基于 RSA 的门限密钥托管方案 [1]

孙晓蓉 王育民

(西安电子科技大学 ISN 国家重点实验室 西安 710071)

摘 要 本文借鉴 D.Boneh(1997) 中密钥产生和 Y.Desmedt(1991) 中的密钥分拆思想，提出了一种有 t 个容错能力的 $(t+1,n)$ 门限托管方案，方案可以避免阈下攻击，验证用户的托管密钥正确性，有效地检查出失效的托管代理，并具有密钥备份的能力．方案可用于多种通信方式．

关键词 门限方案，密钥托管，密钥产生，密钥分拆，密钥备份，容错

中图号 TN918.1

1 引 言

密钥托管是目前密码学界的重要研究课题．密钥托管的主要思想是将用户密钥分拆为数个片段，由托管代理分别保管，获得授权的第三方可以利用托管代理保存的片段恢复出用户密钥，解密通信．密钥分拆是密钥托管的关键环节之一，(k,n)-门限分拆法是较常用的一种密钥分拆方法，它的最大优点是恢复密钥时只需要部分 $(\geq k)$ 密钥片段，这样在某几个托管代理失效时密钥恢复工作仍然可以正常进行，这里“失效”是指托管代理不能正常工作或者不愿合作支持托管机制．现有的一些利用 Shamir(k,n)-门限法的分拆密钥进行密钥托管的方案 [1] 存在两个缺点：一是不能确切地知道工作失效的托管代理，密钥恢复时判断重构的密钥是否正确需要试凑解密，这样会影响实时性；二是为了防止托管代理合作非法恢复密钥而必须保证各个托管代理相互独立时，很难验证各托管代理保存的密钥片段的正确性，而使用户有可能逃避托管．

RSA 体制是目前使用的最为广泛的一种公钥密码算法，密码学家也提出了一些将该公钥算法用于密钥托管的建议，但这些建议均存在有一定的缺陷．如 Micali 在文献 [2] 中提出的公正的 RSA 密钥托管方案，密钥由用户单独生成，容易受到“阈下攻击 (subliminal attack)” [3]；Yaksha 方案 [4] 中由第三方将 RSA 密钥分拆为两个不能互相推算的乘积因子由用户和托管代理分别保存，因为很难保证第三方的绝对可信，而使方案存在很大的安全隐患．

本文提出的基于 RSA 体制的门限密钥托管方案，方案具有以下特点：(1) 密钥由托管机构和用户共同产生，可以有效地防止“阈下攻击”；(2) 可以验证用户托管密钥片段的正确性；(3) 密钥恢复时可以检查出失效的托管代理；(4) 支持实时通信解密．

2 密钥托管方案

方案涉及三个实体：用户、托管机构 (包括相互独立的密钥托管中心 KEC 和托管代理 EA) 和政府监听机构 LA ．用户和托管机构之间、监听机构和托管机构之间分别建立有安全信道．

2.1 密钥产生 防止“阈下攻击”可以采用由托管机构为用户选择密钥或由托管机构和用户共同产生密钥．借鉴文献 [5] 的密钥产生思想，方案中由用户和 KEC 共同产生密钥．

首先给出下面的引理及其简要证明．

2.1.1 引理

(1) 通信双方 A 和 B 分别选择整数 (p_A,q_A) 和 (p_B,q_B)，满足 $p_A = q_A = 3 \bmod 4$，$p_B = q_B = 0 \bmod 4$ ．设通信双方均可以获知 N，$N=(p_A+q_A)(p_B+q_B)$ ．

(2) 通信双方 A 和 B 商定满足 $(g/N)=1$ 的随机数 $g \in Z_N^*$，其中 $(\cdot)$ 表示 Jacobi 符号．

[1] 1997-12-16 收到，1999-02-22 定稿
国家自然科学基金和国家相关密码基金资助课题

(3) 用户 A 计算 $v_A = g^{(N-p_A-q_A+1)/4} \mathrm{mod} N$，用户 B 计算 $v_B = g^{(p_B+q_B)/4} \mathrm{mod} N$，双方交换计算结果，并验证：$v_A = \pm v_B \mathrm{mod}\ N$ 是否成立。如果成立，则用户 A 和 B 认为 N 是两个素数 p、q 的乘积。

证明 验证 $v_A = \pm v_B$ 相当于检验 $g^{(N-p-q+1)/4} = \pm 1 \mathrm{mod} N$ 成立。如果 p 和 q 是素数，则 $(g/N) = 1$，即 $(g/p) = (g/q)$。又因 $(p-1)/2$ 和 $(q-1)/2$ 是奇数，所以

$$g^{\phi(N)/4} = (g^{(p-1)/2})^{(q-1)/2} \equiv \left(\frac{g}{p}\right)^{(q-1)/2} = \left(\frac{g}{p}\right) \mathrm{mod} p,$$

$$g^{\phi(N)/4} = (g^{(q-1)/2})^{(p-1)/2} \equiv \left(\frac{g}{q}\right)^{(p-1)/2} = \left(\frac{g}{q}\right) \mathrm{mod} q,$$

因为 $(g/p) = (g/q)$，所以 $g^{\phi(N)/4} = \pm 1 \ \mathrm{mod} N$。当 p 和 q 为奇数时，$\phi(N) = N - p - q + 1$，所以 $g^{(N-p-q+1)/4} = \pm 1 \ \mathrm{mod}\ N$ 成立。

如果 p 和 q 不是素数，需要分几种情况讨论，可以参见文献 [5]

证毕

2.1.2 密钥产生

(1) 用户 A 和 KEC 分别选择两个互异的随机数 p_A, q_A 和 p_K,q_K，满足 $p_A = q_A = 3 \mathrm{mod} 4$, $p_k = q_K = 0 \ \mathrm{mod}\ 4$。KEC 将 (p_K, q_K) 发送给用户 A，用户 A 将 $z(p_A, q_A)$ 发送给 KEC，$z(\cdot)$ 表示零知识比特交换协议 [6] 证明算法。采用零知识比特交换协议是为了保证 KEC 参与密钥产生的有效性。零知识比特交换协议的构造及其证明过程比较复杂，限于篇幅此处不作详细介绍和讨论。

(2) 用户 A 收到消息后检测 $p = p_A + p_K$, $q = q_A + q_K$, $p' = (p-1)/2$, $q' = (q-1)/2$ 是否均为素数，如不全为素数，向 KEC 发回否定信号，KEC 必须重新选择随机数；如均为素数，令 $N = pq$，选择 $g \in Z_N^*$，满足 $(g/N) = 1$。计算 $v_A = g^{(N-p_A-q_A+1)/4} \ \mathrm{mod}\ N$。将 (N, g, v_A) 回送给 KEC。

(3) 根据零知识比特交换协议的性质，KEC 计算 $z^{-1}(z(p_A, q_A))$，验证用户 A 确实拥有 (p_A, q_A)，$z^{-1}(\cdot)$ 表示零知识比特交换协议验证算法；KEC 计算 $v_K = g^{(p_K+q_K)/4} \mathrm{mod}\ N$，比较 $v_A = \pm v_K \mathrm{mod} N$ 是否成立，如成立，则认为 N 是 KEC 参与构成，向用户 A 发回认可信息。

(4) 令 $\lambda(N) = l\mathrm{cm}, (p-1, q-1) = 2p'q'$，用户 A 选取错乱指数 $e \in Z_{\lambda(N)}^*$，并求出 e 的逆 d，即 $ed \equiv 1 \mathrm{mod} \lambda(N)$。用户 A 以 (e, N) 作为 RSA 公钥，以 d 为私钥。

2.2 密钥分拆 设系统的设计容错能力为 t，即 $n(n \geq 3t+1)$ 个托管代理中的最多有 t 个失效。

用户 A 选择 t 个随机数 $c_k = Z_{\lambda(N)}$, $k = 1, \cdots, t$，构造 t 阶多项式 $f(x) = c_t x^t + c_{t-1} x^{t-1} + \cdots + cx + d - 1$, $f(0) = d - 1$，并按文献 [7] 的密钥分拆方法产生 $n(n \geq 3t+1)$ 个密钥片段：

$$d_i = \frac{f(x_i)}{\prod_{\substack{j=1 \\ j \neq i}}^{n} (x_i - x_j)} \overset{x_i \equiv 2i}{=} \frac{f(2i)}{\prod_{\substack{j=1 \\ j \neq i}}^{n} (2i - 2j)} \ \mathrm{mod}\ p'q', \qquad i = 1, \cdots, n.$$

所有 n 个密钥片段 d_i 的序号 i 构成集合 $\boldsymbol{A}$, $|\boldsymbol{A}| = n$，则任选集合 $\boldsymbol{B} \subset \boldsymbol{A}$, $|\boldsymbol{B}| = t+1$，

可重构多项式：

$$f(x)=\sum_{i\in B} d_i \prod_{\substack{j\notin B\\ j\in A}}(x_i-x_j)\prod_{\substack{j\in B\\ j\neq i}}(x-x_j)$$

$$=\sum_{i\in B} d_i \prod_{\substack{j\notin B\\ j\in A}}(2i-2j)\prod_{\substack{j\in B\\ j\neq i}}(x-2j)\bmod\lambda(N)$$

令 $b_{i,B}=\prod_{\substack{j\notin B\\ j\in A}}(2i-2j)\prod_{\substack{j\in B\\ j\neq i}}(0-2j)$，则 $f(0)=\sum_{i\in B} d_i b_{i,B}=d-1 \bmod\lambda(N)$。

2.3 密钥托管和验证

(1) 用户 A 将 (e,N) 送至 KEC，申请托管业务并获得身份代码 ID_A。

(2) 用户 A 将托管证书和 $(\mathrm{ID}_A,i,d_i,i=1,\cdots,n))$ 送至 n 个托管代理 EA_i, $i=1,\cdots,n$。

(3)EA_i 存储 i, d_i，然后向 KEC 发送 ID_A 及证书。KEC 验证通过后，选择一随机数 $r\in Z^*_{\lambda(N)}$，计算 r^e 并分别送至 EA_i, $i=1,\cdots,n$。

(4)EA_i 计算 $\beta_i=r^{ed_i}\bmod N$, $i=1,\cdots,n$，之后将 $\{\beta_i,i\}$ 回送给 KEC。

(5)KEC 任选 $t+1$ 个 β_i，其序号构成集合 $\boldsymbol{B}$, $|\boldsymbol{B}|=t+1$，验证 $r^e\prod_{i\in B}(\beta_i)^{b_{i,B}}=r \bmod N$ 是否成立。若成立，KEC 向用户发出托管认可证书并将 A 的公钥编入系统的公钥手册。若不成立，则认为 A 托管了无效的密钥，应给予惩罚措施。

2.4 用户加密 / 解密　假设用户 A 欲和用户 B 进行通信。

加密　用户 A 产生一随机数 KS 作为该次通信的会话密钥加密消息 m, $c=f(m,KS)$, f 表示任何一种单钥算法 (如 triple-DES 或 IDEA)；从公钥手册查询到用户 B 的公钥 (N_B,e_B)，计算 EKS=$KS^{e_B}\bmod N_B$，并将密文 c 和法律执行接入字段 Leaf 一起发送给 B。

$$\mathrm{Leaf}=\mathrm{ID}_A,\mathrm{ID}_B,t_A,\mathrm{EKS},(h(\mathrm{ID}_A,\mathrm{ID}_B,t_A,\mathrm{EKS}))^{d_A}$$

其中 t_A 为 A 加盖的时戳，h 为安全单向杂凑函数。

解密　B 收到密文 c 和 Leaf 之后，首先根据 ID_A 查询到 A 的公钥 (N_A,e_A)，验证签名函数 $(h(\mathrm{ID}_A,\mathrm{ID}_B,t_A,\mathrm{EKS})))^{d_A}$ 的正确性，验证通过后，解密 EKS 得到会话密钥 KS，解密密文 c。

2.5 电子监视

(1) 政府监听机构首先提前向法庭申请监听证书，证书上规定监听时间。

(2) 监听机构截获密文 c 和 Leaf，并将 Leaf 和监听证书交给 KEC。KEC 验证 t_A 的有效性，验证签名函数的正确性。验证通过后，将 EKS 和 ID_B 发送给托管代理 EA_i, $i=1,\cdots,n$。

(3)EA_i 计算 $\gamma_i=\mathrm{EKS}^{d_i}\bmod N_B$，并将 $\{\gamma_i,i\}$ 回送给监听机构。

(4) 如果失效托管代理的个数为 t，则共有 $t+1$ 种可能的重构密钥，监听机构无法判断恢复出的密钥的正确性。为了获得正确的密钥，必须判断 γ_i 的真实性。方法如下：

监听机构收到的 n 个 γ_i 进行分组检验，每组 $(n-t)$ 个元素，共分为 C_n^{n-t} 组。

(a) 从某一组中任选 $t+1$ 个 γ_i，其序号构成集合 $\boldsymbol{B}_1$, $|\boldsymbol{B}_1|=t+1$，计算 $\mathrm{EKS}\prod_{i\in B_1}(\gamma_i)^{b_{i,B_1}}=KS_1 \bmod N_B$。

(b) 再任选 $t+1$ 个 γ_i，其序号构成集合 $\boldsymbol{B}_2\neq\boldsymbol{B}_1$, $|\boldsymbol{B}_2|=t+1$，计算 $\mathrm{EKS}\prod_{i\in B_2}(\gamma_i)^{b_{i,B_2}}=KS_2 \bmod N_B$，与 KS_1 比较是否相等。如不相等，终止对该组的检验；若相等，继续对该组的检验。

(c) 一组中这样的集合 B_i 有 C_{n-t}^{t+1} 个，可以只验证其中 $n-2t$ 个．设 $n-t$ 个 γ_i 的序号为 $\{1,2,\cdots,n-t\}$，选取 $B_1=\{1,\cdots,t+1\}$, $B_2=\{2,\cdots,t+2\}$, $\cdots$, $B_{n-2t}=\{n-2t,\cdots,n-t\}$ 可获得与验证全部 C_{n-t}^{t+1} 个 B_i 相同的安全性．若对这 $(n-2t)$ 个集合的计算结果均相等，该结果则为真实的 KS．

(5) 监听机构解密密文 c．

2.6 密钥备份　当用户 A 的密钥受损不能使用时，用户可以向 KEC 发出备份申请，并向 KEC 出示其托管许可证书．KEC 验证证书并批准其申请后，将 ID_A 送至 n 个托管代理，各托管代理向用户回送 d_i, $i=1,\cdots,n$, 用户可恢复出私钥 d_i．

3 安全性分析

安全基础：方案的安全基础是 RSA 中大合数分解的困难性以及门限体制的通过不在任意 $t+1$ 个密钥片段中放入足够多的用于重构密钥的信息实现的无条件安全性．

阈下攻击：用户密钥 N 由用户和 KEC 共同产生，并经 KEC 验证，可以防止阈下攻击．

密钥验证：KEC 对密钥片段的验证避免用户托管无效的密钥片段．2.3 节中，KEC 只对 $(t+1)$ 个 d_i 进行了验证，这样用户欺骗 KEC 的概率为 $P_f=1/C_n^{t+1}$, $(t=1,n=4)$ 时 $P_f=1/6$, $(t=2,n=7)$ 时 $P_f=1/35$, $(t=3,n=10)$ 时 $P_f=1/210$．因此，系统应选择 $t\geq 2$．

Leaf 安全：用户签名函数保证了 Leaf 不能被篡改或伪造．

电子监听：政府监听时只能获得 EKS 的解密片段，而不能获得 d_i，防止了政府滥用．

密钥泄漏：KEC 和托管代理的工作是独立的，它们不会得到足够多的能够重构密钥的信息．$\leq t$ 个托管代理的密钥泄漏也不会导致用户私钥泄漏．

4 结束语

本文提出基于 RSA 的门限密钥托管方案，由用户和托管机构共同产生密钥，防止了阈下攻击；采用 (k,n) 的门限方案进行密钥分拆，在密钥托管时可以验证用户托管的密钥片段的正确性；在密钥恢复时可以判断重构密钥的正确性，检验出失效的托管代理，具有较高的安全性．方案的密钥备份功能可以使用户在密钥无法使用时可以重构密钥．方案可用硬件或软件实现．方案可用于话音、email、传真以及文件加 / 解密等通信方式，可用软件或硬件实现．

参 考 文 献

[1] Denning D E. Description of Key Escrow System. Version of May, 1996.

[2] Micali S. Fair pubic-key cryptosystem. Proceedings of Crypto'92, 1992, 209–221.

[3] Kilian J, Leighton T. Fair cryptosystem, revisited. Proc. of Crypto'95, 1995, 113–137.

[4] Ganesan R. The Yaksha security system. Communication of ACM, 1996, 39(3): 55–60.

[5] Boneh D, Franklin M. Efficient generation of shared RAS key. Proc. of Crypto'97, Springer-Verlag, 1997, 425–439.

[6] Goldwasser S, Micali S, Rockoff C. The knowledge complexity of interactive proof system. Proc. of the 17th ACM Symposium on Theory of Computing, 1985, 291–304.

[7] Desmedt Y, Frankel Y. Shared generation of authentication and signature. Proc. of Crypto'91, Springer-Verlag, 1991: 457–469.

A THRESHOLD KEY ESCROW SYSTEM BASED ON RSA

Sun Xiaorong Wang Yumin

(*National Key Laboratory on ISN of Xidian University, Xi'an* 710071)

Abstract This paper presents a $(t+1,n)$-threshold key escrow system with tolerance of t failure or withholding escrow agents on the basis of key generating of Ref.[1] and key spitting of Ref.[2], which can prevent from subliminal key attack, verify the pieces of the user's private key, effectively discover the failure or corrupted escrow agents and backup key for users. This scheme can used to real-time communication encryption and file encryption.

Key words Threshold system, Key escrow, Key generation, Key splitting, Key backup, Tolerance

孙晓蓉： 女，1972 年生，博士生，研究方向为通信网的安全保密.

王育民： 男，1936 年生，教授，博士生导师，长期从事信息论、密码学以及通信网的安全等方面的研究.