一种软件密钥托管设计方案 [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 近年来密钥托管算法受到了广泛的关注. Clipper 、 Capstone 等硬件芯片均采用保密的加密算法，而遭到公众的不满和怀疑. 1993 年 8 月， NIST 宣布了一项工业合作计划，考虑用开发软件技术实现密钥托管. 本文设计了一种用软件实现的密钥托管方案，采用单钥密码算法加密消息，并利用公钥密码算法、单向杂凑函数算法等实现用户识别和密钥检验.

关键词 软件密钥托管，单钥密码算法，公钥密码算法，单向杂凑函数，法律执行

中图号 TN918

1 引 言

密钥托管算法是目前密码学界的研究热点之一. 密钥托管 (Key Escrowing) 概念的出现是在 1993 年 4 月美国政府公布的联邦加密标准议案中，议案制定了托管加密标准 (EES, Escrow Encryption Standard)，旨在为用户提供更为安全的通信方式，同时保证政府执行机构合法的电子监听 [1] . 实际上，密钥托管最早可追溯到 Shamir 和 Blakley 提出的门限方案 [2] . 他们提出的密钥门限法在一些密钥托管系统中正在运用. 而 Micali 在文献 [3] 中提出的公正公钥密码体制 (Fair Public-Key Cryptosystem) 则提出了对 Diffie-Hellman 方案和 RSA 方案等公钥算法的公正思想. EES 的硬件产品是 Clipper 芯片和 Capstone 芯片，其中采用的加密算法是保密的，而不是像 DES(Data Encryption Standard) 那样是公开的，从而使用户深感怀疑和不满；而且由于这两种芯片都只能保证政府的合法解密，而在用户密钥丢失或损坏时无所作为，加剧了公众的不满情绪，使这两种硬件芯片的推广受到了严重阻碍. 考虑到软件系统的透明性，使用灵活性和低价位、易推广的特性， 1993 年 8 月美国国家标准技术所 (NIST) 宣布了一项工业合作计划，提出了开发软件密钥托管产品的议向. 众多密码学家在设计新的密钥托管方案上作了许多努力，提出了诸多方案，但是其中绝大多数是关于硬件实现的 [4] . 文献 [5] 中提出了两种软件密钥托管方案，文献 [6] 继承了文献 [5] 的思想提出了商业密钥恢复体制. 但是文献 [5] 的两种方案均存在不可忽视的缺陷，如软件易于被盗用，协议中容易泄露用户的托管信息，政府监听没有时间限制以及不能检验通信双方的身份等，影响了方案的安全性和可行性. 文献 [7] 中的方案是对文献 [5] 的改进设计，但也有着几点不安全因素，如用户的公钥、私钥均由托管机构分配，用户没有任何主动性；政府向托管机构出示法律执行接入域 LEAF(Law Enforcement Access Field) 时，有可能更改时戳，窃听监听有效期以外的通信等. 此外，方案的协议设计复杂，许多步骤功能重复，这将使软件产品占用很多的计算空间，而且计算时间长，效率低.

本文第 2 节提出了一个设计简单的软件密钥托管方案，第 3 节对方案进行了安全性分析.

[1] 1997-11-11 收到， 1998-09-10 定稿
国家密码发展基金资助课题

2 软件密钥托管协议设计

为了便于对协议进行描述，首先对后面采用的符号作如下定义： KEC：密钥托管中心 (Key Escrow Center)，可信赖的商业机构. EA1,EA2：用户自己选择的可信赖的密钥托管代理 (Escrow Agents)。LA：法律执行机构 (Law Authority)。$E_x(y)$：采用公钥 x 对消息 y 进行公钥加密。$e_x(y)$：采用单钥 x 对消息 y 进行单钥加密。$\mathrm{MAC}_x(y)$：含有密钥 x 的消息 y 的杂凑函数值 (MAC, Message Authentication Code)。$T(\bullet)$：时戳产生函数。P_M, S_M：用户 M 的公钥和私钥. ID_M：用户 M 的身份号码. SP_M：用户 M 的软件 (SP, Software Program)。KEK：密钥加密密钥 (Key Encryption Key)。EVS：托管验证字串 (Escrow Verification String)。LEAF：法律执行接入域 (LEAF, Law Enforcement Access Field)。$A \to B$： A 向 B 发送消息。

用户注册

(1) 用户 U 加入系统，首先向 KEC 提出申请， KEC 批准后，

KEC→U: N, α, ID_U。

(2) 用户任选 $S_{U1}, S_{U2} \in [1, N-1]$，计算 $P_{U1} = \alpha^{S_{U1}}$, $P_{U2} = \alpha^{S_{U2}} \bmod N$，然后

U→ EA_1: P_{U1}, S_{U1}, ID_U； U→ EA_2: P_{U2}, S_{U2}, ID_U。

(3)EA1, EA2 分别验证 $P_{U1} = \alpha^{S_{U1}}$, $P_{U2} = \alpha^{S_{U2}} \bmod N$ 是否成立，如成立，将 P_{U1}, P_{U2} 交给 KEC 建立公钥数据库。

EA_1 →KEC: P_{U1}, ID_U， EA_2 →KEC: P_{U2}, ID_U， KEC→ U: N, α, ID_U。

购买软件

用户 U 向软件发行商 V(Vender) 出示 ID_U， V 查询公钥数据库，将相应的 ID_U, P_{U1}, P_{U2} 嵌入到软件 SP_U 中，交付给用户 U。V 的这一系列操作应对用户透明。

设通信双方为 A 和 B， A 为发送者， B 为接收者。

加密通信

(1)A →SP_A: ID_A。SP_A 验证 ID_A 是否相互符合，如符合 A 可继续使用软件，否则，拒绝 A 使用软件。

(2)A →SP_A: N, α, S_{A1}, S_{A2}。SP_A 验证 $P_{A1} = \alpha^{S_{A1}}$, $P_{A2} = \alpha^{S_{A2}} \bmod N$ 是否成立，若成立， A 可继续使用软件，否则，拒绝 A 使用软件。

(3) 通信前， A 从公钥数据库查询 B 的身份代码 ID_B 和公钥 P_{B1}, P_{B2}。

A →SP_A:

ID_B, P_{B1}, P_{B2}。

(4)SP_A 根据当前时间 t_A 计算 32-bit 的时戳 $T(t_A) = N_A$。 加盖时戳是为了表示通信发生的时间，以帮助 KEC 确定通信是否在 LA 的监听有效期内。一般法庭给 LA 的监听有效期以 “天” 或 “小时” 为单位 [8]，因此系统可采用 “天” 或 “小时” 作为时戳. 为了保证时钟基本同步，建议采用 UTC(Universal Coordinated Time) 时间。

(5)A 任选 64-bit 的随机数 k。 A →SP_A: k。

(6) A 发送消息 m。

(7) 软件 SP_A 完成以下计算：

$$\mathrm{KEK}=(P_{B1}\cdot P_{B2})^{S_{A1}+S_{A2}}=\alpha^{(S_{B1}+S_{B2})(S_{A1}+S_{A2})}\mathrm{mod}N,\qquad K_B=e_{\mathrm{KEK}}(k),$$

$$H=\mathrm{MAC}_k(\mathrm{ID}_A,\mathrm{ID}_B,N_A),\qquad \mathrm{EVS}=e_k(\mathrm{ID}_A,\mathrm{ID}_B,N_A,H),$$

$$\mathrm{LEAF}=\mathrm{ID}_A,\mathrm{ID}_B,P_{A1},P_{A2},K_B,N_A,\mathrm{EVS},\qquad C=e_k(m).$$

然后向 B 发送 LEAF 和密文 C 。

加密消息的单钥算法可采用了 DES 或 IDEA ，单向杂凑函数可采用 MD5 算法，这主要是考虑到技术出口限制问题.

解密通信

(1)B 收到 LEAF 和密文 C 后，向 SP_B 输入 ID_B, N, α, S_{B1}, S_{B2}, ID_A, P_{A1}, P_{A2}, LEAF, C .

(2)SP_B 首先完成对 B 的检验，如加密通信中的 (1),(2) 两步. 检验通过后， B 可使用软件进行解密操作.

(3)SP_B 首先比较输入的 ID_A, P_{A1}, P_{A2} 与 LEAF 中的 ID_A, P_{A1}, P_{A2} 是否符合，不符合则中断软件的运行.

(4)SP_B 计算 $\mathrm{KEK}'=(P_{A1}\cdot P_{A2})^{S_{B1}+S_{B2}}=\alpha^{(S_{A1}+S_{A2})(S_{B1}+S_{B2})}\mathrm{mod}N$, 解密 K_B 得到 k' .

(5)SP_B 用 k' 解密 EVS ，将解密结果与 LEAF 中的明文 ID_A, ID_B, N_A 相比较，并重新计算 $H'=\mathrm{MAC}_k(\bullet)$ ，与 H 比较. 如全部符合，说明 k' 是正确的；如有一项不符合，中断通信.

(6) 解密 C .

政府电子窃听

(1)LA 向法庭申请监听证书，证书上加盖监听有效期 t_C . 法庭可以对有效期 t_C 进行数字签名，以保证 t_C 不能被伪造或窜改.

(2)LA 截获 LEAF 和密文 C ，并向 KEC 提交监听证书和 LEAF .

(3)KEC 根据 t_C 计算时戳 N_C ，与 LEAF 中的 N_A 比较是否相符，若不相符则拒绝 LA 的请求；若相符， KEC 根据 ID_A 查询公钥数据库得到 P_{A1}, P_{A2} ，验证与 LA 提供的 LEAF 中的 P_{A1}, P_{A2} 是否一致，不一致则拒绝 LA 的请求；若一致，将 ID_B 和监听证书送给 EA1 和 EA2 ； EA1 和 EA2 验证证书，分别向 KEC 返回 S_{B1}, S_{B2} ， KEC 计算 $(P_{A1}\cdot P_{A2})^{S_{B1}+S_{B2}}=\alpha^{(S_{A1}+S_{A2})(S_{B1}+S_{B2})}\mathrm{mod}N$ ，并解密 K_B 项得到 k ，重新计算 $H''=\mathrm{MAC}(\bullet)$ ，解密 EVS ，将解密结果和 H'' 与明文和 H 相比较，如全部符合，将 k 回送给 LA ；如有不符合的情况，则认为 LA 有欺骗行为，拒绝 LA 的请求，并记录在案.

(4)LA 解密密文 C .

3　安全性分析

在密钥托管体制中，最主要的矛盾是个人隐私保护与政府监听之间的利益冲突，用户担心的是政府滥用监听的权利，而政府也必须防止用户设法逃避托管. 本文第 2 节提出的软件密钥托管设计上述两种可能都做了考虑，设计方案具有以下安全特性：

(1) 防止 LA 的滥用 (abuse) 攻击：由于 LEAF 同时包含接收者和发送者的身份代码以及时戳. LA 向托管机构申请会话密钥时，其转发的 LEAF 要经过 KEC 的验证，因而限制 LA 只能在某一个特定的有效期内，监听既定的通信，而不能监听通信者任何时刻的通信或转而去窃听另外的通信.

(2) 防止假冒用户攻击：假设用户 C 盗版了 A 的软件 SP_A，但是由于 C 不了解 A 的私钥 S_{A1}, S_{A2} 和身份代码 ID_A，不能通过 SP_A 的检验，无法使用该软件. 如果 A 企图将其 SP_A 的拷贝出售给其他用户，必须附加上身份代码 ID_A 和 S_{A1}, S_{A2}，否则软件不能启用. 而私钥的泄露将给用户 A 带来不堪设想的后果.

(3) 防止用户逃避托管：方案中，公钥是由用户自己产生的，因而无须证件机构签发公钥证件，简化了系统机构. 如果托管机构和软件发行商是可信赖的，则公钥数据库是由用户的真实的公钥组成的，嵌入用户软件 SP 的公钥也为用户真实的公钥. 如果用户在使用软件时，企图改变密钥逃避托管，则不能通过软件的检验，不能使用.

(4) 会话密钥是在通信的过程中产生的，而不是用户事先商定的，这样会话密钥泄露的机会较少. 在通信过程中，会话密钥 k 由接收者的公钥加密以保证其安全，单向杂凑函数用于验证通信双方的身份以及检验会话密钥的正确性. 当然，如果发送者 A 加密消息 m 的密钥与传递给接收者 B 的密钥不一致，单向杂凑函数是无法检验出来的. 但是，由于方案中没有采用数字签名技术，A 和 B 无法利用阈下信道交换另外的会话密钥，因而 B 也将无法解密密文.

(5) 方案的基础是密钥机构 (包括托管中心 KEC 和托管代理 EA) 以及软件发行商是可以信赖的. 我们知道，软件系统的开发、应用、普及、销售实际上是一种商业运作，在软件密钥托管体制中，密钥托管机构和软件发行商均是商业运作的一部分. 如果一个软件开发公司建立的托管机构和软件发行机构有商业欺诈行为，这是很容易被发现的，比如，一个诚实的用户发现公钥数据库中自己的公钥被窜改，或发现购进的软件不能使用等，那么，这个软件将很快失去客户和市场，公司的可信度也将受到很大的影响，造成的损失是不言而喻的. 因此，我们可以假设密钥机构和软件发行商是可信. 但是，需要指出的是，如果个别 EA 背叛用户，对公司的声誉影响很小，而用户的利益将受到严重损失. (k,n) 门限方案是一种解决该问题有效方法，文献 [2, 3, 9] 对此有较深入的研究.

(6) 对软件密钥托管系统最大的威胁是用户更改软件的原代码，破坏其某些函数功能，如对身份的验证、私钥的验证、甚至 LEAF 的产生流程，而使软件可以被盗版使用或使用户能够逃避密钥托管. 因此，软件中必须设计防窜改的软件保护措施. 这不属于本文的讨论范围，这里不做详细论述. 本文的软件系统是建立在这种软件保护措施之上的.

(7) 公钥数据库的安全：文中的公钥数据库实际上是公开的，允许正常的访问. 但是，为了防止某些恶意用户为了逃避密钥托管更改数据库中的公钥数据，需要对公钥数据库设置安全保护系统，保证数据库的安全.

4 结束语

由于目前硬件实现的密钥托管采用的是保密的加密算法，这引起了公众的不安和疑虑，阻碍了其商业发展. 软件系统的透明性、使用灵活性和低价位，是软件密钥托管有更好的发展前景. 本文设计了一个软件密钥托管方案，采用单钥算法加密消息，以公钥算法和单向杂

凑函数来保证会话密钥的安全，在用户权益保障和政府法律监听之间取得了平衡. 由于方案采用了计算量较大的 Diffle-Hellman 公钥算法，所以为了提高运算速度，可以采用压缩指数算法 [8]、分批 Diffie-Hellman 算法 [10] 以及椭圆曲线算法来实现.

参 考 文 献

[1] Denning D E, Smid M. Key escrowing today. IEEE Communication Magazine, 32(9): 55–68.

[2] Denning D E. Cryptography and Data Security. Addision-Wesley Publishing Company, 1982, 211–218.

[3] Micali S. Fair pubic-key cryptosystem. Proceeding of Crypto'92, Santa Barbara: Aug, 1992, Berlin: Springer-Verlag, 1993: 113–137.

[4] Denning D E. Descriptions of Key Escrow System. Internet Document, Version of May 1, 1996.

[5] Balenson D M, Elison C, Lipner S, Walker S. A New Approach to Software Key Escrow Encryption. Trusted Information System, 3060, Washington RD, Glenwood, Draft of August 15, 1994.

[6] Walker S, Lipner S, Elison C, Balen D. Commercial key recovery. Communication of the ACM, 39(3): 41–47.

[7] Dawson Ed, Jingming He. Another approach to software key escrow encryption. Information security and privacy: First Australasian Conference, ACISP'96, Woltongong, June 1996, Berlin: Springer-Verlag, 1996: 87–95.

[8] Yacobi Y. Exponentially faster with addition chains. Proceeding of Eurocrypt'90, Aarhas, May 1990, Berlin: Springer-Verlag, 1991: 222–229.

[9] Pederson T P. Distributed provens with application undeniable signature. Proceeding Eurocrypto'91, Berlin: Springer-Verlag, 1992: 221–242.

[10] Beller M J, Yacobi Y. Batch Diffie-Hellman key agreement systems and their application to portable communication. Proceeding of Eurocrypt'92, Balatonfired: May 1992, Berlin: Springer-Verlag 1993: 208–220.

DESIGN OF SOFTWARE KEY ESCROW SYSTEM

Sun Xiaorong　　Wang Yumin

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

Abstract Key escrow systems have gained much attention in recent years. A key escrow system can provide cryptographic protection to sensitive data ,while at the same time, allows for the decryption of encrypted message under lawful authorization. Since the public were not satisfied with the classified encryption method used in key escrow system in hardware, such as Clipper and Capstone, NIST of US announced a cooperative program with industry to explore the possibilities of performing key escrow cryptography using software-only techniques. This paper proposes a scheme for supporting the implementation of key escrow systems in software, which employs symmetric encryption algorithm for securing communication, and one-way hash function and asymmetric cryptography for identifying the users and verifying session key.

Key words Software key escrow, Symmetric cryptography, Asymmetric cryptography, One-way hash function, Law enforcement

孙晓蓉：女，1972 年生，博士生，从事密码学专业，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、信道编码、密码学以及通信网的安全等方面的研究.