基于多项式秘密共享的前向安全门限签名方案

芦殿军，张秉儒，赵海兴
(青海师范大学 数学与信息科学系，青海 西宁 810008)

摘 要：采用多项式秘密共享的方法，提出了一种新的前向安全的门限数字签名方案。该方案有如下特点：即使有多于门限数目的成员被收买，也不能伪造有关过去的签名；保持了公钥的固定性；在规则的时间间隔内更新密钥；可抵御动态中断敌手。假设因式分解是困难的，证明了该方案在随机预言模型中是前向安全的。

关键词：通信技术；门限签名方案；前向安全性；多项式秘密共享
中图分类号：TN918 **文献标识码**：A **文章编号**：1000-436X(2009)01-0045-05

Forward-secure threshold signature scheme based on polynomial secret sharing

LU Dian-jun, ZHANG Bing-ru, ZHAO Hai-xing
(Department of Mathematics and Information Science, Qinghai Normal University, Xi'ning 810008,China)

Abstract: Based on polynomial secret sharing, a new forward-secure threshold digital signature scheme was proposed. The scheme included a lot of property as follows: it was impossible to forge signature relating players to the past even if more than the threshold numbers of players are compromised, keep the public key fixed and update the secret keys at regular intervals, the scheme can tolerate mobile halting adversaries. Moreover, for assuming factoring is hard, it is forward-secure in random oracle model.

Key words: communications technology; threshold digital signature scheme; forward security; polynomial secret sharing

1 引言

一个密钥因“非密码学”的原因而泄露，如一个潜在的机器或者系统的泄密，是对很多密码学进程的最大威胁。通常，大部分补救措施是利用秘密共享[1,2]的方法将密钥分布式地存放在多台服务器上，这种思想的具体体现就是门限签名方案[3~6]。由于签名是在一种分布式的基于秘密共享的环境中进行的，敌手为了获得密钥和产生签名必须收买足够多的成员。

尽管密钥的分布式存放使得它更难被敌手获得，但是风险依然存在，例如方案在执行过程中可能存在某种缺陷或操作系统运行在所有的服务器上，这意味着攻破多个服务器不比攻破一个服务器困难太多，所以甚至一个分布式存放的密钥也可能被泄露。作为安全性的第二道防线，提供前向安全性到门限方案中[7]，可以减轻由密钥完全暴露所引起的危险。

数字签名的前向安全性由 Anderson[8]首次提出，解决问题的方法由 Bellare 和 Miner[9]等人设计，它的思想是：当前签名密钥的泄露不能使一个敌手伪造适合过去的签名。通过密钥演化运算，Bellare 和 Miner[9]针对单签名者设计达到了目的：使用者在不同的时间段中使用不同的密钥来产生签名，从一个初始的密钥开始，使用者利用“演化”程序在每一个时间段的最后更新当前的密钥以阻止后来

收稿日期：2008-04-24；**修回日期**：2008-11-05
基金项目：国家自然科学基金资助项目(60863006)
Foundation Item: The National Natural Science Foundation of China (60863006)

成功进入系统的敌手获得它，所以敌手最多只能够伪造密钥暴露以后时间段的签名。继 Bellare 和 Miner[9]之后，Abdalla M[10]、Itkis G[11]和 Kozlov A[12]等人又提出了一些其他改进方案，这些改进旨在使方案更为简单、实用。

将数字签名的前向安全性结合到门限签名中的思想由 Abdalla[7]等人最先提出。他们认为即使一个敌手控制了所有的服务器并且完全掌握了密钥，结合前向安全性和门限密码学，也能够得到一种提高安全性保证的方案。实际上，在他们所构造的方案中，敌手不能用攻破后时间段的密钥来伪造攻破前时间段的签名，所以，有关“过去”的签名及密钥的所有知识对于敌手来说是无用的。Abdalla 的方案成功地解决了门限签名的前向安全性，但它只能抵御窃听敌手，同时要求在签名和密钥更新阶段均有 n（成员总数为 n）个诚实的成员参加。

本文在 Abdalla[7]等人的基础上，采用多项式秘密共享的方法，提出了一种新的前向安全的门限数字签名方案，该方案可抵御动态中断敌手，它的主要优点是在签名或密钥更新阶段不需要所有的成员都参加，而仅仅需要 2/3 的成员在任意的情况下参加。

2 定义和模型

2.1 通信模式

方案的参与者有 n 个成员，他们被一个广播信道连接，并且能够通过秘密信道进行私人的点到点之间的通信（这样的信道可以由广播信道使用密码学技术得到），假设在设置阶段存在一个可信中心，每个成员均可以进行广播通信和点到点通信。

2.2 敌手的类型

按照敌手的攻击能力，敌手可分为：①窃听敌手。可以获得一个成员的秘密信息，但不能影响其行为的敌手。②可中断敌手。不仅能够窃听，而且能够影响该成员对进程参与度的敌手。③恶意敌手。可以引起成员在进程中任意违反常规的敌手。

按照敌手的行为模式，敌手可分为：①静态敌手。在进程开始之前就已经决定了被攻击成员的敌手。②可适应敌手。可以在进程中随着信息的获得即时地决定被攻击成员的敌手。③动态敌手。不仅是可适应的，而且能够决定在不同的时间段控制不同成员的敌手。

2.3 前向安全的门限签名方案

本文所讨论的基于多形式秘密共享的前向安全的门限签名方案 $(t,2t+1,2t+1,3t+1)$（简称为 PFST）是这样的：密钥被分配在 $3t+1$ 个成员中，由 $2t+1$ 个成员执行密钥更新算法，$2t+1$ 个成员执行签名运算，任意掌握了少于 t 个密钥份额的敌手不能伪造签名；使用密钥演化算法，整个生存期被分成若干个时间段，密钥在不同的时间段内不同，而公钥则是固定的。方案由密钥生成阶段、密钥更新阶段、签名生成阶段、签名验证阶段等 4 个阶段组成。

2.4 注释

方案中有 $3t+1$ 个成员，时间段的总数被标记为 T，全部的公钥记为 PK，且由 l 个值构成，记为 $U_1,U_2,\cdots,U_l$。在每一个时间段 j，相应密钥的 l 个组成部分记为 $S_{1,j},S_{2,j},\cdots,S_{l,j}$，它们被所有的成员共享，成员 ρ 掌握的第 j 个时间段的第 i 个密钥值的份额 $S_{i,j}$ 记为 $S_{i,j}^{(\rho)}$，全部（l 个值）秘密信息记为 $SK_j^{(\rho)}$。一般地，记号 $X^{(\rho)}$ 指的是 ρ 掌握的 X 的份额。

3 方案描述

3.1 主模块构造

可信中心执行钥匙生成算法，它在所有的 $3t+1$ 个参与者中使用 Shamir-SS 算法来分享密钥。成员 ρ 的基本密钥份额 $SK_0^{(\rho)}$ 包括每一个 $S_{i,0}$ 值的份额（共有 l 个份额），他的密钥是 $\left(N,T,0,S_{1,0}^{(\rho)},S_{2,0}^{(\rho)},\cdots,S_{l,0}^{(\rho)}\right)$，在每一个时间段的开始，恰好有 $2t+1$ 个成员参与，利用密钥更新算法完成密钥的演化。在时间段 j 的开始，参与了先前更新进程的每一个成员 ρ 均有 $SK_{j-1}^{(\rho)}$ 等先前时间段的密钥份额，使用 Modified-Mult-SS 算法将其平方 l 次后可计算出新的密钥 $SK_j^{(\rho)}$。并且每一个成员立即删除先前时间段的秘密份额，于是所有未被中断的成员，包括那些在先前更新进程中已经被中断的成员将获得一个新的秘密份额。

3.1.1 PFST.keygen(k,T) 初始化算法

1) 可信中心随机地选取不同的 $\frac{k}{2}$ bit 的素数 p,q，均满足 $p\equiv q\equiv 3(\bmod 4)$，设置 $N=pq$；

2) 对于 $i=1,2,\cdots,l$ 循环

① 可信中心随机地选取：$S_{i,0}\in_R Z_N^*$，计算 $U_i\equiv\left(S_{i,0}^{2^{(T+1)}}\right)^{-1}(\bmod N)$；

② 可信中心计算：在 Z_N 中使用 Shamir-SS 创建 $S_{i,0}$ 的份额 $S_{i,0}^{(1)},S_{i,0}^{(2)},\cdots,S_{i,0}^{(3t+1)}$；

3) 对于 $\rho=1,2,\cdots,(3t+1)$ 循环，可信中心计算并对成员 ρ 发送：$SK_0^{(\rho)}=(N,T,0,S_{1,0}^{(\rho)},S_{2,0}^{(\rho)},\cdots,S_{l,0}^{(\rho)})$；

4) 可信中心设置：$PK=(N,T,U_1,U_2,\cdots,U_l)$，且公布 PK。

3.1.2 PFST.sing(m,j)签名算法

1) 利用 Joint-Shamir-RSS 算法生成随机值 $R\in_R Z_N$，其中成员 ρ 得到 R 的份额为 $R^{(\rho)}$；

2) 所有成员利用 Modified-Mult-SS 和 R 的值计算 $Y\equiv(R)^{2^{(T+1-j)}}(\bmod N)$；

3) 每一个成员 ρ 计算 $\delta=H(j,Y,m)$；

4) 每一个成员 ρ 执行 $Z^{(\rho)}\equiv R^{(\rho)}(\bmod N)$ 使得 Z 被 R 初始化；

5) 对于 $i=1,2,\cdots,l$ 循环：$Z\equiv R\prod_{i=1}^{l}S_{i,j}^{\delta}(\bmod N)$，在此使用 Modified-Mult-SS；

6) 公布对消息 m 的签名集合 $\langle j,\langle Y,Z\rangle\rangle$。

3.1.3 PFST.verify$_{PK}(m,\gamma)$验证算法

1) 将 γ 当作 $\langle j,\langle Y,Z\rangle\rangle$。

2) 如果 $Y\equiv 0(\bmod N)$则返回 0。

3) $\delta=H(j,Y,m)$。

4) 如果 $Y\equiv Z^{2^{(T+1-j)}}\prod_{i=1}^{l}U_i^{\delta}(\bmod N)$，则返回 1；否则返回 0。

3.1.4 PFST.update(j) 密钥更新算法

1) 如果 $j=T$，则返回一个空串，否则继续；

2) 成员使用 Modified-Mult-SS 对先前的值 $S_{1,j-1},S_{2,j-1},\cdots,S_{l,j-1}$ 模 N 平方后计算出更新后的密钥份额 $S_{1,j},S_{2,j},\cdots,S_{l,j}$；

3) 每一个成员 ρ 从他们的计算机中删除 $SK_{j-1}^{(\rho)}$。

3.2 子模块构造

3.2.1 Shamir-SS 子模块

Shamir 标准秘密分享方案[13]在有限域 Z_N^* 上实现。可信中心选择一个秘密值 $S_{i,0}$ 和次数为 t 的随机多项式 $p(x)$，它的系数被标记为 $a_0,a_1,\cdots,a_t$，然后将 $S_{i,0}$ 固定在系数 a_0 上，并对第 $\rho(\rho=1,2,\cdots,(3t+1))$ 个成员分配份额 $S_{i,0}^{(\rho)}$。秘密值 $S_{i,0}$ 可由 $3t+1$ 个成员中的任意 $t+1$ 个成员的秘密份额恢复。注意到本文的方案在 Z_N 中运算，而它不是一个域，为了确保其在 Z_N 中有惟一的结果，首先要求成员的数目必须小于 p 和 q，这样任何份额都不能重构包含 p 和 q 的因子，不妨设恢复秘密的 $t+1$ 个成员的私钥为 $x_{i_j}(j=1,2,\cdots,t+1)$，则由他们所构成的方程组的系数矩阵为 $t+1$ 阶范德蒙矩阵，其中所有的元素均与 N 互素，而范德蒙行列式的值由 $\prod_{1\le j<k\le t+1}(x_{i_k}-x_{i_j})(\bmod N)$ 给出，该值也与 N 互素，所以矩阵模 N 可逆，于是我们保证了该系统在 Z_N 中有惟一的结果。

3.2.2 Modified-Mult-SS 子模块

在签名的生成阶段和密钥的更新阶段，都需要在成员中分享 2 个秘密的乘积，即若 2 个秘密 α 和 β 通过 t 次多项式 $f_\alpha(x)$ 和 $f_\beta(x)$ 在 $2t+1$ 个成员中分享，且相应地有 $f_\alpha(0)=\alpha$ 和 $f_\beta(0)=\beta$，通过次数的约化和随机化，成员共同计算出一个新的多项式 G，使得 $G(0)=\alpha\beta$。

文献[14]描述了在仅有窃听敌手的模式下一步完成次数约化和随机化的方法，而与此对比，我们的方案对其进行了修改，允许可中断敌手，也就是说敌手可能随意地选择任意成员来中断，不能假设参与者是一个特定成员的子集，在方案的运行期，为了决定哪一个成员可以参加，要求任意的正在起作用的和在最近的更新阶段中没有被中断的成员 P_i 广播一个“我还存在”的信息。从这些有回应的集合中，选择 $2t+1$ 个参与成员来实际执行这个计算。然后，在时间 $O(2t+1)$ 中，那些成员的子集所对应的常数将被有效地计算出来。

应该指出，在执行 Modified-Mult-SS 算法的任意时间，如果一个参与的成员被敌手所中断，至少有一个另外的参与者的子集注意到它，进程被中断；并且有一个不同的当前参与者的子集重新开始，而且乘法运算重新开始的次数决不会超过 t 次，这是由于敌手在一个时间段中能够中断的成员数目的界限是 t。

3.2.3 Joint-Shamir-RSS 子模块

文献[15]和文献[16]提出了一个在没有可信中心的情况下允许一个成员群共同生成一个秘密随机数的方法。在本文中，每一个参与者选择一个随机数和一个多项式，并且扮演可信中心的角色以使用 Shamir 秘密分享方案分配他的秘密，最终确定的秘密值是所有参与者的秘密的总和。同时，要求成员 P_i 的秘密份额在一个广播信道中被分发出去，并且每一个成员 P_j 的份额均隐藏在 P_j 的公钥之下。如果这样的信息没有从一个特定的成员 P_j 处广播，可以假设他被中断了，并且对任意单独成员所分享的

份额的总和中显然不包括 P_j 的份额。

4 安全性

定理 1 在所提出的门限签名方案 PFST $(t,2t+1,2t+1,3t+1)$ 中，若 PFST.keygen(k,T) 初始化算法产生的公钥为 $PK=(N,T,U_1,U_2,\cdots,U_l)$，为每个成员 ρ 产生的份额为 $SK_0^{(\rho)}=(N,T,0,S_{1,0}^{(\rho)},S_{2,0}^{(\rho)},\cdots,S_{l,0}^{(\rho)})$，$\rho=1,2,\cdots,(3t+1)$，PFST.update$(j)$ 算法为密钥份额进行更新，PFST.sign(m,j) 为消息 m 产生的签名是 $\langle j,\langle Y,Z\rangle\rangle$，则 $\langle j,\langle Y,Z\rangle\rangle$ 是对消息 m 的有效签名。

证明 为了验证 $\langle j,\langle Y,Z\rangle\rangle$ 是对消息 m 的有效签名，需要验证 $Y\equiv Z^{2^{(T+1-j)}}\prod_{i=1}^{l}U_i^{\delta}\pmod N$ 成立，若所得信息均由方案产生，则 $\delta=H(j,Y,m)$，$Y\equiv R^{2^{(T+1-j)}}\pmod N$，$Z\equiv R\prod_{i=1}^{l}S_{i,j}^{\delta}\pmod N$，$U_i\equiv\left(S_{i,0}^{2^{(T+1)}}\right)^{-1}\pmod N$ 因为：$S_{i,j}^{(\rho)}\equiv\left(S_{i,j-1}^{(\rho)}\right)^{2^1}\equiv\left(S_{i,j-2}^{(\rho)}\right)^{2^2}\equiv\cdots\equiv\left(S_{i,0}^{(\rho)}\right)^{2^j}\pmod N$

所以：$\left(S_{i,0}^{(\rho)}\right)\equiv\left(S_{i,j}^{(\rho)}\right)^{2^{-j}}\pmod N$，进而

$$S_{i,0}\equiv\prod_{\rho=1}^{n}S_{i,0}^{(\rho)}\equiv\left(\prod_{\rho=1}^{n}S_{i,j}^{(\rho)}\right)^{2^{-j}}\pmod N$$

因此：

$$\begin{aligned}Z^{2^{(T+1-j)}}\prod_{i=1}^{l}U_i^{\delta}&\equiv\left(R\prod_{i=1}^{l}S_{i,j}^{\delta}\right)^{2^{(T+1-j)}}\left(\prod_{i=1}^{l}\left(S_{i,0}^{2^{(T+1)}}\right)^{-1}\right)^{\delta}\\&\equiv R^{2^{(T+1-j)}}\prod_{i=1}^{l}S_{i,j}^{-\delta 2^{(T+1-j)}}\prod_{i=1}^{l}\left(S_{i,0}\right)^{-\delta 2^{(T+1)}}\\&\equiv R^{2^{(T+1-j)}}\prod_{i=1}^{l}S_{i,j}^{-\delta 2^{(T+1-j)}}\prod_{i=1}^{l}\left(S_{i,j}\right)^{-\delta 2^{(T+1-j)}}\\&\equiv R^{2^{(T+1-j)}}\equiv Y\pmod N\end{aligned}$$

我们证明了 $Y\equiv Z^{2^{(T+1-j)}}\prod_{i=1}^{l}U_i^{\delta}\pmod N$ 成立，所以 $\langle j,\langle Y,Z\rangle\rangle$ 是对消息 m 的有效签名。

定理 2 令 PFST $(t,2t+1,2t+1,3t+1)$ 表示本文所提出的方案，也就是说，当成员总数为 3t+1 时，它可以容忍 t 个中断敌手的攻击，有 $2t+1$ 个成员进行密钥演化，并且有 $2t+1$ 个成员生成一个有效签名。令 FS 表示由 Brllare 和 Miner[9]给出的单用户数字签名方案，那么只要 FS 在标准的单用户方面是一个前向安全的签名方案，PFST 在有动态中断敌手存在的情况下，是一个前向安全的门限数字签名方案。

证明 本定理的证明建立在单成员 FS 签名方案基础之上，采用文献[9]中的证明思想,并使用模拟敌手在协议中的观察方法。令 F 表示对方案攻击的敌手，希望构造出一种针对前向安全性攻击的算法。F 的攻击分 3 个阶段进行：选择明文攻击阶段 cma；超门限阶段 overthreshold；伪造阶段 forge。假设 F 可以访问随机签名预言 S 和随机散列预言 H，令公钥 $PK=(N,T,U_1,U_2,\cdots,U_l)$为在 cma 阶段算法的输入。构造过程概要描述如下：

1) cma 阶段

该阶段允许 F 查询签名预言 PFST.sign(m, j)和散列预言 H，所以需要以 F 的视角来模拟这 2 个预言。在这个阶段允许 F 每次收买的成员为 1，2，…，t（要么单纯窃听，要么实际地中断他们的通信），进而，我们需要能够模拟这些被收买的成员的行为。

在 cma 阶段的开始,F 可以选择是留在该阶段还是进入 overthreshold 阶段。如果它选择前者，算法不变。这里仅考虑当 F 选择进入 overthreshold 阶段的行为。

2) overthreshold 阶段

密钥份额的分配：令 B_j 表示时间段 j 中被收买的成员的集合。按照方案设计，有$|B_j|\leqslant t$。可以简单地为每个成员 $b\in B_j$ 随机地选取 $S_{i,j}$ 的份额 $S_{i,j}^{(b)}$ 作为当前密钥的一个份额 $S_{i,j}(i=1,\cdots,l)$。

签名预言的模拟：令 m 表示被查询的签名预言中的消息明文，通过查询随机签名预言 S，可以容易地以 F 的视角来模拟他的签名预言 PFST.SIG，得到<j,(Y,Z)> 作为签名预言的输出结果。

PFST.sign(m,j)的模拟：首先需要模拟 R 的生成，然后模拟 Modified-Mult-SS 协议的连续运行过程。在 R 的生成中，每个成员取一个随机值且与其他成员分享它。因为在 B_j 中至多有 t 个成员，所以可以随机地取他们在 R 中的份额。

运行 Modified-Mult-SS 算法模拟从 R 的份额中得到 R^2 的份额的过程，也可以用与更新协议相似的方式完成。令 $R^{(b)}$表示成员 $b\in B_j$ 得到的 R 的份额，可以如下计算 R^2 份额：对于每个成员 $b\in B_j$，使用 Shamir-SS 协议创建份额$(R^2)^{(b)}=(R^{(b)})^2$ 并且将它们发送给 B_j 中的其他成员，为了计算所有 $R^4,\cdots,R^{2^{T+1}}\equiv Y\pmod N$的份额，以 B_j 中被收买成员的视角重复同样的处理过程。令 $R^{2^{T+1}(b)}$ 表示被成员 $b\in B_j$

执行的 $R^{2^{T+1}}$ 的份额。可以通过使用上述得到的 Y 值和每个成员 $b\in B_j$ 的 $R^{2^{T+1}(b)}$ 份额的值求出其他 $R^{2^{T+1}}$ 的份额。如果 $|B_j|<t$，那么对 $R^{2^{T+1}}$ 随机地选取 $t-|B_j|$ 个份额，从已有的那些份额中计算其余的份额后，将 Y 和 $R^{2^{T+1}}$ 发送给 F。Z 的份额从 $c_1\cdots c_l$ 及 R 和 $S_{i,j}(i=1,\cdots,l)$ 的份额中计算。

散列预言的模拟：对于每次被 F 查询的 (j,Y,m)，可以使用自己的随机预言 H 查询得到结果 H，将其提供给 F。

3) forge 阶段：令 a 表示 F 决定转到 overthreshold 阶段的时间段，将该阶段的密钥 $(S_{1,a},\cdots,S_{l,a})$ 提供给 F，令 $(m,<a,(Y,Z)>)$ 为被 F 伪造的输出，我们简单地返回 $(m,<a,(Y,Z)>)$ 作为伪造签名的输出，伪造结束。

以上的攻击算法的构造与文献[9]相同，这就是说，对于动态中断敌手来说，如果本文提出的方案不满足前向安全性，则 FS 方案也不满足前向安全性。换言之，若 FS 方案是前向安全的门限签名方案，则本文提出的 PFST 门限数字签名方案也必是前向安全的。而由文献[9]可知，假定因式分解难解，在随机预言模型中，FS 为前向安全的签名方案，所以 PFST 方案是抵御动态中断敌手的前向安全的门限数字签名方案。

5　结束语

本文在 Abdalla[7]等人基础上，采用多项式秘密共享的方法，提出了一种新的前向安全的门限数字签名方案。该方案具备如下优势：收买任意少于门限数目个成员的敌手不能伪造签名；收买大于或等于门限数目个成员的敌手虽然可以知道当前密钥，却不能伪造当前时间周期之前的签名；在签名或密钥更新阶段只需要 2/3 而不必所有的成员都参加；可抵御动态中断敌手。

参考文献：

[1] ITO M, SAITO A, MATSUMOTO T. Secret sharing scheme realizing general access structure[A]. Proceedings IEEE Globecom'87[C]. 1987. 99-102.

[2] SIMMONS G. An introduction to shared secret and/or shared control schemes and their application in contemporary cryptology[J]. The Science of Information Integrity, 1992, 441-497.

[3] DESMEDT Y, FRANKEL Y. Shared generation of authenticators and signatures[A]. Advances in Cryptology-CRYPTO'91[C]. Santa Barbara, 1991. 457-469.

[4] DESMEDT Y. Threshold cryptosystems[A]. Advances in Cryptology-AUSCRYPT'92[C]. Berlin Germany, 1993.718.

[5] PEDERSEN T P. A threshold cryptosystem without a trusted party[A]. Advances in Cryptology-EUROCRYPT'91[C]. Brighton UK, 1991. 522-526.

[6] SHOUP V. Practical threshold signatures[A]. Advances in Cryptology-EUROCRYPT'2000[C]. Bruges Belgium, 2000. 207-220.

[7] ABDALLA M, MINER S, NAMPREMPRE C. Forward-secure threshold signature schemes[A]. Cryptology-CT-RSA'2001[C]. Berlin Germany, 2001.

[8] ANDERSON R. Two remarks on public-key cryptology[A]. Relevant material presented by the author in an invited lecture at the ACM CCS'97: 4th Conference on Computer and Communications Security[C]. Zurich, Switzerland, 1997.1-4.

[9] BELLARE M, MINER S. A forward-secure digital signature scheme[A]. Advances in Cryptology-CRYPTO'99[C]. Santa Barbara, 1999. 431-448.

[10] ABDALLA M, REYZIN L. A new forward-secure digital signature scheme[A]. Asiacrypt 2000[C]. Berlin Germany, 2000, 1976. 116-129.

[11] ITKIS G, REYZIN L. Forward-secure signatures with optimal signing and verifying[A]. CRYPTO 2001[C]. Berlin Germany, 2001. 499-514.

[12] KOZLOV A, REYZIN L. Forward-secure signatures with fast key update[A]. Security in Communication Networks[C]. Berlin Germany, 2002.247-262.

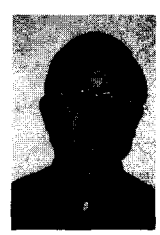
[13] SHAMIR A, How to share a secret[J]. Communication of the ACM, 1979, 22: 612-613.

[14] GENNARO R, RABIN M O, RABIN T. Simplified VSS and fast-track multiparty computations with applications to threshold cryptography[A]. 17th ACM Symposium Annual on Principles of Distributed Computing[C]. Puerto Vallarta, Mexico, 1998.101-111.

[15] INGEMARSSON I, SIMMONS G J. A protocol to set up shared secret schemes without the assistance of a mutually trusted party[A]. Advances in Cryptology-EUROCRYPT'90[C]. Aarhus, Denmark, 1990. 266-282.

[16] GENNARO R, JARECKI S, KRAWCZYK H, *et al*. Secure distributed key generation for discrete-log based cryptosystems[A]. Advances in Cryptology-EUROCRYPT'99[C]. Prague, Czech Republic, 1999. 295-310.

作者简介：



芦殿军（1970-），男，甘肃永昌人，硕士，青海师范大学副教授，主要研究方向为代数组合与密码学。

张秉儒（1949-），男，青海湟中人，青海师范大学教授、硕士生导师，主要研究方向为代数组合与密码学。

赵海兴（1967-），男，青海湟中人，博士，青海师范大学教授、硕士生导师，主要研究方向为图与计算机网络。