周期更新的可验证秘密共享方案*)

魏仕民[1] 许春香[2]
(淮北煤炭师范学院计算机科学与技术系 淮北 235000)[1]
(西安电子科技大学信息保密研究所 西安 710071)[2]

摘 要 在(t,n)秘密共享方案中,秘密的安全性是建立在攻击者在秘密的整个生命周期最多只能获取 t 个秘密碎片。对于长生命周期和敏感的秘密来说,这种保护是不够的。本文基于离散对数问题的难解性,提出一个周期更新的可验证秘密共享方案,方案在秘密信息保持不变的情况下,定期对秘密碎片进行更新。参与者可以对自己的秘密碎片和其他成员出示的秘密碎片进行验证。为了防止秘密碎片的毁坏和丢失,即保证秘密的完整性和可用性,方案还具有检测损坏的秘密碎片和恢复正确的秘密碎片的功能。

关键词 密码学,秘密共享,周期更新,离散对数难题

对于秘密信息,我们可以设想用安全的方法,将其存储在一个地方。但无论保护工作做得多好,只要能够侵入该点就意味着所存储的信息的保密性或认证性降低。如果该存储设备遭受破坏就会使得所存储的信息不可用。改进的最好方法之一就是将秘密信息由不同的成员分享,每一个成员只得到秘密信息的一部分。这就是秘密共享的基本思想。

秘密共享的思想是由 Shamir[1]和 Blakley[2]分别独立提出的。秘密共享系统是将秘密以一种指定资格成员子集可以恢复的方式在一个团体中分享。Shamir 和 Blakley 在提出秘密共享概念的同时,也分别给出了(t,n)门限秘密共享方案。简单地说,设秘密 s 通过秘密分享算法被分发给 n 个成员共享,每一个成员持有一个秘密碎片(也称为影子或子秘密),任何多于 t 个的合格成员通过所持有的正确的秘密碎片都可以重构 s,并且任何 t 个以下的成员集都无法重构 s;称这种方案为(t, n)门限秘密共享方案, t 称为方案的门限值。

Shamir 门限方案是最早提出的,也是研究与应用最广泛的门限方案。在 Shamir 及其他一些门限方案中,秘密的安全性是建立在攻击者在秘密的整个生命周期最多只能获取 t 个秘密碎片。对于长生命周期和敏感的秘密来说,这种保护是不够的。秘密共享方案的一个基本假设是在恢复秘密时,所有成员都给出自己真正的秘密碎片。但是,如果某个或某些成员出示虚假的秘密碎片,则在他们自己得到真正的秘密的同时,可以使其他人得到的却是错误的秘密。因此,在秘密共享体制中还必须设计防欺骗的机制。文[3~7]已经对这些问题进行了探讨。本文基于离散对数问题的难解性,提出一个周期更新的可验证秘密共享方案,方案在秘密信息保持不变的情况下,定期对秘密碎片进行更新;为了防止欺骗,方案还具有验证功能,成员除了可以对自己的秘密碎片进行验证,也能对其他成员和出示的秘密碎片进行验证。为了防止秘密碎片的毁坏和丢失,方案还具有检测损坏的秘密碎片和恢复正确的秘密碎片的功能。

1 新方案

本节我们提出一个新的定期更新的可验证秘密共享方案。我们考虑一个特殊成员 D(可信的分发者)和 n 个秘密共享成员 $P=\{P_1, P_2, \cdots, P_n\}$。通信网络分两部分:各成员间是完全的点到点保密信道,即该信道上传输的消息不能被其他成员阅读或修改;全体成员共享一个广播信道,即若成员 P_i 在该信道上发出一个消息,则所有成员都能收到并且相信消息是来自 P_i 的。该网络是同步的,即全体成员共享同一个时钟。时间间隔单位称为周期,可以为一天、一周、一个月等等。在一个时间周期的开始,成员执行方案约定他在该周期的动作,这段时间称为更新时段。

设 p 是一个大素数,使得 Z_p 中的离散对数问题是难解的,$p=mq+1$,其中 q 也是一个大素数,m 是小的正整数。设 h 是 Z_p 中阶为 q 的元素。

秘密碎片产生和分发:分发者 D 在开始时间周期 $k=0$ 时任选 t 次随机多项式 $f_0 \in Z_q[x]$ 使得 $f_0(0)=s$。也就是在 Z_q 上任选 t 个随机数 $a_{0,1}, a_{0,2}, \cdots, a_{0,t}$,构造 $f_0(x)=a_{0,0}+a_{0,1}x+\cdots+a_{0,t}x^t$,其中 $a_{0,0}=s$。计算 $v_{0,i}=h^{a_{0,i}}$, $s_{0,i}=f_0(i) \bmod q$, $i=1,2,\cdots,n$, $h^{a_{0,0}}, h^{a_{0,1}}, \cdots, h^{a_{0,t}}$。

*)本文受到国家自然科学基金(60172015)、安徽省自然科学基金(03042204)和安徽省教育厅自然科学基金(2004kj317)资助。魏仕民 教授,博士,主要研究方向为网络与信息安全。许春香 副教授,博士研究生,主要研究方向为网络与信息安全。

分别通过保密信道发送秘密碎片 $s_{0,i}$给成员 P_i，$i=1,2,\cdots,n$；并广播 $h^{a_{0,0}}, h^{a_{0,1}}, \cdots, h^{a_{0,t}}$。每个成员 P_i 根据下式验证收到的秘密碎片 $s_{0,i}$是否正确：

$$h^{s_{0,i}} = h^{a_{0,0}}(h^{a_{0,1}})^i(h^{a_{0,2}})^{i^2}\cdots(h^{a_{0,t}})^{i^t} \pmod p,\ i=1,2,\cdots,n, \quad (1)$$

如果等式成立，则收到的碎片 $s_{0,i}$是正确的；否则，则不正确。

秘密碎片更新：分发者在时间周期 $k(k=1,2,3,\cdots)$的更新时段，任选 t 次随机多项式 $f_k(x)\in Z_q[x]$为 $f_k(x)=a_{k,1}x+\cdots+a_{k,t}x^t$，分别保密发送更新碎片 $u_{k,i}=f_k(i) \bmod q$ 给成员 P_i；并广播 $h^{a_{k,1}},\cdots, h^{a_{k,t}}, v_{k,i}=v_{k-1,i}h^{u_{k,i}}, i=1,2,\cdots,n$。

每个成员 P_i 在时刻 k 如下验证收到的秘密碎片 $u_{k,i}$是否正确：

$$h^{u_{k,i}} = (h^{a_{k,1}})^i(h^{a_{k,2}})^{i^2}\cdots(h^{a_{k,t}})^{i^t} \pmod p,\ i=1,2,\cdots,n \quad (2)$$

如果等式成立，则收到的碎片 $u_{k,i}$是正确的；否则，则不正确。当等式成立时，更新自己持有的秘密碎片如下：$s_{k,i}=s_{k-1,i}+u_{k,i}$，并销毁 $s_{k-1,i}$，$i=1,2,\cdots,n$。

秘密重构：如果想重构秘密，各成员 P_i 只需要提供他们在此时刻(不妨为 k)的秘密碎片 $s_{k,i}$。每个成员可以如下验证成员 P_i 提供的秘密碎片 $s_{k,i}$：

$$h^{s_{k,i}} = v_{k,i},\ i=1,2,\cdots,n。 \quad (3)$$

如果等式成立，则收到的碎片 $s_{k,i}$是正确的；反之，则不正确。记 $g_k(x)=f_k(x)+f_{k-1}(x)+\cdots+f_0(x)=b_{k,0}+b_{k,1}x+\cdots+b_{k,t}x^t$，其中，$b_{k,0}=a_{0,0}=s$，$b_{k,1}=a_{k,1}+a_{k-1,1}+\cdots+a_{1,1}$，$\cdots$，$b_{k,t}=a_{k,t}+a_{k-1,t}+\cdots+a_{1,t}$。

则 $g_k(0)=b_{k,0}=s$，且 $g_k(i)=f_k(i)+f_{k-1}(i)+\cdots+f_0(i)=u_{k,i}+u_{k-1,i}+\cdots+u_{1,i}+s_{0,i}=s_{k,i}$，$i=1,2,\cdots,n$。

由于 $g_k(x)$的次数为 t，因而在拥有至少 t+1 个正确秘密碎片的情况下，采用 Langrange 插值法可以重构秘密 s，而且任意少于 t+1 个正确秘密碎片都无法重构秘密 s。

2 新方案的安全性

2.1 新方案的安全性

方案的安全性是基于离散对数难解问题。在秘密分发阶段，如果攻击者想获取秘密，只能够从公开的 $h^{a_{0,0}}, h^{a_{0,1}}, \cdots, h^{a_{0,t}}$开始想办法获取，这需要解离散对数问题。每个成员可以利用式(1)对收到的秘密碎片进行验证，这防止了外部入侵者对合法成员的欺骗。如果这时要恢复秘密，则每个成员都可以利用式(1)对成员 P_i 提供的秘密碎片进行验证，因而可以防止内部成员之间的欺骗。在秘密更新时刻 k，如果攻击者想获取秘密，只能够从公开的 $h^{a_{k,1}}, \cdots, h^{a_{k,t}}, v_{k,i}=v_{k-1,i}h^{u_{k,i}}$，获取相关信息，这需要解离散对数问题。每个成员可以利用式(2)对收到的秘密碎片进行验证，这防止了外部入侵者对合法成员的欺骗，因而每个成员都能够安全更新自己的秘密碎片，并销毁原有碎片。在秘密恢复阶段，每个成员都能够利用式(3)验证成员 P_i 提供的秘密碎片，因而可以防止内部成员之间的欺骗。从式(1)、(2)和(3)攻击者要获取相关秘密信息，也必须解离散对数问题。

2.2 新方案的进一步改进

在我们上面提出的方案中，成员的秘密碎片不能丢失或毁坏。否则，攻击者可以逐渐地毁坏 n-k 个秘密碎片将导致系统无法重构秘密。为此，我们对方案进行进一步的改进，在每个时间周期的更新时段的开始，插入秘密碎片检测和损坏的秘密碎片恢复过程。

在时间周期 k，如何检测成员的秘密碎片呢？最简单的方法就是让各成员通过保密信道出示他们的秘密碎片给分发者，分发者可以通过式(3)来验证，但这使得分发者在此时可以重构秘密。为防止在此时暴露秘密，只要求各成员 P_i 广播 $h^{s_{k-1,i}}$，分发者同样可以通过式(3)来验证。

秘密碎片检测：每个成员 P_i 广播 $h^{s_{k-1,i}}$，分发者验证 $v_{k-1,i}=h^{s_{k-1,i}}$。如果等式成立，则 P_i 的秘密碎片正确，否则被损坏。设秘密碎片正确的成员的下标集为 M，则不正确成员的下标集为 P-M。现在假设成员 P_i 的秘密碎片是不正确的($i\in P-M$)。如何恢复成员 P_i 的秘密碎片呢？最简单的方法就是让各成员通过保密信道出示他们的秘密碎片给分发者，分发者重构多项式 $g_{k-1}(x)$并计算 P_i 的秘密碎片 $s_{k-1,i}=g_{k-1}(i) \bmod q$，但这再次使得分发者可以重构秘密。代替的方法是：当分发者发现 P_i 的秘密碎片不正确时，与其他成员合作，想办法构造 P_i 的秘密碎片 $s_{k-1,i}$的秘密共享，然后将 $s_{k-1,i}$的秘密碎片发送给 P_i，P_i 恢复自己的秘密碎片 $s_{k-1,i}$。

损坏秘密碎片恢复：对每个秘密碎片被损坏的 P_i，$i\in P-M$，分发者选择随机 t 次多项式 $c^i_{k-1}(x)\in Z_q[x]$为 $c^i_{k-1}(x)=c^i_{k-1,0}+c^i_{k-1,1}x+\cdots+c^i_{k-1,t}x^t$，使得 $c^i_{k-1}(i)=0$。分别通过保密信道发送碎片 $w^i_{k-1,j}=c^i_{k-1}(j) \bmod q$ 给成员 $P_j(j\in M)$，并广播 $h^{c^i_{k-1,0}}, h^{c^i_{k-1,1}}, \cdots, h^{c^i_{k-1,t}}$。每个成员 P_j 如下验证收到的碎片 $w^i_{k-1,j}$是否正确：

$$h^{w^i_{k-1,j}} = h^{c^i_{k-1,0}}(h^{c^i_{k-1,1}})^j\cdots(h^{c^i_{k-1,t}})^{j^t} \pmod p \quad (4)$$

如果等式成立，则收到的碎片 $w^i_{k-1,j}$是正确的；否则，则不正确。当等式成立时，P_j 构造 $s_{k-1,i}$的秘密碎片 $s^i_{k-1,i}=w^i_{k-1,j}+s_{k-1,j}$，保密发送 $s_{k-1,i}$的秘

密碎片 $s^j_{k-1,i}$给 P_i，广播 $h^{c^j_{k-1,i}}$。P_i 如下验证收到的碎片 $s^j_{k-1,i}$是否正确：

$$h^{c^j_{k-1,i}} = v_{k-1,i} h^{c^i_{k-1,0}} (h^{c^i_{k-1,1}})^j \cdots (h^{c^i_{k-1,t}})^{j^t} \pmod p, j \in M \quad (5)$$

如果等式成立，则收到的碎片 $s^j_{k-1,i}$是正确的；否则，则不正确。P_i 利用至少 $t+1$ 个正确的碎片可以重构 $s_{k-1,i}$的共享多项式 $d^i_{k-1}(x) = c^i_{k-1}(x) + g_{k-1}(x)$，因为 $d^i_{k-1}(j) = c^i_{k-1}(j) + g_{k-1}(j) = w^j_{k-1,i} + s_{k-1,j} = s^j_{k-1,i}(j \in M)$，且 $d^i_{k-1}(i) = c^i_{k-1}(i) + g_{k-1}(i) = 0 + s_{k-1,i} = s_{k-1,i}$。这恢复了 P_i 的秘密碎片 $s_{k-1,i}$。

现在我们将方案组合为一个完整的方案如下：在时间周期 $k=0$，秘密分发者进行秘密碎片的产生和分发；在时间周期 $k(=1,2,\cdots)$，顺序执行秘密碎片检测、损坏秘密碎片恢复和秘密碎片更新；当系统需要重构秘密时，执行秘密重构。

容易看出，在秘密碎片检测和损坏秘密碎片恢复过程中，攻击者要获取相关秘密信息，也必须解离散对数问题。因此，整个方案的安全性是基于离散对数问题的难解性的。即使在一个时间周期内，$n-k-1$ 个秘密碎片被敌手毁坏或丢失，其余 $k+1$ 个秘密碎片完好无损但有 k 个被敌手偷听，我们的方案仍然可以安全地进行下去。

结论 基于离散对数问题的难解性，提出一个周期更新的可验证秘密共享方案，方案在秘密信息保持不变的情况下，定期对秘密碎片进行更新。参与者可以对自己的秘密碎片和其他成员出示的秘密碎片进行验证。为了保证秘密的完整性和可用性，方案还具有检测损坏的秘密碎片和恢复正确的秘密碎片的功能。方案最大可以预防在一个时间周期内，$n-k-1$ 个秘密碎片被敌手毁坏或丢失，其余 $k+1$ 个秘密碎片完好无损但有 k 个被敌手偷听。由于方案始终有一个可信的分发者参与，与文[3]相比，我们方案的通信量很小，有效性大大提高。

参考文献

1 Shamir A. How to share a secret. Communications of the ACM, 1979, 22(11): 612~613

2 Blakley G R. Safeguarding cryptographic keys. In: Proc. of the National Computer Conf. 1979. Vol. 48 of American Federation of Information Processing Societies Proceedings. 1979. 313~317

3 Herzberg A, Jarecki S, Krawczyk H. Proactive secret sharing or how to cope with perpetual leakage. in Advances in Cryptology: Coppersmith D. In: Proc. of Crypto'95 (vol. 963 of Lecture Notes in Computer Science). Springer-Verlag, 1995. 339~352

4 Gennaro R, Jarecki S, Krawczyk H, Rabin T. Secure distributed key generation for discrete-log based cryptosystems. in Advances in Cryptology: Stern J. In: Proc. of Euro-crypt '99 (vol. 1592 of Lecture Notes in Computer Science). Springer-Verlag, 1999. 295~310

5 张建中，肖国镇. 可防止欺诈的秘密共享方案. 通信学报，2000, 21(5): 81~83

6 张建中，肖国镇. 一个可防止欺诈的秘密分享方案. 电子科学学刊，1999, 21(4): 516~521

7 许春香，魏仕民，肖国镇. 定期更新防欺诈的秘密共享方案. 计算机学报，2002, 25(6): 666~669

(上接第 111 页)

结束语 本文基于 XML 多人数字签名方案实现的电子公文签发系统，已在某市建设规划委员会的电子办公系统中试运行，运行结果表明该多人数字签名方案能够实现多方签名和多方的不可否认性，而且支持同时签名、按任意顺序签名、对部分文档签名。但因为在整个系统内部，所有用户数据信息均以明文形式进行传输，所以本文提出的方案不具有保密功能，对系统内部人员公开。在"电子公文签发系统"中以明文形式传输用户数据，基于两个方面的考虑：(1)如果对全部的用户数据进行加密，势必会增加整个系统的负荷，降低系统的性能；(2)在一般部门的局域网电子办公系统中，电子公文的签名主要是提供电子公文的真实性、权威性，无需对内部人员保密。

如果将该系统扩展到 Internet 上或是应用到某些高度机密的国家安全部门，则可以考虑对明文数据进行加密，使其以密文的形式传送，进一步提高系统的整体安全性。具体方法可选用对称密钥算法中的 DES(或 AES)算法，任何一次通信，都是由发方生成会话密钥，用该密钥对用户数据进行加密生成数据密文，再用接收方的公开密钥(基于 RSA 算法)对会话密钥加密生成密钥密文，然后将得到的密钥密文和数据密文一起发送给接收方。接收方用自己的私有密钥解密密钥密文得到会话密钥，然后再用会话密钥解密数据密文得到用户数据信息，从而实现了数据的保密传输，提高了系统的安全性。

随着 Internet 的发展，数字签名技术的广泛应用以及 XML 文档描述语言的日趋普及，XML 多人数字签名技术将得到进一步的应用与发展。

参考文献

1 陈晋大，郑纪蛟. 用数字签名保护网络通信安全. 计算机应用研究，2000, 9: 43~44

2 王志巍，吴丽红，王天青. 两方间收方不可否认的数字签名. 计算机工程，2001, 27(7): 107~108

3 李凤银，鞠宏伟，刘培玉. 基于 XML 的数字签名技术研究. 山东师范大学学报(自然科学版)，2003, 18(1): 21~23

4 Laurent S S. XML 基础教程[M]. 北京：电子工业出版社，