ISSN 1000-9825 Journal of Software 软件学报 2000,11(10):1344~1347

防欺诈的二方共享 RSA 密钥*

王宏[1] 肖鸿[2] 肖国镇[1]

[1](西安电子科技大学 ISN 国家重点实验室 西安 710071)

[2](空军工程大学电信工程学院 西安 710077)

E-mail: ispi@xidian.edu.cn

摘要 二方共享 RSA 密钥产生协议是很重要的一个密码协议,在密钥托管及其他许多方面都有重要的应用. Niv Gilboa 提出了一个二方共享 RSA 密钥产生协议,其效率较高,但不能防止任何一方恶意欺骗.基于该协议,给出了一个能防欺诈的二方共享 RSA 密钥产生协议.

关键词 秘密共享,门限密码学,二方共享 RSA 密钥产生协议,非交互不经意传送,零知识证明.

中图法分类号 TP309

本文解决的是二方共享 RSA 密钥的产生问题.我们提出的协议,能使双方得到 RSA 公钥 $N=PQ$ 和 e,而不知道 N 的分解及解密密钥 d,并且能够防止任何一方进行欺骗.

共享密钥产生协议与秘密共享理论[1]及门限密码学[2]密切相关.我们所研究的协议,其首要应用是密钥托管,另外还有其他的许多方面的应用,如 Fiat-Shamir 签名的参数产生等.

分布式产生 RSA 密钥协议是很重要的一个密码协议.在 RSA 门限签名方案中,有 k 个成员,他们以如下方式共享 RSA 密钥:任意 t 个成员能够对一个消息签名,而少于 t 个成员则都不能生成签名.文献[3]提出了一种解决方案,但该方案需要一个 dealer,产生公共模数 N 及私钥 d,并发送给各成员.对 dealer 的攻击成功就导致该体制的完全崩溃.Boneh 和 Franklin 在文献[4]中提出了一个没有 dealer 的 RSA 密钥分布式产生协议.因此,一个攻击者欲伪造签名,必须勾结或破坏足够多的成员.

二方共享 RSA 密钥产生协议是分布式 RSA 密钥产生的重要协议.文献[4]提出的协议至少需要三方:Alice 和 Bob 共享密钥以及一个帮助方 Henry(与 dealer 不同).Henry 在协议完成后仅知道公共模数 N.而在实际应用中,这样的帮助方也是需要避免的.Cocks 在文献[5]中提出了一个二方共享协议,并可推广到任意多方,但是该协议无论在安全性还是在效率上都存在许多缺点.

在 Crypto'99 会议上,Niv Gilboa 提出了一个二方 RSA 密钥产生协议[6],其效率较高,并证明能防止被动攻击,但不能防止任何一方主动攻击或恶意欺骗.我们基于文献[6]中的协议,给出了一个能防止欺骗的二方共享 RSA 密钥产生协议.

1 准备知识

1.1 非交互不经意传送(non-interactive oblivious transfer)协议 NIOT(i,b_0,b_1)[7]

双方事先分布选取素数 L,Z_L^* 上生成子 β(参见第 2 节的协议 Step 2)以及 Z_L^* 上某元 C,使得双方都不知 $\log_\beta^C$[7].Alice 随机选取 $i\in\{0,1\}$ 及 $x_i\in\{0,1,\ldots,L-2\}$,计算 $\beta_i=\beta^{x_i}$,$\beta_{1-i}=C\cdot(\beta^{x_i})^{-1}$.Alice 的公钥为 (β_0,β_1),私钥为 (i,x_i).Bob 等人可以通过 $\beta_0\beta_1\stackrel{?}{=}C$ 来验证 Alice 的公钥.

---

* 本文研究得到国家自然科学基金(No.69673025)资助.作者**王宏**,1972 年生,博士生,助理研究员,主要研究领域为秘密共享,门限密码学,公钥密码学,网络安全.**肖鸿**,1967 年生,讲师,主要研究领域为公钥密码体制的设计与分析,网络安全.**肖国镇**,1934 年生,教授,博士生导师,主要研究领域为信息论,密码学,网络安全.

本文通讯联系人:王宏,西安 710071,西安电子科技大学 ISN 国家重点实验室信息安全保密所

本文 2000-05-31 收到原稿,2000-06-30 收到修改稿

(1) Bob随机选取 $y_0,y_1\in\{0,1,\ldots,L-2\}$,计算 $\gamma_0=\beta_0^{y_0},\gamma_1=\beta_1^{y_1}$.再随机选取 $r_0,r_1\in\{0,1\}^k$(其中 $k=\log L$),使得 $\langle\gamma_0,r_0\rangle=b_0,\langle\gamma_1,r_1\rangle=b_1$(其中 $\langle\cdot,\cdot\rangle$ 为比特串内积 mod 2 运算).Bob发送 $\alpha_0=\beta^{y_0},\alpha_1=\beta^{y_1}$ 和 r_0,r_1 给Alice.

(2) Alice用其私钥计算 $\alpha_i^{P_a}=\gamma_i$,然后计算出 $b_i=\langle\gamma_i,r_i\rangle$.

该协议是非交互式的,Alice与Bob都不容易进行欺诈行为.协议完成后,Alice得不到 b_{1-i},而Bob对 i 也一无所知.

1.2 不经意多项式估值(oblivious polynomial evaluation)协议 OPE(B,α)

Alice持有域 F 的一个元素 α,Bob持有域 F 上的次数为1的多项式 $B(x)$,令 m 和 $d_{Q,x}$ 为如文献[8]中所选取的安全参数.双方公共拥有伪随机数产生器 G.

(1) Bob随机选取二元多项式 $Q(x,y)$,使其对于 y 是一次的,对于 x 是 $d_{Q,x}$ 次的,且 $Q(0,\cdot)=B(\cdot)$.Alice随机选取次数为 $d_{Q,x}$ 的多项式 S,使得 $S(0)=\alpha$.

(2) Alice选择 $2d_{Q,x}+1$ 个相异非零点 $x_j(j=1,2,\ldots,2d_{Q,x}+1)$.对每个点 x_j,随机选取 $m-1$ 个域元 $y_{j,1},\ldots,y_{j,m-1}$.对 $S(x_j),y_{j,1},\ldots,y_{j,m-1}$ 作随机置换,置换后记作 $z_{j,1},\ldots,z_{j,m}$,并发送给Bob.

(3) 对每个 x_j,Bob计算 $Q(x_j,z_{j,i}),i=1,2,\ldots,m$,记 $L=\max\log Q(x_j,z_{j,i})$.随机选取 $\log m$ 对种子 $s_l^b(1\leqslant l\leqslant\log m,b\in\{0,1\})$,对每个种子通过伪随机产生器 G 得到长为 $m\cdot L$ 的比特串 $G(s_l^b)$,且记 $G(s_l^b)$ 的第 i 段为 $G(s_l^b)_i$.计算 $Q_i=Q(x_j,z_{j,i})\oplus(\bigoplus_{l=1}^{\log m}G(s_l^{i_l})_i)$,其中 $i\in\{1,2,\ldots,m\}$,其二进制表示为 $i=(i_{\log m}\ldots i_1)_2$.Bob发送 $Q_1,Q_2,\ldots,Q_m$ 给Alice.

(4) 对每个 x_j,由于(2),Alice知道 $S(x_j)$ 在 $z_{j,1},\ldots,z_{j,m}$ 中的位置,设 $z_{j,k}=S(x_j)$.令 $k=(k_{\log m}\ldots k_1)_2$,执行 NIOT($k_l,s_l^0,s_l^1$),Alice得到 $s_l^{k_l}$,再通过公共伪随机产生器产生 $G(s_l^{k_l}),l=1,2,\ldots,\log m$.Alice计算 $Q_k\oplus(\bigoplus_{l=1}^{\log m}G(s_l^{k_l})_k)$,由于(3)即得到了 $Q(x_j,z_{j,k})$,也即 $Q(x_j,S(x_j))$.

(5) Alice对 $Q(x_j,S(x_j)),j=1,2,\ldots,2d_{Q,x}+1$ 进行拉格朗日插值计算,求出 $B(\alpha)=Q(0,S(0))$.

该协议与文献[6]中的不同,我们使用非交互的协议提高了效率与安全性.可以看出,该协议总的交互次数是较少的,在如此少的交互中进行欺诈是很困难的.该协议执行后,Alice只得到 $B(\alpha)$,而Bob得不到任何新的信息.

1.3 乘性shares秘密转换加性shares协议 MsCAs($a,b;x,y$)

R 为双方事先设定的环,$\rho=\log\#R$.Alice持有碎片 a,Bob持有碎片 b.

(1) Bob随机独立选取 $s_0,s_1,\ldots,s_{\rho-1}\in R$.

(2) 设 $a=(a_{\rho-1},\ldots,a_0)_2$.Alice与Bob执行协议 NIOT($a_i,s_i,2^ib+s_i$),协议执行之后,Alice得到 $2^ia_ib+s_i$,$i=0,1,\ldots,\rho-1$.

(3) Alice计算 $x=\sum_{i=0}^{\rho-1}(2^ia_ib+s_i)$,Bob计算 $y=-\sum_{i=0}^{\rho-1}s_i$.

该协议使得双方秘密地得到加性碎片 $x,y;x+y=ab \bmod R$,且交互次数很少.

2 防欺诈的二方共享RSA密钥产生协议

基于文献[6]的协议,我们给出防止欺诈的二方共享RSA密钥产生协议.

设RSA模数 N 长为 $\sigma=1024$ bits.

Step 1.[候选值选取]与文献[4,6]相同,Alice持有 P_a,Q_a,Bob持有 P_b,Q_b.

Step 2.[对候选值的承诺]与DSS相同,Alice与Bob事先选好适当的安全素数 M,使得 $L=(M-1)/2$ 也为素数,L 至少为 $\sigma-3$bits 且 $L-1$ 至少有一个大素数因子.α 为 Z_M^* 的 L 阶元,β 为 Z_L^* 的生成元.

Step 2.1. Alice计算 $\alpha^{P_a} \bmod M,\alpha^{Q_a} \bmod M$.Bob计算 $\alpha^{P_b} \bmod M,\alpha^{Q_b} \bmod M$.双方交换各自的Hash值,然后发送给对方这些值.

Step 2.2. 用文献[9]中的零知识证明技巧,Alice向Bob证明她知道 P_a,Q_a,Bob向Alice证明他持有

P_b,Q_b.

Step 2.3. 双方各自计算值 $I=\alpha^{P_aP_b} \bmod M$.

Step 2.4. Alice 计算并发送 $I^{Q_a} \bmod M$,且向 Bob 证明正确执行.Bob 计算并发送 $I^{Q_b} \bmod M$,且向Alice 证明正确执行.

Step 2.5. 双方都能计算候选值 $N=(P_a+P_b)(Q_a+Q_b)$的承诺值 $\mathrm{Com}(N)=I^{Q_aQ_b}=\alpha^N \bmod M$.

Step 3. [计算 N]

方法一:环 R 设定为 Z_{2^σ},运算为 $\bmod 2^\sigma$.

Step 3.1. Alice 与 Bob 执行两次 MsCAs 协议,得到加性 shares. Alice 得到 x_1,x_2,Bob 得到 y_1,y_2:

$$x_1+y_1=P_aQ_b \bmod 2^\sigma,\quad x_2+y_2=P_bQ_a \bmod 2^\sigma.$$

Step 3.2. Bob 发送 $y=y_1+y_2+P_bQ_b \bmod 2^\sigma$ 给 Alice,Alice 向 Bob 发送 $x=x_1+x_2+P_aQ_a \bmod 2^\sigma$.

Step 3.3. Alice 计算出 $N=P_aQ_a+y \bmod 2^\sigma$,Bob 计算出 $N=P_bQ_b+x \bmod 2^\sigma$.

Step 3.4. 双方各自利用 Step 2 的承诺对 N 作检验,若不符,则公布对对方的 accusation,停止协议.

方法二:设域 F 为 $GF(L)$.

Step 3.1. Bob 随机选取域元 $r\in F$,并准备域 F 上的两个多项式 $B_1(x)=P_bx+r$,$B_2(x)=Q_bx-r+P_bQ_b$.

Step 3.2. Alice 利用协议 OPE 得到 $B_1(Q_a)$,$B_2(P_a)$,计算出 $N=P_aQ_a+B_1(Q_a)+B_2(P_a)$,并发送给 Bob.

Step 3.3. 双方各自利用 Step 2 的承诺对 N 作检验,若不符,则停止协议.

方法三:设 $p_1,p_2,\ldots,p_j$ 为最小 j 个不同素数,使 $\prod_{i=1}^{j} p_i>2^\sigma$.

Step 3.1. 令 $t-p_i,i\in\{1,2,\ldots,j\}$,Alice 建立 Benaloh 加密系统[10]:表示为 $T_M(M\in Z_t)$,p,q,y 并发送公钥 $y,m=pq$ 给 Bob. Alice 计算 $z_1=y^{P_a}u_1^t \bmod m$,$z_2=y^{Q_a}u_2^t \bmod m$,其中 $u_1,u\in_R Z_m^*$.

Step 3.2. Bob 计算 $z=z_1^{Q_b}z_2^{P_b}y^{P_bQ_b}u_3^t \bmod m$,其中 $u_3\in_R Z_m^*$,将 z 发送给 Alice.

Step 3.3. Alice 解密 z,将结果加上 $P_aQ_a \bmod t$ 就得到了 $N \bmod t$.

Step 3.4. 对 $i=1,2,\ldots,j$,重复上述 3 个步骤. Alice 得到 $N \bmod p_i$,$i=1,2,\ldots,j$,运用中国剩余定理即可计算得到 N,并发送给 Bob.

Step 3.5. 双方各自利用 Step 2 的承诺对 N 作检验,若不符,则停止协议.

Step 4. [素性测试]与文献[11]相同,若测试出 N 不是两个大素数的乘积,则退回 Step 1.

Step 5. [共享私钥 d]设公钥为 e,$\eta=\lceil\log e\rceil$,$k>4e\cdot 2^\sigma$. Alice 持有 $\varphi_a=N-P_a-Q_a+1$,Bob 持有 $\varphi_b=-P_b-Q_b$. φ_a和 φ_b 为 $\varphi(N)$的加性 shares.

Step 5.1. Bob 随机一致选取 $r\in Z_e^*$. 设环 R 为 Z_e,Alice 与 Bob 执行协议 MsCAs(φ_a,r),得到 φ_ar 的加性 shares x,y. Bob 发送 $y+\varphi_br \bmod e$ 给 Alice.

Step 5.2. Alice 计算 $\zeta_a=x+y+\varphi_ar \bmod e$,Bob 计算 $\zeta_b=-r \bmod e$. 于是 Alice 与 Bob 得到 $\zeta=-\varphi(N)^{-1} \bmod e$的乘性 shares.

Step 5.3. Alice 与 Bob 执行协议 MsCAs(ζ_a,ζ_b),得到 ζ 的加性 shares x,y. Bob 随机一致选取 $\psi_b\in Z_e$,发送 $y-\psi_b \bmod e$ 给 Alice. Alice 计算 $\psi_a=x+y-\varphi_a \bmod e$. 于是 ψ_a,ψ_b 为 ζ 的加性 shares.

Step 5.4. Alice 与 Bob 执行协议 MsCAs(ψ_a,ζ_b),MsCAs(ζ_a,ψ_b),得到加性 shares. Alice 得到 x_1,x_2,Bob 得到 y_1,y_2:$x_1+y_1=\psi_a\zeta_b \bmod k$,$x_2+y_2=\psi_b\zeta_a \bmod k$.

Step 5.5. Bob 随机一致选取 $y'<k/2$,发送 $y=y'+y_1-y_2+\psi_b\zeta_b \bmod k$ 给 Alice. Alice 计算出 $\Omega_a=\psi_a\zeta_a+x_1+x_2+y \bmod k$,Bob 计算出 $\Omega_b=-y' \bmod k$.

Step 5.6. Alice 计算 $d_a=\lceil\Omega_a/e\rceil$,Bob 计算 $d_b=\lfloor(\Omega_b+1)/e\rfloor$:$d_a+d_b\equiv d \bmod \varphi(N)$.

Step 5.7. Alice 随机选取 $M\in Z_N$,$M\neq 0,1$,发送$(M^{ed_a} \bmod N,M)$给 Bob;Bob 随机选取 $\widetilde{M}\in Z_N$,$\widetilde{M}\neq 0,1$,发送$(\widetilde{M}^{ed_b} \bmod N,\widetilde{M})$给 Alice. Alice 验证 $\widetilde{M}\stackrel{?}{\equiv}\widetilde{M}^{ed_a}(\widetilde{M}^{ed_b}) \bmod N$,Bob 验证 $M\stackrel{?}{\equiv}$

$M^{ed_b}(M^{ed_a}) \bmod N$. 若不符,则公布对对方的 accusation,停止协议.

3 性能及安全性分析

我们就所提出的二方共享 RSA 密钥产生协议作简要的性能及安全性分析. 与 N. Gilboa 的协议相比较[6],我们的协议主要增加了对 N 的承诺以及对 d 的验证,但是对协议中使用较多的 OT 协议,我们采用了 NIOT 协议,因此,该协议并不会比文献[6]的效率低. 同时,采用 NIOT 协议,极大地减少了交互次数,并且对 N 的承诺以及对 d 的验证使得 Alice 或 Bob 的欺诈很难得逞而不被发现. 与 N. Gilboa 的协议一样,我们的协议基于 Benaloh加密方案计算 N 效率也是较高的,并且,由于可离线预计算,效率还能再得到提高. 文献[7,9,11]的安全性证明保证了我们的协议是安全的,并能成功地防止被动和主动攻击.

4 结束语

二方共享 RSA 密钥产生协议是很重要的一个密码协议. 我们基于 N. Gilboa 在 Crypto'99 会议上提出的协议[6],给出了防止欺诈的二方共享 RSA 密钥产生协议. 我们的协议的效率与文献[6]的相差不大,安全性更好,并能防上任何一方的欺诈行为. 但是,现在的二方共享 RSA 密钥产生协议的效率仍然不能令人满意,如何提出更高效率的协议需要进一步的研究.

参考文献

1 Shamir A. How to share a secret. Communications of the ACM, 1979,22(11):612~613
2 Desmedt Y. Threshold cryptography. European Transactions on Telecommunications, 1994,5(4):449~457
3 Desantis A, Desmedt Y, Frankel Y *et al*. How to share a function securely. In: Proceedings of the 26th Annual ACM Symposium Theory of Computing (STOC). New York: ACM Press, 1994. 522~533
4 Boneh D, Franklin M. Efficient generation of shared RSA keys. In: Burton S, Kaliski J eds. Proceedings of the Crypto'97. Berlin: Springer-Verlag, 1997. 425~439
5 Cocks C. Split knowledge generation of RSA parameters. In: Darnell M ed. Cryptography and Coding: the 6th IMA International Conference. Berlin: Springer-Verlag, 1997. 89~95
6 Gilboa N. Two party RSA key generation. In: Wiener M ed. Proceedings of the Crypto'99. Berlin: Springer-Verlag, 1999. 116~129
7 Bellare M, Micali S. Non-Interactive oblivious transfer and applications. In: Brassard G ed. Proceedings of the Crypto'89. Berlin: Springer-Verlag, 1989. 547~557
8 Naor M, Pinkas B. Oblivious transfer and polynomial evaluation. In: Proceedings of the 31st STOC. New York: ACM Press, 1999
9 Goldreich O, Micali S, Wigderson A. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. Journal of the ACM, 1991,38(1):691~729
10 Benaloh J. Dense probabilistic encryption. In: Proceedings of the Workshop on Selected Areas of Cryptography. Berlin: Springer-Verlag, 1994. 120~128
11 Blackburn S, Blake-Wilson S, Burmester M. Shared generation of shared RSA keys. Technical Report, Canada: CORR 98-19, Department of C&O, University of Waterloo, 1998

Two-Party Shared RSA Key Against Cheater

WANG Hong[1] XIAO Hong[2] XIAO Guo-zhen[1]

[1](*State Key Laboratory of Integrated Service Networks Xidian University Xi'an* 710071)
[2](*Telecommunication Engineering Institute Air Force Engineering University Xi'an* 710077)

Abstract Generation of two-party shared RSA keys is an important cryptographic protocol. The protocol is applied in key escrow and has a number of other important applications. N. Gilboa presented a protocol of two-party shared RSA key generation. The protocol is efficient, but it cannot preclude either party from active cheating. Based on that protocol, a new protocol of two-party shared RSA key generation is presented against cheater.

Key words Secret sharing, threshold cryptography, two-party shared RSA key generation protocol, non-interactive oblivious transfer, zero knowledge proof.