一种新型的撤销成员的无加密短群签名方案

马海英[1,2,3] 曾国荪[1,2]
(同济大学计算机科学及技术系 上海 201804)[1]
(嵌入式系统与服务计算教育部重点实验室 上海 201804)[2]
(南通大学计算机科学与技术学院 南通 226019)[3]

摘 要 针对撤销成员的群签名中如何降低群成员的计算量、缩短签名长度等问题,提出了一种新型的撤销成员的无加密短群签名方案,并证明了其安全性。基于XDDH,LRSW和SDLP假设,通过将有效期属性编入签名钥来实现成员的有效撤销;为了提高签名的效率,没有使用加密算法,而是采用签名随机化的方法来保持签名者的匿名性。在成员的通信和计算开销方面,本撤销方案比以往撤销方案有很大的优势,成员可以错过任意多次更新,签名时只需下载最新更新值即可,群公钥保持不变,签名和验证的计算开销与撤销成员数无关,签名长度仅为1195bits。

关键词 群签名,撤销成员,知识签名,IND-CCA2匿名性,安全性证明
中图法分类号 TP393 **文献标识码** A

Novel Revocable Short Group Signatures Scheme without Encryption

MA Hai-ying[1,2,3] ZENG Guo-sun[1,2]
(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)[1]
(The Key Laboratory of Embedded System and Service Computing, Ministry of Education, Shanghai 201804, China)[2]
(College of Computer Science and Technology, Nantong University, Nantong 226019, China)[3]

Abstract Aiming at the intrinsic problems in revocation group signatures, such as reducing group member's computational costs, shortening the signature length and so on, a novel revocation short group signature scheme without encryption was proposed based on the XDDH, LRSW and SDLP assumptions, and it's security was proven. Member revocation was implemented by encoding the validity time into group signature key. In particular, our scheme does not use standard encryption and relies on re-randomizable signature schemes that hide the signed message so as to preserve the anonymity of signers. Our solution outperforms all prior solutions for member revocation in terms of communication and computational costs for the members. Group public key remains constant, and computational costs of signing and verifying are independent of the revocable number, and the signature is only 1195 bits in size.

Keywords Group signatures, Revocation, Signature-proof-of-knowledge, IND-CCA2 anonymity, Security proof

1 引言

1991年,Chaum和Heyst提出群签名的概念[1]。群签名允许任何群成员代表群进行匿名签名,必要时群管理员能够揭露签名者的真实身份。此后,提出大量群签名方案,其安全性主要包括不可伪造性、匿名性、追踪性、无关联性和防陷害性。2003年,Bellare等[2]将静态群签名方案诸多的安全特性标准化,并指出一个安全的群签名方案必须包括完全匿名性和完全追踪性。2005年,Bellare等人[3]将群签名扩展为动态,其安全性主要包括正确性、匿名性、追踪性和防陷害性。群签名同时提供了匿名性和可追踪性,其匿名性为合法用户提供隐私保护,其可追踪性又使得可信机构及时跟踪违法行为,这使得它在不同领域得到了更为广泛的应用,如车载网、电子支付、电子选举及网上拍卖等。

Ateniese和Tsudik指出成员撤销是群签名应用中的一个重要问题[4],随后有大量成员撤销方案被提出[5-7]。Bresson和Stern首次提出采用成员撤销列表方法实现可撤销的群签名方案[5];西安电子科技大学王尚平教授通过更新算子的方法极大地改进Bresson-Stern方案;中科院武传坤教授提出后向无关性的撤销方案。但是此类方案中验证算法的计算量和被撤销成员数线性相关,不适合撤销成员数较多的群。Camenisch等人提出基于累加器[7]的撤销方案,群管理员公

到稿日期:2011-05-02 返修日期:2011-08-01 本文受863项目(2009AA012201),973计划课题(2007CB316502),国家自然基金项目(90718015),NSFC-微软亚洲研究院联合资助项目(60970155),教育部博士点基金项目(20090072110035),上海市优秀学科带头人计划项目(10XD1404400),高效能服务器和存储技术国家重点实验室开放基金项目(2009HSSA06),同济大学青年基金(0800219105,2009kj030),南通大学自然科学基金(13040024)资助。

马海英(1977-),女,博士生,讲师,主要研究方向为密码学、网络安全,E-mail:m_hying@163.com;曾国荪(1964-),男,博士,教授,博士生导师,主要研究方向为信息安全、并行计算。

布累加值，群成员通过零知识证明的方式来证明他拥有相应于累加值的证据，验证算法独立于被撤销成员的个数。但是，不管群中增加还是撤销成员，签名者都需要更新自己的签名钥，在最坏的情况下，更新签名钥的计算量与成员数相关。在其改进方案中，更新签名钥的计算量与撤销成员数相关。最近，Camenisch 等人在匿名证书系统中通过引入一种新的非交互技术[7]来更新由群管理员控制的用户属性，通过将有效期属性编入证书来实现用户的有效撤销。Bichsel 等在素数阶群上构造了无需加密方案且具有 IND-CCA2 匿名性的短群签名方案[8]。

本文给出了一种新型撤销成员的无加密短群签名方案，其通过将有效期编入签名钥来实现成员的有效撤销；为了提高签名的效率，没有使用加密算法，而是采用签名随机化的方法来保持签名者的匿名性。群管理员可以周期性地离线更新未撤销成员的签名钥，并发布与每个签名钥对应的更新值和当前有效期，群成员下载其更新值并重新有效化他的签名钥，从而证明自己拥有当前时间段有效的签名钥。在成员的通信和计算开销方面，本文的撤销方案比以往撤销方案有很大的优势，成员可以错过任意多次更新，签名时只需下载最新更新值即可，群公钥保持不变，签名长度仅为 1195bits。

2 预备知识

本节介绍将用到的标号、术语、双线性群和困难性假设，知识签名与文献[8]类似。

2.1 标号和术语

假定 S 表示集合，$x\leftarrow S$ 表示随机从 S 中均匀选取一个元素赋给变量 x。$\{0,1\}^*$ 和 $\{0,1\}^l$ 分别表示任意长度二进制串和长度为 l 的二进制串，$H:\{0,1\}^*\rightarrow Z_q$ 是一个防碰撞单向哈希函数，符号 $\|$ 表示两个数据串的级联。η 为系统给定的安全参数，素数 $q=\Theta(2^\eta)$。假定 A 是一个随机算法，那么 $z\leftarrow_R A(x,y,\cdots)$ 表示随机算法 A 在输入 $x,y\cdots$ 的情况下按一定的概率输出 z。为了方便表示，将附加了信誉值和有效期的消息记为有效消息，即 m_t。

2.2 双线性群和困难性假设

本文利用了椭圆曲线上非对称双线性群，其中 G_1，G_2 和 G_T 为 q 阶乘法循环群。一般地，用英文字符表示群 G_1 中的元素，用带波浪的英文字符表示群 G_2 中的元素，用小写希腊字符表示 Z_q 中的元素，g 和 $\tilde{g}$ 分别为群 G_1 和 G_2 的生成元。

可计算的映射 $\hat{e}:G_1\times G_2\rightarrow G_T$，该映射具有两个性质：(1)双线性，即对所有的 $g\in G_1,\tilde{g}\in G_2$ 及 $x,y\in Z_q$，有 $\hat{e}(g^x,\tilde{g}^y)=\hat{e}(g,\tilde{g})^{xy}$；(2)非退化性，即 $\hat{e}(g,\tilde{g})\neq 1$，则称群 (G_1,G_2) 是一对双线性群。双线性群包含 4 种类型，其中 3 型双线性群要求 $G_1\neq G_2$，且不存在 $\psi:G_2\rightarrow G_1$ 的同态映射。事实上，根据带宽和计算效率，它能提供最有效的实现。

假设 1(LRSW)　任取 $x,y\rightarrow Z_q$，令 $\tilde{X}\leftarrow\tilde{g}^x,\tilde{Y}\leftarrow\tilde{g}^y$，预言机 $O_{\tilde{X},\tilde{Y}}(\cdot)$，输入 $\mu\in Z_q$，输出 $(a,a^y,a^{x+xy\mu})\in G_1^3$，其中随机选择 $a\in G_1$，将 $\tilde{X},\tilde{Y},O_{\tilde{X},\tilde{Y}}(\cdot)$ 给任意多项式时间 (ppt) 敌手 $\mathcal{A}$，$\mathcal{A}$ 输出一个未询问预言机的三元组 $(a,a^y,a^{x+xy\mu})$ 的概率是可以忽略的。

假设 2(XDDH)　对双线性群 G_1,G_2,G_T，如果 G_1 中 DDH 问题是困难的，则称 XDDH 假设在双线性群中成立。

假设 3(SDLP)　给定 $(g^\mu,\tilde{g}^\mu)\in G_1\times G_2$，计算 μ 是困难的。

假设 4(q-SDH)　给定上述双线性群，如果对任意 ppt 敌手 $\mathcal{A}$，给定 q 元组 $(\tilde{g}^\gamma,\tilde{g}^{\gamma^2},\cdots,\tilde{g}^{\gamma^q})$，任意 $\alpha\in Z_q$，输出 $(\tilde{g}^{\frac{1}{(\gamma+\alpha)}},\alpha)$ 是困难的。

3 撤销成员的短群签名方案

这里提出一种新型的撤销成员的无加密短群签名方案。其主要思想是：用户首先和群管理员执行一次加入交互协议，获得签名钥。群管理员周期性地离线更新未撤销成员的签名钥，成员必须下载最新更新值并重新有效化签名钥，才能产生有效签名。撤销成员因无法更新签名钥不能产生有效签名。本方案适合于周期性撤销成员的情况，具体方案如下：

Setup　参数 $\eta,q,g,\tilde{g},G_1,G_2,G_T$ 和 $\hat{e}$ 的选择如 2.1 节所示。此外，定义两个哈希函数 $H_1:\{0,1\}^*\rightarrow Z_q,H_2:\{0,1\}^*\rightarrow Z_q$。$DSSign(\cdot,\cdot)$ 和 $DSVerify(\cdot,\cdot,\cdot)$ 是一个安全签名和验证算法。群管理员随机选择 $x,y,z\leftarrow Z_q$，计算 $\tilde{X}\leftarrow\tilde{g}^x,\tilde{Y}\leftarrow\tilde{g}^y,\tilde{Z}\leftarrow\tilde{g}^z$，群公钥为 $gpk\leftarrow(\tilde{X},\tilde{Y},\tilde{Z})$，群管理员私钥为 $gmsk\leftarrow(x,y,z)$。

UKey　用户 u_i 利用数字签名算法为自己生成公/私钥对 $(upk[i],usk[i])$，将 $upk[i]$ 发送给群管理员并公开。

GJoin　成员 u_i 和管理员执行加入交互协议，假设协议是在安全信道上执行。如果协议在执行过程中一方验证失败，将会通知另一方协议失败并终止。

①群管理员随机选择 $\alpha\leftarrow Z_q$，计算 $\acute{\alpha}\leftarrow H_2(\alpha)$，将 $\acute{\alpha}$ 发送给用户 u_i。

②用户 u_i 选择 $\tau\leftarrow Z_q$，计算 $s\leftarrow g^\tau,\tilde{r}\leftarrow\tilde{X}^\tau,k\leftarrow\hat{e}(g,\tilde{r}),\tilde{\sigma}\leftarrow DSSign(usk[i],k)$，将 $(s,\tilde{r},\tilde{\sigma})$ 发送给管理员，用户 u_i 和管理员执行 $FPK\{(\tau):s\leftarrow g^\tau,\tilde{r}\leftarrow\tilde{X}^\tau\}$。

③群管理员用 $DSVerify(upk[i],\hat{e}(g,\tilde{r}),\tilde{\sigma})$ 来验证签名 $\tilde{\sigma}$。如果验证通过，计算 $z\leftarrow s\cdot g^\alpha,\tilde{\omega}\leftarrow\tilde{r}\cdot\tilde{x}^\alpha$，授予 u_i 的当前有效期 t，并将 $(1,\tilde{\omega},\tilde{r},\alpha,\tilde{\sigma},t)$ 存储在注册表 $reg[i]$ 中。选择 $\rho\leftarrow Z_q$，计算 $a=g^\rho,A=a^x,b=a^y,B=A^y,P=a^{xy}\rho^{xy},c=PA^{y\rho'}$，并将 (a,A,b,B,c,α,t) 发送给用户 u_i。此外，群管理员与用户 u_i 执行协议：

$$FPK\{(x,y,\rho,\gamma,z):c=a^xz^\gamma A^{xy\rho'}\wedge a=g^\rho\wedge\tilde{X}=\tilde{g}^x\wedge\tilde{Y}=\tilde{g}^y\wedge\tilde{Z}\leftarrow\tilde{g}^z\wedge b^\gamma/g^\gamma=1\}$$

式中，$\gamma=\rho xy$。注意该证明允许用户验证 $x,y\neq 0$。$state_o=(P,A,t)$，群管理员将 $(1,u_i,state_o)$ 存储在成员状态信息表 $List[i]$。

④用户 u_i 计算 $\xi=\tau+\alpha \bmod q$，并验证 $\acute{\alpha}=H_2(\alpha)$，$\hat{e}(a,\tilde{Y})=\hat{e}(b,\tilde{g})$，$\hat{e}(A,\tilde{Y})=\hat{e}(B,\tilde{g})$，$\hat{e}(A,\tilde{g})=\hat{e}(a,\tilde{Z})$ 是否相等，如果所有等式验证成功，用户 u_i 存储私钥 $gsk[i]\leftarrow(\xi,(a,A,b,B,c,t))$。

Revoke　群管理员用 $(0,u_i,state_o)$ 代替 $(1,u_i,state_o)$，第一个元素表示该用户 u_i 已被撤销。

GUpdate　管理员定期更新所有未撤销成员的签名钥，t' 为当前有效期，计算 $c'=PA^{y\rho't'}$，输出 $update_o=(c',t')$，未撤销的成员通过下载自己最新 $update_o$ 来更新签名钥，从而得到自己有效的签名钥 (a,A_1,b,B_1,c',t')。

GSign　成员 u_i 利用 $gsk[i]$ 对有效消息 m_t 签名。首先，用户随机选择 $\zeta \leftarrow Z_q$，计算 $\bar{a}=a^{\zeta}, \bar{A}=A^{\zeta}, \bar{b}=b^{\zeta}, \bar{B}=B^{\zeta}, \bar{c}=c^{\zeta}$，然后计算 Σ[8]，从而证明 $(\bar{a},\bar{A},\bar{b},\bar{B},\bar{c})$ 是一个对 (ξ,t) 有效的 CL-C[9] 签名。最后用户 u_i 输出签名 $\sigma \leftarrow (\bar{a},\bar{A},\bar{b},\bar{B},\bar{c},\Sigma) \in G_1^5 \times Z_q^2$。

$$\Sigma \leftarrow SPK\{(\xi): \frac{e(\bar{c},\tilde{g})}{e(\bar{a},\tilde{X})e(\bar{B},\tilde{X})^{t}} = e(\bar{b},\tilde{X})^{\xi}\}(m)$$

GVerify　为了验证 σ 是对有效消息 m_t 的签名，验证者首先验证等式 $\hat{e}(\bar{a},\tilde{Y})=\hat{e}(\bar{b},\tilde{g})$，$\hat{e}(\bar{A},\tilde{Y})=\hat{e}(\bar{B},\tilde{g})$，$\hat{e}(\bar{A},\tilde{g})=\hat{e}(\bar{a},\tilde{Z})$ 是否相等，然后验证 Σ 是否有效。如果任何一个等式验证失败，输出 0；否则输出 1。

GOpen　给定有效消息 m_t 的签名 σ，在签名通过验证之后，对注册表中每一项 $reg[i]=(\tilde{\omega}_i,\tilde{r}_i,\alpha_i,\bar{\sigma}_i,t)$，验证等式 $\hat{e}(\bar{c},\tilde{g})=\hat{e}(\bar{a},\tilde{X})\hat{e}(\bar{B},\tilde{X})^{t}\hat{e}(\bar{b},\tilde{\omega}_i)$ 是否成立。对于满足该等式的 $\tilde{\omega}_i$，群管理员获取 $\alpha_i,\bar{\sigma}_i$，计算 $k_i \leftarrow \hat{e}(g,\tilde{r}_i)$ 和 Π，输出 $(u_i,\pi=(k_i,\bar{\sigma}_i,\Pi))$。

$$\Pi \leftarrow SPK\{(\alpha_i,\overline{\omega_i}): \frac{\hat{e}(\bar{c},\tilde{g})}{\hat{e}(\bar{a},\tilde{X})\hat{e}(\bar{B},\tilde{X})^{t}} = \hat{e}(\bar{b},\tilde{\omega}_i) \wedge k_i = \frac{\hat{e}(g,\tilde{\omega}_i)}{\hat{e}(g,\tilde{X})^{\alpha_i}}\}$$

GJudge　验证者首先利用 $DSVerify(upk[i],k,\bar{\sigma})$ 验证签名 $\bar{\sigma}$ 有效，然后输入 $gpk, m\_t, \sigma=(\bar{a},\overline{A_1},\bar{b},\overline{B_1},\bar{c},\Sigma)$ 和 π。如果 $GVerify(gpk,m\_t,\sigma)=1$，且 π 有效，输出 1。否则，输出 0。

4　群签名方案的安全性分析

撤销成员的短群签名方案必须满足 4 个安全属性：(1)正确性；(2)匿名性；(3)追踪性；(4)防陷害性。从方案的具体描述中，不难得出本文方案的正确性，本文群签名方案的匿名性、可追踪性和防陷害性可由以下 3 个定理得出。

定理 1　在 XDDH 和 SDLP 假设下，本文提出的基于随机模型的撤销成员的短群签名方案具有匿名性。

证明：设敌手 $\mathcal{A}$ 能够以 $Adv^{anon}_{\mathcal{S},\mathcal{A}}(\eta)\geqslant\varepsilon$ 的优势打破匿名性，我们构造一个仿真器 $\mathcal{S}$，以 $\varepsilon/2$ 优势解决群 G_1 中的 DDH 问题。设 $\mathcal{S}$ 的挑战 DDH 元组为 $(g_0,g_1,g_2,g_3)=(g,g^{\mu},g^{\nu},g^{\omega})$，$\mathcal{S}$ 输出 δ(当 $\omega \leftarrow \mu\nu, \delta=1$；当 $\omega \leftarrow Z_q, \delta=0$)的猜测 $\delta' \in \{0,1\}$。

给定群 G_1，G_2 和 G_T，仿真器 $\mathcal{S}$ 从敌手 $\mathcal{A}$ 中检索出 HU，$DU \subseteq \{1,\cdots,n\}$ 并取 $g \leftarrow g_0$，选择生成元 $\tilde{g} \in G_2$ 和 $x,y,z \leftarrow Z_q$，计算 $\tilde{X} \leftarrow \tilde{g}^{x}, \tilde{Y} \leftarrow \tilde{g}^{y}, \tilde{Z} \leftarrow \tilde{g}^{z}$。然后，$\mathcal{S}$ 将管理员的私钥设置为 $gmsk \leftarrow (x,y,z)$，公钥为 $gpk \leftarrow (g,\tilde{g},\tilde{X},\tilde{Y},\tilde{Z})$，并将其公钥提供给敌手 $\mathcal{A}$。

仿真器 $\mathcal{S}$ 为所有诚实成员生成密钥对 $(usk[i],upk[i])$，并选择 $\rho_i,n_i \leftarrow Z_q$，计算他们的群签名密钥为 $(a_i,A_i,b_i,B_i,c_i,t)=(g_0,a^{z_1},a^{y},A^{y},a^{x}g_1^{\rho_i x y n_i}A^{xyt},t)$。注意，诚实成员的签名私钥为 $\xi_i=\rho n_i$，其中 $n_i \leftarrow Z_q$，这样的元组与本方案的用户密钥具有相同的分布。此外，$\mathcal{S}$ 随机选择 $k_i \in G_T$，计算签名 $\bar{\sigma}_i \leftarrow DSSign(usk[i],k_i)$，将 $(a_i,A_i,b_i,B_i,c_i,n_i,t)$ 存储在 $gsk[i]$。

$\mathcal{S}$ 和 $\mathcal{A}$ 进行交互并仿真下列预言机询问值：

$H_1(R)$：$\mathcal{S}$ 维护一个列表 L_1 存储已询问过的签名预言机询问值 (R,Cha)，对于每一个新的签名 R，$\mathcal{S}$ 指定一个新的随机值 $Cha \leftarrow \{0,1\}^{l}$ 与其对应。

$H_2(\alpha)$：$\mathcal{S}$ 维护一个列表 L_2 存储已询问过的哈希函数返回值 $(\alpha,\dot{\alpha})$，对于每一新的 α，$\mathcal{S}$ 指定一个新的随机值 $\dot{\alpha} \leftarrow \{0,1\}^{l}$ 与 α 对应。

$UKey(i,upk)$：$\mathcal{S}$ 执行 CA 的角色，发布 $upk[i] \leftarrow upk$。注意每个用户都可以得到一个经过认证的 upk 的副本。

$GJoin_{DM}(i)$：对于每一个非诚实用户 $i \in DU$，$\mathcal{S}$ 利用 $gmsk$ 仿真执行群签名方案中的管理员角色，$\mathcal{S}$ 将 $(1,u_i,\tilde{\omega}_i,\tilde{r}_i,\alpha_i,\bar{\sigma}_i,state_{\sigma},t)$ 存储到 $reg[i]$。

$Revoke(i)$：当撤销成员 i 时，仿真器 $\mathcal{S}$ 用 $(0,u_i,state_{\sigma})$ 代替 $(1,u_i,state_{\sigma})$，第一个元素表示成员 u_i 已被撤销。

$GUpdate$：对于非撤销成员，$\mathcal{S}$ 利用 $gmsk$ 仿真执行群签名方案中的管理员角色，输出 (c',t')，成员下载更新自己的签名钥 (a,A,b,B,c',t')。

$GSign$：对于 $\mathcal{A}$ 给定用户 $i \in HU$ 和消息 m_t，$\mathcal{S}$ 检索到 $gsk[i]=(a_i,A_i,b_i,B_i,c_i,n_i,t)$。$\mathcal{S}$ 任取 $\zeta \leftarrow Z_q^*$ 重新随机化签名钥，得到 $(\bar{a},\bar{A},\bar{b},\bar{B},\bar{c})=(a^{\zeta},A^{\zeta},b^{\zeta},B^{\zeta},c^{\zeta})$。$\mathcal{S}$ 利用随机预言机 $H_1(\cdot)$ 仿真 Σ 的知识签名，并将 $\sigma=(\bar{a},\bar{A},\bar{b},\bar{B},\bar{c},\Sigma)$ 返回给 $\mathcal{A}$。$\mathcal{S}$ 将 (i,m_t,σ) 添加到 sgn 列表中。

$GOpen$：仿真器 $\mathcal{S}$ 区别对待由诚实成员和非诚实成员产生的签名。

1)如果 σ 由非诚实成员生成，则 $\mathcal{S}$ 利用 reg 中的信息打开签名并生成证据 π。

2)如果 σ 由诚实成员生成，则 σ 要么是由 $\mathcal{S}$ 利用 $GSign(\cdot,\cdot)$ 生成的，要么是由敌手 $\mathcal{A}$ 伪造的。从定理 3 的防陷害性可以排除第二种情况，在第一种情况中，$\mathcal{S}$ 利用签名列表 sgn 中 m_t 和 σ 查找出相应的用户 i，并查找出相应的 $gsk[i]=(a_i,A_i,b_i,B_i,c_i,n_i,t,k_i)$，$\mathcal{S}$ 利用随机预言机 $H_1(\cdot)$ 仿真 Π 的知识签名：

$$\Pi \leftarrow SPK\{(\alpha,\overline{\omega_i}): \frac{\hat{e}(\bar{c},\tilde{g})}{\hat{e}(\bar{a},\tilde{X})\hat{e}(\overline{B_1},\tilde{X})^{t}} = \hat{e}(\bar{b},\tilde{\omega}_i) \wedge k_i = \frac{\hat{e}(g,\tilde{\omega}_i)}{\hat{e}(g,\tilde{X})^{\alpha}}\}$$

$\mathcal{S}$ 将 $(u_i,\pi=(k_i,\bar{\sigma}_i,\Pi))$ 发送给敌手 $\mathcal{A}$。

$Ch_b(i_0,i_1,m)$：首先，$\mathcal{S}$ 任取 $b \leftarrow \{0,1\}$，并查找 $gsk[i]=(a_{ib},A_{ib},b_{ib},B_{ib},c_{ib},n_{ib},t,k_{ib})$，然后构造 $(a^*,A^*,b^*,B^*,c^*)=(g_2,g_2^{x},g_2^{y},g_2^{yx},g_2^{z}g_3^{xyn_{ib}}g_2^{xyxt})$。如果给定的挑战 DDH 元组是一个 DDH 对，即 $\delta=1$，则 $c^*=g^{\mu x+xyn_{ib}\mu n_{ib}}g^{\mu xyxt}$，挑战签名和真实签名具有相同的分布。如果 $\delta=0$，则无论 b 取任何值，c^* 都服从 G_1 中的平均分布。最后，$\mathcal{S}$ 利用 $H_1(\cdot)$ 仿真 Σ 的知识签名。

执行完上述过程，敌手 $\mathcal{A}$ 将输出一个猜测值 b'。如果 $b'=b$，仿真器 $\mathcal{S}$ 输出 $\delta'=1$，否则输出 $\delta'=0$。$\mathcal{S}$ 解决 DDH 对的优势为 $Adv_{\mathcal{S}}^{XDDH}=\Pr[\delta'=1|\delta=1]-\Pr[\delta'=1|\delta=0]$。当 $\delta=0$ 时，c^* 独立于 b 选择服从 G_1 中的均匀分布，因此 $b'=b$ 的概率为 $\Pr[b'=b|\delta=0]=1/2$。由于当 $b'=b$，$\mathcal{S}$ 猜测 $\delta'=1$，于是 $\Pr[\delta'=1|\delta=0]=1/2$。当 $\delta=1$，挑战签名与真实攻击环境的签名具有相同的分布。根据 δ' 的选择规则，可得 $\Pr[\delta'=1|\delta=1]=\Pr[b'=b|\delta=1]$，上述等式中的值也为敌手 $\mathcal{A}$ 赢得匿

名性的概率。因此

$$\Pr[b'=b\mid\delta=1]=\frac{1}{2}(\Pr[\mathrm{Exp}_{CS,A}^{anon-1}(\eta)=1]+\Pr[\mathrm{Exp}_{CS,A}^{anon-1}(\eta)=0])=\frac{Adv_{CS,A}^{anon}(\eta)+1}{2}$$

敌手 $\mathcal{A}$ 打破 XDDH 的优势为 $Adv_{\mathcal{S}}^{XDDH}\geqslant\frac{\varepsilon}{2}$。

定理 2 在 LRSW 假设下，本文提出的基于随机模型的群签名方案具有追踪性。

证明：如果存在一个敌手 $\mathcal{A}$ 能够赢得可追踪性，那么可以构造一个敌手 $\mathcal{B}$ 打破 CL-C 签名方案的不可伪造性。

首先，为了应用 Forking 引理[11]，将敌手 $\mathcal{A}$ 改造成 $\mathcal{A}'$，给定敌手 $\mathcal{A}$，构造算法 $\mathcal{A}'$，给定 $\mathcal{A}'$ 输入 $(\tilde{X},\tilde{Y},\tilde{Z})$ 和随机值 R，$\mathcal{A}$ 和 $\mathcal{A}$ 执行以 $(\tilde{X},\tilde{Y},\tilde{Z})$ 和随机值 R'（R' 由 R 确定）为输入的交互。$\mathcal{A}'$ 仿真下列预言机询问，同时维护指针 ctr、成员状态信息表 $List$、列表 L_1、L_2 和相应的注册表 reg：

$H_1(R)$：$\mathcal{A}'$ 在列表 L_1 中查找 (R,j,Cha) 并返回 Cha。如果这样的元组不存在，将 ctr 加 1，选择随机值 $Cha\leftarrow\{0,1\}^t$，并将 (R,ctr,Cha) 添加到列表 L_1 中。$H_2(\alpha)$ 和 $Revoke(i)$：同定理 1。

$GJoin_{DM}(i)$：$\mathcal{A}'$ 设置 $upk[i]\leftarrow upk$，作为用户 i 的经过认证的 upk。$\mathcal{A}'$ 选择 $\alpha\leftarrow Z_q$，计算 $\hat{\alpha}\leftarrow H_2(\alpha)$ 将 $\hat{\alpha}$ 发送给 $\mathcal{A}$，$\mathcal{A}'$ 收到 $(s,\tilde{r},\tilde{\sigma})$ 后，$\mathcal{A}$ 重置 $\mathcal{A}$ 以便从 $FPK\{(\tau)\}$ 提取 τ。τ 不能被成功提取的概率为 $1/q$。当 $\mathcal{A}'$ 失败时，输出 $(0,\varepsilon)$ 并终止。否则，$\mathcal{A}'$ 计算 $\xi=\tau+\alpha \bmod q$，并询问签名预言机，得到关于 (ξ,t,r) 的一个 CL-C 签名 (a,A,b,B,c,t)。然后，将 (a,A,b,B,c,t,α) 发送给 $\mathcal{A}$，并利用零知识仿真器来模拟 $FPK\{(x,y,\rho,\gamma,z)\}$，由于零知识证明完备性，仿真器不会造成概率损失。最后，$\mathcal{A}'$ 将 $(\tilde{\omega}_i=\tilde{X}^{\xi},\tilde{r}_i,\alpha_i,\tilde{\sigma}_i,t)$ 存储在 $reg[i]$ 中。

$GUpdate$：对于非撤销用户，$\mathcal{A}'$ 询问签名预言机得到 (ξ,t') 的 CL-C 签名 (a,A_1,b,B,c',t')，并以此作为更新后的签名钥发送给成员。

$GOpen$：如果 $GVerify(gpk,m\_t,\sigma)=0$，$\mathcal{A}'$ 返回 $\perp$，否则将 σ 解析成 (a,A,b,B,c,t,r,Σ) 并寻找 $(\tilde{\omega}_i,\tilde{r}_i,\alpha_i,\tilde{\sigma}_i,t)\in join$，使得 $\hat{e}(c,\tilde{g})/(\hat{e}(a,\tilde{X})\,\hat{e}(B_1,\tilde{X})^t)=\hat{e}(a',\tilde{\omega}_i)$。如果找到这样的元组，用实际的打开算法和 $\tilde{\omega},\tilde{r},\alpha$ 构造证据 π，返回 $(u_i,(k,\tilde{\sigma},\pi))$。

当 $\mathcal{A}$ 输出伪造签名 $(m\_t,\sigma=(a,A,b,B,c,(Cha,Rsp)))$，依据伪造签名的有效性，将其区分为两种情况。如果伪造签名无效，即 $GVerify(gpk,m\_t,\sigma)=0$ 或利用 $\tilde{\omega}$ 可使 σ 追踪到一个注册用户，$\mathcal{A}'$ 输出 $(0,\varepsilon)$ 并终止。否则，$\mathcal{A}'$ 查找索引 j 使得 $(R,j,Cha)\in L_1$，其中 $R=D\|E\|F\|m\_t$，

$$D=\frac{\hat{e}(c,\tilde{g})}{\hat{e}(a,\tilde{X})\,\hat{e}(B_1,\tilde{X})^t},\ E=\hat{e}(b,\tilde{\omega}_i),\ F=D^{Cha}E^{Rsp}$$

由于 $\mathcal{A}$ 生成的伪造签名能通过验证，并被打开到非注册用户或撤销用户，因此上述元组一定存在。称 $\mathcal{A}$ 的第 j 次 $H_1(\cdot)$ 询问为关键哈希询问，$\mathcal{A}'$ 输出 (j,σ) 并终止。

以算法 $\mathcal{A}'$ 和由输入生成器 IG 生成的 gpk 作为 forking 引理的输入。如果敌手 $\mathcal{A}$ 赢得追踪性的概率为 ε，即 $Adv_{CS,A}^{trace}(\eta)=\varepsilon$，则 $\mathcal{A}'$ 输出 (j,σ) 的概率 $acc\geqslant\varepsilon-n/q$，其中 $j\geqslant 1$，n/q 对应于加入协议中提取器失败的概率。

应用 forking 引理可以得到一个算法 $F_{\mathcal{A}}$，它以 frk[8] 概率输出 $(1,\sigma_1,\sigma_2)$。下面证明 $\mathcal{S}$ 可以利用 $F_{\mathcal{A}}$ 生成的两个签名 $\sigma_1=(a',A',b',B',c',(Cha',Rsp'))$，$\sigma_2=(a'',A'',b'',B'',c'',(Cha'',Rsp''))$ 来伪造一个 CL-C 签名。令

$$D'=\frac{\hat{e}(c',\tilde{g})}{\hat{e}(a',\tilde{X})\,\hat{e}(B',\tilde{X})^t},\ E'=\hat{e}(b',\tilde{X}),\ F'=D'^{Cha'}E'^{Rsp'}$$

$$D''=\frac{\hat{e}(c'',\tilde{g})}{\hat{e}(a'',\tilde{X})\,\hat{e}(B'',\tilde{X})^t},\ E''=\hat{e}(b'',\tilde{X}),\ F''=D''^{Cha''}E''^{Rsp''}$$

由于 $\mathcal{A}'$ 的两次执行在输入、随机值和直到关键点处哈希返回值都是相同的，因此两次哈希询问值的证据必须相同，即 $D'=D'',E'=E'',F'=F''$。由 $E'=E''$，得 $b'=b''$；由 $\hat{e}(a',\tilde{Y})=\hat{e}(b',\tilde{g})$ 和 $\hat{e}(a'',\tilde{Y})=\hat{e}(b'',\tilde{g})$，得 $a'=a''$。同理，由本方案的验证等式可得 $A'=A'',B'=B''$。由 $D'=D''$，得 $c'=c''$。由 forking 引理可得 $Cha'\neq Cha''$ 且都小于 q，所以 $Cha'-Cha''\neq 0 \bmod q$。由 F' 和 F'' 的方程可得 $\xi=(Rsp''-Rsp')/(Cha'-Cha'') \bmod q$ 满足 $\hat{e}(c',\tilde{g})=\hat{e}(a',\tilde{X})\,\hat{e}(B',\tilde{X})^t\hat{e}(a',\tilde{X})^{\xi}$，该等式是 CL-C 签名验证的最后一个，由群签名 σ_1 的有效性可得 CL-C 签名验证的其余等式，所以 (a',A',b',B',c') 是 (ξ,t) 的有效签名。其次，由于敌手 $\mathcal{A}$ 打破了追踪性，因此 σ_1 被打开到非注册用户或撤销用户，即 (ξ,t) 不出现在注册表中。最后，$\mathcal{S}$ 通过签名预言机询问 CL-C 签名的所有消息 (ξ,t) 都在注册表中，因此 $((\xi,t),\sigma_1)$ 为 $\mathcal{S}$ 输出的伪造签名。

定理 3 在 SDLP 假设和 CL-C[9] 签名的不可伪造性条件下，本文提出的基于随机模型的群签名方案具有防陷害性。

证明：在不可陷害性攻击中，敌手 $\mathcal{A}$ 的目标是生成消息 m 的群签名 σ 和有效证据 π，满足：1）π 证明了 σ 是由诚实用户 i 生成的；2）σ 不是由敌手 $\mathcal{A}$ 将 i,m 输入签名预言机生成的。区分两种不同类型的攻击。1）$\pi=(k,\tilde{\sigma},\Pi)$ 中的 k 与用户加入时所签的 k 不同，由加入时所用签名方案的不可伪造性可得这种情况是不可能的。2）$\pi=(k,\tilde{\sigma},\Pi)$ 中的 k 与用户加入所签的 k 相同。下面我们讨论第二种攻击。

构造一个可以解决 SDLP 问题 $(g_1,\tilde{g}_1)=(g^a,\tilde{g}^a)$ 的仿真器 $\mathcal{S}$。注意假定 SDLP 问题中的 $g,\tilde{g}$ 与群签名方案中所采用的 $g,\tilde{g}$ 相同。对于基底不同的情况，可以做一个变换，将其转化为相同的情况。证明思路是如果存在一个敌手 $\mathcal{A}$ 能够以不可忽略的优势 $Adv_{\mathcal{S},A}^{cf}(\eta)=\varepsilon$ 攻破不可陷害性，则 $\mathcal{S}$ 可以解决 SDLP 问题。

首先构造一个算法 $\mathcal{A}'$，给定 gpk 和随机串 R 作为 $\mathcal{A}'$ 的输入，$\mathcal{A}'$ 和 $\mathcal{A}$ 执行以 gpk 和随机值 R'（R' 由 R 确定）为输入的交互。$\mathcal{A}'$ 维护指针 ctr、列表 L_1、L_2 和成员状态信息表 $List$。$H_1(R)$、$H_2(\alpha)$ 和 $Revoke(i)$ 同定理 1。

$GJoin_{UD}(i)$：$\mathcal{A}'$ 设置 $upk[i]\leftarrow upk$，作为用户 i 的经过认证的 upk。给定一个诚实用户 $i\in HU$，$\mathcal{A}'$ 通过查找预言机 $H_2(\cdot)$ 列表 L_2 中的 $(\alpha,\hat{\alpha})$ 提取出 α，$\mathcal{A}'$ 选择 $n_i\leftarrow Z_q$，计算 $s\leftarrow g^{n_i}/g^{\alpha}$，$\tilde{r}\leftarrow\tilde{g}_i^{n_i}/\tilde{g}^{\alpha}$，$\tilde{\sigma}\leftarrow\hat{e}(g,\tilde{r})$，将 $(s,\tilde{r},\tilde{\sigma})$ 发送给敌手 $\mathcal{A}$。$\mathcal{A}'$ 仿真 τ 的知识证明：由于零知识证明的完备性，该仿真不会导致概率损失。$\mathcal{A}$ 提供 α 并证明 (a,A,b,B,c,t) 的正确性，由其正确性 $\mathcal{A}'$ 确信 (a,A,b,B,c,t) 是 $\xi=\mu n_i$ 的 CL-C 签名，然后 $\mathcal{A}'$ 将 (ξ,a,A,b,B,c,t) 存储在 $reg[i]$ 中。

$GUpdate$：对于非撤销成员，$\mathcal{A}'$ 将 $(1,u_i,state_o)$ 发送给敌手 $\mathcal{A}$，$\mathcal{A}$ 利用 $gmsk$ 生成更新后的签名钥 (a,A,b,B,c',t')，$\mathcal{A}'$

以此作为更新后的签名钥发送给成员。

GSign：当 $\mathcal{A}$ 给定用户 $i\in HU$ 和消息 m 后，$\mathcal{A}'$ 从 gsk 检索 $gsk[i]$，如果 $gsk[i]=\perp$，$\mathcal{A}'$ 输出 $\perp$。否则，$\mathcal{A}'$ 计算 $\sigma\leftarrow GSign(gsk[i],m)$，将 (m_t,σ) 添加到 sgn 列表中，并返回 σ。算法 $\mathcal{A}$ 输出伪造签名 $(m\_t,\sigma=(a,A,b,B,c,(Cha,Rsp)))$、证据 π 和 u_i（表示敌手 $\mathcal{A}$ 伪造了 u_i 签名）。如果 $GVerify(gpk,m\_t,\sigma)=0, i\notin HU,(m\_t,\sigma=(a,A,b,B,c))\in sgn$ 或者 $GJudge(gpk,m\_t,\sigma,u_i,upk[i],\pi)=0$，其中任何一个不满足，$\mathcal{A}'$ 输出 $(0,\varepsilon)$ 并终止。否则，$\mathcal{A}'$ 查找 $(D||E||F||m\_t,j,Cha)$，其中，

$$D=\frac{\hat{e}(c,\tilde{g})}{\hat{e}(a,\tilde{X})\hat{e}(B_1,\tilde{X})^t},E=\hat{e}(b,\tilde{\omega}_i),F=D^{Cha}E^{Rsp}$$

由于伪造签名能够通过验证，因此上述的元组一定存在。称 $\mathcal{A}$ 的第 j 次 $H_1(\cdot)$ 询问为关键哈希询问，$\mathcal{A}'$ 输出 (j,σ)。以算法 $\mathcal{A}'$ 和由输入生成器 IG 生成的 gpk 作为 forking 引理的输入，如果敌手 $\mathcal{A}$ 赢得不可陷害性的概率为 ε，即 $Adv^{trace}_{\mathcal{A}}(\eta)=\varepsilon$，则 $\mathcal{A}'$ 输出 (j,σ) 的概率 $acc=\varepsilon$，其中 $j\geqslant 1$。应用 forking 引理可以得到一个算法 $F_{\mathcal{A}'}$，它以 frk 概率输出 $(1,\sigma_1,\sigma_2)$，q_H 为敌手 $\mathcal{A}$ 询问 $H_1(\cdot)$ 的最大次数。

$$frk\geqslant\frac{acc^2}{q_H}-\frac{1}{q}=\frac{\varepsilon^2}{q_H}-\frac{1}{q}$$

从算法 $F_{A'}$ 的输出中，$\mathcal{S}$ 可以得到两个签名 $\sigma_1=(a',A',b',B',c',(Cha',Rsp'))$，$\sigma_2=(a'',A'',b'',B'',c'',(Cha'',Rsp''))$。用追踪性证明中类似的方法可得 $(a',A',b',B',c')=(a'',A'',b'',B'',c'')$，$Cha'\neq Cha''\bmod q$。$\mathcal{S}$ 计算 $\xi=(Rsp''-Rsp')/(Cha'-Cha'')\bmod q$。则 (ξ,t) 满足 $\hat{e}(c',\tilde{g})=\hat{e}(a',\tilde{X})\hat{e}(B_1',\tilde{X})^t\hat{e}(a',\tilde{X})^{\xi}$，由于签名构造方法可得 $\xi_i=\mu n_i$ 成立，其中 n_i 可以查找出，因此 $g_1=g^{\xi/n_i}$ 并且 $\tilde{g}_1=\tilde{g}^{\xi/n_i}$，即 ξ/n_i 为 $SDLP$ 问题的解。所以 $\mathcal{S}$ 得到该解的概率至少为 frk。

5 签名方案的性能分析

5.1 签名长度和计算开销

本节从签名长度、签名与验证计算量 3 个方面进行叙述。EXP_{G1} 和 EXP_{GT} 分别表示群 G_1 和 G_T 中的幂乘运算，P 表示双线性对运算。在本方案中，签名算法输出一个随机化的 CL-C[9] 签名和知识签名 Σ，由 Σ 的生成算法得到

$$D\leftarrow\frac{\hat{e}(\bar{c},\tilde{g})}{\hat{e}(\bar{a},\tilde{x})\hat{e}(\bar{B},\tilde{x})^t}=\left(\frac{\hat{e}(c,\tilde{g})}{\hat{e}(a,\tilde{x})\hat{e}(B,\tilde{x})^t}\right)^{\xi}$$

$$E\leftarrow\hat{e}(\bar{b},\tilde{x})=\hat{e}(b,\tilde{x})^{\xi}$$

则知识签名需要证明 $D=E^{\xi}$。签名者随机选择 $rnd\in Z_q$，计算 $Comm\leftarrow Er^{rnd}$，$Cha\leftarrow H(\emptyset||D||Comm||m\_t)$，$Rsp\leftarrow rnd-Cha\cdot\xi(\bmod q)$，$\Sigma$ 为 (Cha,Rsp)，验证者检验 $Cha=H(\emptyset||D||E^{Rsp}D^{Cha}||m\_t)$。因此一个群签名由 5 个群 G_1 中的元素和 2 个 Z_q 中的元素组成。取 $q=170$bit，群 G_1 中的元素为 171bit，即签名长度为 1195bits。

下面考虑签名代价。由于 $\hat{e}(c,\tilde{g})$，$\hat{e}(a,\tilde{X})$，$\hat{e}(B,\tilde{X})^t$，$\hat{e}(B,\tilde{X})^r$，$\hat{e}(b,\tilde{X})$ 可预计算，因此签名代价为 $7EXP_{G1}+2EXP_{GT}$。通过优化哈希函数的计算来进一步缩短签名计算开销，用 $\bar{a},\bar{b},\overline{A_1},\overline{B_1},\bar{c}$ 来代替 D，即 $Cha\leftarrow H(\emptyset||\bar{a}||\bar{b}||\bar{A}||\bar{B}||\bar{c}||Comm||m\_t)$，签名者不需要计算 D，因此签名生成代价为 $5EXP_{G1}+1EXP_{GT}$。

最后考虑验证代价。验证者要验证 $\hat{e}(\bar{a},\tilde{Y})=\hat{e}(\bar{b},\tilde{g})$，$\hat{e}(\bar{A},\tilde{Y})=\hat{e}(\bar{B},\tilde{g})$，$\hat{e}(\bar{A},\tilde{g})=\hat{e}(\bar{a},\tilde{Z})$ 成立，需要 $5P$，还需计算 $\hat{e}(\bar{c}^{Cha},\tilde{g})/\hat{e}(\bar{a}^{Cha}\overline{B_1}^{rCha}\bar{b}^{-Rsp},\tilde{x})$ 来验证知识签名，需要 $1P+2EXP_{G1}$，验证总代价为 $6P+2EXP_{G1}$。

5.2 与现有撤销方案的性能比较

本小节将现有两种典型的可撤销方案和所提方案中的具体性能指标在表 1 中分别列出，其中 R 表示已撤销成员的个数。与其他撤销方案相比，本方案的签名、验证和签名钥更新均与 R 无关。虽然签名计算量比 BS 多 3 EXP_{G1}，但验证计算量比 BS 少得多，而且群成员无需下载成员撤销列表，签名长度依然很短，仅为 1195bits。

表 1 本文方案与其他撤销方案比较

方案	签名长度	签名过程计算量	验证过程计算量
BS[5]	$5G_1+2Z_q$ (1195bits)	$2EXP_{G1}+1P$	$4EXP_{GT}+(2+R)P$
CKS[10]	$7G_1+4Z_q$ (1877bits)	$19EXP_{G1}+12EXP_{GT}+9P$	$12EXP_{G1}+16EXP_{GT}+15P$
本方案	$5G_1+2Z_q$ (1195bits)	$5EXP_{G1}+1EXP_{GT}$	$2EXP_{G1}+6P$

结束语 本文提出了一种新型的撤销成员的无加密短群签名方案，其在 XDDH、LRSW 和 SDLP 假设下，满足 IND-CCA2 匿名性、可追踪性和防陷害性。与其他撤销方案相比，本方案的签名、验证和签名钥更新均与 R 无关，其极大地减少了群成员的计算开销，签名长度依然很短，仅为 1195bits。因此，本方案特别适合于计算和存储能力差的轻量级移动设备，可应用于 IEEE802.1x 等移动环境。本文方案可在满足上述假设下的任何代数群上实现，限于篇幅在此不做详细说明。本方案中打开算法效率较低，但这并不影响系统的整体效率，打开签名是一种异常情况，很少使用。此外，本方案不具有前向匿名性，下一步工作是加强其安全性。

参 考 文 献

[1] Chaum D, Heyst E. Group signatures [C]// Proc of EUROCRYPT 91. New York: Springer-Verlag, 1991: 257-265

[2] Bellare M, Micciancio D, Warinschi B. Foundations of group signatures: formal definitions, simplified requirements, and a construction based on general assumptions [C]// Proc of EUROCRYPT 2003. Berlin: Springer-Verlag, 2003: 614-629

[3] Bellare M, Shi H, Zang C. Foundations of group signatures: The case of dynamic groups[C]// Proc of the Cryptographers. Berlin: Springer-Verlag, 2005: 136-153

[4] Ateniese G, Tsudik G. Some open issues and new directions in group signature schemes[C]// Financial Cryptography(FC'99). Berlin: Springer-Verlag, 1999: 196-211

[5] Bresson E, Stern J. Efficient revocation in group signatures[C]// Proc of PKC01. Berlin: Springer, 2001: 190-206

[6] Camenisch J, Lysyanskaya A. Dynamic accumulators and application to efficient revocation for anonymous credentials[C]// Proc of crypto2002. Berlin: Springer-Verlag, 2002: 61-76

(下转第 66 页)

下面再用反证法证明。假设敌手 $\mathcal{A}$ 可以在多项式时间内，以不可忽略的成功概率修改承诺 e 得到 e^*，并正确揭示一个与 m 相关的消息 m^*，则构造敌手 $\mathcal{C}_{\mathcal{A}}$ 和 $\mathcal{C}_{\mathcal{B}}$，分别模拟承诺者和验证者执行本文承诺方案，并以 $\mathcal{A}$ 为子程序求解 aprCVP。假设挑战者 $\mathcal{C}_{\mathcal{B}}$ 从挑战预言机 $\mathcal{O}$ 处得到一个挑战实例 $\tilde{c}$，需要向 $\mathcal{O}$ 返回在格中靠近向量 $\tilde{c}$ 的向量 $\tilde{x}$，于是进行下面的游戏：

1）$\mathcal{C}_{\mathcal{B}}$ 通过秘密渠道将 $\tilde{c}$ 交给 $\mathcal{C}_{\mathcal{A}}$；

2）$\mathcal{C}_{\mathcal{A}}$ 开始执行承诺协议，将 $\tilde{c}$ 作为自己对任意消息 m 的承诺发送，且被 $\mathcal{A}$ 得到；

3）$\mathcal{A}$ 从 $\tilde{c}$ 中解出 m，并延展为一个与 m 有多项式因子关系的 m^*，生成承诺 e^* 发给 $\mathcal{C}_{\mathcal{B}}$；

4）揭示阶段结束后，$\mathcal{C}_{\mathcal{B}}$ 可得到 $\mathcal{A}$ 给出的 m^*；

5）$\mathcal{C}_{\mathcal{B}}$ 将 m^* 作为自己对挑战 $\tilde{c}$ 的响应 $\tilde{x}$ 发给 $\mathcal{O}$；

6）多次重复上述过程，直到 $\mathcal{O}$ 满意，即 $\mathcal{O}$ 认为向量 $\tilde{x}$ 在格中靠近向量 $\tilde{c}$。

由于 $\mathcal{A}$ 可对消息 m 延展得到 m^*，且成功概率不可忽略，因此如果 m 是最靠近于向量 $[0,\tilde{c}]$ 的向量，那么 m^* 与向量 $[0,\tilde{c}]$ 间的距离将是最近距离的多项式因子。重复多次，直到 $\mathcal{O}$ 认为多项式因子足够小，于是 $\mathcal{C}_{\mathcal{B}}$ 以不可忽略概率在 aprCVP 挑战游戏中胜出。

因此，本文方案是与揭示有关的不可展承诺方案。

证毕。

定理 3 上述承诺可以抵抗信道窃听攻击。

证明：假设有敌手 $\mathcal{A}$ 可以窃听 Alice 与 Bob 的所有通信，于是可得知 Alice 发送的密文 e，但他没有 Bob 的私钥，也不能求解 CVP 问题，于是无法从 e 得知明文 m。在揭示阶段，敌手 $\mathcal{A}$ 窃听到随机数 u，但对他获得 m 也毫无帮助，敌手 $\mathcal{A}$ 在整个协议运行过程中得不到关于 m 的任何信息。因此，该文方案可以抵抗信道窃听攻击。

证毕。

定理 4 上述承诺可以抵抗消息重放攻击。

证明：假设敌手 $\mathcal{A}$ 曾经窃听并记录了 Alice 与 Bob 的一次通信过程，在以后某个时间企图冒充 Alice 向 Bob 再次做出承诺。于是他将 m 的密文 e 及求出 m 所需的随机数 u 再次发给 Bob，显然 Bob 可以正确验证并解开这个承诺。但方案中使用的 u 是一次性随机数，因此 Bob 在验证之前就会发现这个 u 已经存在于他的字典中了。因此终止整个协议，敌手 $\mathcal{A}$ 的消息重放攻击失败。

证毕。

定理 5 上述承诺可以抵抗复制承诺攻击。

证明：假设敌手 $\mathcal{A}$ 曾经窃听并记录了 Alice 与 Bob 的一次通信过程，然后企图冒充 Carol 向 Bob 做出承诺，于是他将 m 的密文 e 及求出 m 所需的随机数 u 发给 Bob，他期望 Bob 解承诺后会认为 Carol 与 Alice 的承诺相同。但在 Bob 验证时发现 $H(id_A \| m \| id_B) \neq H(id_C \| m \| id_B)$，于是拒绝。复制承诺攻击失败。

证毕。

结束语 作为基于格的公钥加密系统的代表，NTRU 主要用于公钥加密及数字签名。本文在标准模型下，利用 NTRU 构建了一个非交互承诺方案，其不仅实现了绑定性和隐藏性，也实现了与揭示有关的不可展性，对于敌手的信道窃听攻击、消息重放攻击及复制承诺攻击有免疫能力。本文方案高效快速，可作为零知识证明及多方安全计算的基本模块。

参 考 文 献

[1] Damgard I, Groth J. Non-interactive and reusable non-malleable commitment schemes[C]// Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing (STOC'03). San Diego, California, USA, 2003: 426-437

[2] Fischlin M, Fischlin R. Efficient non-malleable commitment schemes[C]// Proceedings of Advances in Cryptology CRYPTO. LNCS, vol. 1880. New York: Springer Verlag, 2000: 413-431

[3] 唐春明，裴定一，姚正安. 基于单向函数的完全隐藏承诺方案的构造及应用 [J]. 应用数学学报，2008，31(4)：663-670

[4] Goldreich O, Goldwasser S, Halevi S. Public-key cryptosystems from lattice reduction problems[C]// Proceedings of Advances in Cryptology CRYPTO'97. LNCS, vol. 1294. Santa Barbara: Springer Verlag, 1997: 112-131

[5] Hoffstein J, Pipher J, Silverman J H. NTRU: A Ring-based Public Key Cryptosystem[C]// Third International Symposiun of Algorithmic Number Theory. 1998, LNCS 1423. Portland, Springer Verlag, 1998: 267-288

[6] Hoffstein J, Graham N H, Pipher J, et al. NTRUSign: Digital signatures using the NTRU lattice[C]// The Cryptographers' Track at the RSA Conference. LNCS 2612. Springer Verlag, 2003: 122-140

[7] 胡予濮. 一个新型的 NTRU 类数字签名方案 [J]. 计算机学报，2008，31(9)：1661-1666

[8] 张文芳，余位驰，何大可，等. 一种基于格理论的数字签名方案 [J]. 计算机科学，2006，33(3)：93-96

[9] 蔡庆玲，詹宜巨，余松森，等. 基于 NTRU 公钥密码系统的 RFID 通信安全协议的研究 [J]. 中山大学学报：自然科学版，2009，48(5)：6-11

[10] 张文芳，何大可，缪祥华，等. 基于 NTRU 公钥密码体制的无线局域网安全方案 [J]. 计算机科学，2006，33(1)：111-113

[11] 步山岳，王汝传. 一种可验证和高效的多秘密共享门限方案 [J]. 计算机科学，2011，38(1)：100-103

(上接第 45 页)

[7] Camenisch J, Kohlweiss M, Soriente C. Solving revocation with efficient update of anonymous credentials[C]// SCN2010. Berlin: Springer-Verlag, 2010: 454-471

[8] Bichsel P, Camenish J, Neven G, et al. Get shorty via group signatures without encryption[C]// Proc. of EUROCRYPT 2010. Berlin: Springer-Verlag, 2010: 381-398

[9] Camenisch J, Lysyanskaya A. Signature scheme and anonymous credentials from bilinear maps[C]// CRYPTO 2004. Berlin: Springer-Verlag, 2004: 56-72

[10] Camenisch J, Kohlweiss M, Soriente C. An Accumulator Based on Bilinear Maps and Efficient Revocation for Anonymous Credentials[C]// PKC2009. Berlin: Springer-Verlag, 2009: 481-500