等级加密体制中的密钥管理研究*

蒙 杨, 卿斯汉, 刘克龙

(中国科学院 软件研究所,北京 100080);
(中国科学院 信息安全技术工程研究中心,北京 100080)
E-mail: mengy@ercist.iscas.ac.cn
http://www.ercist.ac.cn

摘要: 分别利用对称加密技术、非对称加密技术以及对称加密与非对称加密结合技术,提出在等级加密下的密钥管理体制,同时对其安全性与效率进行分析.与现有的体制相比,这种体制采用树状模型,其各等级用户之间的密钥通过关系参数连接起来,使得该体制不但适合于一般的等级体制,同时也适合于密钥频繁改变、用户动态变化的应用环境.

关键词: 对称加密;非对称加密;密钥管理
中图法分类号: TP309 **文献标识码**: A

在公钥加密体制被提出以后,多个用户参加的加密体制的研究进展很快,如门限加密[1].但是,这些加密体制把参加者看成是平等的,而在实际应用中,用户的组织往往是一种等级体制,所谓等级体制,即指用户按其安全等级分成等级组,上级用户可以得到下级用户的加密信息,反之一定不成立.关于等级加密的研究归根到底是密钥的管理问题,关于类似问题的研究有文献[2~6],但是它们各有缺点.文献[2]通过构造不同组的关系参数来完成密钥分发,尽管其构造参数的思路比较新奇,但实质上,他以一种未经讨论的加密算法用高级别用户的密钥加密低级别用户的密钥.所以,文献[3]发现了只要已知某一安全类别的上一次的密钥,则新选择的密钥可以轻易地得到.文献[4,5]中提出的等级体制基于RSA(Ron Rivest, Adi Shamir, Leonard Adleman)体制,作为一种加密体制,只采用一种算法是不够的.文献[6]采用传送ticket的方法来完成,每次完成一次加密,需要多个组之间相互协作,但这种方法不适合交互式服务.本文提出基于密钥加密、公钥加密、公钥加密与密钥加密相结合的等级加密下的密钥管理体制,该体制利用树作为等级体制的管理模型,树中每条边所对应的是不同等级组之间的密钥关系,每一个组的有关变化只影响与该组有边相连的组.本文第1节介绍等级模型.第2节介绍密钥管理体制.

1 等级模型

在等级体制中,我们把所有用户按安全等级分成组,组之间是按外向树组织,树中每一个节点对应一个组,父节点对应的组一定能解密子孙节点对应组所加密的消息,反之则一定不成立.由于节点与组之间一一对应,所以,为叙述方便,在下文中,节点与组为同一概念.我们以图1为例来说明.所有用户被分为6个组,组 g_0 的安全级别最高,组 g_3, g_4, g_5 的安全级别最低.也就是说, g_0 可

· 收稿日期: 1999-07-21; 修改日期: 2000-04-19
基金项目: 国家自然科学基金资助项目(69673016)

作者简介: 蒙杨(1972~),男,甘肃平凉人,博士,讲师,主要研究领域为信息安全理论与技术;卿斯汉(1939~),男,湖南隆回人,研究员,博士生导师,主要研究领域为信息安全理论与技术;刘克龙(1971~),男,安徽桐城人,博士,讲师,主要研究领域为信息安全理论与技术.

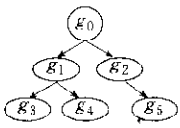

Fig. 1 The tree constructed from groups

图1 分组构成的树

以解密任何一个组加密的消息,g_3,g_4,g_5只能解密自己组加密的消息.这里,消息是指文件内容、电子邮件、同组用户之间的会话等.对于外向树中的每一个节点所对应的组,我们把父节点对应的组称为父组,把了孙节点对应的组称为子孙组.用$g_{par(j)}$表示g_j的父组,用$g_{des(j)}$表示g_j的子孙组,用$g_{son(j)}$表示g_j的子组.根节点对应的组称为根组,把叶子节点对应的组称为叶子组.

2 密钥管理体制

我们用$E_k\{x\}$表示用密钥K利用对称体制对消息x加密,$E_k[x]$表示用公钥K对x利用非对称体制加密,$D_k\{x\}([x])$表示相应的解密运算,$\{x\}\mathrm{sig}(A)$表示x连同签名的消息,ID_j表示组g_j的属性字符串.同样,我们有一个密钥管理中心,作为可信任的第三方存在,我们把该中心称为KMC(key management center).在我们的体制中有一个证书当局CA(certification authoriy)存在.

2.1 利用非对称加密技术的体制

2.1.1 会话密钥的产生与发布

在此体制下,KMC用来存放组公钥以及存放组之间的关系参数.会话密钥的产生与发布的步骤如下:

步骤1. 每一个子组g_j选择公钥pk_j和私钥s_j,把$\{\mathrm{ID}_j,pk_j\}\mathrm{sig}(g_j)$送交KMC,KMC对其签名并发布出来.这里,如果该组是第一次参与到该等级加密体制中来,则$\{\mathrm{ID}_j,pk_j\}$需通过安全信道传送至KMC.这是由于不诚实的组假扮成另一个组与KMC通信,而KMC无法判断其真伪所至.同时,把pk_j及相关的信息传送到CA,由其建立相应的证书.

步骤2. 每一个非根组g_j,从KMC得到$g_{par(j)}$的由KMC签名的公钥,然后计算:

$$E_{pk_{par(j)}}[s_j],$$

并传送到KMC,KMC签名并发布出来,我们把上式称为$g_{par(j)}$与g_j的关系参数.很显然,树中的每一条边对应于一个关系参数.

步骤3. g_j选择会话密钥k_j,计算:

$$E_{pk_j}[k_j],$$

然后将上式及有关的必须信息(如管理信息等)传送到KMC,由KMC签名并发布出来.

2.1.2 会话密钥的推导

由于任何一个组到其子孙组都有一条直接的通路,通路上每一条边都有对应的关系参数,只要已知父节点的私钥,就能推导出子孙节点的私钥.所以任何的父组都能得到子孙组的私钥,这样,通过解密由子孙组的公钥加密的会话密钥,就可以得到相应的会话密钥,也就是说,每一个祖先组都能够计算得到子孙组的会话密钥.

2.1.3 用户私钥的改变

为安全起见,组可能改变自己的公、私钥.以组g_j为例,将其公、私钥改为pk'_j,s'_j的过程如下:

步骤1. 把$E_{pk_{KMC}}[pk'_j,s_j]$传送到KMC,KMC对仍有效的会话密钥重新加密,比如会话密钥k_j仍有效,则KMC计算:

$$D_{s_j}[k_j],\quad E_{pk'_j}[k_j],$$

然后将加密结果签名并公布出来.

步骤 2. g_j 计算该组与其父组的关系参数:

$$E_{pk_{par(j)}}[s'_j],$$

然后将 $E_{pk_{par(j)}}[s'_j]$及有关的必需信息送到 KMC.

步骤 3. KMC 将 pk'_j,$E_{pk_{par(j)}}[s'_j]$分别签名后发布出来.

2.1.4 基于 RSA 算法的等级加密体制

每一个组 g_j 选择素数 p_j,q_j,计算 $n_j=p_jq_j$,随机选择 e_j,使得 $gcd(e_j,\varphi(n_j))=1$,使用 Euclidean 算法计算 $d_j=e_j^{-1} \bmod \varphi(n_j)$,将$\{n_j,e_j\}sig(g_j)$送交 KMC,如果是第一次加入到等级加密体制中来,则通过安全信道将$\{n_j,e_j\}$送交 KMC. 同时,将$\{n_j,e_j\}$及相关信息送交到 CA,由 CA 产生相应的证书.

g_j 从 KMC 得到$\{n_{par(j)},e_{par(j)}\}$sig(KMC),计算$\{d_j\}^{e_{par(j)}} (\bmod n_{par(j)})$,如果$\{d_j\}$的位数大于$\log_2^{n_{par(j)}}$,则对 d_j 进行分块加密,然后将结果送交 KMC,再由 KMC 签名后发布出来.

组 g_j 选择会话密钥 k_j,计算 $k_j^{e_j} (\bmod n_j)$,然后将计算结果及相关的信息送交 KMC,再由 KMC 签名并且发布出来.

注意:在使用基于 RSA 的等级加密体制时,所有组不能选取同样的素数 p,q.

2.1.5 基于 ElGamal 算法的等级加密体制

KMC 选择大素数 p,在乘群 z_p^* 中选择本原根 α,然后将 p,α 公布出来供所有的用户使用.

每一个组 g_j 随机选择 $x\in z_{p-1}^*$,计算 $y_j=\alpha^{x_j} (\bmod p)$,然后将$\{ID_j,y_j\}sig(g_j)$传送到 KMC. 如果是第一次参与到等级加密体制中来,则通过安全信道传送,在 KMC 验证签名的正确性之后,重新签名并将其发布出来. 同时,将$\{y_j\}$及相关信息送交到 CA,由 CA 产生相应的证书.

组 g_j 从 KMC 得到$\{y_{par(j)}\}$sig(KMC),随机选择 $r_j\in z_{p-1}^*$,计算

$$g^{r_j},y_{par(j)}^{r_j}x_j \quad (\bmod p),$$

然后将结果送交 KMC,再由 KMC 签名后发布出来.

会话密钥的推导与组公、私钥的改变这里不再加以描述.

2.1.6 安全性分析

以上的体制具有如下的安全特性:下级组要得到祖先组的会话密钥,面对离散对数问题(ElGamal)或素因子分解(RSA)问题;对于外部用户(不属于会话组的用户),要得到任何子组的会话密钥,也面对同样的问题.

这是因为:我们假设子组 g_j 企图得到 $k_{par(j)}$,他只能通过获得 $s_{par(j)}$ 去解密 $E_{pk_{par(j)}}[E_{par(j)}]$,要获得 $s_{par(j)}$,首先,g_j 可以通过改变 s_j 得到自己想要的密文 $E_{pk_{par(j)}}[s_j]$. 很显然,得到选择的明密文对对 RSA 和 ElGamal 的安全性不构成任何威胁,所以,通过改变 s_j 获得明密文对无助于得到 $s_{par(j)}$. 其次,g_j 只有通过解密 $E_{pk_{par(par(j))}}[s_{par(j)}]$,才能获得其父组的私钥,而解密该密文只能面对离散对数问题或者素因子分解问题. 所以,对于 g_j 要获得祖先组的会话密钥或者对于外部用户要获得任何组的会话密钥,在目前看来是不可能的.

2.2 基于对称加密技术的体制

与公钥体制不同,若采用密钥加密体制,则同 Kerberos[7]一样,KMC 在产生不同组的会话密钥之前,与每一个组有一个预共享密钥 sh_i.

2.2.1 会话密钥的产生

步骤 1. 子组 g_j 选择密钥 $k_j^{(0)}$,计算:

$$E_{sh_j}\{k_j^{(0)}\},$$

并传送至 KMC.

步骤 2. KMC 用 $g_{par(j)}$ 的密钥 $k_{par(j)}^{(0)}$,计算:

$$E_{k_{par(j)}^{(0)}}\{k_j^{(0)}\},$$

并发布出来.

步骤 3. 每一个子组 g_j 选择会话密钥 $k_j^{(1)}$,然后计算:

$$E_{k_j^{(0)}}\{k_j^{(1)}\},$$

并将计算结果传送到 KMC.

步骤 4. KMC 将 $E_{k_j^{(0)}}\{k_j^{(1)}\}$ 发布出来.

2.2.2 会话密钥的推导

与基于公钥的体制一样,任何一组到其子孙组都有一条通路,沿着该通路的对应边逐一解密,即可得到子孙组密钥,从而得到其会话密钥.

2.2.3 组密钥的改变

设 g_j 要改变密钥为 $k_j^{(0)'}$,则 g_j 计算:

$$E_{sh_j}\{k_j^{(0)'}\},$$

并传送到 KMC,KMC 计算:

$$E_{k_{par(j)}^{(0)}}\{k_j^{(0)'}\},$$

并发布出来,对于 g_j 所有的子组 $g_{son(j)}$,计算:

$$E_{k_j^{(0)'}}\{k_{son(j)}^{(0)}\},$$

并发布出来.对仍有效的会话密钥,KMC 先解密得到会话密钥,然后用 $k_j^{(0)'}$ 加密后重新发布出来.

2.2.4 共享密钥的改变

设 g_j 要改变其与 KMC 之间的会话密钥为 sh'_j,则 g_j 计算:

$$E_{sh_j}\{sh'_j\},$$

传送到 KMC,KMC 解密得到 sh'_j.

2.2.5 安全性分析

如果等级加密体制采用的对称算法可以抵抗明文的已知攻击,则以上体制可以做到:下级组要得到任何其祖先组的会话密钥只能使用穷搜索攻击;对于外部用户,他要得到任何组的会话密钥也只能采用穷搜索攻击.

这是由于,一个内部用户可以通过不停地改变其密钥来获得由其父组密钥加密的明密文对,表面上看来,该用户可以采取选择明文攻击的策略.到目前为止,由于选择明文攻击所需的明密文对数量非常大,我们可以通过有效手段控制任何组得到足够多的明密文对,使得内部用户不可能实施选择明文攻击,所以,只要我们的对称算法抵抗已知明文攻击,则下级组要得到任何其祖先组的会话密钥只能使用穷搜索攻击.同样,外部用户要得到某一组的会话密钥,也只能使用穷搜索攻击.

2.3 采用混合算法的体制

我们知道,公钥加密的安全性要高于密钥加密,但其运算效率较低.而密钥算法的效率虽然高,但其安全性却较低,同时也不具备签名等特性.下面,我们利用公钥与密钥的优点,提出采用混合算法的管理体制.同时,该体制由一密钥来管理中心 KMC.

2.3.1 会话密钥的产生

步骤1. 每一个组 g_j 选择公钥 pk_j 和私钥 s_j,把 ID_j,pk_j 送交KMC.KMC对其签名,并将其发布出来.与基于公钥的体制一样,在第一次使用该等级体制时,通过安全信道传送,同时将 $\{pk_j\}$ 连同相关信息传送到CA,由CA产生相应的证书.

步骤2. 由 g_j 及其父组 $g_{par(j)}$ 通过公钥体制协商产生他们的共享密钥 $k_{j,par(j)}$.

步骤3. 如果 g_j 为非叶子节点,则对于 g_j 的每一个子节点 $g_{son(j)}$,计算:

$$E_{k_{j,par(j)}}\{k_{j,son(j)}\},$$

并发送到KMC,由KMC签名发布出来.

步骤4. g_j 选择会话密钥 k_j,计算:

$$E_{k_{j,par(j)}}\{k_j\},$$

并发送到KMC,由KMC签名发布出来.

2.3.2 共享密钥的改变

出于安全方面的考虑,组与其父组之间可能需要更换共享密钥,其步骤如下:

步骤1. 组和父组利用 $k_{j,par(j)}$ 通过密钥加密传送新的共享密钥 $k'_{j,par(j)}$.

步骤2. 传送 $E_{pk_{KMC}}[k_{j,par(j)},k'_{j,par(j)}]$ 到KMC,KMC对仍有效的会话密钥重新加密,签名后重新发布出来,然后丢弃 $k_{j,par(j)}$,$k'_{j,par(j)}$.

步骤3. 对任意的非叶子组 $g_{son(j)}$,g_j 计算:

$$E'_{k_{j,par(j)}}\{k_{j,son(j)}\},$$

传送到KMC并签名发布出来.

步骤4. 如果 $g_{par(j)}$ 是非根组,$g_{par(j)}$ 计算:

$$E_{k_{par(j),par\,par(j)}}\{k'_{j,par(j)}\},$$

传送到KMC并签名发布出来.

2.3.3 会话密钥的推导

由于从任何上一级组到子孙组之间都有一条通路,通路中的每个组既与父组共享一个密钥,又与儿子组共享一个密钥,而用前一个密钥加密后一个密钥的信息由KMC公布出来,所以任何上一级组可以得到子孙组与其父组的共享密钥,从而可以得到子孙组的会话密钥.

2.3.4 基于"自证明"的等级加密体制

由于在我们的体制中涉及到密钥的协商,同时,由于子组之间的共享密钥需要不停地加以改变,所以,我们采用适合于这种体制的"自证明"密钥协商协议[8],相关的文献有文献[9~11].具体步骤如下:

KMC随机选择素数 $p,q,n=pq,p-1=2p',q-1=2q',p',q'$ 都是素数,选择阶为 $r=p'q'$ 的整数 α,大整数 $u<r$ 和一个单向函数 f,KMC公开 α,u,f,n,保密 r,丢弃 p,q.

步骤1. 组 g_j 随机选择 $x_j<u$,计算 $y_j=\alpha^{x_j}(\mathrm{mod}\ n)$,然后传送 ID_j,y_j 到KMC,KMC计算 g_j 的"证明":

$$I_j=f(ID_j),\quad w_j=y_j^{I_j^{-1}}(\mathrm{mod}\ n),$$

并将 w_j 发送回 g_j.

步骤2. 组 g_j 传送 ID_j,w_j 到其父节点,同样,其父节点组 $g_{par(j)}$ 传送 $ID_{par(j)},w_{par(j)}$ 到 g_j,$g_{par(j)}$ 计算:

$$I_j=f(\mathrm{ID}_j),\ y_j=w_j^{I_j}(\mathrm{mod}\ n),$$

$$k_{j,\mathrm{par}(j)}=y_j^{x_{\mathrm{par}(j)}}(\mathrm{mod}\ n),\quad E_{k_{\mathrm{par}(j),\mathrm{par}(\mathrm{par}(j))}}\{k_{j,\mathrm{par}(j)}\},$$

并将 $E_{k_{\mathrm{par}(j),\mathrm{par}(\mathrm{par}(j))}}\{k_{j,\mathrm{par}(j)}\}$传送到 KMC,并由 KMC 签名发布出来.

步骤 3. g_j 计算

$$I_{\mathrm{par}(j)}=f(\mathrm{ID}_{\mathrm{par}(j)}),\quad y_{\mathrm{par}(j)}=w_{\mathrm{par}(j)}^{I_{\mathrm{par}(j)}}(\mathrm{mod}\ n),$$

$$k_{j,\mathrm{par}(j)}=y_{\mathrm{par}(j)}^{x_j}(\mathrm{mod}\ n),\quad E_{k_{j,\mathrm{par}(j)}}\{k_{j,\mathrm{par}(j)}\},$$

并将 $E_{k_{j,\mathrm{par}(j)}}\{k_{j,\mathrm{par}(j)}\}$传送到 KMC,并由 KMC 签名发布出来.

步骤 4. g_j 选择会话密钥 k_j,计算:

$$E_{k_{j,\mathrm{par}(j)}}\{k_j\},$$

将计算结果送交 KMC,由 KMC 签名公布出来.

2.3.5 共享密钥的改变

设 g_j 要把其与 $g_{\mathrm{par}(j)}$之间的共享密钥 $k_{j,\mathrm{par}(j)}$改变为 $k'_{j,\mathrm{par}(j)}$,则计算:

$$E_{k_{j,\mathrm{par}(j)}}\{k'_{j,\mathrm{par}(j)}\},$$

并传送到 $g_{\mathrm{par}(j)}$,改变与 $g_{\mathrm{son}(j)}$之间的共享密钥采用相同的方法.

对仍有效的会话密钥的重新加密以及对关系参数的重新计算与第 2.3.2 节的计算相同.

注意,一旦共享密钥泄露,则只能重新产生共享密钥,需要重新加密有效的会话密钥.

2.3.6 安全性分析

通过文献[9]我们知道,这种端对端共享密钥的分发体制等价于 diffie-hellman 问题,所以,根据文献[12]中的定理 5,以上的密钥分发体制是"保证保密"的.因此,对于计算能力有限的下级组,要得到上级组的最初的共享密钥在计算上不可能的.同样,对于外部用户(不属于任何组的用户),要得到任何一组的最初的共享密钥在计算上也是不可能的.

下面我们分析共享密钥不停变化的情况.假设组 g_j 企图得到 $g_{\mathrm{par}(j)}$的会话密钥,他可以通过不停地改变 $k_{j,\mathrm{par}(j)}$来得到明密文对:

$$(k_{j,\mathrm{par}(j)},E_{k_{\mathrm{par}(j),\mathrm{par}(\mathrm{par}(j))}}\{k_{p,\mathrm{par}(j)}\}).$$

在我们的体制里稍加限制,攻击者很难得到选择明文攻击的明密文对,所以只要采用的密钥算法抵抗已知明文攻击,则 g_j 要得到 $k_{\mathrm{par}(j),\mathrm{par}(\mathrm{par}(j))}$也只能采用穷搜索攻击.同样,外部用户要得到任何一组的会话密钥,只能采用穷搜索攻击.显然,子孙组联合攻击与外部攻击的难度是一样的.

注意,以上各协议在消息传送时都应加盖时间戳或者 nonce,以防重放攻击.同时,与文献[2]中的体制一样,以上的体制适合于密钥频繁改变和用户动态变化的环境,有关具体的分析,可以参看文献[2].

3 结 论

本文以树作为等级模型,提出了一个比较完备的等级加密体制,通过关系参数来实现不同组之间的存取关系,同时对其安全性进行分析.众多的实际应用,如多安全级的分布式应用、网上会议、电视会议中多级管理、实时网上游戏中的公平竞赛、交互式的实时模拟甚至网上聊天都涉及到类似问题,所以,该体制有较广泛的应用背景.

References:

[1] Dewmedt, Y. Society and grop oriented cryptography: a new concept. In: Odlyzko, ed. Proceedings of CRYPTO'87. Lecture Notes in Computer Science 293, Berlin: Springer-Verlag, 1988. 120~127.

[2] Lin, Chu-hsing. Dynamic key management schemes for access control in a hierarchy. Computer Communications, 1997,20(15):1381~1385.

[3] Lee, Narn-yih, Hwang, Tzonelih. Comments on 'dynamic key management schemes for access control in a hierarchy'. Computer Communications, 1997,22(5):1381~1385.

[4] Sun, Hung-min, Shieh, Shiuh-pyng. Secure broadcasting in large networks. Computer Communications, 1998,21(3):279~283.

[5] Sun, Hung-min, Shieh, Shiuh-pyng, Sun, Hsin-min. A note on breaking and repairing a secure broadcasting in large networks. Computer Communications, 1999,22(5):193~194.

[6] Ghodosi, H., Pieprzyk, J., Charnes, C., *et al*. Cryptosystems for hierarchical groups. In: Varadharajan, Pjeprzyk-Mu, eds. Proceedings of Information Security and Privacy (ACISP'97). Lecture Notes in Computer Science 1270, Berlin: Springer-Verlag, 1997. 275~286.

[7] Neuman, C, Kerboses, Ts'o T. An authentication service for computer network. IEEE Communications, 1994,32(9):33~38.

[8] Shahrokh, Saeednia. Identity-Based and self-certified key-exchange protocols. In: Varadharajan, Pjeprzyk-Mu, eds. Proceedings of Information Security and Privacy (ACISP'97). Lecture Notes in Computer Science 1270, Berlin: Springer-Verlag, 1997. 303~313.

[9] Buchmann, J., Williams, H. A key-exchange system based on imaginary quardratic fields. Journal of Cryptology, 1988,1(2):107~118.

[10] Gunther, C. An identity-based key-exchange protocol. In: Davenport, ed. Proceedings of EuroCrypt'89. Lecture Notes in Computer Science 434, Berlin: Springer-Verlag, 1990. 29~37.

[11] Girault, M. Self-Certified public keys. In: Christoodolakis, ed. Proceedings of EuroCrypt'91. Lecture Notes in Computer Science 547, Berlin: Springer-Verlag, 1991. 490~497.

[12] Burmester, M., Desmedt, Y. A secure and efficient conference key distribution system. In: Pacholski, Tiurgn, ed. Proceedings of CRYPT'94. Lecture Notes in Computer Science 839, Berlin: Springer-Verlag, 1995. 275~286.

Research for Key Management in Cryptosystem for Hierarchy*

MENG Yang, QING Si-han, LIU Ke-long

(*Institute of Software, The Chinese Academy of Sciences, Beijing* 100080, *China*);
(*Engineering Research Center for Information Security Technology, The Chinese Academy of Sciences, Beijing* 100080, *China*)
E-mail: mengy@ercist.iscas.ac.cn
http://www.ercist.ac.cn

Abstract: In this paper, a key management scheme in cryptosystem for hierarchy is proposed using symmetric cryptography technology, asymmetric cryptography technology, symmetric cryptography combined with asymmetric cryptography technology. At the same time, it's security and efficiency are analyzed. Comparing with the exist systems, this scheme adopts tree model, the users of different secure levels contact each other with relation parameters, the system is appropriate for not only the common environments, but also the special environments in which key changes frequently and users change dynamically.

Key words: symmetric cryptography; asymmetric cryptography; key management

* Received July 21, 1999; accepted April 19, 2000
Supported by the National Natural Science Foundation of China under Grant No. 69673016