维普资讯 http://www.cqvip.com

一种基于 RSA 密码体制的门限代理签名方案的设计与分析*)

程 曦 戚文峰

(郑州信息工程大学信息工程学院应用数学系 郑州 450002)

摘 要 提出了一种新的基于 RSA 密码体制的(t,n)门限代理签名方案。此方案不仅满足所有代理要求,即具有保密性、代理保护、不可伪造性、认可性、已知签名者、期限性,并具有签名过程简单、低交互、不需通过安全信道、可以及时发现不合法的代理签名者、最终签名长度短的优点。

关键词 代理签名,门限代理签名,RSA 密码体制

Design and Analysis of a Threshold Proxy Signature Scheme Based on the RSA Cryptosystem

CHENG Xi QI Wen-Feng

(Department of Applied Mathematics, Zhengzhou Information Engineering University, Zhengzhou 450002)

Abstract In this paper, we propose a new (t,n)threshold proxy signature scheme based on RSA cryptosystem, which satisfies all the proxy requirements and has the advantages of simplicity of signing process, low communications and detecting the illegal proxy signers timely. Besides these, our scheme doesn't need a secure channel and the length of our ultimate signature is very short.

Keywords Proxy signature, Threshold proxy signature, RSA cryptosystem

1 引言

1996 年由 M. Mambo 等人[3]首次提出了代理签名方案,它使得代理者在不暴露原始签名者私钥的情况下以原始签名者的身份对消息进行签名。在实际应用中存在这样一种情况:一个原始签名者指派的代理者不止一个,而是由多个代理者所组成的一个代理群,只有群中一定数量的人合作才能产生一个合法的代理签名。因此结合门限思想,Kang Zhang 提出了(t,n)门限代理签名方案[2]。一个(t,n)门限代理签名是指代理密钥由 n 个成员组成的群共享,群中 t 或更多成员合作才能获得代理密钥从而产生代理签名。代理密钥通过原始密钥计算,但从代理密钥无法还原出原始密钥。

一个有效、安全的(t,n)门限代理签名方案必须满足以下要求[1],即具有保密性、代理保护、不可伪造性、认可性、已知签名者、期限性。

目前为止,绝大多数门限代理签名方案都是基于 ElGamal 体制,近期的签名方案[4,9]中主要缺陷[10]在于不能防止代理者勾结以及原始签名者或其它伪造者伪造代理签名,文[12]给出了比较完善的签名方案,但在签名执行过程中需要大量交互式的认证协议和指数乘运算,同时也要求部分代理信息的传输需要使用安全信道,从而增加了通信成本。

2003 年,M. S. Hwang 等人[6]首次提出了基于 RSA 的(t,n)门限代理签名方案,但该方案中的拉格朗日系数在计算上不可行,虽然结合 Shoup 提出的 RSA 门限方案[7]可以解决这个问题,但还是存在不能防止原始签名者伪造代理签名的缺点[5],H. F. Huang 等人[8]在此基础上提出了一种新的基于 RSA 密码系统的门限代理签名。该方案可识别签名者,并且解决了文[6]中体制的缺陷。但该签名方案在代理签名产生的过程中,没有验证代理共享的步骤,这使得在整个签名过程中,无法察觉代理签名者的恶意行为。除此之外,该方案还存在着代理签名长度较长的缺点,因此不利于在对带宽要求较高的网络中传输。

基于目前并没有较完善的基于 RSA 密码体制的门限代理签名方案提出,本文提出了一种新的基于 RSA 的门限代理签名方案,该方案满足所有代理要求[1],并且方案具有低交互,计算复杂度低,签名长度短和及时发现恶意代理签名者的优点。另外,通过结合 M. Bellare 等人提出的 PSS 签名方案[11]中消息处理的思想使得安全性增强。本文第 2 节给出整个方案具体步骤,第 3 节对本方案的安全性和可行性以及与现有签名方案进行分析和比较。

2 基于 RSA 密码体制的(t,n)-门限代理签名

本节给出一种新的基于 RSA 系统的(t,n)门限代理签名方案。在整个方案中我们假设 RSA 模数是安全的。系统参与者有:原始签名者,n 个代理签名者,一个组合器(the combiner)。原始签名者指派这 n 个代理者有权产生代理签名,组合器可以是由原始签名者指派的一个秘书,用来收集共享信息从而合成最终的代理签名。

整个代理签名体制分为三大部分:密钥共享,签名产生,签名验证。

2.1 系统建立

设原始签名者为 P_0,n 个代理签名者为 P_1,P_2,…,P_n,组合器记为 DC (designated combiner)。在系统建立过程中,原始签名者和代理签名者分别产生各自的公钥和私钥。

2.1.1 原始签名者

产生安全 RSA 模:$N_0 = p_0 \cdot q_0$,其中 $p_0 = 2p_0' + 1$,q_0

*)基金项目:国家自然科学基金资助项目(批准号:60373092)。程 曦 研究生,研究方向为密码学和信息安全。戚文峰 教授,博士生导师,研究方向为密码学和信息安全。

维普资讯 http://www.cqvip.com

$=2'q_0+1$ 为大素数，且 p'_0 和 q'_0 都为素数。基于密码体制的安全性，N_0 至少为 1024 比特数。

选择公钥 e_0 满足 $\gcd(e_0, \phi(N_0))=1$，且 e_0 为大于 n 的素数，计算私钥 d_0 满足 $d_0 \cdot e_0=1 \bmod p'_0q'_0$。则原始签名者的公钥为 e_0，私钥为 d_0。原始签名者产生许可证 m_w，包含代理密钥的有效期限，原始签名者以及代理签名者的身份信息等重要的信息。

计算 $K=h_1(m_w, e_0)^{d_0} \bmod N_0$，其中 $h_1:\{0,1\}^* \to Z^*_{N0}$ 为公开的安全 Hash 函数。公布 (m_w, K)。

2.1.2 *代理签名者*

每个 P_i 产生各自 RSA 模 $N_i=p_iq_i$，且满足 $N_i<N_0$，同理产生公钥 e_i，私钥 d_i。所有 n 个代理者共同协商产生公开参数：(1)p 为大素数，$2^{511}<p<2^{512}$；(2)$q \mid p-1$，$2^{159}<q<2^{160}$；(3)$\alpha=h^{(p-1)/q} \bmod p$，$h$ 为$[1, p-1]$中的随机整数，且 $h^{(p-1)/q} \bmod p>1$，则 α 为 Z^*_p 中 q 阶元。

2.2 代理密钥共享

2.2.1 *原始签名者* P_0

计算群代理密钥：$D=d_0^{m_w} \bmod p'_0q'_0$，$E=e_0^{m_w} \bmod p'_0q'_0$

产生随机多项式：

$$f(x)=D+a_1x+\cdots+a_{t-1}x^{t-1} \bmod p'_0q'_0,$$

其中 a_i 为随机数。计算：$\sigma_i=f(i) \bmod p'_0q'_0 (1 \leqslant i \leqslant n)$。

设 $L=\min_{1 \leqslant i \leqslant n}\lceil \log^2 N_i \rceil-1$，将 σ_i 写成每 L 比特级联的形式：$\sigma_{it_i} \| \cdots \| \sigma_{i1}$，其中 σ_{it_i} 为高位，σ_{ij} $(1 \leqslant j \leqslant t_i)$长为 L 比特，高位不足补零，计算 $\sigma_{ij}^{e_{ij}} \bmod N_0$ 并发送给 $P_i(1 \leqslant i \leqslant n, 1 \leqslant j \leqslant t_i)$。

注：将 σ_i 写成按 L 比特级联形式的目的是为了 P_i 能恢复出原 σ_i(见 2.2.2 中(1))。

设 Q 为 Z^*_{N0} 中平方元的集合，随机选取 $v \in Q$，计算 $v_i=v^{\sigma_i} \bmod N_0 (1 \leqslant i \leqslant n)$，有 $v_i \in Q$，则验证密钥为：$VK=v$，$VK_i=v_i$。

2.2.2 *代理签名者* $P_i(1 \leqslant i \leqslant n)$

(1)计算 $(\sigma_{ij}^{e_{ij}})^{d_i} \bmod N_i (1 \leqslant j \leqslant t_i)$，则得 $\sigma_i=\sigma_{it_i} \| \cdots \| \sigma_{i1}$

(2)随机选择：$z_i \in [1, q-1]$，计算并公开：$y_i=\alpha^{z_i} \bmod p$，则每个成员都可以计算群公钥：$y=\sum_{i=1}^{n} y_i \bmod p$

(3)产生随机 $t-1$ 次多项式 $f_i(x)$ 满足 $f_i(0)=z_i \bmod q$，将 $f_i(j)$ 直接用 $P_j(1 \leqslant j \leqslant n, j \neq i)$ 的公钥 e_j 加密后发送给 P_j，并公开 $y_{i,j}=\alpha^{f_i(j)} \bmod p$(一般情况下，$p$，$q$ 都小于 N_i)

到目前为止，可以公开在列表中的验证密钥为：

$(e_i, N_i, m_w, K, E, p, q, \alpha, y, y_i, y_i, j, v, v_i)$

其中 $1 \leqslant i \leqslant n$，$1 \leqslant j \leqslant n$。

2.3 代理签名产生

不失一般性，设由原始签名者指定的代理签名者为 $P_1, \cdots, P_t$，签名消息为 m。整个代理签名的产生分成三部分：

Part Ⅰ

每个 P_i 随机产生 $k_i \in [1, q-1]$，计算 $r_i=\alpha^{k_i} \bmod p$，公布 r_i，并计算：$s_i=(z_i+\sum_{j=t+1}^{n} f_j(i) \cdot L_i) \cdot M-k_i r \bmod q$，其中 $M=h_2(m, K, APSID)$，$L_i=\prod_{\substack{j'=1 \\ j' \neq i}}^{t} \frac{-j'}{i-j}$，$r=\prod_{i=1}^{t} r_i \bmod p$，$h_2:\{0,1\}^* \to Z_{q-1}^*$ 为公开的安全 Hash 函数，$APSID$ 为实际参与本次代理签名的代理者身份。P_i 将部分代理信息 (M, s_i, r_i) 发送给 DC。

DC 收到后验证：

$\alpha^{s_i} \cdot r_i^r=(y_i \cdot (\prod_{j=t+1}^{n} y_{j,i})^{L_i})^M \bmod p$，对所有 $P_i(1 \leqslant i \leqslant t)$都通过后，计算并公布：$s=\sum_{i=1}^{t} S_i \bmod q$

Part Ⅱ

DC 将 s 输入由代理签名者产生的 PSS 签名方案(见附录)，输出 1024 比特的 $m' \in Z^*_{N0}$

Part Ⅲ

令 $\Delta=n!$，每个代理者 $P_i(1 \leqslant i \leqslant t)$计算：$x_i=m'^{2\Delta s_i} \in Q$，$P_i$ 的共享签名包含 x_i 和一个正确性证明参数(z,c)。

下面设 $x=m'^{4\Delta}$，正确性证明即要证明离散对数：

$\log_x x_i^2=\log_v v_i$。

参数(z, c)产生过程如下：

成员 P_i 随机选取 $r' \in \{0, \cdots, 2^{L(N_0)+2L'}-1\}$，($L'$为安全参数 128 比特)，计算：

$v'=v^{r'}$，$x'=x^{r'}$，$c=H(v, x, v_i, x_i^2, v', x')$，$z=\sigma_i c+r'$

P_i 将共享签名(x_i, z, c)及公开的 hash 函数 H(H 函数的输出为 L'比特数)发送给 DC。

DC 验证等式：

$c=H(v, x, v_i, x_i^2, v^z v_i^{-c}, x^z x_i^{-2c})$

当所有的共享签名都通过验证后，DC 计算：$w=x_1^{2\lambda_{0,1}} \cdots x_t^{2\lambda_{0,t}} (=m'^{4\Delta^2 D})$。其中 $\lambda_{j,i}=\Delta \cdot \prod_{\substack{j'=1 \\ j' \neq i}}^{t} \frac{j-j'}{i-j'} \in Z (1 \leqslant i \leqslant t, 0 \leqslant j \leqslant n)$。设 $E'=4\Delta^2$，则有 $w^E=m'^{E'} \bmod N_0$。由 $\gcd(E', E)=1$，则存在 a, b 使得 $E'a+Eb=1$，计算 $S=w^a m'^b \bmod N_0$(此时有：$S^E=m' \bmod N_0$)。

最终，消息 m 的代理签名为$(r, S, APSID)$。

2.4 代理签名验证

验证者验证：$K^{e_0} \bmod N_0=h_1(m_w, e_0)$，计算：$M=h_2(K, m, APSID)$，

$m'=S^E \bmod N_0$，由 m' 恢复出 s(恢复方法见附录 PSS 签名方案的验证)，验证代理签名：$y^M=\alpha^s r^r \bmod p$

定理 1 若签名合法，则有 $y^M=\alpha^s r^r \bmod p$

证明：因为有

$$\begin{aligned}\sum_{i=1}^{t} S_i &= \sum_{i=1}^{t}((z_i+\sum_{j=t+1}^{n} f_j(i) \cdot L_i) \cdot M-k_i r) \\ &= (\sum_{i=1}^{t} z_i+\sum_{i=1}^{t}\sum_{j=t+1}^{n} f_j(i) \cdot L_i) \cdot M-\sum_{i=1}^{t} k_i \cdot r \quad (z_j=\sum_{i=1}^{t} f_j(i) \cdot L_i) \\ &= (\sum_{i=1}^{t} z_i+\sum_{j=t+1}^{n} z_j) \cdot M-\sum_{i=1}^{t} k_i \cdot r \\ &= \sum_{i=1}^{n} z_i \cdot M-\sum_{i=1}^{t} k_i \cdot r\end{aligned}$$

所以有

$$\begin{aligned}\alpha^s r^r &= \alpha^{\sum_{i=1}^{t} s_i} \cdot r^r=(\prod_{i=1}^{n} \alpha^{z_i})^M \cdot (\prod_{i=1}^{t} \alpha^{k_i})^{-r} \cdot r^r \\ &= (\prod_{i=1}^{n} y_i)^M \cdot (\prod_{i=1}^{t} r_i)^{-r} \cdot r^r=y^M r^{-r} r^r=y^M \bmod p\end{aligned}$$

3 分析

在这一节中，我们将分析上述方案的安全性及可行性。

3.1 原始签名者保护

代理者无法以原始者身份产生签名，即无法产生 m^{d_0}。因为，即使 t 或更多代理者勾结起来，他们只能计算出 $\Delta \cdot D$，

维普资讯 http://www.cqvip.com

但由于不知道 N_0 的分解，从而无法计算出 D，更不能计算出 d_0，并且最后签名形式是 $m^D \bmod N_0$，而不是 $m^{d_0} \bmod N_0$，所以，代理者无法产生原始签名。

3.2 代理保护

尽管部分代理签名密钥 σ_i 是由原始签名者产生，但原始签名者不知道代理者的另一代理私钥 z_i，从而无法计算部分代理签名 $s_i(=(z_i+\sum_{j=t+1}^{n} f_j(i)\cdot L_i)\cdot M-k_i r)$。因此，原始签名者不能伪装成代理者产生代理签名。并且，虽然代理者将部分代理信息都发送给合成者，但部分代理签名仍是代理者用自己的私钥产生，合成者并不能从部分代理信息中计算出代理者的私钥，因此也不能伪造代理者产生签名。

3.3 不可伪造性

一个伪造者成功伪造了签名，是指他对一个未被签过名的消息产生了合法签名。

定理 2 假设伪造者至多能腐化 $t-1$ 个代理者，并且在整个签名方案运行前，伪造者已选好 $t-1$ 个腐化代理者。则在随机预言模型中，此方案是安全的。

证明：设 $i_1,\cdots,i_{t-1}$ 为被腐化的代理者。则伪造者可得到 $\sigma_{i_j}=f(i_j)\bmod p'_0q'_0$，并对于满足 $S^e{}_0=m'$ 的已知签名对 (m',S)，他可计算未被腐化的代理者 i 的共享签名：

$$x_i=S^{2(\lambda_{i,0}+\varepsilon_0(\lambda_{i,i_1}\sigma_{i_1}+\cdots+\lambda_{i,i_{t-1}}\sigma_{i_{t-1}}))}(=m'^{2\Delta\sigma_i})$$

但他只能得到已知签名对的共享签名，对未被签名过的消息不能计算出正确的共享签名。下面说明伪造者对一个不正确的共享签名给出一个正确性证明的概率是可忽略的。

若已知 m'，x_i 和 (z,c)，则有 $c=H(v,x,v_i,x_i^2,v',x')$，

其中 $x=m'^{4\Delta}$，$v'=v^z\cdot v_i^{-c}$，$x'=x^z\cdot x_i^{-2c}$。

易见 $x,v_i,x_i^2,v',x'\in Q$，假设 v 是 Q 的生成元，则有：

$x=v^\rho,v_i=v^{\rho i},x_i^2=v^\beta,v'=v^\gamma,x'=v^\delta$

可得：$z-c\sigma_i=\gamma \bmod p'_0q'_0$，$z\alpha-c\beta=\delta \bmod p'_0q'_0$。

在第一个等式两边同乘 α 并与第二个相减得：

$$c(\beta-\sigma_i\alpha)=\alpha\gamma-\delta \bmod p'_0q'_0 \qquad (1)$$

结论：一个共享签名正确当且仅当：$\beta=\sigma_i\alpha \bmod p'_0q'_0$。否则，则对模 p'_0 和 q'_0 都不成立，则(1)唯一确定了 c 模 p'_0（或 q'_0）的值。但在随机模型中，c 值的分布和 Hash 函数的输入值是独立且唯一的。

除此之外，即使伪造者造出 m' 使得 $S^E=m' \bmod N_0$，他还必须产生一个与 m' 对应的 s，使验证时有 $y^M=\alpha^s r^r \bmod p$ 成立，但由于伪造者不知道 PSS 方案中的随机数 l（见附录），因此找到这样的 s 也是困难的。

3.4 认可性

原始签名者和真实的代理签名者都无法否认一个合法的代理签名。首先，一个合法的代理签名 (r,S) 必须由 t 或更多的代理者产生，因为只有代理者 P_i 有私钥 z_i，从而能产生部分代理签名 s_i。另外，最终代理签名的验证密钥为 E，是与原始签名者产生的代理私钥 D 对应的，所以，原始签名者和代理者都无法否认一个合法签名。

3.5 期限性

在方案中，许可证 m_w 由原始签名者签过名，里面包含了代理的有效期限。在验证过程中，验证者可验证许可证是否过期，若过期了，整个代理签名就无效。

3.6 已知签名者

在部分代理签名产生和最终代理签名中，都包含实际参与签名的代理者的身份信息 APSID，验证者可以通过此信息来确定实际签名者的身份。

3.7 签名长度

在文[8]方案中，最终签名包含一个数 R，其比特值范围为 $[1,\prod|N_i|]$，长度较长。而本方案最终传给验证者的签名为 $(r,S,APSID)$，则其长度比特数至多为 $|q|+|N_0|+|APSID|$（符号 $\|$ 表示比特数）。

3.8 可执行性

本签名方案与文[8]比较，其优越性在于在部分代理签名产生的同时附上了部分代理签名的验证，即 DC 在收到部分代理签名的同时可进行部分代理签名的验证，若发现某部分代理签名不合理，可立即公布其代理者的身份并终止签名。

另外与现有的基于离散对数的门限代理签名相比，其优点是可在一般的信道上传输而不需通过安全信道；并且最后代理签名产生过程是非交互的，在执行和验证过程中有较低的指数运算量和计算复杂度。与文[4]中方案的比较见表 1。

表 1

	Hsu 等人提出的方案[4]	本方案
代理密钥共享阶段	所有参与签名者： $n^2T_E(3n^2t-2nt-t-3n^2+3n+1)T_M$ 除此以外，原始签名者： $tT_E+(nt-n+1)T_M+T_h$ 每个代理签名者： $(t+1)T_E+(2t-1)T_M+T_h$	原始签名者： $(2+n)T_E+(nt-1)T_M$ 每个代理签名者： $(2n+2)T_E+(n-1)T_M$
代理签名产生阶段	生成部分代理签名： $T_E+(3t-1)T_M+(t-1)T_I+T_h$ 生成最终代理签名： $(2t^2+t-3)T_E+(5t^2-7t+2)T_M+(t^2-2t+1)T_I+2T_h$	生成部分代理签名： $5T_E+nT_M+T_I+2T_h$ 生成最终代理签名： $(4t+7)T_E+(tn+2t+3)T_M+(t^2+2)T_I+3T_h$
代理签名验证阶段	$4T_E+(t+4)T_M+2T_h$	$5T_E+T_M+4T_h$

参 考 文 献

1 Kim S, Park S, Won D. Proxy signature, revisited. In: Proc. of ICICS'97, LNCS, 1334. 223～232
2 Zhang K. Threshold proxy signature scheme. Information Security Workshop, Japan, 1997. 191～197
3 Mambo M, Usuda K, Okamoto E. Proxy signature: Delegation of the power to sign message. IEICE Trans. Fundamentals, 1996, 9: 1338～1354
4 Hsu C L, Wu T S. Efficient nonrepudiable threshold proxy signature scheme with known signers against the collusion attack. Applied Mathematics and Computation, 2004
5 Wang G, Bao F, Zhou J, Deng R H. Comments on a threshold proxy signature scheme based on the RSA cryptosystem. IEEE Transaction on knowledge and data engineering, 2004, 16(10): 1309～1311
6 Hwang M S, Lu E J L, Lin I C. A practical (t, n) threshold proxy signature scheme based on the RSA cryptosystem. IEEE Transaction on Knowledge and Data Engineering, 2003, 15(6): 1552～1560
7 Shoup V. Practical threshold signature. In: Proc. Advanced in Cryptoltgy- Eurocrypt, 2000. 207～220
8 Huang Hui-Feng, Chang Chin-Chen. An efficient and practical (t,n) threshold proxy signature scheme with known signers. Fundamenta Informatica, 2003, 56(3): 243～253

（下转第 159 页）

维普资讯 http://www.cqvip.com

始位置域为[−5.12,5.12]；

6. 针对 Shaffer's F6 函数，粒子的最大速度为 2.0，粒子的初始位置域为[−100,100]；

7. 逆反比例：指逆向思维粒子数在总群体中占的比例；

8. 结果的得出是针对每个函数每种算法每种参数各运行 50 次，对它们取平均值。

表 2 Maurice Clerc 和 James Kennedy 的最优实验结果

函数名	最优结果
Rosenbrock	39.118488
Rastrigin	46.4689
Shaffer's F6	0.000247

Maurice Clerc 和 James Kennedy 在文[9]中的实验中也用到了上面提到的函数，在他们的实验中，粒子的群体规模是 20。他们得出的最好结果如表 2 所示。

我们选取计算 Rastrigin 最小值的运行结果曲线，作为运算结果示意图的比较(图 4 所示)，其中最左边的是经典方法的 PSO，运算结果为 32.1132，中间的是逆反比例为 10%的 PSORTP，运算结果为 30.8226，最右边的是逆反比例为 30%的 PSORTP，运算结果为 23.1431(图中上面的曲线是群体最优值 G，下面的曲线是粒子的平均适应度值(函数值)。运算参数：粒子数：100；循环次数：2000；初始值 $x_i \in [-5.12, 5.12]$；V_{max}：0.5)。

图 4 运行结果曲线示意

3.3 结果分析

由上面的结果可以看出 PSORTP 算法的性能好于经典的 PSO 算法，同时也明显优于 Maurice Clerc 和 James Kennedy 的最优实验结果。当粒子数较少时(20 个)，PSORTP 与经典方法的 PSO 算法的性能相差不大，尤其是当逆反比例很小时，最优值通常是由具有 50%逆反比例的 PSORTP 获得。当粒子数较大时(100 个)，PSORTP 的优势就较为明显了，尤其是逆反比例为 30%时，对于三个函数的优化都取得了很好的效果，结果要比经典方法的 PSO 算法的结果好许多。

结束语 本文对 PSO 算法进行了改进，提出了一种能够保证全局收敛的 PSO 算法—PSORTP 算法。该算法引入逆向思维粒子，与传统 PSO 算法相比，提高了算法全局寻优的能力；在理论上，我们也对它做了分析，证明它是以概率 1 收敛到全局最优解的。最后通过实验和与其它方法的对比，表明了本方法的有效性。

在逆向思维粒子的选取上，本文采用的是固定比例选取的方式，这样就不能保证在任何情况下都能选择最佳比例，如果能让算法自动确定逆向思维粒子的最佳比例，算法的性能将会变得更好。因此，下一步，我们将会把研究的重点放在对逆向思维粒子比例的自适应确定上，进一步提高算法的性能。

参 考 文 献

1 Bonabeau E, Dorigo M, Theraulaz G. Inspiration for optimization from social insect behaviour[J]. Nature, 2000, 406(6): 39～42

2 Kennedy J, Eberhart R C. Particle swarm optimisation [A]. In: Proc. of the IEEE International Conference on Neural Networks [C]. Piscataway, NJ, USA: IEEE, 1995. 1942～1948

3 李宁，付国江，等. 粒子群优化算法的发展与展望. 武汉理工大学学报，27(2)：26～29

4 柯晶，钱积新，乔谊正. 一种改进粒子群优化算法[J]. 电路与系统学报，2003, 8(5): 87～91

5 李沛君. 逆向思维在企业经营创新中的应用. 重庆商学院学报，2002(1)

6 Cui Zhihua, Zeng Jianchao. A Guaranteed Global Convergence Particle Swarm Optimizer. RSCTC'2004 [C]. Sweden, 2004. 762～767

7 Solis F, Wets R. Minimization by random search techniques. Mathematics of Operations Research, 1998, 6(1): 19～30

8 http://ocw. mit. edu/NR/rdonlyres/Sloan-School-of-Management/15-099Fall2003/594A2FDC-A9B1-4336-AFDC-E2298F3C0DC4/0/ses5_solis_wets. pdf

9 Clerc M, Kennedy J. The Particle Swarm-Explosion, Stability, and Convergence in a Multidimensional Complex Space[J]. IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, 2002, 6(1): 58～73

(上接第 110 页)

9 Tzeng S F, Hwang M S, Yang C Y. An improvement of nonrepudiable threshold proxy signature scheme with known signers. Computer and Security, 2004, 23: 174～178

10 Tan Z W, Liu Z J. Cryptanalysis of threshold proxy signature scheme. MM Research Preprints. Dec. 2004(23): 226～233

11 Bellare M, Rogaway P. The exact security of digital signature—How to sign with RSA and Rabin. In: Proc. Advance in Cryptology- Eurocypt'96, 1996. 399～416

12 卢建朱，陈火炎，沈建军. 高效的不可否认的门限代理签名新方案. 计算机工程与应用，2005，31：14～17

附录

PSS(Probabilistic Signature Scheme)签名方案

PSS$[k_0, k_1]$是由公开参数 k_0, k_1 刻画的签名方案，满足 $k_0 + k_1 \leqslant k - 1$。其中，$k$ 为 N_0 的特长，$k = 1024$，下设 $k_0 = k_1 = 128$。(MD5 的输出为 128 比特)

算法中的 Hash 函数：$h's: \{0,1\}^* \to \{0,1\}^{k_1}$；$g: \{0,1\}^{k_0} \to \{0,1\}^{k-k_1-1}$；$g_1$：输入 $b \in \{0,1\}^{k_0}$，输出 $g(b)$的前 k_0 比特；g_2：输入 $b \in \{0,1\}^{k_0}$，输出 $g(b)$的余下 $k - k_0 - k_1 - 1$ 比特。

签名时：$l \xleftarrow{R} \{0,1\}^{k_0}$；$b \leftarrow h'(s \| l)$；$l^* \leftarrow g_1(b) \oplus l$；$M^* \leftarrow g_2(b) \oplus s$；$m' \leftarrow 0 \| b \| l^* \| M^*$。

验证时：将 m' 写成 $a \| b \| l^* \| M^*$ 的形式；$l \leftarrow l^* \oplus g_1(b)$；$s \leftarrow M^* \oplus g_2(b)$ 判断 $h'(s \| l) = b, a = 0$；若成立，输出 s。

注： 上述 PSS 消息处理方案由 n 个代理者在系统建立时产生，可以作为一个整体用程序编入黑盒子，并植入一个随机数产生器，每签一次名，随机数发生器产生一个随机数 l，输入一个 s，结果输出一个 m'。此方案的安全性是基于 PSA 模的分解和安全 Hash 函数。