Journal of Computer Applications
计算机应用,2013,33(9):2536-2539,2545
ISSN 1001-9081
CODEN JYIIDU
2013-09-01
http://www.joca.cn

文章编号:1001-9081(2013)09-2536-04
doi:10.11772/j.issn.1001-9081.2013.09.2536

非线性一次一密(*t*,*n*)门限秘密共享方案

范　畅*,茹　鹏
(电子科技大学成都学院 计算机系,成都 611731)
(*通信作者电子邮箱 fan_ccn@yahoo.com.cn)

摘　要:针对本身不安全的线性算法构造的门限秘密共享方案存在安全漏洞的问题,以及可信方的参与容易导致单点故障和不可靠情形,结合非线性算法和密码学理论,提出一种无可信方的非线性门限秘密共享方案。方案基于混沌算法和有限状态自动机两种非线性结构,子密钥的产生具有随机性和动态性,参与者可控制每一轮的子密钥来实现一次一密或*N*次一密安全级别。秘密恢复由拉格朗日插值公式来实现。安全多方计算使各参与者相互牵制,不需可信方参与,满足弹性均衡,可防欺骗与合谋攻击。

关键词:门限秘密共享;非线性;一次一密;混沌算法;有限状态自动机
中图分类号:TP309　　**文献标志码**:A

(*t*, *n*) threshold secret sharing scheme for nonlinear one-time pad

FAN Chang, RU Peng
(*Department of Computer, Chengdu College of University of Electronic Science and Technology of China, Chengdu Sichuan* 611731, *China*)

Abstract: To address the problem that secret sharing scheme constructed by linear algorithm has security vulnerabilities, and to solve the problem that it easily leads to a single point of failure and unreliable situations with trusted party, this paper proposed a nonlinear threshold secret sharing scheme which combined nonlinear algorithm and cryptography. The scheme was based on two nonlinear structures of chaos algorithm and finite state automata, so it can generate random and dynamic shares. Participants can control each round shares to achieve the security level of once or *N* times a password. Secret was recovered by the Lagrange interpolation formula. Secure multiparty computation restricted every participant so that the scheme satisfied resilient equilibrium and could withstand chicanery or conspiracy attack.

Key words: threshold secret sharing; nonlinear; once time once password; chaos algorithm; Finite State Automata (FSA)

《计算机应用》Journal of Computer Applications

0　引言

门限秘密共享是指将一秘密分割为*n*个子秘密,分别由*n*个参与者秘密保管,至少*t*个参与者提供真实子密钥才能恢复秘密,称为(*t*,*n*)门限秘密共享。许多秘密共享方案[1-5]由线性算法实现,不具有一次一密特性,并且没有解决子密钥复用和秘密更新问题。线性算法的缺点是除初值a_0的选取具有随机性外,算法本身不具有随机性,a_0一旦选定,由线性关系式后续数容易确定;一次一密指每次产生的数具有良好的"随机性"和"不可预测性",且与之前的所有数都不同;秘密更新问题指如果不修改子密钥就不能更新共享秘密;子密钥复用问题指旧的子密钥仍然可以恢复新的共享秘密。为了达到一次一密安全级别,本文基于混沌算法和有限状态自动机两种非线性结构,根据安全多方计算和Lagrange插值公式原理提出了一种非线性一次一密(*t*,*n*)门限秘密共享方案,解决了上述存在的问题,安全性和灵活性都提高了。

1　相关工作

通过对这些秘密共享方案[1-15]的研究与分析,结论是它们既不是非线性结构,也达不到一次一密安全级别。文献[3]中的访问结构由于基于线性结构,所以不安全。文献[11-13]描述了非线性机制的理论,特别是文献[13]与线性算法相比,证实非线性算法的优势所在。文献[14]主要解决参与者删添管理问题,不具有系统性。文献[15]主要解决门限值更改问题,动态性不完整。

以文献[1]EMSSS(An Efficient Multistage Secret Sharing Scheme Using Linear One-way Functions and Bilinear Maps)方案为例,EMSSS建立在椭圆曲线密码体制(Elliptic Curve Cryptosystem,ECC)、线性单向Hash函数和双线性映射基础上,其修改版[1,7-8]简述如下:

子密钥分配阶段:

1)分发者计算$Q=rP$,其中$r\in\mathbf{Z}_q^*$为随机数,$P\in G$为生成元,且公开P、Q;

2)每个参与者$P_j(j=1,2,\cdots,n)$随机选择$s_j\in\mathbf{Z}_q^*$作为自己的子密钥,并计算s_jQ传给分发者;

3)分发者验证所有的$s_jQ(j=1,2,\cdots,n)$都不同后,计算$r^{-1}(s_jQ)=r^{-1}(s_jrP)=s_j(r^{-1}rP)=s_jP$;

4)分发者计算$Q(0)P$和$Q(d_k)P$如下列等式:

$$Q(0)P=\sum_{j=1}^{n}\left(\prod_{\substack{i=1\\i\neq j}}^{n}\frac{x_i}{x_i-x_j}\right)s_jP \tag{1}$$

$$Q(d_k)P=\sum_{j=1}^{n}\left(\prod_{\substack{i=1\\i\neq j}}^{n}\frac{d_k-x_i}{x_i-x_j}\right)s_jP,k\in\{1,2,\cdots,n-t\} \tag{2}$$

收稿日期:2013-03-19;**修回日期**:2013-05-02。

作者简介:范畅(1974-),男,四川广安人,讲师,硕士研究生,主要研究方向:秘密共享、门限群签名;　茹鹏(1982-),女,甘肃酒泉人,讲师,硕士研究生,主要研究方向:代理数字签名。

其中:$Q(x)$ 为不超过 $n-1$ 次的多项式,且经过点 (x_1,s_1P),$(x_2,s_2P),\cdots,(x_n,s_nP)$,$x_1,x_2,\cdots,x_n \in \mathbf{Z}_q^*$ 为各个参与者不同标识。

5)分发者计算 $\alpha_1 = h(Q(0)P,1),\alpha_2 = h(Q(0)P,2),\cdots,\alpha_m = h(Q(0)P,m)$,且公开向量 $(s_iQ,S_j-\alpha_j,Q(d_k)P)$。

子密钥验证阶段:

在恢复秘密之前,$P_j(j=1,2,\cdots,n)$ 检验以下等式来验证参与者提供的伪子密钥是否真实:

$$e(h(s_jP,i),Q) = e(h(P,i),s_jQ);\quad 1 \leqslant i \leqslant m,\ j=1,2,\cdots,t \tag{3}$$

秘密恢复阶段:

令 $P_1,P_2,\cdots,P_t$ 为秘密 $S_i(i \in \{1,2,\cdots,m\})$ 恢复的参与者,用他们的伪子密钥 $h(s_jP,i)(j \in \{1,2,\cdots,t\})$ 和公开参数 $Q(d_k)P$ 可计算 $\alpha_i = h(Q(0)P,i)$,从而可恢复秘密 $S_i = \alpha_i + (S_i-\alpha_i)$,其中 $(S_i-\alpha_i)$ 为公开向量中的数据。

EMSSS 方案性能不够强壮,这在第 3 章与本文方案的比较中体现出来。非线性理论知识少存于文献[11-12]中,而系统完整的非线性秘密共享方案笔者目前还未找到。

2　方案描述

方案由系统初始化、子密钥产生、秘密恢复、成员添加删除、一次一密实现及门限 t 更新子协议组成,并由这些子协议共同执行完成。成员 P_i 第 j 轮产生的子密钥 s_{ij} 及恢复秘密 s_j 的总体思路如图 1 所示。

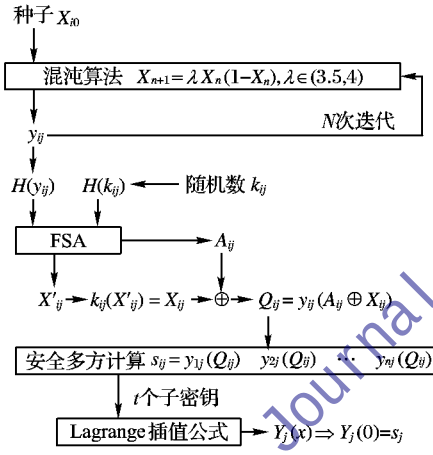

图 1　方案总思路

2.1　系统初始化

系统参数有:有限域 $GF(p)$ 上的多项式 $k():[0,p)\to[0,p)$ 和 $y():[0,p)\to[0,p)$,其中 p 为大正整数;有限域 $GF(q^t)$ 上的多项式 $Y()$,其中 q 为大素数,为 $Y()$ 系数的模,t 为门限值,$Y()$ 最高次数为 $(t-1)$,模乘运算满足 $a_ix^i \cdot b_jx^j \bmod p = (a_i \cdot b_j \bmod p)x^{(i+j) \bmod t}$;非线性迭代函数 $X_{n+1} = \lambda X_n(1-X_n),X_n \in (0,\infty),\lambda \in (3.5,4)$;迭代次数 N(大整数);单向安全杂凑函数 $H():[0,\infty)\to[0,p)$;FSA 的转移函数 $f_1():[0,p)\times[0,p)\to[0,\infty)$ 和 $f_2():[0,p)\times[0,p)\to[0,\infty)$,输入为 $H(y_{ij})$ 和 $H(k_{ij})$,输出分别为矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 中的某个元素,这些元素都是随机数,

$$\boldsymbol{A} = \begin{pmatrix} A_{11} & A_{12} & \cdots & A_{1n} \\ A_{21} & A_{22} & \cdots & A_{2n} \\ \vdots & \vdots & & \vdots \\ A_{m1} & A_{m2} & \cdots & A_{mn} \end{pmatrix}$$

$$\boldsymbol{X}' = \begin{pmatrix} X_{11} & X_{12} & \cdots & X_{1n} \\ X_{21} & X_{22} & \cdots & X_{2n} \\ \vdots & \vdots & & \vdots \\ X_{m1} & X_{m2} & \cdots & X_{mn} \end{pmatrix}$$

成员保密的参数:初态值 X_{i0}、y_{ij}、k_{ij}、$y_{ij}()$、$k_{ij}()$、$Y_j()$ 和 s_{ij};系统公告牌上公开的参数:授权集 $GS=\{P_1,P_2,\cdots,P_i,\cdots,P_n\}$、秘密恢复者 D、门限结构 (t,n)、p、q、N、$X_{n+1} = \lambda X_n(1-X_n),\lambda \in (3.5,4)$、$H()$、矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$、Q_{ij} 及每个成员(含 D)签名的公钥列表 $SK_i(PK_i)-PK_i(i=1,2,\cdots,n$ 和 $D)$。

2.2　子密钥产生

以成员 $P_i(i=1,2,\cdots,n)$ 为例,其第 j 轮的子密钥 s_{ij} 的产生步骤如下:

1)选择随机数 $N_{i0} \in (0,\infty)$,附上日期时间 DT_{i0} 和 MAC_{i0} 地址,作为种子 $X_{i0} = (N_{i0} \parallel DT_{i0} \parallel MAC_{i0}) \in (0,\infty)$,代入 $X_{n+1} = \lambda X_n(1-X_n),\lambda \in (3.5,4)$,经 N 次迭代后,输出 $y_{ij} = f(\cdots f(f(X_{i0}))\cdots)$。

2)选择随机数 $k_{ij} \in (0,\infty)$,计算 $H(k_{ij})$ 和 $H(y_{ij})$。

3)查矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 得:$A_{ij} = f_1(H(y_{ij}),H(k_{ij}))$,$X'_{ij} = f_2(H(y_{ij}),H(k_{ij}))$,若无输出,则返回 1)。

4)将 k_{ij} 和 y_{ij} 表示成 $GF(2^q)$ 上的多项式 $k_{ij}()$ 和 $y_{ij}()$,例:$k_{ij}=10=(1010)_2 \Rightarrow k_{ij}(x)=x^3+x$,计算 $k_{ij}(X'_{ij}) = X_{ij}$ 和 $y_{ij}(A_{ij} \oplus X_{ij}) = Q_{ij}$,$X_{ij}$ 保密,Q_{ij} 公布到公告牌上。

5)P_i 可通过公开信道获得 $P_k(k \neq i,k=1,2,\cdots,n)$ 签密的 $de_{ij} = PK_i(SK_k(y_{kj}(Q_{ij})))$,$P_i$ 验证出 $y_{kj}(Q_{ij}) = PK_k(SK_i(de_{ij}))$,若验证失败,停止协议;否则,计算并保密其子密钥 s_{ij}:

$$s_{ij} = \prod_{k=1}^{n} y_{kj}(Q_{ij}) \bmod q \tag{4}$$

GS 中的其他成员 $P_k(k \neq i,k=1,2,\cdots,n)$ 也以 1)至 5)产生各自保密的子密钥 s_{kj}。

2.3　秘密恢复

1) GS 中至少有 t 个成员用各自私钥和 D 的公钥签密提供子密钥 $PK_D(SK_i(s_{ij}))(i=1,2,\cdots,t)$,放于公告牌上;

2) GS 中任何成员在 D 允许的情形下,即 D 解密出 $SK_i(s_{ij}) = SK_D(PK_D(SK_i(s_{ij})))$,可用这 t 个成员的公钥验证其提供的子密钥 $s_{ij} = PK_i(SK_i(s_{ij}))(i=1,2,\cdots,t)$,若验证失败,停止协议;

3) D 找到 $(Q_{ij},s_{ij})(i=1,2,\cdots,t)$ t 个点,若有 $r(2 \leqslant r \leqslant t)$ 个相同的点,则让 t 个成员中的任意 $(r-1)$ 个成员返回 2.2 节的 2)以重新产生 t 个不同的点;

4) D 基于 Lagrange 插值公式,由 $(Q_{ij},s_{ij})(i=1,2,\cdots,t)$ t 个点构造出 $(t-1)$ 次多项式:

$$Y_j(x) = \sum_{l=0}^{t-1} a_lx^l \bmod p \tag{5}$$

计算秘密 $s_j = Y_j(0)$。

2.4　成员的加入和退出

1)设 P_{n+1} 要加入授权集 $GS=\{P_1,P_2,\cdots,P_i,\cdots,P_n\}$,$D$ 操作如下:

①在 GS 中添加该成员,添加后的授权集为 $GS'=\{P_1,P_2,\cdots,P_i,\cdots,P_n,P_{n+1}\}$,门限结构改为 $(t,n+1)$;

②将 P_{n+1} 签名的公钥 PK_{n+1} 放于公钥列表中,即公钥列表中有 $SK_{n+1}(PK_{n+1})-PK_{n+1}$;

③系统其他参数不变。

《计算机应用》Journal of Computer Applications

2)设 P_i 要退出授权集 $GS=\{P_1,P_2,\cdots,P_i,\cdots,P_n\}$,$D$ 操作如下:

①在 GS 中删除该成员,删除后的授权集为 $GS'=\{P_1,P_2,\cdots,P_j,\cdots,P_{n-1}\}(j\neq i)$,门限结构改为 $(t,n-1)$;

②将公钥列表中有关 P_i 的项删掉,即删掉之后的公钥列表中没有 $SK_i(PK_i)-PK_i$;

③系统其他参数不变。

2.5 一次一密的实现

$P_i(i=1,2,\cdots,n)$ 第 $j(j=1,2,3,\cdots)$ 次产生的子密钥 s_{ij} 和恢复的秘密 s 变化关系分为三种情形:1)s_{ij} 和 s 都变化;2)s_{ij} 变化而 s 不变且没泄漏;3)s 变化而 s_{ij} 不变且没泄漏。情形 1)实现了一次一密,同时解决了子密钥复用问题;情形 2)解决了秘密更新问题;情形 3)增加了方案其他灵活特性。

2.5.1 s_{ij} 和 s 一次一密

1) s_{ij} 一次一密:即 P_i 第 $l(l\neq j)$ 次产生的子密钥 $s_{il}\neq s_{ij}$,实现操作为下列三种情形之一:

① P_i 通过更新种子 X_{i0} 或选择随机数 $k_{il}\neq k_{ij}$ 或两者组合操作,不难从 2.2 节的 1)至 5)可知,中间运算的许多参数将发生变化,主要有 $y_{il}\neq y_{ij}$(因 X_{i0} 变)$\Rightarrow A_{il}\neq A_{ij}$(或因 $k_{il}\neq k_{ij}$)$\Rightarrow Q_{il}\neq Q_{ij}$,从而 $s_{il}\neq s_{ij}$;

② D 更新矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$,易实现 $A_{il}\oplus X_{il}\neq A_{ij}\oplus X_{ij}\Rightarrow Q_{il}\neq Q_{ij}$,从而 $s_{il}\neq s_{ij}$;

③前两种情形的组合操作,易实现 $s_{il}\neq s_{ij}$。

2) s 一次一密,即第 $l(l\neq j)$ 次恢复的秘密 $s_l\neq s_j$。实现操作为:

$P_i(i=1,2,\cdots,n)$ 中至少有 1 个成员在产生其子密钥之前,须更新种子 X_{i0},从而使 $y_{il}\neq y_{ij}$,并保证多项式 $y_{il}()$ 与 $y_{ij}()$ 常系数不相等 $\Rightarrow Y_l()\neq Y_j()$ 且常系数不相等,从而 $Y_l(0)=s_l\neq s_j=Y_j(0)$。

2.5.2 s_{ij} 一次一密而 s 不变

1) $P_i(i=1,2,\cdots,n)$ 不更新种子 X_{i0},即对同一个成员来讲,第 $l(l\neq j)$ 次和第 j 次的种子相同,故 $y_{il}=y_{ij}\Rightarrow y_{il}()=y_{ij}()\Rightarrow Y_l()=Y_j()\Rightarrow Y_l(0)=s_l=s_j=Y_j(0)$;$P_i(i=1,2,\cdots,n)$ 也可以更新种子 X_{i0},但必须保证对于每个成员来讲,多项式 $y_{il}()$ 与 $y_{ij}()$ 常系数相等 $\Rightarrow Y_l()$ 与 $Y_j()$ 的常系数相等 $\Rightarrow Y_l(0)=s_l=s_j=Y_j(0)$;

2) $P_i(i=1,2,\cdots,n)$ 选择随机数 $k_{il}\neq k_{ij}$,或 D 更新矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$,或两者组合操作,均可实现 s_{ij} 一次一密,见 2.5.1 节的 1)。

2.5.3 s 一次一密而 s_{ij} 不变

1) $P_i(i=1,2,\cdots,n)$ 实现 s 一次一密,见 2.5.1 节的 2)。

2)由上可知:更新种子 $X_{i0}\Rightarrow y_{il}()\neq y_{ij}()$ 且易使 $Q_{il}\neq Q_{ij}$,而要保证 $s_{ij}(j=1,2,\cdots)$ 不变,则须授权集 GS 中每个成员交互信息来“确认”,以 P_i 为例,P_i 计算出第 l 次的 $s_{il}=Y_l(Q_{il})\bmod p$,比较 $s_{il}\overset{?}{=}s_{ij}$,若相等,则 s_{ij} 不变;若不等,则 P_i 通过再次更新种子 X_{i0} 改变 $y_{il}()$,重新计算 s_{il},直到 $s_{il}=s_{ij}$。不相等的情形又分为两种情形:当 $Q_{il}\neq Q_{ij}$,P_i 须让授权集 GS 中的其他所有成员重新计算 $y_{kl}(Q_{il})(k\neq i,k=1,2,\cdots,n)$ 并秘密传给 P_i,P_i 计算 $s_{il}=Y_l(Q_{il})=\prod_{k=1}^{n}y_{kl}(Q_{il})\bmod p$,直到 $s_{il}=s_{ij}$ 结束交互。当 $Q_{il}=Q_{ij}$,P_i 无须与授权集 GS 中的其他所有成员交互,只须自己通过更新种子 X_{i0} 改变 $y_{il}()$ 重新计算 $y_{il}(Q_{il})$,使 $s_{il}=Y_l(Q_{il})=Y_j(Q_{ij})=s_{ij}$ 为止。

2.6 门限 t 更新

门限 t 的更新可由 $P_i(i=1,2,\cdots,n)$ 和 D 协商完成,以下操作完成 $t\rightarrow t'(t'\neq t$ 且 $t'\leqslant n)$ 的门限更新:

1) $P_i(i=1,2,\cdots,n)$ 和 D 确定一个新门限 t',D 将门限结构改为 (t',n);

2)在计算子密钥 s_{ij} 和恢复秘密 s_j 时,多项式 $Y_j()$ 的构造改为在有限域 $GF(q^{t'})$ 上完成,即构造的 $Y_j()$ 最高次数由原来的 $(t-1)$ 变为 $(t'-1)$;

3)系统其他参数不变。

3 性能分析

3.1 安全性分析与比较

下面通过对可能的攻击进行分析,说明本文方案的安全性。

攻击 1 s_{ij} 的泄漏攻击分析。

分析 可产生一次一密或 $N(N\geqslant 2)$ 次一密的 s_{ij},取决于更新 X_{i0}、k_{ij} 这些参数和刷新矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 的频率,见 2.5.1 节的 1),N 值越小,安全性越高,素数模 q 可以设置为很大,使产生的 s_{ij} 序列为整周期,即实现周期最大化;若之前的子密钥 s_{ib} 泄漏,攻击者不会由 s_{ib} 得到 s_{ij} 和 s_j,因其不知道 $y_{1j}()$、$y_{2j}()$、…、$y_{nj}()$,从而不知道 $Y_j(Q_{ij})=s_{ij}$ 及 $Y_j(0)=s_j$。

攻击 2 s 泄漏攻击分析。

分析 同上,可产生一次一密或 $N(N\geqslant 2)$ 次一密的 s_j,主要取决于更新 X_{i0} 的频率,见 2.5.1 节的 2),s_j 的取值范围增大和随机性好也可以用增加素数 q 的值来实现;若之前的秘密 s_b 泄漏,攻击者不会由 s_b 得到 s_j 和 s_{ij},因其不知道 $Y_b()$,而 $Y_j()$ 与 $Y_b()$ 没有必然联系,故也不知道 $Y_j()$,从而不知道 $s_j=Y_j(0)$,也不知道 $s_{ij}=Y_j(Q_{ij})$。

攻击 3 参数值相同攻击。

分析 就同一轮(第 j 轮)不同成员来讲,$P_i(i=1,2,\cdots,n)$ 各自选择了含随机数、日期时间和 MAC 地址的种子 $X_{i0}(i=1,2,\cdots,n)$,故各自的种子都不同,再加上选择随机数 $k_{1j}\neq k_{2j}\neq\cdots\neq k_{nj}$ 的概率很大,故图 1 中各个输出参数 $y_{1j}\neq y_{2j}\neq\cdots\neq y_{nj}$、$X_{1j}\neq X_{2j}\neq\cdots\neq X_{nj}$、$A_{1j}\neq A_{2j}\neq\cdots\neq A_{nj}$、$Q_{1j}\neq Q_{2j}\neq\cdots\neq Q_{nj}$、$s_{1j}\neq s_{2j}\neq\cdots\neq s_{nj}$ 的概率很大;即便 $k_{ij}(i=1,2,\cdots,n)$ 中有相等的情形,如 $k_{ij}=k_{mj}(i\neq m,i,m\in[1,n])$,由于 $X_{i0}\neq X_{m0}$,故 $H(y_{ij})\neq H(y_{mj})\Rightarrow$FSA 输出不同 $\Rightarrow Q_{ij}\neq Q_{mj}\Rightarrow S_{ij}\neq S_{mj}$。就同一成员($P_i$)不同轮(第 j 次和第 b 次,$j\neq b$)来讲,P_i 对种子 X_{i0} 可以变也可以不变:若 X_{i0} 变,则第 j 次和第 b 次产生的主要参数不同;若 X_{i0} 不变,因选择随机数 $k_{ij}\neq k_{ib}$,则第 j 次和第 b 次产生的主要参数也不同。此外,D 也可以不定期刷新矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 来防止参数相同。

攻击 4 由公告牌上的公开信息不能推出保密参数。

分析 由于 X_{i0} 和 k_{ij} 是随机选择的数,且可选范围很大,没有确定的算法能推出或预测已选的或将选的 X_{i0} 和 k_{ij},所以输出 y_{ij}、X_{ij} 和 A_{ij} 都具有“不确定性”而使 s_{ij} 和 $Y_j()$ 具有“不确定性”;若攻击者知道了中间输出结果 X'_{ij} 和 A_{ij},也不能由矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 获得 y_{ij} 和 X_{ij},因 $H()$ 函数的单向安全性;由公开的 Q_{ij} 显然不能推出多项式 y_{ij};又由于许多保密参数可由协议实现一次一密,且每一轮是独立的,所以泄漏也不会对以前和未来那些轮的保密参数有影响。

攻击 5 参与者的欺骗与合谋攻击。

《计算机应用》Journal of Computer Applications

分析　恢复秘密时，参与者提供的子密钥须验证，见 2.3 节的 2)；设 $P_A \in GS$ 不诚实，经过 r 轮安全多方计算后，P_A 获得了 GS 中其他成员 $y_i(\,)(i=1,2,\cdots,n,\ i\neq A)$ 上的点 $(Q_{ij}, y_i(Q_{ij}))(j=1,2,\cdots,r)$，就某个成员来讲，$P_A$ 获得了 $y_i(\,)$ 上的 r 个点，若 $y_i(\,)$ 最高次数小于等于 $(r-1)$ 次，则 P_A 可由 Lagrange 插值公式获得 $y_i(\,)$，针对这种攻击的措施是：要么将 $y_i(\,)$ 最高次数变很大，要么每轮重选种子 X_{i0} 或至少在 $(r-1)$ 轮之后更新种子 X_{i0} 来改变 $y_i(\,)$；D 解密出第 j 轮的 $s_{ij}(i=1,2,\cdots,t)$，选择全部或部分用于第 $l(l\neq j)$ 轮秘密恢复的子密钥，因 $s_{ij}\neq s_{il}$ 且 $s_j\neq s_l$ 的可能性很大，即第 j 轮和第 l 轮比较，只要至少有一个成员的子密钥更新，或秘密更新，那么 D 这种攻击“恢复”的秘密是经不起数学正确性验证的，况且 P_i 在 D 授权下，得到 $SK_i(s_{ij})$ 可以自己或让其他成员验证其子密钥 s_{ij} 是否为当前的子密钥；少于 t 个成员不能恢复秘密 s，即系统少于 t 个不诚实成员的合谋攻击是不能恢复秘密的，这由 Lagrange 插值公式性质所决定。

本文方案与 EMSSS 方案安全性对比如表 1 所示。

3.2　开销分析与比较

本文方案的主要运算集中在 X_{n+1} 函数迭代、矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 的查询和求多项式 $Y(\,)$ 的模乘，计算开销取决于迭代次数 N、矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 的大小和 $Y(\,)$ 复杂度，可以通过减少 N、矩阵 $\boldsymbol{A}$、$\boldsymbol{X}'$ 大小、$y_i(\,)(i=1,2,\cdots,n)$ 复杂度及模数 q 减小开销。系统初始化工作后，每一轮主要是安全多方计算的通信开销，其他开销可以忽略。计算开销与参数更新频率有关，更新越频繁开销越大，安全性也越高。设 T_{modexp}、T_{EC}、T_{h}、T_{e} 分别是模乘、ECC 群加法、单向 Hash 函数、双线性映射运算的单位时间开销，其中 kP 与 $g^k \bmod p$ 所花费时间关系是：$8T_{\text{EC}}=T_{\text{modexp}}$，本文方案与 EMSSS 方案在主要开销上的对比如表 2 所示。

表 1　安全性比较

安全特性	本文方案	EMSSS 方案
非线性性	有：混沌算法和 FSA 实现	无：依赖线性单向 Hash
随机性	强：每轮种子、随机数、$\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 的随机性	弱：系统中，可充当随机数的参数较少
动态性	有：随机特性可使 s_{ij} 和 s 具有生命期，见 3.1 节的攻击 1～2 及其分析	无：协议中未体现
一次一密	有：动态性的实现可使系统达到一次一密安全级别，见 2.5 节	无：协议中未体现
保密性和认证性	有：2.2 节的 5)以签密方式完成，其他保密参数也以加密方式保存；另见 3.1 节的攻击 4 及其分析	无：协议中，没有体现加密和签名手段来实现这类特性
子密钥产生的安全性	强：通过安全多方计算产生，不仅成员间相互牵制以抗欺诈，而且成员间或一个成员多轮间参数重复概率较小，见 2.2 节	弱：直接选择随机数产生，成员间或一个成员多轮间参数重复概率较大，难以抗成员欺诈或合谋，见第 1 章的 2)
子密钥可验证性	有：2.3 节的 2)	有：见第 1 章的子密钥验证阶段
防参与者欺诈	好：见 3.1 节的攻击 5 及其分析	较好：见第 1 章的子密钥验证和秘密恢复阶段
防 D 欺诈	好：安全多方计算特性决定，见 2.2 节的 5)	差：因 D 权力偏大，如 $Q(x)$ 函数的定义和其他参数的计算偏多
防合谋攻击	好：见 3.1 节的攻击 5 及其分析	差：见第 1 章的子密钥验证可以防成员合谋攻击，但不能防与 D 合谋的攻击，因许多参数为 D 事先定义
抗消息重放攻击	有：见 2.2 节的 5)和 2.2 节的 1)	无：协议中未体现
前向安全性	高：随机数和动态机制确保攻击者在获得保密参数情况下也实施无效	低：因无动态特性，故没有严格意义的前向安全性
可信方参与	无：分布式安全多方计算实现，避免权力集中化	倾向：D 权力偏大，有可信方角色的倾向
强壮性	高：见 3.1 节的攻击 3～4 及其分析	较高：文献[1]分析中提及

表 2　两种方案开销比较

运算	本文方案	EMSSS 方案
模乘运算	$(n+1)T_{\text{modexp}}$，见式(4)～(5)	$2(n+1)T_{\text{modexp}}$，见式(1)～(2)
ECC 群加法运算	无	$(2n+6)T_{\text{EC}}$，若恢复 1 个秘密；$(2n+2m+4)T_{\text{EC}}$，若恢复 m 个秘密
单向 Hash 函数运算	某一轮运算，至少须 $2T_{\text{h}}$，若随机参数延续保持不变，则可减少 Hash 运算量	$(2n+m+1)T_{\text{h}}$，若恢复 1 个秘密；$2m(n+1)T_{\text{h}}$，若恢复 m 个秘密
双线性映射运算	无	$2nT_{\text{e}}$，若恢复 1 个秘密；$2mnT_{\text{e}}$，若恢复 m 个秘密。见式(3)
$D\rightarrow P_i$	若 m 个成员加入或退出，则 m 次，见 2.4 节	n 次，见第 1 章的 5)
$P_i\rightarrow D$	t 次，见 2.3 节的 1)和 3)	n 次，见第 1 章的 2)
P_i 之间	$n(n-1)$ 次，见 2.2 节的 5)	t 次，见第 1 章的秘密恢复阶段，若须恢复 m 个秘密，则为 $t\times m$ 次

4　结语

本文方案在安全性、灵活性和统领性方面较好，s_{ij} 一次一密的实现，可通过每轮更新种子 X_{i0}，更新 k_{ij}，更新矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 或任意组合完成；s 一次一密的实现，可通过每轮更新种子 X_{i0} 完成；s_{ij} 和 s 还可实现“N 次一密”，即通过 N 轮后才更新上述参数来完成。更新种子 X_{i0} 或 k_{ij} 由 $P_i(i=1,2,\cdots,n)$ 操作完成，更新矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{X}'$ 可由 D 操作完成，D 还可以设置为通过改变迭代函数的结构或 λ 值、迭代次数和改变门

(下转第 2545 页)

Journal of Computer Applications 计算机应用

and opinion mining [C]// Proceedings of the International Conference on Language Resources and Evaluation Conference. Malta: LREC, 2010: 1320 - 1326.

[2] MELVILLE P, GRYC W, LAWRENCE R D. Sentiment analysis of blogs by combining lexical knowledge with text classification[C]// Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2009: 1275 - 1284.

[3] GO A, BHAYANI R, HUANG L. Twitter sentiment classification using distant supervision[EB/OL]. [2012-10-10]. http://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

[4] 庞磊, 李寿山, 周国栋. 基于情绪知识的中文微博情感分类方法[J]. 计算机工程, 2012, 38(13): 156 - 158.

[5] 刘鲁, 刘志明. 基于机器学习的中文微博情感分类实证研究[J]. 计算机工程与应用, 2012, 48(1): 1 - 4.

[6] 张珊, 于留宝, 胡长军. 基于表情图片与情感词的中文微博情感分析[J]. 计算机科学, 2012, 39(Z3): 146 - 148.

[7] 徐琳宏, 林鸿飞, 赵晶. 基于语义理解的文本倾向性识别机制[J]. 中文信息学报, 2007, 21(1): 96 - 101.

[8] 徐琳宏, 林鸿飞, 潘宇. 情感词汇本体的构造[J]. 情报学报, 2008, 27(2): 180 - 185.

[9] 崔大志, 孙立伟. 在线评论情感词汇模糊本体库构建[J]. 辽宁工程技术大学学报, 2010, 12(4): 395 - 398.

[10] 邹忠民. 作家的情绪心理[J]. 昭通师范高等专科学校学报, 1992, 14(2): 64 - 71.

[11] 林传鼎. 社会主义心理学中的情绪问题——在中国社会心理学研究会成立大会上的报告(摘要)[J]. 社会心理科学, 2006, 21(1): 37 - 37.

[12] 许小颖, 陶建华. 汉语情感系统中情感划分的研究[C]// 第一届中国情感计算及智能交互学术会议论文集. 北京: 中国中文信息学会, 2003: 199 - 205.

[13] EKMAN P. Facial expression and emotion[J]. American Psychologist, 1993, 48(4): 384 - 392.

[14] ZHANG Y, LI Z M, REN F J. *et al.* Semi-automatic emotion recognition from textual input based on the constructed emotion thesaurus [C]// Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering. Piscataway, NJ: IEEE Press, 2005: 571 - 576.

[15] NECHES R, FIKES R E, GRUBER T R, *et al.* Enabling technology for knowledge sharing[J]. AI Magazine, 1991, 12(3): 36 - 56.

[16] 宋炜, 张铭. 语义网简明教程[M]. 北京: 高等教育出版社, 2004.

[17] GRUBER T R. Toward principles for the design of ontologies used for knowledge sharing[J]. International Journal of Human Computer Studies, 1995, 43(5): 907 - 928.

[18] 杨卓. 本体理论研究初探[J]. 中小企业管理与科技, 2011(21): 142 - 143.

[19] 滕悦明. 基于本体的远程教学辅助系统的设计与实现[D]. 北京: 北京邮电大学, 2007.

[20] 梅家驹, 竺一鸣, 高蕴琦, 等. 同义词词林[M]. 上海: 上海辞书出版社, 1996.

[21] 韩容洙. 现代汉语的程度副词[J]. 汉语学习, 2000(2): 12 - 15.

[22] 王海, 冯向前, 钱钢. 网页在线评论情感倾向的直觉模糊分类[J]. 计算机工程与应用, 2013, 49(1): 148 - 152.

(上接第 2539 页)

限 t, 来实现对系统参数的控制和对成员的进一步约束,二者可相互配合和相互约束,在安全性和效率两方面达到最佳平衡点,但该方案也有系统参数较多、成员数量大时开销会增加的不足,这是以后需完善的方向。

参考文献:

[1] FATEMI M, EGHLIDOS T, AREF M. An efficient multistage secret sharing scheme using linear one-way functions and bilinear maps [EB/OL]. [2012-03-02]. http://eprint.iacr.org/2012/121.

[2] CARLES R, LEONOR Y, YANG J. Finding lower bounds on the complexity of secret sharing schemes by linear programming[EB/OL]. [2012-03-02]. http://eprint.iacr.org/2012/464.

[3] TANG C, GAO S, ZHANG C. The optimal linear secret sharing scheme for any given access structure[EB/OL]. [2012-03-02]. http://eprint.iacr.org/2011/147.

[4] CRAMER R, DAMGARD I, MAURER U. General secure multi-party computation from any linear secret-sharing scheme[C]// EUROCRYPT 2000: Proceedings of the 19th International Conference on Theory and Application of Cryptographic Techniques. New York: ACM Press, 2000: 316 - 334.

[5] NIKOY V, NIKOVA S, PRENEEL B. Multi-party computation from any linear secret sharing scheme secure against adaptive adversary: the zero-error case[EB/OL]. [2012-03-02]. http://eprint.iacr.org/2003/006.

[6] YUEN K, CHEONG S W. A secret sharing scheme of prime numbers based on hardness of factorization[EB/OL]. [2012-03-02]. http://eprint.iacr.org/2012/222.

[7] KAYA K, SELCUK A. Secret sharing extensions based on the chinese reminder theorem[EB/OL]. [2012-03-02]. http://eprint.iacr.org/2010/096.

[8] FATEMI M, EGHLIDOS T, AREF M. A multi-stage secret sharing scheme using all-or-nothing transform approach[C]// Proceedings of ICICS 2009. New York: ACM Press, 2009: 449 - 458.

[9] WANG S J, TSAI Y R, SHEN J. Dynamic threshold multi-secret sharing scheme using elliptic curve and bilinear maps[C]// FGCN 2008: Proceedings of the Second International Conference on Future Generation Communication and Networking. Piscataway, NJ: IEEE Press, 2008, 2: 405 - 410.

[10] WONG T M, WANG C X, WING J M. Verifiable secret redistribution for threshold sharing schemes[EB/OL]. [2012-03-02]. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.160.4811.

[11] PAUL M B, ANYI V. Algorithmic chaos[EB/OL]. [2010-07-05]. http://arxiv.org/abs/nlin/0303016v1.

[12] LANGLOIS D, VERNIZZI F. Nonlinear perturbations of cosmological scalar fields[EB/OL]. [2010-07-05]. http://arxiv.org/pdf/astro-ph/0610064.pdf.

[13] BEIMEL A, ISHAI Y. On the power of nonlinear secret-sharing [EB/OL]. [2012-01-12]. http://eprint.iacr.org/2001/030.

[14] 张利远, 张恩. 基于中国剩余定理的可验证理性秘密共享方案[J]. 计算机应用, 2012, 32(11): 3143 - 3146.

[15] 罗黎霞, 张峻. 基于双线性映射的动态门限签名方案[J]. 计算机应用, 2010, 30(3): 677 - 679.

《计算机应用》Journal of Computer Applications