具有前摄能力的可公开验证秘密共享

陈养奎 于 佳 郝 蓉 刘红艳 许曰滨
(青岛大学信息工程学院 青岛 266071)

摘 要 可公开验证秘密共享是一种特殊的秘密共享,由分发者分发的秘密份额不仅能被份额持有者自己验证,而且可以被其他任何成员验证。然而,对于一般的可公开验证秘密共享,敌手可能使用很长的时间才能攻破门限个份额服务器,获得秘密。为了解决这个问题,提出了第一个具有前摄能力的可公开验证的秘密共享方案,该方案不仅能够公开验证份额的正确性,而且具有份额定期更新的性质,比其它一般可公开验证秘密共享方案更安全,能够更好地满足各种应用的安全需求。

关键词 秘密共享,门限方案,可公开验证,前摄性
中图法分类号 TP309 **文献标识码** A

Publicly Verifiable Secret Sharing Scheme with Proactive Ability

CHEN Yang-kui YU Jia HAO Rong LIU Hong-yan XU Yue-bin
(College of Information and Engineering, Qingdao University, Qingdao 266071, China)

Abstract A publicly verifiable secret sharing(PVSS) scheme is a special secret sharing scheme in which the shares distributed by the dealer can be verified not only by shareholders themselves but also by any other party. However, in a normal publicly verifiable secret sharing scheme, an adversary may get the secret by attacking threshold shareholder servers for a long time. In order to deal with this problem, a publicly verifiable secret sharing scheme with proactive ability was newly proposed, which not only can publicly verify the validity of shares, but also has the property of periodically renewing shares. This makes the proposed scheme more secure than other common publicly verifiable secret sharing schemes, and makes it better satisfy security demand of various applications.

Keywords Secret sharing, Threshold scheme, Publicly verifiable scheme, Proactive property

秘密共享是信息安全方向的一个重要分支,它是安全多方计算的理论基础,在保证计算机及网络安全方面具有重要的作用。1979年,Shamir[1]和Blakley[2]分别基于Lagrange插值多项式和射影几何理论最先提出了门限秘密共享方案。文献[3,4]分别提出可验证和可公开验证秘密共享,它解决了分发过程中分发者的诚实性问题和秘密重构过程中份额持有者的诚实性问题。Herzberg等人[5]最先提出前摄秘密共享,可定期地对每个秘密份额进行更新,从而使得攻击者在前一个时间周期内获得的秘密份额完全失效。文献[6]则提出了先动的可公开验证服务器辅助秘密共享。

本文提出了一种具有前摄能力的可公开验证秘密共享方案。首先,执行一个初始化协议,分发者生成秘密 x 的 n 个秘密份额 $x_i(i=1,\cdots,n)$,使用变形的ElGamal加密分发将其分发给 n 个服务器成员 P_i,每个服务器成员都可以对任意一个份额进行验证。每过一个时间周期,执行一个秘密份额更新协议,这 n 个服务器成员更新他们的份额 x_i,而对传送的信息的验证是可公开的。如果验证某个服务器成员发送的信息是错误的,就把它放入坏服务器集合,并执行重启操作,调用恢复损坏秘密份额协议重新生成秘密份额。这种前摄的属性对于保护长时间有效的秘密(如CA的签名密钥)十分有效。

1 预备知识

1.1 符号定义

p,q 是大素数,满足 $q|(p-1)$。G_q 是 Z_p^* 中唯一的 q 阶乘法子群,$g,h\in G_q$ 是 G_q 中的两个生成元。P_i 是表示第 i 个服务器成员($i=1,\cdots,n$)。$f^{(t)}$ 表示第 t 个时间周期中的函数,$x^{(t)}$ 表示第 t 个时间周期中的变量。$B_i(i=1,\cdots,n)$ 表示第 i 个服务器成员用来记录非法服务器成员的集合,初始值 $B_i=\phi$。$(a_i^{(t)},b_i^{(t)})$ 表示在第 t 时期的公私钥对,$a_i^{(t)}$ 是 P_i 的私钥,$b_i^{(t)}$ 是 P_i 的公钥。

1.2 模型和假设

假定系统由 n 个服务器成员 $\{P_1,P_2,\cdots,P_n\}$ 构成,使用的是 $(k+1,n)$ 门限方案共享的密钥 x,其中 $n\geqslant 2k+1$。假定每个服务器成员都与公共通道相接。整个生命周期分成若干

到稿日期:2009-07-12 返修日期:2009-10-07 本文受国家自然科学基金资助项目(60703089),山东省自然科学基金资助项目(ZR2009GQ008),山东省教育厅科技计划项目(J08LJ02)资助。

陈养奎(1984—),男,硕士生,主要研究方向为信息安全;于 佳(1976—),男,博士,副教授,主要研究方向为信息安全,E-mail:qduyujia@gmail.com(通讯作者);郝 蓉(1976—),女,硕士,讲师,主要研究方向为信息安全;刘红艳(1986—),女,硕士生,主要研究方向为信息安全;许曰滨(1950—),男,教授,主要研究方向为网格计算。

时间段，每个时间段包含一个短时间的更新阶段，此阶段执行一个交互的更新协议。敌手在每个时间段的任何时刻都可以攻破服务器，但不能在一个时间段内攻破多于 k 个的服务器。假定当某个服务器被检测到异常时，可以重启此服务器使其恢复正常。上述的假设是常用的假设，见文献[5]。

2 提出的具有前摄能力的可公开验证秘密共享方案

定义 1 一个具有前摄能力的可公开验证秘密共享方案是由子协议构成的六元组(Init, Pku, Sku, Acc, Bsd, Ssr)，其中：

1. Init，初始化：

输入：安全参数 k 和秘密 x。

输出：初始秘密份额 $x_i^{(0)}$。

2. Pku，私密钥更新协议：

输入：公共密钥 $a_i^{(t-1)}$，私密钥 $b_i^{(t-1)}$。

输出：公共密钥 $a_i^{(t)}$ 和私密钥 $b_i^{(t)}$。

3. Sku，秘密份额更新协议：

输入：秘密份额 $x_i^{(t-1)}$。

输出：新的秘密份额 $x_i^{(t)}$。

4. Acc，指控核实协议：

输入：服务器成员 P_i 对 P_j 的指控。

输出：不诚实的服务器 P_i。

5. Bsd，损坏份额发现协议：

输入：每个服务器成员 P_i 都拥有的所有服务器成员份额的承诺 $\{\psi_j^{(t)}\}_{j\in\{1,\cdots,n\}}$。

输出：通过比较每一服务器和大多数服务器的不同决定的那些不诚实的服务器 P_i。

6. Ssr，恢复损坏份额协议：

输入：份额被破坏的服务器成员 P_r。

输出：服务器成员 P_r 的正确份额。

下面对所提方案的 6 个子协议(Init, Pku, Sku, Acc, Bsd, Ssr)进行详细描述。

2.1 初始化(Init)

每个服务器成员 P_i 任意选取自己的私密钥 $z_i\in\mathbb{Z}_q$ 并且公开他的公钥 $y_i=h^{z_i}\pmod p$。

分发者 D 选择随机多项式 $f^{(0)}(x^{(0)})=\sum_{j=0}^{k}a_j(x^{(0)})^j\pmod q$，其中 $a_0=x$，计算每个服务器成员的份额 $x_i^{(0)}=f^{(0)}(i)=\sum_{j=0}^{k}a_j i^j$(其中 $i=1,\cdots,n$)。

令 $x_i=x_i^{(0)}$，分发者 D 对 x_i 进行加密。D 任意选取 $\alpha\in\mathbb{Z}_q$ 计算 $(A,B_i)=(h^\alpha,x_i^{-1}y_i^\alpha)$。设 $V_i=g^{x_i}\pmod p$，那么 $V_i^{B_i}=g^{x_iB_i}=g^{y_i^\alpha}$，于是 $(A,V_i^{B_i})=(h^\alpha,g^{y_i^\alpha})$。

分发者 D 利用知识证明来验证 $(A,B_1,B_2,\cdots,B_n)$ 的正确性。

D 任意选取 $l(l\approx100)$ 个数 $w_j\in\mathbb{Z}_q$，计算 $t_{hj}=h^{w_j}\pmod p$，$t_{gj}=g^{y_i^{w_j}}$，$j=1,\cdots,l$，$i=1,\cdots,n$。

计算 $R=(r_1,\cdots,r_l)=(w_1-c_1\alpha\pmod q,w_2-c_2\alpha\pmod q,\cdots,w_l-c_l\alpha\pmod q)$，其中 c_j 表示下式中的第 j 比特位，

$$c_j=H_l(g\|h\|V_1\|V_2\|\cdots\|V_n\|A\|B_1\|B_2\|\cdots\|B_n\|t_{h1}\|\cdots\|t_{hl}\|t_{g1}\|\cdots\|t_{gl}) \quad (1)$$

验证者利用已知的 $c_j,r_j,A,B_i,h,g,V_i,y_i$，其中 $i=1,\cdots,n$，$j=1,\cdots,l$，计算：$t_{hj}=h^{r_j}A^{c_j}\pmod p$，$t_{gj}=(g^{1-c_j}V_i^{c_jB_i})^{y_i^{r_j}}$，将其代入式(1)，看是否成立。若成立，则说明分发者 D 分发的份额是正确的。

P_i 生成其公私钥对 $(a_i^{(0)},b_i^{(0)})$，广播公钥 $b_i^{(0)}$。

每个 P_i 都拥有一个所有服务器成员份额承诺的集合 $\{\psi_j=g^{x_j^{(0)}}\pmod p,(j=1,\cdots,n)\}$。

2.2 私密钥更新协议(Pku)

此协议在份额更新协议和份额重建协议之前运行，运行如下：

P_i 选择新的公私钥对 $(a_i^{(t)},b_i^{(t)})$ 并广播公共密钥 $b_i^{(t)}$，用 $a_i^{(t-1)}$ 签名 $b_i^{(t)}$ 并广播，其他服务器成员用 $b_i^{(t-1)}$ 验证 P_i 对 $b_i^{(t)}$ 的签名。

2.3 秘密份额更新协议(Sku)

协议中度为 k 的多项式的构造方法如下：

选择度为 k 的多项式 $\delta(\cdot)$，其中 $\delta(0)=0$，那么 $f^{(t)}(0)=f^{(t-1)}(0)+\delta(0)=x+0$，每个服务器成员的份额是 $x_i^{(t)}=f^{(t)}(i)=f^{(t-1)}(i)+\delta(i)\pmod q$。构造 $\delta(\cdot)=\delta_1(\cdot)+\delta_2(\cdot)+\cdots+\delta_n(\cdot)\pmod q$，其中 $\delta_i(\cdot)$ 是每一位服务器成员 P_i 自己构造的度为 k 的常数项为零的多项式，也即 $\delta_i(0)=0$。

下面详细介绍在第 t 个时间段可公开验证秘密份额更新协议：

(1)P_i 任意构造度为 k 的多项式 $\delta_i(z)=\delta_{i1}z^1+\delta_{i2}z^2+\cdots+\delta_{ik}z^k$。

(2)P_i 计算 $u_{ij}=\delta_i(j)\pmod q$，并根据 u_{ij} 计算 $V_{ij}=g^{u_{ij}}\pmod p$，其中 $j=1,\cdots,n$。

(3)服务器成员 P_i 公开广播 $(h^{z_i},u_{ij}^{-1}y_j^{z_i})\pmod p$，其中 z_i 是 P_i 的私密钥，y_j 是 P_i 的公共密钥。

(4)当 P_j 接收到 (A_i,B_{ij})，解密 $u_{ij}=A_i^{z_j}/B_{ij}$。

(5)如果 (A_i,B_{ij}) 等于 $(h^{z_i},u_{ij}^{-1}y_j^{z_i})$，那么 $V_{ij}^{B_{ij}}=(g^{u_{ij}})^{u_{ij}^{-1}\cdot y_j^{z_i}}=g^{y_j^{z_i}}$。

(6)P_i 随机取 l 个数 $(l\approx100)w_{ijk}(k=1,\cdots,l)$ 计算

$$t_{hk}=h^{w_{ijk}}\pmod p,t_{gk}=g^{y_j^{w_{ijk}}},$$

$$R=(r_{ij1},r_{ij2},\cdots,r_{ijl})=(w_{ij1}-c_{ij1}z_i\pmod q,w_{ij2}-c_{ij2}z_i\pmod q,\cdots,w_{ijl}-c_{ijl}z_i\pmod q)$$

式中，c_{ijk} 是下列式中的第 k 比特位。

$$c_{ij}=H_{il}(V_{ij}\|A_i\|B_{ij}\|t_{h1}\|\cdots\|t_{hl}\|t_{g1}\|\cdots\|t_{gl}) \quad (2)$$

式中，H_{il} 是输出为 l 位的 Hash 函数。

(7) 验证者计算：$t_{hk}=h^{r_{ijk}}A_i^{c_{ijk}}\pmod p$，$t_{gk}=(g^{r_{ijk}c_{ijk}}V_{ij}^{c_{ijk}B_{ij}})^{y_j^{r_{ijk}}}$ $(k=1,\cdots,l)$。将其代入式(2)，看是否成立。

(8)如果 P_i 验证所有信息都是合法的，他就会向其他人广播说“他得到的信息是正确的”。若不正确就执行指控核实协议。

(9)如果所有服务器收到的信息都是正确信息，那么他们会各自更新自己的份额 $x_i^{(t)}=x_i^{(t-1)}+(u_{1i}+\cdots+u_{ni})\pmod q$，并计算 $\psi_j^{(t)}=\psi_j^{(t-1)}\prod_{l\in A-B_i}g^{u_{lj}}\pmod p$，销毁 $x_i^{(t-1)}$ 和接收到的其它有关信息。

(10)若有服务器成员发现另外一个服务器成员给他传送的信息不正确，他将有两种处理方法，其一：不让其参与第(9)步；其二：重启他的服务器让其重新生成多项式。

2.4 指控核实协议(Acc)

当 P_i 发出的消息错误时，第一种是错误的时间段、数据越界等；第二种是同一服务器发送多条含有有效签名的信息或根本就没有发信息；第三种是等式(2)不成立。

这3种错误任何其他服务器成员都可以检测。对于第一种和第二种错误，每个服务器成员 $P_i(i\neq j)$ 把 j 放入"坏服务器"集合中，即 $B_i \leftarrow B_i \cup \{j\}$，启动"reboot"操作，并调用恢复损坏秘密份额协议。对于第三种错误：当 P_i 发现 P_j 发出的信息不满足式(2)时，其他服务器也能够通过式(2)验证出谁在说谎，若是 P_i 说谎，对 P_i 启动 reboot 操作，并调用恢复损坏秘密份额协议；否则，对 P_j 进行以上操作。

2.5 损坏秘密份额发现协议(Bsd)

若某些服务器没有参与份额更新阶段，或被敌手攻击但没有被发现，则需要定期地对所有服务器进行检测。

每个 P_i 都广播 $\{\psi_j^{(t)}\}_{j\in\{1,\cdots,n\}}$ 和对其的签名。P_i 得到所有的广播后，检测签名的正确性，再通过比较每一个服务器和大多数服务器的不同决定哪个服务器是不诚实的，并将这些服务器加入到 B_i 中，调用恢复损坏秘密份额协议进行恢复。

2.6 恢复损坏秘密的份额(Ssr)

需要恢复的秘密份额 x_r，$r\in B$，假定集合 $D=A\backslash B$。

(1)每个服务器成员 P_i $i\in D$ 在 $\mathbb{Z}_q$ 上构造一个最高次项为 k 阶的多项式 $\delta_i(\cdot)$ 且满足 $\delta_i(i)=0$。构造方法：任取多项式系数 $\{\delta_{ij}\}_{j=\{1,\cdots,k\}}\subset\mathbb{Z}_q$ 那么 $\delta_{i0}=-\sum_{j\in\{1,\cdots,k\}}\delta_{ij}r^j \pmod q$。对每个服务器成员 $P_i(i\in D)$，把 $\delta_i(j)$ 作为秘密份额利用可公开验证思想分发给每个 P_j，$j\in D$ 且 $j\neq i$。

秘密份额分发如下：

①P_i 任意构造度为 k 的多项式 $\delta_i(z)=\delta_{i0}+\delta_{i1}z^1+\delta_{i2}z^2+\cdots+\delta_{ik}z^k \pmod q$。

②P_i 计算 $u_{ij}=\delta_i(j) \pmod q$，$V_{ij}=g^{u_{ij}} \pmod p$，其中 $j=1,\cdots,n$。

③服务器成员 P_i 公开广播 $(h^{z_i},u_{ij}^{-1}y_j^{z_i},V_{ij}) \pmod p$，其中 z_i 是 P_i 的私密钥，y_j 是 P_i 的公共密钥。

④当 P_j 接收到 (A_i,B_{ij},V_{ij}) 解密 $u_{ij}=A_i^{z_j}/B_{ij}$。

⑤如果 (A_i,B_{ij}) 等于 $(h_j^{z_i},u_{ij}^{-1}y_j^{z_i})$，那么 $V_{ij}^{B_{ij}}=(g^{u_{ij}})^{u_{ij}^{-1}\cdot y_j^{z_i}}=g^{y_i^{z_i}}$。

⑥P_i 随机取 l 个数 $(l\approx 100)$ w_{ijk} $(k=1,\cdots,l)$ 计算

$t_{hk}=h^{w_{ijk}} \pmod p$，$t_{gk}=g^{y_j^{w_{ijk}}}$。

$R=(r_{ij1},r_{ij2},\cdots,r_{ijl})=(w_{ij1}-c_{ij1}z_i \pmod q, w_{ij2}-c_{ij2}z_i \pmod q,\cdots,w_{ijl}-c_{ijl}z_i \pmod q)$

式中，c_{ijk} 是下列式中的第 k 比特位。

$$c_{ij}=H_{il}(V_{ij}\|A_i\|B_{ij}\|t_{h1}\|\cdots\|t_{hl}\|t_{g1}\|\cdots\|t_{gl}) \quad (3)$$

式中，H_{il} 是输出为 l 位的 Hash 函数。

⑦验证者计算：$t_{hk}=h^{r_{ijk}}A_i^{c_{ijk}} \pmod p$，$t_{gk}=(g^{r_{ijk}c_{ijk}}V_{ij}^{c_{ijk}B_{ij}})^{y_j^{r_{ijk}}}$ $(k=1,\cdots,l)$。代入式(3)中验证是否成立。

⑧如果 P_i 验证所有信息都是合法的，他就会向其他人广播说"他得到的信息是正确的"。若不正确，执行指控核实协议。

(2)对每个服务器成员 $P_i(i\in D)$，根据接收到的 u_{ji}，计算 $x_i'=x_i+\sum_{j\in D}\delta_j(i)$，并把 $ENC_{b_r^{(t)}}(x_i')$ 发送给 P_r。

(3)P_r 根据 $P_i(i\in D)$ 发送来的 x_i'，插值算出 x_r。

3 安全性分析和相关工作比较

下面分析所提方案的安全性。

(1)初始化阶段的安全性主要是基于 Shamir 秘密共享的安全性和离散对数问题的难解性。由于离散对数问题是难解的，敌手从公共通道获取的承诺信息难于计算出份额；另外，即使敌手获得了 k 个份额，由于他只知道度为 k 的多项式上的 k 个解，因此他仍然不能构造出秘密多项式和隐含的秘密，所以，初始化阶段是安全的。

(2)私密钥更新协议，服务器成员 P_i 用自己旧的私密钥对新公共密钥进行签名，其他服务器成员解密验证，将得到的结果与 P_i 广播的内容进行比较，若相同，则说明得到的新的公共密钥确实是 P_i 广播的，因为其他人不知道 P_i 的私密钥，不可能伪造，所以是安全的。

(3)秘密份额更新协议和恢复损坏秘密的份额协议，与初始化阶段的安全性一样也是基于 Shamir 秘密共享的安全性和离散对数问题的难解性，所以也是安全的。

(4)指控核实协议，因为所有的验证过程都是可公开验证，其他诚实的服务器成员都可以验证谁在说谎。

(5)因为 $n\geqslant 2k+1$，敌手在每个时间段最多收买 k 个服务器成员，所以绝大多数服务器是诚实的，通过比较每一个服务器与大多数服务器的不同就能够知道哪些服务器是不诚实的，就可以恢复损坏的秘密份额。

表1给出了几种秘密共享方案在相关方面的比较，提出的方案既具有验证功能和可公开验证功能，又具有前摄功能，安全性更高。

表1 相关工作比较

加密方案	是否能验证份额	是否能可公开验证	是否具有前摄功能
Shamir[1]	否	否	否
Stadler[5]	是	是	否
Herzberg[7]	是	否	是
提出的方案	是	是	是

结束语 提出一种新的门限秘密共享方案，方案同时具有可公开验证和前摄能力。提出的方案满足：如果敌手在一个时间段内没有攻破足够多(多于门限)的份额，那么在下一个时间段他得到的份额信息完全失效，使得秘密信息更加安全。我们的方案的安全性是基于离散对数问题、计算安全的。

参考文献

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

[2] Blakley G R. Safeguarding cryptographic keys [C]// Proc. of National Computer Conference. Montvale NJ, AFIPS, 1979

[3] Feldman P. A Practical Scheme for Non-Interactive Verifiable Secret Sharing [C]// Proc. of the 28th IEEE Symposium on the Foundations of Computer Science. California, USA, 1987

[4] Stadler M. Publicly verifiable secret sharing [C]// Advances in Cryptology-EUROCRYPT'96. Saragossa, Spain, 1996

[5] Herzberg A, Jarecki S, Krawczyk H, et al. Proactive secret sharing[C]// Advances in Crytptolgy-Crypto'95. California, USA, 1995

[6] 于佳，郝蓉，孔凡玉，等. 先动的可公开验证服务器辅助秘密共享[J]. 北京邮电大学学报，2008，31(5)：13