矩阵 $\boldsymbol{M}$，$\boldsymbol{M} = \begin{bmatrix} h(ss_0 g) & h^2(ss_0 g) & \cdots & h^t(ss_0 g) \\ h(ss_1 g) & h^2(ss_1 g) & \cdots & h^t(ss_1 g) \\ \cdots & \cdots & \cdots & \cdots \\ h(ss_{n-1} g) & h^2(ss_{n-1} g) & \cdots & h^t(ss_{n-1} g) \end{bmatrix}$，秘密分发者将 sg 分发到公告栏。

(2) 每个参与者得到随机 ID，分别表示为 u_j。秘密分发者将共享秘密分成 t 份，根据每个参与者所处的等级 j，分发者 D 秘密地给各参与者分发不同的子秘密，分别表示为 $\sigma(u) = P_{k_{j-1}}(u)$，使其构成一个 $t \times 1$ 的列矩阵 $\boldsymbol{X} = \begin{pmatrix} \sigma(u_1) & \sigma(u_2) & \cdots & \sigma(u_t) \end{pmatrix}^{\mathrm{T}}$。其中

$P_0(x) = P(x); P_1(x) = f^1(P(x)) = f(P(x)); P_j(x) = f(P_{j-1}(x)) = f(f(P(x))$

计算

$$\boldsymbol{V} = \boldsymbol{M} * \boldsymbol{X} = \begin{bmatrix} h(ss_0 g) & h^2(ss_0 g) & \cdots & h^t(ss_0 g) \\ h(ss_1 g) & h^2(ss_1 g) & \cdots & h^t(ss_1 g) \\ \cdots & \cdots & \cdots & \cdots \\ h(ss_{n-1} g) & h^2(ss_{n-1} g) & \cdots & h^t(ss_{n-1} g) \end{bmatrix} \begin{bmatrix} \sigma(u) \\ \sigma(u_2) \\ \cdots \\ \sigma(u_t) \end{bmatrix} = \begin{bmatrix} I_0 \\ I_1 \\ \cdots \\ I_{n-1} \end{bmatrix}$$

秘密分发者将 $I_i(i = 1,2,\cdots,n-1)$ 公布在布告栏上。

各参与者在收到自己的子秘密后，可根据以下方法来验证所收到的子秘密 $\sigma(u)$ 是否正确。首先设 r 是一个 $2k-1$ 次的多项式，写成向量的形式为 $\boldsymbol{r}(x) = (1, x^1, x^2, \cdots, x^{2k-2})$，则分配给用户的子秘密可以表示为 $\sigma(u) = r_{k_{j-1}}(u) \cdot a^{\mathrm{T}}$。其中 $a = (a_0, a_1, \cdots, a_{2k-2})$。属于第 j 等级的用户在收到子秘密后，对于所有的 $i \in \{k_{j-1}, k_j, \cdots, 2k-2\}$，计算 $\varepsilon_i = (u)i - k_{j-1}$。用户验证等式 $g^{\sigma(u)} \equiv \sum_{i=k_{j-1}}^{2k-2} (g^{a_i})\varepsilon_i$ 是否成立，如果成立，则参与者可知自己收到的子秘密 $\sigma(u)$ 是否正确。如果子秘密不正确，可要求秘密分发者重新发送，并发起投诉。

1.3 秘密重构

为了将 t 份秘密重构，需要 t 个参与者提供他们的 $ss_i g$ $(i = H_1, H_2, \cdots, H_t)$ 值。每个参与者 $U_i(i = 1, 2, \cdots, t-1)$ 首先从公告栏上下载 sg，并计算 $ss_i g$，然后利用哈希函数 h 生成矩阵 $\boldsymbol{M}$ 的第 i 行，最后参与者将该行信息发送给秘密合成者。这样，授权子集内的各参与者合作，可列出方程 $\boldsymbol{M} * \boldsymbol{X} = \boldsymbol{I}$。

$$\begin{bmatrix} h(ss_0 g) & h^2(ss_0 g) & \cdots & h^t(ss_0 g) \\ h(ss_1 g) & h^2(ss_1 g) & \cdots & h^t(ss_1 g) \\ \cdots & \cdots & \cdots & \cdots \\ h(ss_{n-1} g) & h^2(ss_{n-1} g) & \cdots & h^t(ss_{n-1} g) \end{bmatrix} \begin{bmatrix} \sigma(u) \\ \sigma(u_2) \\ \cdots \\ \sigma(u_t) \end{bmatrix} = \begin{bmatrix} I_0 \\ I_1 \\ \cdots \\ I_{n-1} \end{bmatrix}$$

如果 $|\boldsymbol{M}| \neq 0$，则方程有且仅有唯一解。这样合成者就可以重构所有的 t 份秘密。

1.4　秘密更新

秘密分发者重新选择共享秘密的门限和参数 s，秘密更新如下：

(1) 秘密分发者选择新的门限 t'，$s' \in Z_\alpha^*$，构造矩阵 $\boldsymbol{M}'$

$$\boldsymbol{M}' = \begin{bmatrix} h(s's_0g) & h^2(s's_0g) & \cdots & h^{t'}(s's_0g) \\ h(s's_1g) & h^2(s's_1g) & \cdots & h^{t'}(s's_1g) \\ \cdots & \cdots & \cdots & \cdots \\ h(s's_{n-1}g) & h^2(s's_{n-1}g) & \cdots & h^{t'}(s's_{n-1}g) \end{bmatrix}$$

，然后秘密分发者在公告栏上公布 $s'g$。

(2) 秘密分发者将共享秘密分成 t' 份，使其构成一个 $t' * 1$ 的列矩阵 $\boldsymbol{X} = \begin{pmatrix} \sigma_1' & \sigma_2' & \cdots & \sigma_{t'}' \end{pmatrix}^{\mathrm{T}}$，并计算 $\boldsymbol{V}' = \boldsymbol{M}' * \boldsymbol{X}'$。

$$\begin{bmatrix} h(s's_0g) & h^2(s's_0g) & \cdots & h^{t'}(s's_0g) \\ h(s's_1g) & h^2(s's_1g) & \cdots & h^{t'}(s's_1g) \\ \cdots & \cdots & \cdots & \cdots \\ h(s's_{n-1}g) & h^2(s's_{n-1}g) & \cdots & h^{t'}(s's_{n-1}g) \end{bmatrix} \begin{bmatrix} \sigma' \\ \sigma'_2 \\ \cdots \\ \sigma'_{t'} \end{bmatrix} = \begin{bmatrix} I'_0 \\ I'_1 \\ \cdots \\ I'_{n-1} \end{bmatrix}$$

，秘密分发者将 $I'_i(i = 0, \cdots, n-1)$ 和 sg' 公布在公布栏上。

1.5　成员的增加和删除

在秘密没有恢复前，如果有另外的参与者加入，则新成员选择自己的秘密份额，并计算 $P(x) = \sum_{i=0}^{2k-2} a_i x^i$，然后秘密分发者随机选择 $s \in Z_\alpha^*$，对所有的 $U_i(i = 0, 1, \cdots, n)$，使用哈希函数 h，构造一个 $(n+1) * t$ 的矩阵 $\boldsymbol{M}$，重复执行 1.2 步骤。

在秘密没有恢复前，如果有参与者因某些原因要被删除。此时秘密分发者将其公开信息从公告栏上删除，然后秘密分发者随机选择 $s \in Z_\alpha^*$，对所有的 $U_i(i = 0, 1, \cdots, n-2)$，使用哈希函数 h，构造一个 $(n-1) * t$ 的矩阵 $\boldsymbol{M}$，重复执行 1.2 步骤。

2　分析与讨论

2.1　安全性分析

(1) 参与者提供的秘密分量是可验证的，并且验证并不需要复杂的算法或更多的信息。秘密的重构者只需验证 $e(ss_ig, g) \equiv \mathrm{e}(\mathrm{sg}, \mathrm{U}_i)$ 是否成立。如果成立，则参与者提供的信息是有效的，否则信息是无效的。同时也可以识别出无效的参与者。

(2) 当多于 t 个参与者提供其秘密信息时，秘密是可重构的。首先通过等式 $e(ss_ig, g) \equiv \mathrm{e}(\mathrm{sg}, \mathrm{U}_i)$ 来验证参与者提供的信息是否有效，然后将参与者提供的信息进行重构，将秘密恢复。

(3) 参与者提交给秘密分发者的信息 $U_i(= s_ig)$ 是安全的。由于算法是基于椭圆曲线离散对数问题，所以参与者的个人信息 s_i 是不会泄密的。

(4) 可检验秘密分发者是否诚实。各参与者在收到自己的子秘密之后，可根据公开的信息 g^{a_i}，验证式 $g^{\sigma(u)} \equiv \sum_{i=k_{j-1}}^{2k-2} (g^{a_i})\,\varepsilon_i$ 是否成立。如果成立，说明秘密分发者分发的是正确子秘密。这是因为，第 j 等级的用户的多项式进行 k_{j-1} 次 f 函数运算后，多项式中原本次数低于 k_{j-1} 的项降为 0，其后各项分别为 $x^{t-k_{j-1}}$，$k_{j-1} \leqslant \mathrm{t} \leqslant \mathrm{k}-1$。所以有：

$$\prod_{t=k_{j-1}}^{k-1} (g^{a_t})\,\varepsilon \equiv \prod_{t=k_{j-1}}^{k-1} (g^{a_t\varepsilon}) \equiv \mathrm{g}_{t=k_{j-1}}^{\sum^{2k-2}((u_i)t-k_{j-1}\cdot a_t)} \equiv \mathrm{g}^{P(k_{j-1})(u_i)} \equiv \mathrm{g}^{\sigma(u_i)}$$

如果等式 $g^{\sigma(u)} \equiv \sum_{i=k_{j-1}}^{2k-2} (g^{a_i})\,\varepsilon_i$ 不成立，说明秘密分发者造假，可要求秘密分发者重新分发子秘密，并发起投诉。

2.2 性能分析

(1) 与 Chen et al.′s 秘密共享机制相比,Chen et al.′s 秘密共享机制中的共享秘密需通过哈希函数 $h(rp)$ 生成。在本文的共享机制中,共享秘密的门限值 t 存储在矩阵 $\boldsymbol{M}$,故大小由矩阵 $\boldsymbol{M}$ 的大小决定,秘密 s 在任何时候均存储在矩阵 $\boldsymbol{X}$ 中,与哈希函数 h 无关。所以计算时间会缩短。

(2) 与 Chen et al.′s 秘密共享机制相比,Chen et al.′s 秘密共享机制中,矩阵 $\boldsymbol{M}$ 会以 $(n-t+1)\times(n+t)$ 增大,而 $\boldsymbol{X}$ 矩阵会以 $(n+t)\times 1$ 增大,此外,其门限值随着秘密的重构会改变为 $2t-1$;在本文的共享机制中,矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{X}$ 的大小不会改变,且秘密更新也非常方便,秘密分发者只需选择新的门限值 t', s',进而构造新的矩阵 $\boldsymbol{M}'$, $\boldsymbol{X}'$, $\boldsymbol{V}'$, $\boldsymbol{I}'$,故多秘密共享在本方案中所需的空间不会发生改变。

(3) 秘密重构的门限是可变的。秘密分发者只需改变矩阵 $\boldsymbol{M}$ 和 $\boldsymbol{X}$ 的大小,秘密重构的门限就可以改变。

表 1 本文协议与 Chen et al.′s 协议的比较

	本文协议	Chen et al.′s 协议
共享秘密数	t	1
矩阵 $\boldsymbol{M}$ 的大小	$n\times t$	$(n-t+1)\times(n+1)$
矩阵 $\boldsymbol{X}$ 的大小	n	$n-t+1$
秘密份额的平均内存大小	$1/t$	1/1
秘密份额的公共信息大小	$(n+(n+1))/nt$	$(n+(n+1-t))/n$
总的内存大小	$(3n+1)/nt$	$(n+(n+1-t))/n$

3 结论

本文提出了一种的新的多秘密共享方案,该方案允许每个参与者只需保存一个秘密信息就能共享多个秘密。降低了秘密分发者的算法难度。秘密重构时,参与者只需提供公共信息和个人的秘密信息,就可实现秘密的重构。当门限发生改变时,秘密分发者只需调整一个矩阵的大小。此外,该方案对参与者的验证加强了该方案的健壮性。由于方案是基于椭圆曲线,故其安全性较高,算法的效率也较高。

参考文献:

[1] CHIEN H Y, JAN J K, TSENG Y M. A practical (t, n) multi-secret sharing scheme[J]. IEICE Transaction on Fundamentals, 2000, 83(12): 2 762-2 765.

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

[3] BLAKLEY G. Safeguarding Cryptographic Keys[C]//New York: Proceedings of the AFIPS 1979 National Computer Conference, 1979: 313-317.

[4] CHOR B, GOLDWASSER S, MICALI S, et al. Verifiable Secret Sharing and Achieving Simultaneity in the Presence of Faults[C]//Proc of the 26th IEEE Symposium on Foundations of Computer Sciences. Los Angeles, USA: IEEE Computer Society, 1985.

[5] TASSA T. Hierarchical threshold secret sharing[J]. Journal of Cryptology, 2007, 20(2): 237-264.

[6] 许春香,肖鸿,肖国镇. 安全的门限秘密共享方案[C]//第八届中国密码学学术会议论文集. 北京:科学出版社, 2004: 120-124.

[7] 庞辽军,李慧贤. 动态门限多重秘密共享方案[J]. 计算机工程, 2008, 34(15): 164-165.

[8] 毛颖颖,毛明,李冬冬. 安全的多等级门限秘密共享[J]. 计算机工程与应用, 2009, 45(32): 90-92.

[9] CHEN W, LONG X, BAI Y B, GAO X P. A New Dynamic Threshold Secret Sharing Scheme from Bilinear Maps[R]. In International Conference on Parallel Processing Workshops, 2007: 19.

Verifiable Dynamic Multi-secret Sharing Scheme

Zhao Liping，Tang Wenliang

(School of Software Engineering，East China Jiaotong University，Nanchang 330013，China)

Abstract：The paper proposes a new verifiable multi-secret sharing scheme of dynamic threshold. The security of the proposed scheme is based on Shamir's secret sharing scheme and the ECIES cryptosystem，and the difficulty in solving the elliptic curve discrete logarithm. In the scheme，the secret will change periodically and the dealer will periodically publish some of the information to increase the robustness of system. The dealer could adjust the threshold value depending on the secure level of different secret. In addition，the efficient solutions against multiform cheating of any participant are proposed，and the participants can verify the information which they have received. Each participant uses his own private secret during different time periods to reconstruct the corresponding shared secrets without revealing their own private information. Public information is renewed periodically in the scheme，which will not influence new secret sharing.

Key words：threshold；elliptic curve；secret sharing；multi-secret

（责任编辑　刘棉玲）

（上接第 47 页）

A Watermarking Algorithm for Color Images Based on Double Scrambling and SVD

Shi Hongqin，Lv Fangliang

(School of Software，East China Jiaotong University，Nanchang 330013，China)

Abstract：A secure watermarking algorithm based on watermark and the double carrier scrambling is proposed. Firstly，the carrier image is transformed from RGB color space into YUV color space. After scrambling Fibonacci transformation，the Y component is divided into 8×8 blocks and singular value decomposition for each block is conducted. Through Arnold，the binary image is embedded into singular value of each block of Y component. Simulating experiments show that the algorithm has better robustness for JPEG compression，cropping，rotation，filtering，and geometric attacks.

Key words：singular value decomposition；color space convert；double scrambling transform

（责任编辑　王建华）