第36卷 第20期 计 算 机 工 程 2010年10月
Vol.36 No.20 Computer Engineering October 2010
·安全技术· 文章编号：1000—3428(2010)20—0153—02 文献标识码：A 中图分类号：TP309

基于密钥协商的门限多秘密共享方案

杨 捷，李继国
(河海大学计算机与信息工程学院，南京 210098)

摘 要：为使攻击者在骗取所有共享者的子秘密时无法恢复秘密，提出一种基于密钥协商的门限多秘密共享方案。该方案使系统内的成员相互制约，子秘密由共享者自己选取，且可重复使用，系统不需要秘密信道。共享者可以方便地动态加入或离开系统，且所有秘密能一次性恢复。

关键词：密钥协商；门限方案；秘密共享；安全性

Threshold Multi-secret Sharing Scheme Based on Key Agreement

YANG Jie, LI Ji-guo
(College of Computer & Information Engineering, Hohai University, Nanjing 210098, China)

【Abstract】 This paper proposes a threshold multi-secret sharing scheme based on key agreement to assure that secret data can not restored by attackers have diddled subsecrets from all the participants. The cheating of dealer and the cheating of participants can be detected. The subsecrets are chosen by the participants themselves and can be used repeatedly. System does not need a security channel. It is easy for the participants to join or leave system dynamically, and all the secrets can be recovered in one time.

【Key words】 key agreement; threshold scheme; secret sharing; security

1 概述

文献[1]利用Lagrange内插多项式给出了第1个(t, n)门限秘密共享方案。该方案把秘密数据分拆成n个子秘密，分交n个人保管，使得当有t个或t个以上的人进行合作提供子秘密时，可恢复原先的秘密数据。该秘密共享的思想为秘密数据的安全存放提供了一个科学框架。三十年来，许多学者对秘密数据安全存放进行了大量研究，秘密共享已成为密码学与信息安全领域一个重要的研究课题。人们针对秘密共享的应用及自身结构上的创新，提出了许多很好的方案。在防止外来人员的攻击、内部成员的欺诈、多秘密的一次性恢复、共享成员的动态变化、计算量的大小、数据传输的安全性等方面不断有改进[2-7]。本文介绍的方案除了兼有上述各个方面的较多优点外，还具有攻击者即使得到足够多共享者的子秘密也无法恢复秘密数据的特点。

2 方案的构成

2.1 系统初始化

设$k_1,k_2,\cdots,k_l$为一组共享秘密，$U_1,U_2,\cdots,U_n$为共享参与者，D为秘密分发者。D取大素数p、q满足$q \mid p-1$，且$g \in Z_p^*$，g的阶为q，$k_1,k_2,\cdots,k_l \in Z_q^*$。$D$取随机数$\alpha \in Z_q^*$，计算$\beta = g^{\alpha} \bmod p$，$D$以$\alpha$为私钥，$\beta$为公钥。$U_i$取随机数$\alpha_i \in Z_q^*$，计算$\beta_i = g^{\alpha_i} \bmod p$，$U_i$以$\alpha_i$为私钥，$\beta_i$为公钥。$U_i$将$\beta_i$传给$D$，$i=1,2,\cdots,n$。秘密恢复者取随机数$\alpha_0 \in Z_q^*$，计算$\beta_0 = g^{\alpha_0} \bmod p$，并将$\beta_0$传给$D$。秘密恢复者可以是$n$个共享者之一，也可以不是。$D$检查$\beta,\beta_0,\beta_1,\cdots,\beta_n$是否互不相同，如果不是，则$D$通知$U_i$重新选取私钥并计算相应的公钥，直到$\beta,\beta_0,\beta_1,\cdots,\beta_n$互不相同为止。$h:(0,1)^* \to Z_q$为抗碰撞的单向的Hash函数。

本方案需要一个公告牌，D可在公告牌上发布或更新信息，他人能阅读下载并可提出质疑。在初始化阶段，D在公告牌上公布(p,q,g,h,β,β_0)。

2.2 插值多项式的生成

插值多项式的生成过程如下：

(1)共享者U_i取$u_i \in Z_q^*$为身份信息，计算$b_i = \beta_0^{\alpha_i} \bmod p$，$U_i$将$(u_i, b_i)$传给$D$，$i=1,2,\cdots,n$。

(2)D检查$u_1,u_2,\cdots,u_n$是否互不相同，如果$i \neq j$、$u_i = u_j$，则通知U_i、U_j重新选取身份信息，直到n个身份信息互不相同为止。

(3)在确认n个身份信息互不相同后，D计算$b_i' = (b_i^{\alpha} \bmod p) \bmod q$，$i=1,2,\cdots,n$。

(4)D以$(1,k_1),(2,k_2),\cdots,(l,k_l)$与$(u_i,b'_i)$$(i=1,2,\cdots,n)$为插值点，作$Z_q$上的$(n+l-1)$次多项式：

$$f(x) = a_0 + a_1 x + \cdots + a_{n+l-1} x^{n+l-1}$$

(5)D取$t<n$并取互不相同的数$r_1,r_2,\cdots,r_{n+l-t} \in Z_q^* \setminus \{1,2,\cdots,l,u_1,u_2,\cdots,u_n\}$，计算$f(r_i)$ $i=1,2,\cdots,n+l-t$，并将$(r_i, f(r_i))$$(i=1,2,\cdots,n+l-t)$公布于公告牌。

(6)D计算$h_i = h(u_i \| \beta_i^{\alpha} \bmod p)$$(i=1,2,\cdots,n)$、$H_i = h(u_i \| b_i')$$(i=1,2,\cdots,n)$和$k_0 = h(k_1 \| k_2 \| \cdots \| k_l)$，将它们公布于公告牌。

2.3 秘密恢复阶段

秘密恢复过程如下：

(1)设参与恢复秘密的t个共享者为$U_i$$(i=1,2,\cdots,t)$。$U_i$

基金项目：国家自然科学基金资助项目(60842002, 60673070)；国家“863”计划基金资助项目(2007AA01Z409)

作者简介：杨 捷(1972－)，男，硕士研究生，主研方向：信息安全，密码学及其应用；李继国，教授、博士、博士生导师

收稿日期：2010-05-22 **E-mail：**lcm0617@126.com

向恢复者提供$(u_i,\tilde{b}_i)$，$\tilde{b}_i=\beta^{\alpha_i} \bmod p$ $(i=1,2,\cdots,t)$。

(2)恢复者或其他任何人都可验证等式$h(u_i \| \tilde{b}_i)=h_i$是否成立。由于$\tilde{b}_i=\beta^{\alpha_i} \bmod p=g^{\alpha\alpha_i} \bmod p=\beta_i^{\alpha} \bmod p$，因此如果验证等式不成立，则说明$(u_i,\tilde{b}_i)$不是真实的。$U_i$对恢复者欺诈。如果成立则相信$(u_i,\tilde{b}_i)$是真实的。

(3)当(2)中的验证等式成立时，恢复者利用$\tilde{b}_i$计算$\tilde{b}_i'=(\tilde{b}_i^{\alpha_0} \bmod p) \bmod q$，并验证$h(u_i \| \tilde{b}_i')=H_i$是否成立。

$$\begin{aligned}\tilde{b}_i' &=(\tilde{b}_i^{\alpha_0} \bmod p) \bmod q=(\beta^{\alpha\alpha_0} \bmod p) \bmod q= \\ &(g^{\alpha\alpha_i\alpha_0} \bmod p) \bmod q=(\beta_0^{\alpha_i\alpha} \bmod p) \bmod q= \\ &(b_i^{\alpha} \bmod p) \bmod q=b_i'\end{aligned}$$

如果验证等式不成立，则说明U_i传给 D 的(u_i,b_i)不真实，U_i对 D 欺诈。如果等式成立则相信$\tilde{b}_i'=b_i'$。

(4)当参与恢复秘密的共享者传给恢复者的数据$(u_1,\tilde{b}_1),(u_2,\tilde{b}_2),\cdots,(u_t,\tilde{b}_t)$在(2)的验证中都通过，且由这些数据产生的$(u_1,\tilde{b}_1'),(u_2,\tilde{b}_2'),\cdots,(u_t,\tilde{b}_t')$在(3)的验证中也都通过时，恢复者就得到了 t 个点：$(u_1,b_1'),(u_2,b_2'),\cdots,(u_t,b_t')$，加上公告牌上的点$(r_i,f(r_i))$ $(i=1,2,\cdots,n+l-t)$，把这 $n+l$ 个点重新记为(x_i,y_i) $i=1,2,\cdots,n+l$，那么：

$$f(x)=\sum_{i=1}^{n+l} y_i \prod_{\substack{j=1 \\ j\neq i}}^{n+l} \frac{x-x_j}{x_i-x_j}$$

恢复者接着计算$h(f(1) \| f(2) \| \cdots \| f(l))$，如果$h(f(1) \| f(2) \| \cdots \| f(l))$为公告牌上的$k_0$，那么可以相信恢复出的$f(x)$是正确的，且$f(1),f(2),\cdots,f(l)$是需要恢复的秘密$k_1,k_2,\cdots,k_l$。如果$h(f(1) \| f(2) \| \cdots \| f(l)) \neq k_0$，则应怀疑在公告牌上公布的点$(r_i,f(r_i))$ $(i=1,2,\cdots,n+l-t)$中有些是不真实的。此时可以向分发者提出质疑。

3 方案分析

本方案可将 l 个秘密一次性恢复。它们的安全性依赖于离散对数的难解性、Hash 函数的抗碰撞性与单向性。

本方案以共享者的私钥为子秘密，该子秘密由共享者自己产生，有别于子秘密通常由 D 分发的模式，增加了子秘密的安全性。

在构造$f(x)$的过程中，插值点$(u_i,b_i')=(u_i,\tilde{b}_i')$的第二分量即$(g^{\alpha_0\alpha_i\alpha} \bmod p) \bmod q=(g^{\alpha\alpha_i\alpha_0} \bmod p) \bmod q$，其中，$g^{\alpha_0\alpha_i\alpha} \bmod p$ $(=g^{\alpha\alpha_i\alpha_0} \bmod p)$是在分发者、恢复者、共享者三方之间通过一种特殊形式的密钥协商产生。其过程可视为在Z_p中，恢复者发送g^{α_0}给U_i，U_i发送$g^{\alpha_0\alpha_i}$给分发者，分发者利用自己的私钥产生$g^{\alpha_0\alpha_i\alpha}$。分发者发送$g^{\alpha}$给$U_i$，$U_i$发送$g^{\alpha\alpha_i}$给恢复者，恢复者利用自己的私钥产生$g^{\alpha\alpha_i\alpha_0}$。这样，$U_i$参与其中，帮助分发者与恢复者完成了密钥协商获得共享密钥$g^{\alpha_0\alpha_i\alpha} \bmod p$ $(=g^{\alpha\alpha_i\alpha_0} \bmod p)$，而$U_i$自己没有得到这一共享密钥。在这密钥协商过程中，如果共享者发送给分发者的数据$b_i(=g^{\alpha_0\alpha_i} \bmod p)$与发送给恢复者的数据$\tilde{b}_i(=g^{\alpha\alpha_i} \bmod p)$都在公共信道上传输，攻击者就容易得到这些数据。但是对于b_i需要由它产生出b_i'。同样，对于$\tilde{b}_i$需要由它产生出$\tilde{b}_i'$才能成为插值点的第二分量用于构造$f(x)$。攻击者要达到这一目的就须在$\beta=g^{\alpha} \bmod p$中解α，或者在$\beta_0=g^{\alpha_0} \bmod p$中解$\alpha_0$，即要解离散对数问题。这是攻击者难以完成的任务。此外，假设攻击者还能得到 t 个甚至全体共享者的子秘密，即U_i的私钥$\alpha_i(i=1,2,\cdots,n)$，但这些私钥不能起到α或α_0的作用，也就是说无法利用它们构成插值点也就不能恢复$f(x)$，最终不能恢复秘密数据。这与传统的(t,n)门限方案获得了 t 个或 t 个以上的子秘密就能恢复秘密数据的情况不同。另一方面，攻击者如果想从公告牌上的H_i去解$u_i \| b_i'$，由 Hash 函数的单向性可知这一设想也是不可行的。总之，攻击难以成功。

共享者发送给分发者的数据(u_i,b_i)与发送给恢复者的数据$(u_i,\tilde{b}_i)$是否真实可靠都可以通过公告牌上h_i与H_i的值加以验证。这在秘密恢复过程中的(2)和(3)中已经讨论。在密钥协商过程中，攻击者的中间入侵与共享者的欺诈行为一样可以用h_i、H_i的数值在验证时发现。要补充说明的是，当分发者公布h_i时，U_i先计算$\tilde{b}_i=\beta^{\alpha_i} \bmod p$，然后计算$h(u_i \| \tilde{b}_i)$，因为$h(u_i \| \tilde{b}_i)=h(u_i \| (\beta_i^{\alpha} \bmod p))$，所以用$h(u_i \| \tilde{b}_i)$的值可以验证$h_i$的正确性。恢复者在收到$U_i$传来的$(u_i,\tilde{b}_i)$后，同样能验证公告牌上$h_i$的正确性。验证正确时，恢复者计算$\tilde{b}_i'=(\tilde{b}_i^{\alpha_0} \bmod p) \bmod q$和$h(u_i \| \tilde{b}_i')$，因为$h(u_i \| \tilde{b}_i')=h(u_i \| b_i')$，所以用$h(u_i \| \tilde{b}_i')$的值可以验证$H_i$的正确性。当以上验证都通过时，利用恢复出的多项式$f(x)$计算$h(f(1) \| f(2) \| \cdots \| f(l))$，看它是否为$k_0$，可以防止分发者在公告牌上对$(r_i,f(r_i))$ $(i=1,2,\cdots,n+l-t)$进行欺诈的行为。综上所述，本方案可对内部成员的双方数据相互验证，对欺诈形成相互制约的结构。

本方案中的恢复者也可以不止一人，假设共享者中的U_1、U_2、U_3均有权恢复秘密$k_1,k_2,\cdots,k_l$，此时，U_1、U_2、U_3可以通过密钥协商获得共享密钥$g^{\alpha_1\alpha_2\alpha_3} \bmod p$，令$\alpha_0=g^{\alpha_1\alpha_2\alpha_3} \bmod p \bmod q, \beta_0=g^{\alpha_0} \bmod p$，这样，$\alpha_0$和$\beta_0$可作为恢复者$U_1$、$U_2$、$U_3$为恢复秘密$k_1,k_2,\cdots,k_l$设置的私钥和公钥。

本方案能在不影响其他共享者的情况下实现成员的删除与新成员的加入。假设有某成员(如U_j)要退出系统，此时只须 D 在公告牌上删除与U_j有关的验证值h_j和H_j，保留其他成员的验证值h_i、H_i $(i=1,2,\cdots n,i\neq j)$便可。相应地，门限方案变为$(t,n-1)$门限方案。假如 D 认为可以批准接纳某个新成员U_{n+1}加入秘密共享集的请求，则U_{n+1}取随机数$\alpha_{n+1} \in Z_q^*$计算$\beta_{n+1}=g^{\alpha_{n+1}} \bmod p$。$U_{n+1}$以$\alpha_{n+1}$为私钥、以$\beta_{n+1}$为公钥，选$u_{n+1} \in Z_q^*$为身份信息，并计算$b_{n+1}=\beta_0^{\alpha_{n+1}} \bmod p$，将$\beta_{n+1}$与$(u_{n+1},b_{n+1})$传给 D。如果 D 检查认为U_{n+1}的公钥β_{n+1}不同于其他公钥且u_{n+1}也不同于其他成员的身份信息，D 就计算$\beta_{n+1}^{\alpha} \bmod p$与$b_{n+1}'=(b_{n+1}^{\alpha} \bmod p) \bmod q$，并将$h_{n+1}=h(u_{n+1} \| \beta_{n+1}^{\alpha} \bmod p)$与$H_{n+1}=h(u_{n+1} \| b_{n+1}')$公布于公告牌，而保留其他成员的验证值$h_i$与$H_i$。相应地，方案变为$(t,n+1)$门限方案。

4 结束语

本文方案与传统门限方案相比，在防止外来攻击上更有效。方案规定公告牌上的内容系统成员不仅能阅读下载并可提出质疑，使得在防止内部欺诈上，系统内的成员可以相互制约。子秘密由共享者自己产生，且可反复使用。数据传输可在公共信道上进行。共享者加入或退出系统时只须更新公告牌的内容。系统中无论是分发者、共享者还是恢复者的计算都不复杂。因此，本方案具有较好的应用前景。

(下转第 163 页)