ISSN1000-9825 Journal of Software 软件学报 2000,11(5):616~619

一种新型的综合型安全系统研究*

蒙杨 刘克龙 卿斯汉

(中国科学院软件研究所 北京 100080)

(中国科学院信息安全工程研究中心 北京 100080)

E-mail: qsiban@yahoo.com

摘要 提出了一种新型的基于ElGamal型算法而非RSA(Rivest r. l., Shamir A., Adleman)算法的综合型Yaksha安全系统.该系统可以用作加密、数字签名、分布式认证、密钥交换.文章当前的热点问题"密钥托管"提出了一种可行的解决方案.

关键词 离散对数,分布式认证,密钥托管,安全系统,证书当局.

中图法分类号 TP309

在1993年4月,克林顿政府宣布了一种新的联邦标准对称加密系统的建议[1].该建议使用防窜改的CLIPPER芯片作为其加密标准的一个实现.由于该建议没有给广大用户带来好处,因此受到广大用户特别是企业界的抵制.在随后的几年中,多种软件托管方案被提了出来,其中的大多数方案都能平衡政府与用户之间的利益关系.文献[2]提出一种Yaksha安全系统,该体系结构是RSA(Rivest r. l., Shamir A., Adleman)公钥密码体制的一个变形.该系统除了能够完成数字签名、密钥交换、分布式认证以外,对当前的热点问题——"密钥托管"提出了一个可行的解决方案.该方案为政府监听提供"后门",同时也能恢复用户密钥.但是,该系统缺少公钥构架(public key infrastructure,简称PKI),使得公钥缺少可信任第三方的公证;同时,该系统的"密钥托管"方案有些部分也不尽人意,因而Yaksha服务器成为本系统最大的安全隐患.一旦Yaksha服务器被攻破,入侵者可以毫无约束地偷听该系统所有用户的谈话,在文件加密服务中,一旦文件服务器被攻破,入侵者可以毫无限制地去解密被加密的用户文件.众所周知,公钥加密有两大主要体系,一种是以素因子分解为基础的以RSA为代表的算法,另一种是以离散对数计算为基础的算法,Elgamal[3]算法是其典型的代表.本文提出了一种以Elgamal算法为基础的安全系统,我们把该系统同样称为Yaksha系统.该系统除具有文献[2]的性能以外,还引入证书当局(CA)机制来有效地管理公钥;引入密钥拆分机制,增加了Yaksha系统的安全;利用该体制的特点改变了其密钥交换的过程.同时,本系统为合法的第三者提供监听手段,也提供了用户或合法的第三者恢复其文件加密密钥的手段.本系统也给出一种新的分布式认证协议.下面就系统安全构架以及用户注册、实时认证、签名、文件加密、密钥托管问题作详细的描述.

1 系统安全构架以及用户注册

本系统除了具有Yaksha服务器以外(以下简称YS),我们还增加了一个证书当局CA.YS随机选择一大素数p,并且$p-1$有大素数因子,选择一本原根$g\in Z_p^*$,公开p,g供本系统的所有用户使用.同时,CA的公钥与公钥加密算法是公开的,即用户在注册之前是知道的.下面,我们用$E_{pk}[D]$表示用公钥pk对数据D加密.用户a注册到该安全系统的具体过程如下:

* 本文研究得到国家自然科学基金(No. 69673016)资助.作者蒙杨,1972年生,博士生,主要研究领域为信息安全理论与技术.刘克龙,1971年生,博士生,主要研究领域为信息安全理论与技术.卿斯汉,1939年生,研究员,博士生导师,主要研究领域为信息安全理论与技术.

本文通讯联系人:卿斯汉,北京 100080,中国科学院软件研究所

本文1999-01-16收到原稿,1999-06-01收到修改稿

·用户 a 从 CA 获得 YS 的公钥证书.

·a 选择一随机数 x_{aa},使得 $x_{aa}\in z_p^*{}_{-1}$,计算,$y_{aa}=g^{x_{aa}}(\mathrm{mod}\ p)$,$x_{aa}$便是用户 a 的私钥.

·a 向 YS 申请托管业务,YS 为用户选择若干个具有一定资格的被委托人,a 将自己的私钥拆分后,将每一个密钥片通过被委托人的公钥加密传送到被委托人.具体的拆分方法可采用 Shamir 于 1979 年提出的基于拉格朗日插值公式的密钥分存思想[4].

·将 a 的用户号 ID_a 和用 YS 的公钥 y_s 加密的 y_{aa}传送至 YS.

$$a\rightarrow YS:ID_a,E_{y_s}[ID_a,T_a,y_{aa}],$$

其中 T_a 是用户 a 加盖的时戳,防止重放攻击.

即 a 选择一随机数 k_0,使得 $k_0\in z_{p-1}^*$,计算

$$y_0=y_s^{k_0},\ c_0=g^{k_0},\ w_0=y_0(ID_a,T_a,y_{aa})(\mathrm{mod}\ p),$$

其中 $y_0(ID_a,T_a,y_{aa})$表示 y_0 乘以(ID_a,T_a,y_{aa}).如果(ID_a,T_a,Y_{aa})的长度大于 p 的长度,则作适当分块即可.

·YS 选择一随机数 x_{ay},使得 $x_{ay}\in z_{p-1}^*$,计算

$$y_a=g^{x_{aa}x_{ay}}=g^{x_a}(\mathrm{mod}\ p),\quad x_a=x_{aa}x_{ay}(\mathrm{mod}\ p-1),$$

使得 $y_a\neq 1$,否则重新选择 x_{ay}.其中 x_{ay}是用户 a 的 YS 私钥,由 YS 保存.y_a 是用户 a 的公钥,由 YS 用 CA 的公钥加密并传送至 CA,由 CA 产生 a 的公钥证书.

·YS 用 a 的公钥 y_a 及 YS 的私钥 x_{ay}加密 T_a 传送至用户 a.

$$YS\rightarrow a:E_{y_a^{x_{aa}}}[T_a],$$

即 YS 选择一随机数 k_1,使得 $k_1\in z_{p-1}^*$,计算

$$y_1=y_a^{k_1},\ c_1=g^{k_1x_{ay}},\ w_1=y_1(T_1)(\mathrm{mod}\ p).$$

YS 将 y_1,w_1 传送至 a,用户 a 从 CA 获得自己的公钥证书,解密收到的消息后,得知已注册成功.

从用户的注册过程来看,用户 a 的私钥 x_{aa}与 x_{ay}是独立的选择,无论是用户 a 还是 YS,只知道 y_a 与两个私钥中的一个以及 g 和 p,从一个推导另一个都是求离散对数问题,同时也等价于 Diffe-Hellman 问题.所以,该系统是安全的.

2 实时认证过程

文献[5]已利用 Yaksha 体制构造了一个比较令人满意的分布式认证协议,同时,文献[6]也已说明了如何把 Yaksha 体制集成到 Kerberos[7]协议中.以下是基于本安全系统的分布式认证协议.

用户 a 欲与 b 通信,执行以下的认证过程:

·如果用户 a 没有 b 的公钥证书,则应首先从 CA 获得.

·用户 a 随机选择一个 a 与 b 的会话密钥 K_{ab},向 YS 明文传送用户 a 的用户号 ID_a,用 a 的公钥 y_a 及私钥 x_{aa}加密传送 ID_a,用户 b 的用户号 ID_b,用 b 的公钥加密传送 a 的请求 R,该请求 R 由 ID_a,ID_b,K_{ab}构成,再对所有的被加密信息加盖时戳 T_a.

$$a\rightarrow YS:ID_a,E_{y_a^{x_{aa}}}[ID_a,ID_b,T_a],E_{y_b}[ID_a,ID_b,T_a,K_{ab}],$$

即 a 选择随机数 k_2,r,使得 $k_2,r\in z_{p-1}^*$,计算

$$y_2=y_a^r,\quad c_2=g^{rx_{aa}},\quad w_2=y_2(ID_a,ID_b,T_a)(\mathrm{mod}\ p),$$
$$y_3=y_b^{k_2},\quad c_3=g^{k_2},\quad w_3=y_3(ID_a,ID_b,T_a,K_{ab})(\mathrm{mod}\ p),$$

a 将 ID_a,c_2,w_2,c_3,w_3 传送至 YS.

·YS 首先根据 ID_a 用 x_{ay}对密文的 $E_{y_a^{x_{aa}}}[ID_a,ID_b,T_a]$部分解密,得知 a 欲与 b 进行通信,然后将密文的 $E_{y_b}[ID_a,ID_b,T_a,K_{ab}]$部分用 b 的 YS 私钥 x_{by}加密.最后将二次加密的密文传送到用户 b.

$$YS\rightarrow b:E_{y_b^{x_{by}}}[ID_a,ID_b,T_a,K_{ab}],$$

即 YS 计算

$c_3=c^{x_{ys}}\pmod p$，YS 将 c_3,w_3 传送至 b.

• 用户 b 解密 YS 发来的消息后得到 T_a,K_{ab}，用 K_{ab} 加密传送 T_a 至用户 a，用户 a 收到 b 发来的消息后就可以与用户 b 通话了. 即

$$b\rightarrow a:E_{k_{ab}}(T_a).$$

从认证过程来看，a 到 YS 和 YS 到 b 传送的消息是非否认的，同时，该认证过程具有文献[5]所提认证协议的安全性.

3 签名算法

用户 a 利用一个在 Z_p^* 内的公开并已知的哈希函数 $H(-)$，计算消息 m 摘要 $H(m)=h$，对于全部消息 m，$H(m)\in z_{p-1}^*$. 具体的签名过程如下：

• 用户选择一随机数 $k,k\in z_{p-1}^*$ 并且 $(k,p-1)=1$

• 用户计算 $r=g^k\pmod p$

• 用户计算 $s=k^{-1}(h-x_a r)\pmod{p-1}$，$t=g^{x_{ys}r}\pmod p$

• 将五元组 (ID_a,m,r,s,t) 送至 YS，YS 计算

$$t=t^{(-x_{ys})^{-1}}\pmod p.$$

YS 传送 (ID_a,m,r,s,t) 至接收者，设接收者得到的五元组为 $(\underline{ID}_a,\underline{m},\underline{r},\underline{s},\underline{t})$. 接收者从 CA 处得到标号为 $\underline{ID}_a$ 的用户的公钥证书，计算 $\underline{h}=H(\underline{m})$，接收者作如下检验：

$$\text{计算 } u=g^{\underline{h}}\pmod p,$$

$$\text{计算 } v=y_{\underline{a}}^{\underline{s}}\underline{r}^{\underline{s}}\underline{t}\pmod p,$$

当且仅当 $u=v$ 时，签名被认为是正确的.

不难看出，这种签名的本质仍是 Elgamal 签名. 所以其安全性由 Elgamal 签名算法保证. 这种通过 YS 进行签名的方法在审计等方面具有优点，文献[2]中已作过详细的说明，此处不再赘述.

4 文件加密

设用户 a 要对自己的文件加密，随机选择一次性加密密钥 k_a，用 a 的公钥 y_a 加密传送 k_a 至 YS，由 YS 用 a 的 YS 私钥 x_{ay} 加密得到的消息，为了能恢复密钥，本系统采用文献[8]中的产生 LEAF 字段的方法，该字段应包含被 a 的公钥与 a 的 YS 私钥 x_{ay} 加密的 k_a 及其他信息. 然后 YS 向 a 传送用 y_a,x_{ay} 加密的 T_a，a 解密收到消息后即可对自己的文件加密. 具体的过程可描述如下：

$$a\rightarrow YS:E_{y_a}[k_a,T_a];\quad YS\rightarrow a:E_{y_a\cdot x_{ay}}[T_a].$$

即 a 选择一随机数 k_5，使得 $k_5\in z_{p-1}^*$，计算

$$y_5=y_a^{k_5}\pmod p,\quad c_5=g^{k_5 r_{ay}},\quad w_5=y_5(k_a,T_a)\pmod p,$$

a 将 c_5,w_5 传送到 YS.

LEAF 字段的内容如下：

$$\text{LEAF}:(ID_{\text{LEAF}},E_{y_a}[k_a,T_a]\pmod p,\ldots),$$

其中 ID_{LEAF} 代表 LEAF 的标识号.

5 密钥托管

5.1 政府监听

众所周知，密钥托管的一个重要目的就是为合法的第三者提供监听手段，而本安全系统的一个重要特点便是提供了此功能. 本系统中所有的监听者都是系统的用户. 具体的监听过程如下：

(1) 合法的监听者向 YS 提交监听请求 R，该请求是被监听者签名的. 同时，该请求还应包括监听的对象、监听的时间期限等信息. 当然，如果是有多个监听者共同签名才能进行的监听，提交的请求 R 应是被多个监听者共

同签名的.其共同签名的算法可以通过多重签名来完成,先计算 $h=H(R)$,再对 h 由第 1 个监听者签名,由第 2 个监听者对第 1 个监听者的签名结果再签名,依此类推.

(2) YS 核验了 R 的真实性以后,合法的监听者向被监听的用户的委托人收集拆分密钥片后得到用户私钥,YS 服务器给合法的监听者发送被用户的公钥及 YS 私钥加密的会话密钥,监听者可以实现监听.从监听的过程来看,监听者尽管获得了用户的私钥,但由于它不可能知道用户的 YS 私钥,所以其监听权限只能限于本次会话.

5.2 密钥恢复

用户 a 欲恢复某一次加密的密钥,首先将自己签名的恢复申请 R 送交 YS,YS 核验 R 以后,根据 R 中提供的要恢复密钥的相关信息,找到该次加密产生的 LEAF,将 LEAF 中的被 a 的公钥与 YS 私钥 x_{ys} 加密的 k_a,传送至用户 a,用户 a 即可用自己的私钥 x_{aa} 解密得到 k_a.当然,如果合法的第三方想恢复被用户加密的文件,则第三方只需从 a 的委托人那里恢复用户私钥 x_{aa},再从 YS 那里得到被 a 的公钥与 YS 私钥 x_{ys} 加密的 k_a 即可.

6 结束语

作为一个安全系统,仅支持一种公钥算法是不够的,本文提出的基于 Elgamal 算法的系统是对这种安全系统的补充和发展.本系统引入了 CA,它的存在使得用户的公钥有可信任第三方的公证.同时,引入密钥拆分机制,降低了特权滥用和攻破的可能性,大大增加了 YS 和文件服务器的安全性.总之,该系统在安全性能上优于原 Yaksha 系统.YS 可能是本系统的一个瓶颈,我们可以采用多个 YS 相互协作的方法或其他 YS 冗余策略来解决这个问题.同时,该系统可以用于防火墙等安全产品中.

参考文献

1 A proposed federal information processing standard for an escrowed encryption standard (EES). Federal Register, 1993

2 Ganesan R. The Yaksha security system. Communications of the ACM, 1996,39(3):55~60

3 ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory, 1985,16(31):469~472

4 Shamir A. How to share a secret. Communication of the ACM, 1979,22(11):612~613

5 Sun Xiao rong. Authentication for network and distributed system. Computer Research and Development, 1998,35(10):865~868
(孙晓蓉.网络和分布式系统的认证.计算机研究与发展,1998,35(10):865~868)

6 Ganesan R. Yaksha, augmenting Kerberos with the public key cryptography. In: Proceedings of the Internet Society Symposium on Network and Distributed System Security. 1995

7 Neuman C, Ts'o T. Kerberos, an authentication service for computer network. IEEE Communication Magazine, 1994,32(9):33~38

8 Walker S T, Lipner S B, Ellision C M *et al*. Commercial key recovery. Communications of the ACM, 1996,39(3):41~47

Research on a New Type Integrated Security System

MENG Yang LIU Ke-long QING Si-han

(*Institute of Software The Chinese Academy of Sciences Beijing* 100080)

(*Engineering Research Center for Information Security Technology The Chinese Academy of Sciences Beijing* 100080)

Abstract In this paper, a new type Yaksha security system is presented based on ELGAMAL(NOT RSA) algorithm. The system is capable of reusing a single security infrastructure to perform various security functions—— cryptography digital signatures, distributed authentication and key exchange. At the same time, how the system can be used for key escrow is also described, one of the discussions which attract public attention.

Key words Discrete logarithm, distributed authentication, key escrow, security system, certification authority.