

基于单向函数的动态密钥托管方案

闫鸿滨^{1,2}, 袁 丁¹

(1. 四川师范大学 计算机科学学院, 四川 成都 610068; 2. 南通职业大学 电子系, 江苏 南通 226007)
(yanbin317@yahoo.com.cn)

摘 要: 动态密钥托管方案, 采用 ElGamal 公钥体制, 利用单向函数不可求逆的安全特性设计, 该方案可用于任意接入结构, 每个托管代理者的子密钥可以多次使用。如果某托管子密钥泄露, 系统只需为其重新分配子密钥, 其他成员的子密钥不必更改。本系统可方便增删托管代理成员, 既具安全性, 又可提高密钥托管的动态性和灵活性。

关键词: ElGamal 公钥体制; 动态密钥托管; 托管代理。

中图分类号: TP309.07 **文献标识码:** A

Dynamic key escrow scheme based on one-way function

YAN Hong-bin^{1,2}, YUAN Ding¹

(1. Department of Computer Science, Sichuan Normal University, Chengdu Sichuan 610068, China;
2. Department of Electron, Nantong Vocational College, Nantong Jiangsu 226007, China)

Abstract: A dynamic key escrow scheme was proposed based on ElGamal public key cryptosystem and one-way function, which could effectually identify cheaters, and could be applicable to arbitrary access structures. The key shadow could be reused for many times. When some key agents' key shadow had been revealed, they could be renewed without any effect on the others. The scheme can accept or fire a key agent easily, which not only is security, but also increases dynamic and flexibility of key escrow.

Key words: ElGamal public key cryptosystem; dynamic key escrow; escrow agent

0 引言

密钥托管的主要思想是将用户密钥分拆为数个片段, 由托管代理分别保管, 获得授权的第三方可以利用托管代理保存的密钥片段恢复出用户的密钥, 解密通信。密钥分拆是密钥托管的关键环节之一。目前已有多种密钥托管方案是基于 (n, n) 门限方案, 如文献[2]中的方案, 这类方案要求所有托管代理共同参与与恢复会话密钥, 在某些条件下这种情况是难以实现的。还有些托管方案使用的是 (k, n) 门限分拆法, 如利用 Shamir (k, n) 门限法分拆密钥进行密钥托管的方案, 这种 (k, n) 门限方案存在如下缺点: 一是不能确切地知道工作失效的托管代理, 密钥恢复时判断重构的密钥是否正确, 需要试凑解密, 这样会影响实时性; 二是为了防止托管代理合作非法恢复密钥而必须保证各个托管代理相互独立时, 很难验证各个托管代理保存的密钥片段的正确性, 而使用户有可能逃避托管。文献[3]中提出的托管方案, 也是基于 (k, n) 门限分拆法, 该方案在保证所有托管者是诚实的前提下, 能成功实施监听, 也能确保用户密钥和托管者子密钥的安全性, 可以避免来自托管者的欺骗, 但是, 由于用户密钥及托管者的子密钥都是由用户独自产生的, 因此, 不能避免来自用户的欺骗行为。另外, (k, n) 门限方案在无形中也增加了各个托管代理者具有完全平等的地位、权利和可靠性的假设, 而现实世界中, 这样的假设往往是难以得到满足的。要满足上述假设必须使用高级门限方案, 而我们需要考虑到在一般的接入结构上进行密钥分拆。文献[4,5]对一般接入结构上的密钥分拆做了研究, 这些方案有较好的安全性, 但因需要托管者保存的信息量太大而效率低, 而且系统也无法灵活地增加和删除新成员。本

文采用文献[6]中给出的密钥分拆体制, 建立了动态密钥托管方案, 弥补了上述不足, 也适于实际应用。

1 动态密钥分拆体制

本方案用到的系统参数: p 为大素数, $GF(p)$ 为相应的有限域, $h(x)$ 为 $GF(p)$ 上的单向函数。 $H = \{H_1, H_2, \dots, H_n\}$ 为系统中的 n 个密钥托管者的集合, Γ 为 H 上的一个单调接入结构, $\Gamma_0 = \{A_1, A_2, \dots, A_t\}$ 是 Γ 的基。密钥分发者 D 首先把 H 以及 A_1, A_2, \dots, A_t 依次向所有托管者公布。

密钥分配:

(1) 分发者 D 随机地选取 n 个互不相同的元素 $s_1, s_2, \dots, s_n \in GF(p)$, 且 $s_i \neq 1 \pmod{p-1}$ ($i = 1, 2, \dots, n$), 并将 s_i 通过安全信道秘密地发送给 H_i , 作为 H_i 拥有的子密钥。

(2) 分发者 D 随机地选取 $GF(p)$ 上的 $n-1$ 次多项式 $F(x) = a_0 + a_1x + \dots + a_{n-1}x^{n-1}$, 使得 $F(0) = s$, 且 $F(x)$ 保密, s 为用户密钥。

(3) 分发者 D 随机地选取 $\alpha \in GF(p)$, 并计算:

$$x_i = h(\alpha + s_i) \pmod{p} \quad (1)$$

$$d_i = (F(I_i) - h(\alpha + s_i)) \pmod{p} \quad (2)$$

$$y_i = h(x_i) \pmod{p} \quad (3)$$

其中: x_i 为 H_i 的屏蔽子密钥, s_i 为 H_i 的秘密子密钥。 I_i 为 H_i 的身份表示符号 (公开), 然后 D 公开参数 α 以及有序数组 (y_1, y_2, \dots, y_n) 与 (d_1, d_2, \dots, d_n) 。 $i = 1, 2, \dots, n$ 。

(4) 分发者 D 对每一个最小合格子集 $A_i = \{H_{i_1}, H_{i_2}, \dots, H_{i_k}\}$, D 由 $(I_{i_1}, F(I_{i_1})), (I_{i_2}, F(I_{i_2})), \dots, (I_{i_k}, F(I_{i_k}))$ 及 $(0, s)$ 共 $k+1$ 个点用 Lagrange 插值公式确定出一个 k 次多项式:

$$F_i(x) = \sum_{j=1}^k F(I_{ij}) \prod_{m=1, m \neq j}^k \frac{x - I_{im}}{I_{ij} - I_{im}} \pmod{p} \quad (4)$$

并计算出 $F_i(\alpha)$, 公开 $F_i(\alpha)$, $i = 1, 2, \dots, t_0$.

密钥恢复:

对任何一个合格子集 A , 设它包含的最小合格子集为 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$, A_i 中的每一个成员把自己的屏蔽子密钥秘密地发送给 A 中的其他各成员, 汇集到 A_i 中所有成员的屏蔽子密钥后, 对每一个屏蔽子密钥 x_{ij} , 可以验证等式 $y_{ij} = h(x_{ij}) \pmod{p}$ 是否成立, 若成立, 则提供的是有效份额。从而 $F_{ij} = (d_{ij} + x_{ij}) \pmod{p}$ 也是有效的, 故由 $(I_{i1}, F(I_{i1})), (I_{i2}, F(I_{i2})), \dots, (I_{ik}, F(I_{ik}))$ 以及 $(\alpha, F_i(\alpha))$ 为 k 次多项式 $F_i(x)$ 上的 $k+1$ 个点。因此, 每一个最小合格子集中的成员可以利用他们的有效份额, 按照 Lagrange 插值公式正确恢复出公式(4), 并不难通过计算得出:

$$\begin{aligned} s &= F_i(0) \\ &= \sum_{j=1}^k F_i(I_{ij}) \prod_{m=1, m \neq j}^k (I_{im}(I_{im} - I_{ij})^{-1}) \pmod{p} \\ &= \sum_{j=1}^k F_i(I_{ij}) C_i \end{aligned} \quad (5)$$

其中, $C_i = \sum_{m=1, m \neq j}^k (I_{im}(I_{im} - I_{ij})^{-1}) \pmod{p}$ 。

份额的验证算法在恢复密钥时, 可以用来检测提供虚假份额的恶意参与者。

2 系统描述

在我们所讨论的密码系统中, 假设用户采用传统的密码体制(比如 DES)来加密消息 M , 所使用的会话密钥是 k_A, k_B 由 ElGamal 公钥密码体制来加密传送。方案系统中有一个密钥管理中心(KMC)负责颁发通信用户的公钥证书; 有 n 个委托代理者的集合 $H = \{H_1, H_2, \dots, H_n\}$ 负责托管用户用来保护 k_A 的 ElGamal 私钥 s ; 有一个法律授权机构负责监听授权; 有一个监听机构负责实施对用户通信的监听。

3 动态密钥托管方案

想利用该系统通信的用户, 首先要向密钥管理中心注册申请公钥证书, 密钥管理中心选择一个大的安全的素数 p 和 $GF(p)$ 的一个本原元 g ; 利用下面的协议生成用户的 ElGamal 私钥及公钥证书, 并实施对用户的私钥 s 进行托管。

第一步: 用户 A 随机选取 $s' \in Z_p$, 计算 $y = g^{s'} \pmod{p}$, 并将 y 传送给密钥管理中心。

第二步: 密钥管理中心随机选取 $t, s'' \in Z_p$, 使 $Y \equiv g^{s''} y \neq 1 \pmod{p}$, $y_1 \equiv g^t \pmod{p}$, $y_2 \equiv (s'' y^t) \pmod{p}$ 。公开 (p, g, Y) , 并将 (y_1, y_2) 传送给用户 A 。

第三步: 用户 A 计算 $s'' \equiv (y_2 (y_1')^{-1}) \pmod{p}$, $s \equiv (s' + s'') \pmod{(p-1)}$, 若 $s = 0$, 则重新申请公钥证书, 否则 s 作为用户 A 的 ElGamal 私钥。然后用户 A 做如下工作:

(1) 用户 A 随机地选取 n 个互不相同的元素 $s_1, s_2, \dots, s_n \in GF(p)$, 且 $s_i \neq 1 \pmod{(p-1)}$ ($i = 1, 2, \dots, n$), 并将 s_i 通过安全信道秘密地发送给 H_i , 作为 H_i 拥有的子密钥。

(2) 用户 A 随机选取 $GF(p)$ 上的 $n-1$ 次多项式 $F(x) = a_0 + a_1 x + \dots + a_{n-1} x^{n-1}$, 使得 $F(0) = s$, 且 $F(x)$ 保密。

(3) 用户 A 随机选取 $\alpha \in GF(p)$, 并依次计算公式(1)、(2)、(3), 其中 x_i 为 H_i 的屏蔽子密钥, s_i 为 H_i 的秘密子密钥。 I_i 为 H_i 的身份表示符号(公开), 然后 D 公开参数 α 以及有序数组 (y_1, y_2, \dots, y_n) 与 (d_1, d_2, \dots, d_n) 。

(4) 对每一个最小合格子集 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$, 用户

A 由 $(I_{i1}, F(I_{i1})), (I_{i2}, F(I_{i2})), \dots, (I_{ik}, F(I_{ik}))$ 及 $(0, s)$ 共 $k+1$ 个点用 Lagrange 插值公式确定出一个 k 次多项式 $F_i(x)$, 并计算出 $F_i(\alpha)$, 公开 $F_i(\alpha)$, $i = 1, 2, \dots, t_0$ 。

第四步: A_i 中的每个托管代理收到 x_{ij} 及 s_{ij} 后, 验证 $x_{ij} = h(\alpha + s_{ij}) \pmod{p}$ 及 $y_{ij} = h(x_{ij}) \pmod{p}$ 是否成立。若成立, 则计算 $ss = \text{sig}H_i(ID_A, y_{ij}, z_{ij} = g^{h(x_{ij})} \pmod{p})$, 并将 $(ID_A, y_{ij}, z_{ij}, ss)$ 传给管理中心 KMC, 否则不进行签名。

第五步: 密钥管理中心 KMC 收到每个委托代理的 $(ID_A, y_{ij}, z_{ij}, ss)$ 后, 通过验证签名和 $z_{ij} = g^{y_{ij}} \pmod{p}$ 是否成立来确定 $(ID_A, y_{ij}, z_{ij}, ss)$ 的有效性, 若全部有效, 则计算签名 $s_k = \text{SigKMC}(h(ID_A, p, g, Y))$, 并颁发用户 A 的公钥证书 $C(A) = (ID_A, p, g, Y, s_k)$, 否则, 则告知用户 A 注册失败。

4 用户间的通信

当用户 B 欲向 A 发送秘密消息 M 时, 用户 B 首先要从 KMC 或用户 A 处获取用户 A 的公钥证书 $C(A)$, 然后在随机的选取 $K_B, L \in Z_p$, K_B 作为加密消息 M 的会话密钥, 计算 $y_1 = g^L \pmod{p}$, $y_2 = (K_B Y^L) \pmod{p}$, $s_B = \text{sig}B(h(y_1, y_2, \text{Time}, ID_A, ID_B))$, 同时计算 $LEAF = (y_1, y_2, \text{Time}, ID_A, ID_B, s_B)$, 用传统密码体制把消息 M 加密 $C = E(M, K_B)$, 并把 $(LEAF, C)$ 传送给 A , 用户 A 收到 $(LEAF, C)$ 后, 计算 $K_B = y_2 (y_1')^{-1} \pmod{p}$ 得到 K_B , 然后用 K_B 解出明文 $M = D(C, K_B)$ 。

5 监听过程

(1) 监听机构首先得到监听用户 A, B 间通信的许可证书, 并将此证书和监听到的 $LEAF = (y_1, y_2, \text{Time}, ID_A, ID_B, s_B)$ 出示给任意一个最小合格子集 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$ 中的每个托管代理 H_{ij} 。

(2) 托管代理 H_{ij} 验证了证书的有效性后, 计算 $G_{ij} = (y_1^{x_{ij}} y_1^{d_{ij}}) \pmod{p}$, 并把 G_{ij} 传送给监听机构。

(3) 监听机构收到 $G_{i1}, G_{i2}, \dots, G_{ik}$ 后, 通过计算:

$$\begin{aligned} G &= \prod_{j=1}^k G_{ij}^{C_i} = \prod_{j=1}^k y_1^{(x_{ij} + d_{ij}) C_i} \\ &= y_1^{\sum_{j=1}^k F_i(I_{ij}) C_i} = y_1^s = g^{sL} \pmod{p} \end{aligned} \quad (6)$$

由 $K_B = (y_2 G^{-1}) \pmod{p}$ 恢复出 K_B , 再用 K_B 解出明文 $M = D(C, K_B)$, 从而实现对用户 A 与用户 B 间通信的监听, 或者用以下方法验证: 监听机构任选两个最小合格子集 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$ 和 $A_j = \{H_{j1}, H_{j2}, \dots, H_{jm}\}$, 把监听许可证书提供给两个最小合格子集中的所有托管代理, 监听机构在汇集到两个最小合格子集中的所有托管代理的屏蔽子密钥并验证有效后, 则由 A_i 中成员的份额恢复出的密钥与由 A_j 中的成员的份额所恢复出的密钥必定相同。(证明与文献[7]中的类似)。如此, 监听机构可以得到 $(I_{i1}, F(I_{i1})), (I_{i2}, F(I_{i2})), \dots, (I_{ik}, F(I_{ik}))$ 及 $(\alpha, F_i(\alpha))$ 为 k 次多项式 $F_i(x)$ 上的 $k+1$ 个点, 按照 Lagrange 插值公式正确恢复出用户的本次通信所用密钥 K_B , 然后从监听到的 $(LEAF, C)$ 中, 取出 $C = E(M, K_B)$, 用密钥 K_B 解出明文 $M = D(C, K_B)$, 从而实现对用户 A 与用户 B 间通信的监听。

6 性能分析

(1) 攻击者模型及困难性假设

我们假定攻击者可以勾结任何分享者, 但每一个合格子集中至少有一名成员不能被勾结, 同时至少有一个合格子集, 其中的所有成员都不会被勾结。攻击者在协议开始之前就已经确定好了要勾结那些分享者。为方便起见, 我们称被攻击者

勾结的分享者是腐败的或不诚实的。攻击者可以得到腐败的分享者所拥有的任何秘密信息。

(2) 安全性

命题1 对任何两个最小合格子集 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$ 和 $A_j = \{H_{j1}, H_{j2}, \dots, H_{jm}\}$, 如果 A_i 和 A_j 中的成员都收到有效的份额, 并且 $F_i(\alpha)$ 和 $F_j(\alpha)$ 也有效, 那么由 A_i 中成员的份额所恢复出的秘密与由 A_j 中成员的份额所恢复出的秘密必定相同。

证明 由份额的分配算法及秘密的恢复算法可知, 如果最小合格子集 A_i 和 A_j 中的成员的份额及公开信息 $F_i(\alpha)$ 和 $F_j(\alpha)$ 都有效, 那么 $(I_{i1}, F(I_{i1})), (I_{i2}, F(I_{i2})), \dots, (I_{ik}, F(I_{ik}))$ 及 $(\alpha, F_i(\alpha))$ 是 k 次多项式 $F_i(x)$ 上的 $k+1$ 个点, $(J_{j1}, F(J_{j1})), (J_{j2}, F(J_{j2})), \dots, (J_{jm}, F(J_{jm}))$ 及 $(\alpha, F_j(\alpha))$ 是 m 次多项式 $F_j(x)$ 上的 $m+1$ 个点, 因而它们各自唯一地确定出 $F_i(x)$ 和 $F_j(x)$ 。于是它们各自恢复出秘密 $F_i(0)$ 和 $F_j(0)$, 由分配算法可知 $F_i(0) = F_j(0) = s_0$ 。证毕

命题2 攻击者无法恢复秘密而且不能阻止由诚实的分享者构成的合格子集正确地恢复秘密。

证明 设 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$ 是任一合格子集, 由于攻击者无法得到 A_i 的全体成员的所有份额, 他最多只能知道与 A_i 相应的 k 次多项式 $F_i(x)$ 上的 k 个点 (其中包含了一个公开点 $(\alpha, F_i(\alpha))$), 由这些点无法确定出 $F_i(x)$, 也无法确定出 $F_i(x)$ 上的其他任何一个点。因此, 攻击者无法恢复出秘密。另外, 由于至少存在一个合格子集, 其中的所有分享者都是诚实的, 攻击者无法阻止这样的合格子集正确地恢复秘密。证毕

命题3 攻击者所获得的信息是独立于被分享的秘密 s 的。即我们设计的密钥托管协议是安全的。

证明 首先, 从分发者分发秘密份额所使用的多项式 $F(x)$ 来说, 由于 $F(x)$ 的次数为 $n-1$, 而攻击者所能知道的 $F(x)$ 上的点的个数不超过 $n-1$, 因此, 攻击者不能获得关于秘密 s 的任何信息; 其次, 从对应于每一最小合格子集 $A_i = \{H_{i1}, H_{i2}, \dots, H_{ik}\}$ 的多项式 $F_i(x)$ 来说, $F_i(x)$ 的次数为 k , 而攻击者最多只能知道 $F_i(x)$ 上的 k 个点, 因此, 他不能得到关于 $F_i(x)$ 的常数项 s 的任何信息。

综上所述, 攻击者所获得的信息是独立于被分享的秘密 s 的。即设计的基于单向函数的密钥托管协议是安全的。证毕

由于在恢复密钥时, 每个托管代理提交的是其屏蔽子密钥, $x_i = h(\alpha + s_i) \pmod{p}$ 。根据单向函数不可求逆的特性, 其他人无法通过 x_i 求出 H_i 的子密钥 s_i , 即每个托管代理的子密钥并没有因为通信密钥的恢复而被公开, 从而可以继续使用。

同样, 任何人也无法通过公开信息 α 及有序数组 $(y_1, y_2,$

$\dots, y_n)$ 与 (d_1, d_2, \dots, d_n) 来获取托管代理的屏蔽子密钥及秘密子密钥。

另外, 当某个成员提供他的屏蔽子密钥后, 可以通过验证等式 $y_i = h(x_i) \pmod{p}$ 是否成立, 来判断该成员提供的份额是否有效, 从而可以检验恶意参与者。

(3) 动态性分析

如果用户 A 要更换密钥, 用户 A 只需和密钥管理中心重新协商密钥 s' , 用户再重新选择一个 $\alpha' (\alpha' \neq \alpha)$ 及一个新的 $n-1$ 次多项式 $F'(x)$, $F'(x) \neq F(x)$, 满足 $F'(0) = s'$ 为新密钥, 然后利用新的 α' 及 $F'(x)$ 更新公开的 α 与 $F_i(\alpha)$ 及有序数组 (y_1, y_2, \dots, y_n) 与 (d_1, d_2, \dots, d_n) , 而不必更换托管代理的子密钥; 当有新成员 H_{n+1} 加入时, 用户 A 只需随机生成一个 s_{n+1} 作为 H_{n+1} 的秘密子密钥。分配密钥时, 完全可以按照文中的方案进行, 而无须更改其他成员的子密钥; 当删除某个成员时, 用户 A 只需在其余 $n-1$ 个成员中分配密钥即可, 也无须更改其他成员的子密钥。

7 结语

本文的动态密钥托管方案采用 ElGamal 公钥体制设计, 由用户和密钥管理中心共同参与来产生用户的密钥, 解决了文献[3]中的方案存在的问题; 在进行密钥分配时, 由用户、密钥管理中心和托管代理共同参与, 既能保证托管内容的有效性、安全性, 又能避免来自托管者的欺骗, 确保了合法监听活动的有效实施; 本方案应用可验证的动态密钥分拆体制, 不但减少了各个托管代理者的地位、所拥有的权利及可靠性的差别, 使得各托管代理在协议中所起的作用完全对等, 能应用于更一般的接入结构; 用户间通信时, 可以动态选择会话密钥和恢复该密钥所用的多项式; 方案恢复会话密钥时, 可以对任意指定的托管方成员增加、减少、更换, 而不必更改托管的密钥碎片, 减少了通信量, 提高了效率。

参考文献:

- [1] DENNING DE, SMID M. Key escrowing today[J]. IEEE Communication Magazine, 1994, 32(9): 55-68.
- [2] 宋荣功, 詹榜华, 胡正名. 基于多层次可验证共享协议的密钥托管方案[J]. 电子学报, 1999, 27(6): 136-137.
- [3] 蒋绍权, 张玉峰. 部分密钥托管的监听体制[J]. 软件学报, 2000, 11(8): 1133-1137.
- [4] GENNARO R. Theory and practice of verifiable secret sharing[D]. Massachusetts Institute of Technology (MIT); Cambridge, 1996.
- [5] 张福泰, 王育民. 无条件安全的广义可验证秘密分享协议[J]. 计算机研究与发展, 2002, 39(10): 1199-1024.
- [6] 何业峰, 张建中. 基于单向函数的广义动态秘密分享方案[J]. 贵州大学学报(自然科学版), 2003, 20(4): 358-360.

(上接第 1083 页)

(3) UTD-IDS 通过协同预警原理可以使低安全级别的 IDS 可以对其未知入侵进行预警, 防止入侵对目标系统的进一步的攻击损害。而传统 IDS 的预警功能实现主要是分析入侵, 并根据入侵的特点来检测将要到来的入侵。

参考文献:

- [1] ZHANG J, DING Y, GONG J. Intrusion detection system based on fuzzy default logic[A]. FUZZ-IEEE 2003[C]. St. Louis Missouri USA, 2003, 2(12243): 1350-1356.
- [2] HUANG MY, WICKS TM. A large scale distributed intrusion detection framework based on attack strategy analysis[J]. Computer Networks, 1999, 31(23-24): 2465-2475.
- [3] YANG W, FANG BX, LIU B, et al. Intrusion detection system for high-speed network[J]. Computer Communications, 2004, 27(13):

1288-1294.

- [4] DASARATHY BV. Intrusion detection[J]. Information Fusion, 2003, 4(4): 243-245.
- [5] ULVILA JW, GAFFNEY J, JOHN E. Evaluation of intrusion detection systems[J]. Journal of Research of the National Institute of Standards and Technology, 2003, 108(6): 453-473.
- [6] SIN LN, LEE MC. Intrusion detection system models[A]. Proceedings of the International Conference on Security and Management [C]. v 1, Proceedings of the International Conference on Security and Management, SAM 2003, 2003. 359-364.
- [7] JIANG WB, SONG H, DAI YQ. Real-time intrusion detection for high-speed networks[J]. Computers & Security, 2005, 24(4): 287-294.