一个标准模型下可证明安全的无证书签名方案

王 旭,钱雪忠
WANG Xu,QIAN Xue-zhong

江南大学 信息工程学院,江苏 无锡 214122
School of Information Technology,Jiangnan University,Wuxi,Jiangsu 214122,China
E-mail:pieces1229@hotmail.com

WANG Xu,QIAN Xue-zhong.Certificateless signature scheme provably secure in standard model.Computer Engineering and Applications,2008,44(11):129-132.

Abstract: Certificateless Public Key Cryptography (CL-PKC) eliminates the inherent key escrow problem of Identity-Based Cryptography (IBC),while preserving the attractive advantage of IBC which dispenses with certificates of traditional Public Key Cryptography(PKC) and their costly management overhead.In this paper,the author propose an efficient concrete certificateless signature scheme that is provably secure against strong adversaries in the standard model,and that can be implemented on wider elliptic curves suitable for pairings.

Key words: certificateless signature;standard model;pairings;ECC

摘 要: 无证书公钥密码体制消除了基于身份公钥密码体制中固有的密钥托管问题,同时还保持了基于身份公钥密码体制的优点,那就是没有传统公钥密码体制中的证书以及证书管理带来的额外开销。提出了一种高效的可使用更多适合配对的椭圆曲线的,同时在标准模型下可证明安全的无证书签名方案。

关键词: 无证书签名;标准模型;配对;椭圆曲线密码体制
文章编号: 1002-8331(2008)11-0129-04 **文献标识码:** A **中图分类号:** TP309

1 引言

Al-Riyami 和 Paterson[1]在 2003 年首次提出的无证书公钥密码体制(CL-PKC),消除了传统公钥密码体制(PKC)中对证书管理的额外开销,同时也解决了基于身份公钥密码体制(IBC)中的密钥托管问题。首个无证书签名方案也在文献[1]中提出,但没有给出安全证明。Huang 等人在文献[3]中提出了针对这个签名方案的一种攻击,并给出一种修正方案,同时在随机预言模型下简单证明了他们方案的安全性。第一个通用的无证书签名的构造方法由 Yum 和 Lee 文献[6]提出,其安全性基于构造的两个基元-传统公钥签名(PKS)和基于身份的签名(IBS)的安全性。Hu 等人[2]指出文献[6]的构造方法不安全,并提出了一种修正方案。Liu 等人在文献[4]中根据文献[5]的 IBS 方案提出了一种在标准模型下可证明安全的无证书签名方案作为构成他们提出的一种新颖的公钥密码体制的一个基元。他们声称这种签名方案比文献[2]的方案有更强的安全性,并且是第一个在标准模型下可证明安全的具体的无证书签名方案。

基于文献[5,4]提出一种更加高效和易于实现的在标准模型下可证明安全的无证书签名方案。本文的方案与文献[4]的签名方案相比的特点有:(1)更短的参数长度;(2)更方便在椭圆曲线密码体制中实现;(3)可以使用更多适合配对的椭圆曲线。

2 相关定义

G_1、G_2 是阶为素数 p 的加群,G_T 是阶同为 p 的乘群,P_1 是 G_1 的一个生成元,P_2 是 G_2 的一个生成元。

定义 1 配对:$e:G_1\times G_2\rightarrow G_T$ 是具有双线性、非退化、e 可有效计算的映射。

为了进行安全证明,要求存在有效的可计算的同构映射 $\psi:G_2\rightarrow G_1$,满足 $\psi(P_2)\rightarrow P_1$。但在实际方案的建立过程中,无需存在这样的同构映射。

定义 2 配对参数生成器:如果随机算法 Ig 能够:(1)接受安全参数 k;(2)运行 k 的多项式时间;(3)输出有效的$<G_1,G_2,G_T,e>$,则称 Ig 为配对参数生成器。

定义 3 co-计算 Diffie-Hellman 问题(co-CDHP):给定$<P_1,P_2,aP_1,aP_2>$,其中 $a、b\in_R Z_p^*$,能够输出 abP_1。

基金项目: 江苏省自然科学基金(the Natural Science Foundation of Jiangsu Province of China under Grant No.BK20003017);教育部<网络教育安全强认证技术>专项基金项目(教技[2001]750 号)。

作者简介: 王旭(1974-),男,助教,主要研究方向:信息安全,计算机网络技术;钱雪忠(1967-),男,副教授,硕士生导师,主要研究方向:数据库技术,Web 服务,网络安全。

收稿日期: 2007-08-07 **修回日期:** 2007-11-14

定义 4 广义 co-计算 Diffie-Hellman 问题（co-GCDHP）：给定$<P_1,P_2,aP_1,aP_2>$，其中 $a、b\in_R Z_p^*$，能够输出$<cp_1,abcP_1>$，$c\in Z_p^*$。

定义 5 广义计算 Diffie-Hellman 问题(GCDHP)：给定$<P_1,aP_1,bP_1>$，其中 $a、b\in_R Z_p^*$，能够输出$<cP_1,abcP_1>$，$c\in Z_p^*$。

定义 6 简化的 Many-Diffie-Hellman 问题（Many-DHP）：给定$<P_1,aP_1,bP_1,cP_1,abP_1,acP_1,bcP_1>$，其中 $a、b、c\in_R Z_p^*$，能够输出 $abcP_1$。

定义 7 简化的广义 Many-Diffie-Hellman 问题（Many-GDHP）：给定$<P_1,P_2,aP_1,bP_2,cP_1,abP_1,bcP_2,bcP_2>$，其中 $a、b、c\in_R Z_p^*$，能够输出 $abcP_1$。

3 方案建立

不同于文献[1]的七个算法，本文具体的签名方案采用精简的五个算法：

Setup：是一个概率多项式时间(*PPT*)算法，以安全参数 k 作为输入，由密钥生成中心(*KGC*)执行以下操作：

运行 $Ig(1^k)$，输出$<G_1,G_2,G_T,e>$；选择两个能抵御碰撞的哈希函数 $H_u:\{0,1\}^*\to\{0,1\}^{n_u}$ 和 $H_m:\{0,1\}^*\to\{0,1\}^{n_m}$，$n_u、n_m\in Z$；让 $\alpha\in_R Z_p^*$，计算 $P_b=\alpha P_2$，让 $P_a\in_R G_1^*$；让 $U'、M'\in_R G_1^*$，$U_i\in_R G_1^*$，$1\leqslant i\leqslant n_u'$，$M_i\in_R G_1^*$，$1\leqslant i\leqslant n_m'$（$n_u'$和 n_m'的定义将在下面描述），让$\hat{U}=\{U_i\}$，$\hat{M},=\{M_i\}$；最后公开参数 $params=<G_1,G_2,G_T,e,P_1,P_2,P_a,P_b,U',\hat{U},M',\hat{M},H_u,H_m>$，其主密钥 $mk=\alpha P_a$ 保密。

Partial-Secret-Key-Extract：是一个 *PPT* 算法，以用户的身份 $ID\in\{0,1\}^*$、参数 *params* 和主密钥 *mk* 作为输入，由 *KGC* 执行以下操作：

计算 $u=H_u(ID)$；将 u 从低位到高位看成是由以 s 位二进制为单位组成的串，即 $u=u[1]\|\cdots\|u[n_u']$，其中 $n_u'=\lceil n_u/s\rceil$，$u[i]\in Z_{2^s}$；计算 $U=U'+\sum_{i=1}^{n_u'}(u[i]U_i)$；让 $r_u\in_R Z_p^*$；最后计算 *ID* 对应的部分私钥 $psk=(psk_1,psk_2)=((mk)+r_uU,r_uP_2)$并输出。

User-Key-Generation：是一个 *PPT* 算法，以 *params* 为输入，由用户执行操作：

选择 $x\in_R Z_p^*$作为用户的私钥 sk；公布他的公钥 $pk=(pk_1,pk_2)=(x\ P_1,x\ P_b)$。

Sign：是一个 *PPT* 算法，以 *params*、用户的私钥 sk、用户的部分私钥 *psk* 和消息 $M\in\{0,1\}^*$ 作为输入，由签名用户执行以下操作：

计算 $m=H_m(M)$；同样将 m 看成 $m=m[1]\|\cdots\|m[n_m']$，其中 $n_m'=\lceil n_m/t\rceil$，$m[i]\in Z_{2^t}$；计算 $M=M'+\sum_{i=1}^{n_m'}(m[i]M_i)$；选择 $r_\pi、r_m\in_R Z_p^*$；用 Partial-Secret-Key-Extract 中的方法计算出 U；最后输出签名 $\sigma=(V,R_\pi,R_m)=((sk)(psk_1)+r_\pi U+r_mM,(sk)(psk_2)+r_\pi P_2,r_mP_2)$。

Verify：是一个确定多项式时间算法，以 *params*、*ID*、公钥 pk、消息 M 和签名 σ 作为输入，由验证签名用户执行以下操作：

检验 $e(pk_1,P_b)\overset{?}{=}e(P_1,pk_2)$并且 $e(V,P_2)\overset{?}{=}e(P_a,pk_2)e(U,R_\pi)e(M,R_\pi)$，如果都相等则签名有效，否则签名无效。

4 安全性分析

显然 Sign 生成的签名可以用 Verify 检验，因此本文的方案是正确的。

安全模型：根据无证书签名的特点，定义两种类型的攻击，简述如下：

TypeI 类型：它不知道 *KGC* 的主密钥 *mk*，但可以替换公钥，它享有查询某 *ID* 的部分私钥 *psk*、私钥 *sk* 和公钥 *pk* 的预言服务。

TypeII 类型：它知道 *KGC* 的 *mk*，因为它拥有 *mk*，它可以构造出任意合法的 *psk*、*sk* 与 *pk*，所以它不允许替换公钥，它享有查询某 *ID* 的 *sk* 和 *pk* 的预言服务。

定理 1（*TypeI* 存在性不可伪造(EU)）。本文的签名方案针对拥有最多 ε 优势和最多运行时间的 *TypeI* 类型的攻击是(ε,t)-存在性不可伪造的，假定 (ε',t')-co-GCDHP 的难解性成立。这里，$\varepsilon'\geqslant\dfrac{\varepsilon}{16(q_e+q_s)(2^{s-1}n_u'+1)q_s(2^{t-1}n_m'+1)}$，$t'=t+O((q_en_u'+q_s(n'_u+n'_m))\rho+(q_k+q_e+q_s)\tau)$，$q_e$ 是 S-Partial-Secret-Key-Extract-Oracle 查询的最大次数，q_s 是 S-Signing-Oracle 查询的最大次数，q_k 是 S-Public-Key-Broadcast-Oracle 和 S-Secret-Key-Extract-Oracle 查询的最大次数之和，ρ 和 τ 分别是 G_1 或 G_2 上的一次点加运算和一次标量乘运算的时间。

证明 假设 *TypeI* 类型的 *PPI* 攻击算法 A 存在，能够建立一个 *PPT* 算法 B，它可以利用 A 最少在 ε'的概率下最多在 t'的时间内解决 co-GCDHP。B 被给定 co-GCDHP 的一个实例$<P_1,aP_1,P_2,bP_2>$，它最终要通过利用 A 来输出$<xP_1,abxP_1>$，B 可以这样做：

Setup：让 $l_u=2(q_e+q_s)$，$l_m=2q_s$。随机选择两个整数 k_u 和 k_m 满足 $0\leqslant k_u\leqslant 2^{s-1}n_u'$和 $0\leqslant k_m\leqslant 2^{s-1}n_m'$，假定 $l_u(2^{s-1}n_u'+1)<p$ 且 $l_m(2^{s-1}n_m'+1)<p$；$x'\in_R Z_{l_u}$，$z'\in_R Z_{l_m}$，$y'、w'\in_R Z_p$；$x_i\in_R Z_{l_u}$，$i=1,\cdots,n_u'$；$z_i\in_R Z_{l_m}$，$i=1,\cdots,n_m'$；$y_i\in_R Z_p$，$i=1,\cdots,n_u'$；$w_i\in_R Z_p$，$i=1,\cdots,n_m'$；$X=\{x_i\}$，$Y=\{x_i\}$，$Y=\{y_i\}$，$Z=\{z^i\}$，$W=\{w_i\}$；定义计算式 $F(U)=x'+\sum_{i=1}^{n_u'}(u[i]x_i)-l_uk_u$，$J(u)=y'+\sum_{i=1}^{n_u'}(u[i]x_i)$，$K(m)=z'+\sum_{i=1}^{n_m'}(m[i]z_i)-l_mk_m$，$L(m)=w'+\sum_{i=1}^{n_m'}(m[i]w_i)$以备后用；让 $P_a=aP_1$，$P_b=bP_2$，$U'=(x'-l_uk_u)P_a+y'P_1$，$U_i=x_iP_a+y_iP_1$，$1\leqslant i\leqslant n_m'$，$M'=(z'-l_mk_m)P_a+w'P_1$，$M_i=z_iP_a+w_iP_1$，$1\leqslant i\leqslant n_u'$，$\hat{U}=\{U_i\}$，$\hat{M}=\{M_i\}$；最后 B 将参数 $params=<G_1,G_2,G_T,e,P_1,P_2,P_a,P_b,U',\hat{U},M',\hat{M},H_u,H_m>$传送给 A。*params* 与 A 真实攻击中获得的参数无从区分。

B 的主密钥 *mk* 的实际值是 $\alpha P_a=bP_a=abP_1$，虽然 B 无法构造此值，但 A 并不知情。通过构造，下面的等式成立：$U=F(u)P_a+J(u)P_1$ 和 $M=K(m)P_a+L(m)P_1$。

B 建立一个四属性列表 $List<ID,psk,sk,pk>$，用于记录 A 所查询的信息。如果一条记录中相关属性下的还没有值，记其值为 $\perp$。B 可以为 A 模拟以下查询：

S-Public-Key-Broadcast-Oracle：B 收到查询 *ID* 后，在 List

中查找,如果存在这样 ID 的记录,将 pk 值返回给 A;如果没有这样的记录,运行 User-Key-Generation 算法得到的公私钥,在 List 中生成一条新记录,将 ID 与公私钥值存入,并返回公钥值给 A。

S-Secret-Key-Extract-Oracle:B 收到 ID 后,在 List 中查找,如果存在这样 ID 的记录,同时 sk 值存在,则将 sk 值返回给 A;如果 sk 不存在,运行 User-Key-Generation 算法得到公私钥,将公私钥值存入相关记录并返回私钥值给 A;如果记录不存在,运行 User-Key-Generation,在 List 中生成一条新记录,将 ID 与公私钥值存入,返回私钥值给 A。

S-Partial-Secret-Key-Extract-Oracle:B 收到 ID 后,在 List 中查找,如果存在这样 ID 的记录,同时 psk 值存在,将其返回给 A,如果 psk 不存在或没有 ID 记录,因为 B 其实没有 mk,也就无法运行 Partial-Secret-Key-Extract 算法,但 B 可以执行以下操作:

计算 $u=Hu(ID)$;如果 $F(u)\neq 0(\mathrm{mod}\, l_u)$,选择 $r_u\in_R Z_p^*$,则 $psk=(psk_1,psk_2)=(-\frac{J(u)}{F(u)}\psi(P_b)+r_uU,-\frac{1}{F(u)}P_b+r_uP_2)$。

部分私钥 psk 有效性证明:$psk_1=-\frac{J(u)}{F(u)}\psi(P_b)+r_uU=-\frac{bJ(u)}{F(u)}P_1+r_uU=-\frac{bJ(u)}{F(u)}P_1+\frac{b}{F(u)}(F(u)P_a+J(u)P_1)-\frac{b}{F(u)}(F(u)P_a+J(u)P_1)+r_uU=bP_a+(r_u-\frac{b}{F(u)})U=(mk)+\tilde{r}_uU$;$psk_2=-\frac{1}{F(u)}P_b+r_uP_2=-\frac{b}{F(u)}P_2+r_uP_2\approx\tilde{r}_uP_2$。因与 Partial-Secret-Key-Extract 生成的 psk 形式一样,A 无从区分,所以 psk 有效。

将 psk 存入 List 中并返回 psk 值给 A;如果 $F(u)=0(\mathrm{mod}\, l_u)$,模拟程序中止。

S-Public-Key-Replace-Oracle:B 收到 ID 与有效公钥 pk' 后,在List 中查找,如果有这样 ID 的记录,就将 pk 值换成 pk',将 sk 的值变为 $\perp$;否则新建一条记录,存入 pk'。

S-Signing-Oracle:B 收到 ID 后,在 List 中检查,如果它的公钥被替换(即存在 pk 值而对应 sk 值为 $\perp$),则仅用公钥 pk 值为 M 生成签名:假定 $K(m)\neq 0(\mathrm{mod}\, l_m)$,计算 $m=H_m(M)$和 M,选择 $r_\pi、r_m\in_R Z_p^*$,则签名 $\sigma=(V,R_\pi,R_m)=((-\frac{L(m)}{K(m)})\psi(pk_2)+r_\pi U+r_mM,r_\pi P_2(-\frac{1}{K(m)})(pk_2)+r_mP_2)$;如果 $K(m)=0(\mathrm{mod}\, l_m)$,则模拟程序中止。

签名有效性证明:$V=(-\frac{L(m)}{K(m)})\psi(pk_2)+r_\pi U+r_mM=(-\frac{bx(m)}{K(m)})P_1+\frac{bx}{K(m)}M-\frac{bx}{K(m)}M+r_\pi U+r_mM=(-\frac{bxL(m)}{K(m)})P_1+\frac{bx}{K(m)}(K(m)P_a+L(m)P_1)+r_\pi U+(r_m-\frac{bx}{K(m)})M=x(bP_a+\tilde{r}_uU)+(r_\pi-x\tilde{r}_u)U+\tilde{r}_mM=(sk)(psk_1)+\tilde{r}_\pi U+\tilde{r}_mM$;$R_\pi=r_\pi P_2=x\tilde{r}_uP_2+(r_\pi-x\tilde{r}_u)P_2=(sk)(psk_2)+\tilde{r}_\pi P_2$;$R_m=(-\frac{1}{K(m)})(pk_2)+r_mP_2=(r_m-\frac{bx}{K(m)})P_2=\tilde{r}_mP_2$。因签名 σ 与 Sign 生成的签名形式一样,A 无法区分,所以签名有效。

签名正确性证明:$e(pk_1,P_b)=e(x\ P_1,P_b)=e(P_1,x\ P_b)=e(P_1,pk_2)$;$e(V,P_2)=e((sk)(psk_1)+\tilde{r}_\pi U+\tilde{r}_mM,P_2)=e(x\ b\ P_a+x\tilde{r}_uU+\tilde{r}_\pi U+\tilde{r}_mM,P_2)=e(x\ b\ P_a,P_2)e((x\tilde{r}_u+\tilde{r}_\pi),U,P_2)e(\tilde{r}_m,M,P_2)=e(P_a,x\ b\ P_2)e(U,r_\pi\ P_2)e(M,\tilde{r}_mP_2)=e(P_a,pk_2)e(U,R_\pi)e(M,R_m)$。签名正确性证明完毕。

如果公钥没有被替换,计算 $u=H_u(ID)$和 U,$m=H_m(M)$和 M,通过 S-Secret-Key-Extract-Oracle 中的方法可以得到 ID 对应的私钥 sk。此时,如果 $F(u)\neq 0(\mathrm{mod}\, l_u)$,通过 S-Partial-Secret-Key-Extract-Oracle 中的方法可以构造有效的部分私钥 psk,有了 sk 与 psk,可以直接用 Sign 算法生成正确有效的签名 σ;如果 $F(u)=0(\mathrm{mod}\, l_u)$,无法构造 psk,但可以采用与公钥被替换时的一样的方法,如果 $K(m)\neq 0(\mathrm{mod}\, l_m)$,仅用 pk 值生成正确有效的签名 σ,如果 $K(m)=0(\mathrm{mod}\, l_u)$,则模拟程序中止。

Challenge:如果 B 一直没有中止,最终 A 将以至少 ε 的概率成功输出$<ID^*,pk^*,M^*,\sigma^*=(V,R_\pi,R_m)>$给 B,σ^* 是 M^* 的能用 pk^* 验证的合法签名。B 计算 $u^*=H_u(ID^*)$和 $m^*=H_m(M^*)$,然后判断 $F(u^*)=0(\mathrm{mod}\ p)$且 $K(m^*)=0(\mathrm{mod}\ p)$是否成立,如果不成立,$B$ 中止;如果成立则计算 $V-\psi(R_\pi)J(u^*)-\psi(R_m)L(m^*)=(sk)(psk_1)+\hat{r}_\pi U+\hat{r}_mM-\psi((sk)(psk_2)+\hat{r}_\pi P_2)J(u^*)-\psi(\hat{r}_mP_2)L(m^*)=abxP_1+x\hat{r}_uJ(u^*)P_1+\hat{r}_\pi J(u^*)P_1+\hat{r}_mL(m^*)P_1-x\hat{r}_uJ(u^*)P_1-\hat{r}_\pi J(u^*)P_1-\hat{r}_mL(m^*)P_1=abxP_1$。写成 $\hat{r}_\pi,\hat{r}_m,\hat{r}_\pi$ 的形式,是为了说明只要 A 能伪造出有效签名且上述判断条件满足,B 总能成功输出$<pk_1^*,abxP_1>=<xP_1,abxP_1>$。

概率分析:在以下三个条件都满足的情况下,B 不会中止:

(1)在 S-Partial-Secret-Key-Extract-Oracle 中,$F(u)\neq 0(\mathrm{mod}\, l_u)$;

(2)在 S-Signing-Oracle 中,如果公钥未被替换,$F(u)\neq 0(\mathrm{mod}\, l_u)$或 $K(m)\neq 0(\mathrm{mod}\, l_m)$,其它情况,$K(m)\neq 0(\mathrm{mod}\, l_m)$;

(3)在 Challenge 中,$F(u^*)=0(\mathrm{mod}\, p)$并且 $K(m^*)=0(\mathrm{mod}\, p)$。

让 $u_1,\cdots,u_{qu}$ 表示在 S-Partial-Secret-Key-Extract-Oracle 和 S-Signing-Oracle 中的 H_u 输出,$m_1,\cdots,m_{qm}$ 表示在 S-Signing-Oracle 中的 H_m 输出。定义事件 $A_i:F(u_i)\neq 0(\mathrm{mod}\, l_u)$ $i=1,\cdots,qu$,事件 $B_j:K(m_j)\neq 0(\mathrm{mod}\ l_m)$ $j=1,\cdots,qm$,事件 $A^*:F(u^*)=0(\mathrm{mod}\ p)$;事件 $B^*:K(m^*)=0(\mathrm{mod}\ p)$;则 $Pr[B\text{ 不会中止}]\geqslant Pr[(\bigwedge_{i=1}^{qu}A_i\wedge A^*)\wedge(\bigwedge_{j=1}^{qm}B_j\wedge B^*)]$,此处$(\bigwedge_{i=1}^{qu}A_i\wedge A^*)$与$(\bigwedge_{j=1}^{qm}B_j\wedge B^*)$相互独立。根据定义,有$-p<F(u)<p$,这意味着如果 $F(u)=0(\mathrm{mod}\ p)$一定有 $F(u)=0(\mathrm{mod}\, l_u)$,相反,如果 $F(u)=0(\mathrm{mod}\, l_u)$,在 $0\leqslant k_u\leqslant 2^{s-1}n_u'$中只有惟一的 k_u 可使得 $F(u)=0(\mathrm{mod}\ p)$,则 $Pr[A^*]=Pr[F(u^*)=0(\mathrm{mod}\ p)\wedge F(u^*)=0(\mathrm{mod}\, l_u)]=Pr[F(u^*)=0(\mathrm{mod}\, l_u)]Pr[F(u^*)=0(\mathrm{mod}\ p)|F(u^*)=0(\mathrm{mod}\, l_u)]=\frac{1}{l_u}\frac{1}{2^{s-1}n_u'+1}$。还有 $Pr[\bigwedge_{i=1}^{qu}A_i|A^*]=1-Pr[\bigvee_{i=1}^{qu}\bar{A}_i|A^*]\geqslant 1-\sum_{i=1}^{qu}Pr[\bar{A}_i|A^*]$,$\bar{A}_i$ 表示事件 $F(u_i)=0(\mathrm{mod}\, l_u)$,因为 $Pr[\bar{A}_i|A^*]=\frac{1}{l_u}$,所以 $Pr[(\bigwedge_{i=1}^{qu}A_i\wedge A^*)]=Pr[A^*]Pr[\bigwedge_{i=1}^{qu}A_i|A^*]=\frac{1}{l_u}\frac{1}{2^{s-1}n_u'+1}(1-\frac{qu}{l_u})\geqslant\frac{1}{l_u}\frac{1}{2^{s-1}n_u'+1}(1-\frac{q_e+q_s}{l_u})=\frac{1}{4(q_e+q_s)(2^{s-1}n_u'+1)}$。同样的方法可以得到 $Pr[(\bigwedge_{j=1}^{qm}B_j\wedge B^*)]\geqslant\frac{1}{4q_s(2^{t-1}n_u'+1)}$,所以 $Pr[B\text{ 不会中止}]\geqslant\frac{1}{16(q_e+q_s)(2^{s-1}n_u'+1)q_s(2^{t-1}n_m'+1)}$,因此 $\varepsilon'\geqslant\frac{\varepsilon}{16(q_e+q_s)(2^{s-1}n_u'+1)q_s(2^{t-1}n_m'+1)}$。

时间复杂度分析:在一次 S-Partial-Secret-Key-Extract-Oracle 中有 $O(n_u')$次点加和 $O(1)$次标量乘,在 S-Signing-Oracle 中有 $O(n_u'+n_m')$次点加和 $O(1)$次标量乘,在 S-Secret-Key-Extract-Oracle 和 S-Public-Key-Broadcast-Oracle 中有 $O(1)$次标量乘,所以 $t'=t+O((q_e n_u'+q_s(n_u'+n_m'))\rho+(q_k+q_e+q_s)\tau)$。定理 1 证明完毕。

定理 2 (*TypeII* 存在性不可伪造(EU))签名方案针对拥有最多 ε 优势和最多 t 运行时间的 *TypeII* 类型的攻击是 (ε,t)-存在性不可伪造的,假定(ε',t')-Many-GDHP 的难解性成立。这里 $\varepsilon' \geqslant \frac{\varepsilon}{16q_s^2(2^{s-1}n_u'+1)(2^{t-1}n_m'+1)q_k}$,$t'=t+O((q_s(n_u'+n_m'))\rho+(q_k+q_s)\tau)$,$q_s$ 是可以向 S-Signing-Oracle 查询的最大次数,q_k 是可以向 S-Public-Key-Broadcast-Oracle 和 S-Secret-Key-Extract-Oracle 查询的最大次数之和,ρ 和 τ 分别是 G_1 或 G_2 上的一次加运算和一次标量乘运算的时间。

证明 方法与定理 1 的证明类似,限于篇幅,只介绍与 *TypeI* 中的不同之处和重要之处。B 可以利用 A 来解决 Many-GDHP,B 被给定一个 Many-GDHP 的实例$<P_1,P_2,aP_1,bP_2,xP_1,abP_1,axP_1,bxP_2>$,其中 $a、b、x\in {}_R Z_p^*$,最终 B 要输出 $abxP_1$。B 可以这样做:

Setup:让 $P_a=aP_1$,$P_b=bP_2$,主密钥 $mk=abP_1$,ID^* 的公钥 $pk^*=(pk_1^*,pk_2^*)=(xP_1,bxP_2)$,其余的设置与 *TypeI* 中一样。最后 B 将参数 $params=<G_1,G_2,G_T,e,P_1,P_2,P_a,P_b,U',\hat{U},M',\hat{M},H_u,H_m>$和主密钥 m_k 传送给 A。

上述各值与 *TypeI* 中其实一样,只不过 mk 与 pk^* 值已给定。

B 有一个与 *TypeI* 中一样的列表 List,首先将 ID^* 和对应的 pk^* 存入。B 为 A 模拟了以下查询:

S-Public-Key-Broadcast-Oracle:与 *TypeI* 过程类似。

S-Secret-Key-Extract-Oracle:与 *TypeI* 类似,但如果查询 ID^* 的私钥,B 中止。

S-Signing-Oracle:与 *TypeI* 类似,如果查询 $ID=ID^*$,因为没有 sk^*,所以只能用 pk^* 模拟签名,签名的有效性和正确性在前面已证明,其余情况完全参照 *TypeI*。

Challenge:如果 B 一直没有中止,最终 A 将以至少 ε 的概率成功输出$<ID^{**},pk^{**},M^{**},\sigma^{**}=(V,R_\pi,R_m)>$给 B,σ^{**} 是 M^{**} 的能用 pk^{**} 验证的合法签名。因为 A 不能换公钥,所以 A 输出的$<ID^{**},pk^{**}>$必在 List 中。B 要首先判断 ID^{**} 是否等于 ID^*,如果成立则用与 *TypeI* 中同样的方法,判断 $F(u^*)=0(\mathrm{mod}p)$且 $K(m^*)=0(\mathrm{mod}p)$是否成立,如果仍然成立则计算和输出 $V-\psi(R_\pi)J(u^*)-\psi(R_m)L(m^*)=abxP_1$;其它情况,$B$ 中止。

概率分析:在以下四个条件都满足的情况下,B 不会中止:

(1)在 S-Secret-Key-Extract-Oracle 中,查询 $ID\neq ID^*$;

(2)在 S-Signing-Oracle 中,如果查询 $ID\neq ID^*$,$F(u)\neq 0(\mathrm{mod}l_u)$或 $K(m)\neq 0(\mathrm{mod}l_m)$,其它情况,$K(m)\neq 0(\mathrm{mod}l_m)$;

(3)在 Challenge 中,$F(u^*)=0(\mathrm{mod}p)$并且 $K(m^*)=0(\mathrm{mod}p)$;

(4)在 Challenge 中,$<ID^{**},pk^{**}>=<ID^*,pk^*>$。

让事件 B_1 表示条件(1)满足,显然 $Pr[B_1]\approx 1$;事件 B_{23} 表示上面的条件(2)和条件(3)都满足,$Pr[B_{23}]$的分析基本等同于 *TypeI*,因此有 $Pr[B_{23}]\geqslant \frac{1}{16q_s(2^{s-1}n_u'+1)q_s(2^{t-1}n_m'+1)}$;事件 B_4 表示条件(4)满足,因为 List 的长度$\leqslant q_k$,所以 $Pr[B_4]\geqslant \frac{1}{q_k}$;而以上事件彼此独立,所以 $Pr[B$ 不会中止$]\geqslant \frac{1}{16q_s^2(2^{s-1}n_u'+1)(2^{t-1}n_m'+1)q_k}$,所以 $\varepsilon'\geqslant \frac{1}{16q_s^2(2^{s-1}n_u'+1)(2^{t-1}n_m'+1)q_k}$。

时间复杂度分析:与类似,只是少了 S-Partial-Secret-Key-Extract-Oracle 查询,因而结论显然成立。定理 2 证明完毕。

5 方案的特点分析

在文献[4]的签名方案中,最坏情况下,*params* 中的$\hat{U}$中群元素的个数是 n_u,$\hat{M}$中群元素的个数是 n_m,*params* 中总的群元素个数为 n_u+n_m+5。本文的方案中,*params* 中的$\hat{U}$中群元素的个数是$\lceil n_u/s\rceil$,$\hat{M}$中群元素的个数是$\lceil n_m/t\rceil$,*params* 中总的群元素个数为$\lceil n_u/s\rceil+\lceil n_m/t\rceil+6$。在资源受限的环境中,这样的性能提升带来的益处是巨大的。不过,参数变短也会带来 ε'比文献[4]的小的副作用,这导致的后果是本文方案的安全强度将会比文献[4]的约有 $s+t-2-\mathrm{lb}(st)$位的降低,但这不影响本文方案的可证明安全性,而且可以通过合适地选择 G_1 与 G_2 来解决。具体可以参见文献[5]。

在本文的方案中,让 $G_2\neq G_1$,所以本文的方案可以使用更多适合配对的椭圆曲线。详细的论述请参见[5]。当然,本文中的 *params* 中要多一个对群 G_2 的描述。本文的方案在 $G_2=G_1$ 时依旧是一个在标准模型下可证明安全的无证书签名方案。此时让 $P_2=P_1$,方案的安全性证明可用同样的方法分别归约到 GCDHP 和 Many-DHP 的难解性假设之上。

6 总结

本文给出了一个高效且易于实现的具体的无证书签名方案。本文的方案参数更短,更适合在椭圆曲线密码体制中实现。另外,该方案可以使用更多适合配对的椭圆曲线,为方案的具体实现带来极大的灵活性。在标准模型而不是随机预言模型下证明了方案的安全性,使得方案的实施更加安全。

参考文献:

[1] Al-Riyami S S,Paterson K.Certificateless public key cryptography [C]//LNCS 2894:ASIACRYPT 2003.Berlin:Springer-Verlag,2003:452-273.

[2] Hu B.Key replacement attack against a generic construction of certificateless signature[C]//LNCS 4058:ACISP'06.Berlin:Springer-Verlag,2006:235-246.

[3] Huang X.On the security of certificateless signature schemes from Aisacrypt 2003[C]//LNCS 2810:CANS 2005.Berlin:Springer-Verlag,2005.

[4] Liu J K.Self generated-certificate public key cryptography and certificateless signature/encryption scheme in the standard model[EB/OL].(2006-10-28)[2006-12-04].http://eprint.iacr.org/2006/373.

[5] Paterson K,Schuldt J.Efficient identity-based signatures secure in the standard mode[EB/OL].(2006-02-28)[2006-04-20].http://eprint.iacr.org/2006/080.

[6] Yum D H,Lee P J.Generic construction of certificateless signature[C]//LNCS 3108:ACISP'04.Berlin:Springer-Verlag,2004.