(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 103986574 B
(45)授权公告日 2017.10.13

(21)申请号 201410209022.9

(22)申请日 2014.05.16

(65)同一申请的已公布的文献号
申请公布号 CN 103986574 A

(43)申请公布日 2014.08.13

(73)专利权人 北京航空航天大学
地址 100191 北京市海淀区学院路37号

(72)发明人 刘建伟 周云雅 伍前红 刘巍然

(74)专利代理机构 北京慧泉知识产权代理有限公司 11232
代理人 王顺荣 唐爱华

(51)Int.Cl.
H04L 9/30(2006.01)
H04L 9/08(2006.01)

(56)对比文件

Weiran Liu, Xiao Liu, Qianhong Wu, Bo Qin.Experimental Performance Comparisons between (H)IBE Schemes over Composite-Order and Prime-Order Bilinear Groups.《Proceedings of 2014 11th International Bhurban Conference on Applied Science & Technology (IBCAST)》.2014,全文.

张新方.“基于身份密码体制的研究”.《中国优秀硕士学位论文全文数据库》.2009,正文第37-39页.

审查员 王卓然

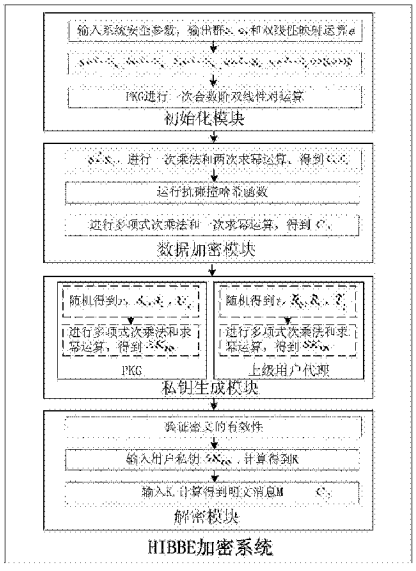
权利要求书3页 说明书9页 附图3页

(54)发明名称

一种基于身份的分层广播加密方法

(57)摘要

一种基于身份的分层广播加密方法，步骤如下：1、PKG输入系统安全系数，输出初始化参数；2、PKG运行随机数生成算法，选择随机数；3、PKG运行双线性对运算，输出公钥，保留主密钥；4、加密方运行随机数生成算法、乘法和求幂运算，输出部分密文；5、加密方运行抗碰撞哈希函数；6、加密方运行乘法和求幂运算，输出完整密文；7、PKG或上级用户运行随机数生成算法，生成私钥所需的随机数；8、PKG或上级用户运行乘法和求幂运算，输出用户私钥；9、解密方运行抗碰撞哈希函数，验证密文的有效性，有效进行下一步骤，无效输出NULL；10、满足解密条件的用户，由用户私钥计算得到K；11、解密用户根据K，运行双线性对运算和乘法运算，输出明文。



CN 103986574 B

1.一种基于身份的分层广播加密方法，其特征在于：该方法是由四个模块共11个步骤实现基本功能，各模块按照“初始化模块”→“数据加密模块”→“私钥生成模块”→“解密模块”顺序执行，其步骤如下：

模块一：初始化模块

私钥生成中心即PKG，在这一模块中将系统安全参数λ、用户分层结构的最大深度D和用户数量的最大值n+1作为输入，输出主密钥MSK和公钥PK；公钥PK可以公开，而主密钥MSK则需由PKG严格保密；该模块功能的具体实现分为三步：

步骤1：PKG首先输入系统安全参数λ，然后运行算法$\mathcal{G}(1^{\lambda})$，输出两个阶数为合数N的群$\mathbb{G}$、$\mathbb{G}_T$和一个双线性映射运算$e:\ \mathbb{G}\times\mathbb{G}\rightarrow\mathbb{G}_T$，其中$N=p_1p_2p_3$，$p_1$，$p_2$，$p_3$分别是N的三个离散的大素数因子；

步骤2：PKG运行随机数生成算法，随机选择阶数为p_1的群$\mathbb{G}_{p_1}$中的一个生成元g，$\mathbb{G}_{p_1}$群中的一个元素h，阶数为p_3的群$\mathbb{G}_{p_3}$中的一个元素X_3以及Z_N:{0,1,…,N-1}域中的一个元素α作为随机指数；并对包括动态虚拟用户在内的所有n+1个用户随机分配$\mathbb{G}_{p_1}$群中的一个元素u_i，$i\in[1,n+1]$内的整数；

步骤3：最后PKG进行一次合数阶双线性对运算，得到$\mathbb{G}_T$群中的一个元素$e(g,g)^{\alpha}$；经过上述三个步骤得到的参数：

$(g,h,u_1,\ldots,u_{n+1},X_3,e(g,g)^{\alpha})$

作为公钥能对外公开，g^{α}作为主密钥由PKG保管；

模块二：数据加密模块

加密方在这一模块中将公钥PK和待加密的消息M以及接收密文用户的身份向量集合V作为输入，输出加密消息M后得到的密文CT；该模块功能的实现分三步：

步骤4：加密方随机选择Z_N:{0,1,…,N-1}域中的一个元素作为指数β，完成1次乘法和2次求幂运算，得到：

$(C_0,C_2)\leftarrow(g^{\beta},e(g,g)^{\alpha\beta}\cdot M)$

对于接收用户的身份向量集合V，我们定义集合$\mathbb{I}=\{i:ID_i\in S_V\}$；

步骤5：加密方运行抗碰撞哈希函数H{•}，计算得到$ID_{n+1}\leftarrow H(C_0,C_2)\in\mathbb{Z}_N$，

其运行抗碰撞哈希函数H{•}的计算方法如下：哈希函数的输入是密文C_0、C_2，输出为映射到Z_N:{0,1,…,N-1}域中的元素，该哈希函数能从Pairing-Based Cryptosystems函数包中的库函数中调用得到；

步骤6：加密方完成多项式乘法和求幂运算，得到C_1，

$$C_1\leftarrow\left(h\cdot u_{n+1}^{ID_{n+1}}\cdot\prod_{i\in\mathbb{I}}u_i^{ID_i}\right)^{\beta}$$

最后的密文输出：$CT=(C_0,C_1,C_2)$，

该密文是面向用户身份向量集合$V\cup\{(ID_{n+1})\}$加密的；

模块三：私钥生成模块

该模块由两部分参与：一部分由PKG承担，模块输入某一用户的身份向量ID和主密钥

MSK，生成对应的私钥SK_{ID}；另一部分由待生成私钥的用户的上层用户承担，代理PKG完成用户私钥的生成和分发任务；模块输入该上层用户的私钥$SK_{ID'}$和下层用户的身份ID，输出ID对应的私钥SK_{ID}；

由PKG承担的私钥生成功能具体分为两步实现：

步骤7：PKG随机选择Z_N：{0，1，…，N-1}域中的一个元素r作为指数，运行随机选择两个$\mathbb{G}_{p_3}$群中的元素A_0，A_1；对于用户的身份向量ID，我们定义集合I＝{i：$ID_i \in S_{ID}$}，并对所有不在集合I中的用户随机分配$\mathbb{G}_{p_3}$群中的一个元素U_j，$j \in [1,n+1] \backslash I$内的整数；

步骤8：PKG完成多项式乘法和求幂运算，得到最后的用户私钥

$$SK_{\mathbf{ID}} \leftarrow \left(g^{\alpha} \left(h \cdot \prod_{i \in \mathrm{I}} u_i^{ID_i} \right)^{r} A_0, g^{r} A_1, \left\{ u_j^{r} U_j \right\}_{j \in [1,n+1] \backslash \mathrm{I}} \right)$$

由上级用户承担的私钥生成功能同样分为两步实现：

步骤7*：上级用户首先随机选择Z_N：{0，1，…，N-1}域中的一个元素t作为指数，运行随机数生成算法随机选择两个$\mathbb{G}_{p_3}$群中的元素R_0，R_1，并对所有不在集合I中的用户随机分配$\mathbb{G}_{p_3}$群中的一个元素T_j，$j \in [1,n+1] \backslash I$内的整数；

步骤8*：上级用户由自身的私钥$SK_{ID'}$出发，令 $a_0 = g^{\alpha} \left(h \cdot \prod_{i \in \mathrm{I}'} u_i^{ID_i} \right)^{r'} A_{0'}$，$a_1 = g^{r'} A_{1'}$，$\left\{ b_j \right\}_{j \in [1,n+1] \backslash \mathrm{I}'} = \left\{ u_j^{r'} U_j \right\}_{j \in [1,n+1] \backslash \mathrm{I}'}$，经过多项式乘法和求幂运算能得到用户身份向量ID＝(ID′，ID)对应的私钥SK_{ID}；

$$SK_{\mathbf{ID}} = \left(a_0 \left(b_i^{ID} \right)_{i \in \mathrm{I} \backslash \mathrm{I}'} \left(h \cdot \prod_{i \in \mathrm{I}} u_i^{ID_i} \right)^{t} R_0, a_1 g^{t} R_1, \left\{ b_j u_j^{t} T_j \right\}_{j \in [1,n+1] \backslash \mathrm{I}} \right)$$

模块四：解密模块

解密用户在这一模块中，首先运行抗碰撞哈希函数H{·}验证密文的有效性，如果密文是有效的，则将接收密文用户的身份向量集合$V \cup \{(ID_{n+1})\}$、加密消息M得到的密文CT和用户私钥SK_{ID}作为输入；如果满足条件$ID \in V$，则该模块输出正确的明文消息M；若密文无效，则系统输出NULL即无效符号；该模块功能的实现具体分为三步：

步骤9：解密方首先验证密文的有效性，运行抗碰撞哈希函数H{·}：$ID_{n+1} \leftarrow H(C_0, C_2) \in \mathbb{Z}_N$，检测下列等式是否成立：

$$e(g, C_1) \stackrel{?}{=} e\left(C_0, \left(h \cdot u_{n+1}^{ID_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_i^{ID_i} \right) \right)$$

若等式成立，进行如下步骤10、11，若不成立，系统输出NULL；

其中，所述的“运行抗碰撞哈希函数H{·}”，其运行和计算的方法与步骤5中所述的相同；

步骤10：对于满足$ID \in Pref(V)$条件的用户，由自身的私钥SK_{ID}首先计算得到

$$K = a_0 \cdot \prod_{j \in \mathbb{I} \setminus \mathrm{I}} b_j^{ID_j}$$

步骤11：解密用户根据上步得到的K，进行两次双线性对运算和一次乘法运算，计算输出明文消息M：

$$M = C_2 \frac{e(C_1, a_1)}{e(K, C_0)}$$

上述方案中解密模块的第一步，能进行公开验证；因为验证的输入为公钥参数和密文，均是对外公开的，因而身份的分层广播加密HIBBE方法能用于构造高级的安全协议；

优化模式下的树形用户身份分层结构：所涉及的基于身份的分层广播加密方法，当用户分层结构深度为d时，每个用户私钥包括n-d+2个元素，密文包括三个群元素；考虑到某些接收密文用户的数据存储能力受限的情况，我们在私钥长度和密文长度之间做出折衷，提出了优化模式下的树形用户身份分层结构；

最原始的系统所有用户按照一棵树T的结构组织排列，树节点总数为n；我们对树T进行分割，分割为$\mathcal{T}$棵子树，每棵子树包含的节点数为n_i个，$i \in [1, \mathcal{T}]$内的整数；分割的原则为：①每棵子树的节点数尽量相等，即$n_i \approx n/\mathcal{T}$；②所有子树尽量分享最少的高层节点；每棵子树均看做独立的HIBBE系统，正常运行本技术方案中的各功能模块；当广播加密密文时，若用户集合来自不同的子树，则向涉及到的所有访问结构树进行广播；

通过上述优化，私钥的长度降至$O(\frac{n}{\mathcal{T}})$数量级，密文的长度增加至$O(\mathcal{T})$数量级；若我们令$\mathcal{T} = \sqrt{n}$，则密文和私钥的长度均变为$O(\sqrt{n})$数量级，大大减少了用户的存储负担。

2.根据权利要求1所述的一种基于身份的分层广播加密方法，其特征在于：在步骤1中所述的“运行算法$\mathcal{G}(1^\lambda)$”，其做法如下：PKG根据输入的安全参数λ的大小，选择预定的椭圆曲线：$Y^2 = X^3 + aX + b$，a和b是系数；根据所选椭圆曲线上的点构成群$\mathbb{G}$、$\mathbb{G}_T$，选择一种函数映射e，将群$\mathbb{G}$中的元素映射到群$\mathbb{G}_T$中去；安全参数数值越大，所选择椭圆曲线上的点也越多，群也越大。

3.根据权利要求2所述的一种基于身份的分层广播加密方法，其特征在于：在步骤2中所述的“随机数生成算法”，其做法如下：根据步骤1中所选的椭圆曲线：$Y^2 = X^3 + aX + b$，随机选择自变量x的一个值x_1，计算对应因变量y的值y_1；若点(x_1, y_1)在我们想要映射的群中，则成功生成了随机元素；若点(x_1, y_1)不在群中，则继续选择x的值，直到找到出现在群中的点。

4.根据权利要求1所述的一种基于身份的分层广播加密方法，其特征在于：在步骤3中所述的“合数阶双线性对运算”，其做法如下：自变量的输入为群$\mathbb{G}_{p_1}$中的一个生成元g，输出为群$\mathbb{G}_T$中的元素：$e(g,g)^\alpha$。

一种基于身份的分层广播加密方法

(一)技术领域：

[0001] 本发明涉及一种基于身份的分层广播加密方法，可用于网络分层结构中的用户之间进行保密通信，属于信息安全中密码学领域。

(二)技术背景：

[0002] 进入21世纪，人类已迈入信息化的时代，各种先进的通信技术、计算机技术以及网络技术渗透到了我们生活的方方面面。信息技术的迅猛发展普及，为我们的生活带来了极大的便利，信息全球化、信息一体化正不断改变着我们的生产和生活方式。但是，信息技术在为我们带来生活、工作等各方面便利的同时，也在极大地威胁着我们的个人隐私安全。如何在开放的网络环境下安全、可靠地进行信息交互，成为信息安全领域一个重要的议题。

[0003] 在信息网络安全需求日益迫切的环境下，设计能够保证数据安全传输的健壮安全的技术方法变得至关重要，因此标准公钥基础设施——Public Key Infrastructure(PKI)应运而生。PKI可以在网络信息交互的过程中，对用户的身份证书进行验证，保障通信双方的可信性。但是，在应用过程中，PKI被发现存在很多不尽人意的地方，考虑到PKI系统的构成异常复杂且成本过高，实用性不强，如何高效地验证用户身份成为新的挑战。

[0004] 1984年，Shamir率先提出了基于身份的加密方法——Identity-based Encryption(IBE)。在IBE体制下，用户的身份信息作为加密时使用的公钥，而不是从公钥证书中获得。用户的身份信息可以是用户的身份证号码、网络IP地址或邮箱地址；私钥则由私钥生成中心——Private Key Generator(PKG)处统一生成并分发给各用户，数据加密方不用获得对方的公钥证书也可以完成数据的加密。

[0005] 传统的IBE加密方法，PKG承担着为所有用户生成私钥并认证用户身份的职能；在计算出私钥后，PKG还须通过秘密信道将私钥传送给用户。这无疑降低了系统的效率，甚至成为整个加解密系统的瓶颈；且因PKG存储着所有用户的私钥，所以极易成为敌人攻击的目标。考虑解决上述问题，分层的身份基加密方案——Hierarchical Identity-based Encryption(HIBE)被提出。HIBE方案将用户以树形结构组织起来，模拟多社会组织机构交叉合作的情况，更加接近现实生活。一方面，PKG可授权有上层用户为直属下层用户计算生成私钥，从而减轻了PKG生成、分发私钥的负担，降低了PKG受到攻击的风险。另一方面，传统的IBE系统在加密过程中，加密方是根据用户的单一身份ID加密；在HIBE方案中加密方则分局对方用户的身份向量ID＝(ID_1,ID_2,...,ID_d)对信息进行加密，仅有身份出现在指定身份向量中的合法用户可以解密。例如，在电子健康网络应用环境中，病人Alice想与某医院的医生Bob秘密安全地分享自己的病历信息，Alice仅须根据指定的身份向量：“医院:XXX-科室:XXX-医生:XXX”加密病历数据，并将密文传送给Bob即可。与上述应用场景类似，加密方可能想同时与多个层次结构中的用户进行保密通信。例如，某公司想与同一所大学来自不同实验室的多个教授合作共同开发一款软件。在IBE加密系统中，该公司需要分别与每位教授进行保密通信，但这须一一指明每个用户的加密路径，这势必会给加密方带来沉重的加密开销和冗长的密文存储负担。此外，在基于IP的多路广播网络环境下，采用现有的IBE系

统看似可行，但当更多来自不同IP路径的子节点加入时，用户数目激增，该系统便丧失了其高效性。

[0006] 在上述应用需求的驱使下，我们发明了一种新的加密方法——分层的身份基广播加密(Hierarchical Identity-based Broadcast Encryption，HIBBE)。广播加密，即一种在非安全信道上同时传输数字信息给多个授权用户的方法，仅在授权子集中的用户可以完成对密文的解密。通过引入广播加密功能，加密方可以根据用户的身份向量，设置多用户接收集合对信息加密，同时与多用户进行保密通信，提高了加密方加密数据的工作效率。

[0007] 本发明提出的基于身份的分层广播加密方法支持对任意大小用户子集的加密，仅有在子集中的用户和该类用户的上层用户可以解密；同时兼具私钥生成代理功能，当系统中用户较多时，可大大减轻PKG生成私钥的负担；此外，我们通过在分层用户结构中的第一层加入动态虚拟用户节点，实现了可抵抗适应性选择身份向量集合和选择密文攻击(IND-CIVS-CCA2)。

(三)发明内容：

[0008] 1、目的：

[0009] 本发明的目的是提供一种基于身份的分层广播加密方法，它是一种应用于网络保密通信中安全高效的数据加密方案，它兼具了已有分层身份基加密和广播加密方法的特点，同时克服了现有技术功能不完备的不足，具有高灵活性与可证明安全的优点。该方法可以实现基于用户身份向量对信息加密，支持对多路径用户广播传输密文，上层用户代替私钥生成中心(Private Key Generator，PKG)为下述用户生成私钥，可公开验证密文有效性，并优化私钥与密文长度的功能。

[0010] 2、技术方案：

[0011] 本发明包括三个实体，1)私钥生成中心(Private Key Generator，PKG)：具有验证用户身份，计算生成、分发用户私钥功能的机构。2)数据加密方(Encrypting Party)：具有加密功能的个人或社会机构；3)用户(User)：具有解密功能的个人或社会机构；一般地，用户包括两类，上层用户与下级用户，这由用户在层次结构中的位置决定；其中上层用户具有代理PKG为直属下级用户生成、分发私钥的功能。

[0012] 首先，为了便于理解我们定义用户在分层结构中的身份向量——ID，代表用户在系统中的身份，表示为$ID=(ID_1,ID_2,\ldots,ID_d)$；如图1中所示，基于身份的分层广播加密方案中，用户是以分层的树形结构组织起来的，用户在每一层都有对应的身份ID_i，每个用户的身份向量ID是由每层的身份ID_i串联组成的向量。

[0013] 其次，我们定义$S_{ID}=\{ID_1,ID_2,\ldots,ID_d\}$和V，$S_{ID}$表示用户身份向量ID关联的所有$ID_i$的集合；V表示加密方广播密文时，所有接收用户身份向量的集合。

[0014] 随后我们引入前缀Pref(•)的概念，用以从用户的身份向量中分离出该用户的所有上层用户的身份向量，表示为

[0015] $Pref(ID)=\{(ID_1,ID_2,\ldots,ID_{d'}):d'\leq d\}$

[0016] 该集合包括用户本身及他的所有上层用户的身份向量。从而用户身份向量集合V中包括的所有用户，以及他们的上层用户的身份向量可以表示为

[0017] $$Pref(V)=\bigcup_{\mathbf{ID}\in V} Pref(\mathbf{ID})$$

[0018] 本发明为一种基于身份的分层广播加密方法，该方法包括初始化模块、数据加密模块、私钥生成模块和解密模块，由该四个模块共11个步骤实现基本功能，如图2所示，各模块按照“初始化模块”→“数据加密模块”→“私钥生成模块”→“解密模块”顺序执行，其步骤如下：

[0019] 模块一：初始化模块

[0020] PKG在这一模块中将系统安全参数λ、用户分层结构的最大深度D和用户数量的最大值n+1作为输入，输出主密钥MSK和公钥PK。公钥PK可以公开，而主密钥MSK则须PKG严格保密。该模块功能的具体实现分为三步：

[0021] 步骤1：PKG首先输入系统安全参数λ，然后运行算法$\mathcal{G}(1^{\lambda})$，输出两个阶数为合数N的群$\mathbb{G}$、$\mathbb{G}_T$和一个双线性映射运算$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$，其中$N=p_1p_2p_3$，$p_1,p_2,p_3$分别是N的三个离散的大素数因子。

[0022] 步骤2：PKG运行随机数生成算法，随机选择阶数为p_1的群$\mathbb{G}_{p_1}$中的一个生成元g，$\mathbb{G}_{p_1}$群中的一个元素h，阶数为p_3的群$\mathbb{G}_{p_3}$中的一个元素X_3以及Z_N：{0，1，...，N-1}域中的一个元素α作为随机指数；并对包括动态虚拟用户在内的所有n+1个用户随机分配$\mathbb{G}_{p_1}$群中的一个元素u_i，$i\in[1,n+1]$。

[0023] 步骤3：最后PKG进行一次合数阶双线性对运算，得到$\mathbb{G}_T$群中的一个元素$e(g,g)^{\alpha}$。经过上述三个步骤得到的参数

[0024] $(g,h,u_1,\ldots,u_{n+1},X_3,e(g,g)^{\alpha})$

[0025] 作为公钥参数可以对外公开，g^{α}作为主密钥由PKG保管。

[0026] 其中，在步骤1中所述的“运行算法$\mathcal{G}(1^{\lambda})$”，其做法如下：PKG根据输入的安全参数λ的大小，选择合适的椭圆曲线：$Y^2=X^3+aX+b$（a和b是系数）。根据所选椭圆曲线上的点构成群$\mathbb{G}$、$\mathbb{G}_T$，选择一种函数映射e，将群$\mathbb{G}$中的元素映射到群$\mathbb{G}_T$中去；安全参数数值越大，所选择椭圆曲线上的点也越多，群也越大。

[0027] 其中，在步骤2中所述的“随机数生成算法”，其做法如下：根据步骤1中所选的椭圆曲线：$y^2=x^3+ax+b$，随机选择自变量x的一个值x_1，计算对应因变量y的值y_1；若点(x_1,y_1)在我们想要映射的群中，则成功生成了随机元素。若点(x_1,y_1)不在群中，则继续选择x的值，直到找到出现在群中的点。下文中涉及到的随机数生成算法，其做法相同。

[0028] 其中，在步骤3中所述的“合数阶双线性对运算”，其做法如下：自变量的输入为群$\mathbb{G}_{p_1}$中的一个生成元g，输出为群$\mathbb{G}_T$中的元素：$e(g,g)^{\alpha}$。

[0029] 模块二：数据加密模块

[0030] 加密方在这一模块中将公钥PK和待加密的消息M以及接收密文用户的身份向量集合V作为输入，输出加密消息M后得到的密文CT。该模块功能的实现分三步：

[0031] 步骤4：加密方随机选择Z_N：{0，1，...，N-1}域中的一个元素作为指数β，完成1次乘法和2次求幂运算，得到

[0032] $(C_0,C_2)\leftarrow(g^{\beta},e(g,g)^{\alpha\beta}\cdot M)$

[0033] 对于接收用户的身份向量集合V，我们定义集合$\mathbb{I}=\{i:ID_i\in S_V\}$。

[0034] 步骤5：加密方运行抗碰撞哈希函数H{·}，计算得到$ID_{n+1} \leftarrow H(C_0, C_2) \in \mathbb{Z}_N$，其运行抗碰撞哈希函数H{·}的计算方法如下：哈希函数的输入是密文C_0、C_2，输出为映射到Z_N：{0,1,...,N-1}域中的元素；该哈希函数可以从Pairing-Based Cryptosystems函数包中的库函数中调用得到；

[0035] 步骤6：加密方完成多项式次乘法和求幂运算，得到C_1，

[0036] $$C_1 \leftarrow \left(h \cdot u_{n+1}^{ID_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{\beta}$$

[0037] 最后的密文输出：CT＝(C_0,C_1,C_2)，该密文是面向用户身份向量集合V∪{(ID_{n+1})}加密的。

[0038] 模块三：私钥生成模块

[0039] 该模块由两部分参与：一部分由PKG承担，模块输入某一用户的身份向量ID和主密钥MSK，生成对应的私钥SK_{ID}。另一部分由待生成私钥的用户的上层用户承担，代理PKG完成用户私钥的生成和分发任务；模块输入该上层用户的私钥$SK_{ID'}$和下层用户的身份ID，输出ID对应的私钥SK_{ID}。

[0040] 由PKG承担的私钥生成功能具体分为两步实现：

[0041] 步骤7：PKG随机选择Z_N：{0,1,...,N-1}域中的一个元素r作为指数，运行随机选择两个$\mathbb{G}_{p_3}$群中的元素A_0,A_1。对于用户的身份向量ID，我们定义集合I＝{i：$ID_i \in S_{ID}$}，并对所有不在集合I中的用户随机分配$\mathbb{G}_{p_3}$群中的一个元素U_j，j∈[1,n+1\I]。

[0042] 步骤8：PKG完成多项式次乘法和求幂运算，得到最后的用户私钥

[0043] $$SK_{\mathbf{ID}} \leftarrow \left(g^{\alpha} \left(h \cdot \prod_{i \in \mathbf{I}} u_i^{ID_i} \right)^{r} A_0, g^{r} A_1, \left\{ u_j^{r} U_j \right\}_{j \in [1,n+1] \setminus \mathbf{I}} \right)$$

[0044] 由上级用户承担的私钥生成功能同样分为两步实现：

[0045] 步骤7*：上级用户首先随机选择Z_N：{0,1,...,N-1}域中的一个元素t作为指数，运行随机数生成算法随机选择两个$\mathbb{G}_{p_3}$群中的元素R_0,R_1，并对所有不在集合I中的用户随机分配$\mathbb{G}_{p_3}$群中的一个元素T_j，j∈[1,n+1]\I。

[0046] 步骤8*：上级用户由自身的私钥$SK_{ID'}$出发，令 $a_0 = g^{\alpha} \left(h \cdot \prod_{i \in \mathbf{I}'} u_i^{ID_i} \right)^{r'} A_{0'}$，$a_1 = g^{r'} A_{1'}$，$\left\{ b_j \right\}_{j \in [1,n+1] \setminus \mathbf{I}'} = \left\{ u_j^{r'} U_{j'} \right\}_{j \in [1,n+1] \setminus \mathbf{I}'}$，经过多项式次乘法和求幂运算可以得到用户身份向量ID＝(ID′,ID)对应的私钥SK_{ID}。

[0047] 模块四：解密模块

[0048] 解密用户在这一模块中，首先运行抗碰撞哈希函数验证密文的有效性，如果密文是有效的，则将接收密文用户的身份向量集合V∪{(ID_{n+1})}、加密消息M得到的密文CT和用户私钥SK_{ID}作为输入。如果满足条件ID∈V，则该模块输出正确的明文消息M。若密文无效，则系统输出NULL(无效符号)。该模块功能的实现具体分为三步：

[0049] 步骤9：解密方首先验证密文的有效性，运行抗碰撞哈希函数H{·}：$ID_{n+1} \leftarrow H(C_0, C_2) \in \mathbb{Z}_N$，检测下列等式是否成立：

[0050] $$e(g,C_1)\overset{?}{=}e\left(C_0,\left(h\cdot u_{n+1}^{ID_{n+1}}\cdot\prod_{i\in\mathbb{I}}u_i^{ID_i}\right)\right)$$

[0051] 若等式成立，进行如下步骤10、11，若不成立，系统输出NULL。

[0052] 其中，所述的“运行抗碰撞哈希函数H{•}”，其运行和计算的方法与步骤5中所述的相同；

[0053] 步骤10：对于满足ID∈Pref(V)条件的用户，由自身的私钥SK_{ID}可首先计算得到

[0054] $$K=a_0\cdot\prod_{j\in\mathbb{I}\setminus\mathrm{I}}b_j^{ID_j}$$

[0055] 步骤11：解密用户根据上步得到的K，进行两次双线性对运算和一次乘法运算，计算输出明文消息M：

[0056] $$M=C_2\frac{e(C_1,a_1)}{e(K,C_0)}$$

[0057] 特别地，上述方案中解密模块的第一步，可以进行公开验证。因为验证的输入为公钥参数和密文，均是对外公开的，因而本发明所述的HIBBE方案可用于构造高级的安全协议。

[0058] 优化模式下的树形用户身份分层结构：本发明涉及的基于身份的分层广播加密方法，当用户分层结构深度为d时，每个用户私钥包括n-d+2个元素，密文包括三个群元素。考虑到某些接收密文用户的数据存储能力受限的情况，我们在私钥长度和密文长度之间做出折衷，提出了优化模式下的树形用户身份分层结构，如图5(b)所示。

[0059] 其中，图5(a)为最原始的系统，所有用户按照一棵树T的结构组织排列，树节点总数为n；图右我们对树T进行分割，分割为$\mathcal{T}$棵子树，每棵子树包含的节点数为n_i个，$i\in[1,\mathcal{T}]$。分割的原则为：①每棵子树的节点数尽量相等，即$n_i\approx n/\mathcal{T}$；②所有子树尽量分享最少的高层节点。每棵子树均可看做独立的HIBBE系统，可正常运行本发明方案中的各功能模块。当广播加密密文时，若用户集合来自不同的子树，则向涉及到的所有访问结构树进行广播。

[0060] 通过上述优化，私钥的长度降至$O(\frac{n}{\mathcal{T}})$数量级，密文的长度略微增加至$O(\mathcal{T})$数量级；若我们令$\mathcal{T}=\sqrt{n}$，则密文和私钥的长度均变为$O(\sqrt{n})$数量级，大大减少了用户的存储负担。

[0061] 3、优点及功效：

[0062] 本发明提供一种基于身份的分层广播加密方法，可用于在复杂的分层网络环境下用户间的保密通信，其优点和功效是：

[0063] 1)该方法首先在已有分层的身份基加密方案基础上，扩展了广播加密功能，可实现对任意用户身份向量集合的加密，仅有在接收者集合中的用户或该类用户的上级用户可以解密；

[0064] 2)本发明方案兼顾了分层结构的优势，即上级用户可代理PKG完成下级用户私钥的生成和分发任务，降低了PKG的工作负担，提高了系统效率；

[0065] 3)通过在分层结构的第一层引入动态虚拟用户节点，将原可抵抗选择身份向量集合和选择明文攻击(IND-CIVS-CPA)的方案转换为可抵抗适应性选择身份向量集合和选择

密文攻击(IND-CIVS-CCA2)的方案,支持密文的有效性公开验证,安全等级更高,可用于构造更高级的协议;

[0066] 4)优化模式下的树形用户身份分层结构的提出,缓解了用户存储长密钥时存储器容量不足问题,使本方案更加灵活多变,实用性更强。

(四)附图说明:

[0067] 图1为一般化的树形用户身份分层结构。

[0068] 图2为本发明所述方法的流程框图。

[0069] 图3为本发明树形用户身份分层结构的一个特例。

[0070] 图4为针对图3的特例,树形结构中的各用户私钥的具体构成。

[0071] 图5(a)为本发明原始系统下的树形用户身份分层结构。

[0072] 图5(b)为本发明优化模式下的树形用户身份分层结构。

[0073] 图中符号说明如下:

[0074] 在图1中,方格的节点代表PKG,斜纹的节点代表在接收用户集合中的用户,白色的节点代表不在接收用户集合中的用户;ID_1,ID_2,...,ID_{11}代表各节点的身份。

[0075] 在图3中,方格的节点代表PKG,斜纹的节点代表在接收密文用户集合V中的用户,白色的节点代表不在接收用户集合中的用户,u_i代表每个身份节点被分配到的随机数,ID_1,ID_2,...,ID_8代表各节点的身份。

[0076] 在图4中,ID_i代表各身份向量,a_0,a_1,b_1,...,b_8代表每个用户私钥的组成部分,Ø表示私钥内容为空。

[0077] 在图5(a)和(b)中,方格的节点代表PKG,斜纹的节点代表在接收用户集合中的用户,白色的节点代表不在接收用户集合中的用户。

(五)具体实施方式

[0078] 见图1-图5(b),主要的数学符号及算法解释:

[0079] (1)合数阶的双线性映射$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$,本发明方案的初始化模块中,通过输入安全系数λ,运行算法$\mathcal{G}(1^\lambda)$,可获得两个阶数为合数N的群$\mathbb{G}$,$\mathbb{G}_T$和一个双线性映射运算$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$,其中$N=p_1p_2p_3$,$p_1$,$p_2$,$p_3$分别是N的三个离散的大素数因子。

[0080] 合数阶的双线性映射满足下述三个特性:

[0081] ①双线性特性:对于$g,h\in\mathbb{G}$,$a,b\in\mathbb{Z}_N$,有$e(g^a,h^b)=e(g,h)^{ab}$成立;

[0082] ②非退化性:$\mathbb{G}$群中至少存在一个元素g,使得计算后的$e(g,g)$为$\mathbb{G}_T$群的某个生成元;

[0083] ③可计算性:存在有效的算法,使得所有的$u,v\in\mathbb{G}$,可以有效计算出$e(u,v)$的值;

[0084] 特别地,$\mathbb{G}$群包含三个阶数为p_1,p_2,p_3的子群,分别表示为$\mathbb{G}_{p_1}$,$\mathbb{G}_{p_2}$和$\mathbb{G}_{p_3}$,对于这三个子群中元素的双线性映射还满足正交特性:

[0085] 对于所有的$h_i\in\mathbb{G}_{p_i}$,$h_j\in\mathbb{G}_{p_j}$,如果$i\neq j$,则有$e(h_i,h_j)=1$。

[0086] (2)抗碰撞哈希算法:$H:\{0,1\}^*\to\mathbb{Z}_N$,本发明中使用的抗碰撞哈希函数具备两个基本特性:单向性和抗碰撞性;单向性是指只能从哈希函数输入推导出输出,而不能从哈希

函数输出计算出输入；抗碰撞性是指不能同时找到两个不同的输入使其哈希结果完全相同。本发明中的哈希算法输入是密文，输出为映射到域Z_N：{0，1，...，N-1}中的元素。

[0087] 本发明为一种基于身份的分层广播加密方法，该方法由初始化模块、数据加密模块、私钥生成模块和解密模块这四个模块实现，见图2。该方法具体实现步骤如下：

[0088] 模块一：初始化模块该模块功能的实现具体分为三步：

[0089] 步骤1：PKG首先输入系统安全参数λ，运行算法$\mathcal{G}(1^{\lambda})$，输出两个阶数为合数N的群$\mathbb{G}$，$\mathbb{G}_T$和一个双线性映射运算$e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$，其中$N=p_1p_2p_3$，$p_1$，$p_2$，$p_3$分别是N的三个离散的大素数因子

[0090] 步骤2：

[0091] $$g\xleftarrow{R}\mathbb{G}_{p_1},\ h\xleftarrow{R}\mathbb{G}_{p_1},\ X_3\xleftarrow{R}\mathbb{G}_{p_3},\ \alpha\xleftarrow{R}\mathbb{Z}_N,\ u_i\xleftarrow{R}\mathbb{G}_{p_1}(i\in[1,n+1]),$$

[0092] PKG运行随机数生成算法，随机选择$\mathbb{G}_{p_1}$群中的一个生成元g，$\mathbb{G}_{p_1}$群中的一个元素h，$\mathbb{G}_{p_3}$群中的一个元素X_3以及$\mathbb{Z}_N$域中的一个元素α作为随机指数；并对包括动态虚拟用户在内的所有(n+1)个用户随机分配$\mathbb{G}_{p_1}$群中的一个元素u_i，$i\in[1,n+1]$，随机分配给真实用户节点的元素为u_i，$i\in[1,n]$，随机分配给动态虚拟用户节点的元素为u_{n+1}。

[0093] 步骤3：PKG进行一次合数阶双线性对运算，得到$\mathbb{G}_T$群中的一个元素$e(g,g)^{\alpha}$。经过上述三个步骤得到的参数

[0094] $$(g,h,u_1,\ldots,u_{n+1},X_3,e(g,g)^{\alpha})$$

[0095] 作为公钥的参数可以对外公开，g^{α}作为主密钥由PKG保管。

[0096] 其中，在步骤1中所述的“运行算法$\mathcal{G}(1^{\lambda})$”，其做法如下：PKG根据输入的安全参数λ的大小，选择合适的椭圆曲线：$Y^2=X^3+aX+b$(a和b是系数)。根据所选椭圆曲线上的点构成群$\mathbb{G}$、$\mathbb{G}_T$，选择一种函数映射e，将群$\mathbb{G}$中的元素映射到群$\mathbb{G}_T$中去；安全参数数值越大，所选择椭圆曲线上的点也越多，群也越大。

[0097] 其中，在步骤2中所述的“随机数生成算法”，其做法如下：根据步骤1中所选的椭圆曲线：$y^2=x^3+ax+b$，随机选择自变量x的一个值x_1，计算对应因变量y的值y_1；若点(x_1,y_1)在我们想要映射的群中，则成功生成了随机元素。若点(x_1,y_1)不在群中，则继续选择x的值，直到找到出现在群中的点。下文中的随机数生成算法原理相同。

[0098] 其中，在步骤3中所述的“合数阶双线性对运算”，其做法如下：自变量的输入为群$\mathbb{G}_{p_1}$中的一个生成元g，输出为群$\mathbb{G}_T$中的元素：$e(g,g)^{\alpha}$。

[0099] 模块二：数据加密模块该模块功能的实现分三步：

[0100] 步骤4：$\beta\xleftarrow{R}\mathbb{Z}_N$，加密方随机选择$Z_N$：{0，1，...，N-1}域中的一个元素作为指数β，完成一次乘法和两次求β幂运算，得到

[0101] $$(C_0,C_2)\leftarrow(g^{\beta},e(g,g)^{\alpha\beta}\cdot M)$$

[0102] 步骤5：加密方运行抗碰撞哈希函数$H\{\cdot\}$，计算得到$ID_{n+1}\leftarrow H(C_0,C_2)\in\mathbb{Z}_N$。

[0103] 步骤6：对于接收密文用户的身份向量集合V，我们令$\mathbb{I}=\{i:ID_i\in S_V\}$。加密方完成多项式次乘法和求幂运算，得到$C_1$，

[0104] $$C_1 \leftarrow \left(h \cdot u_{n+1}^{ID_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{\beta}$$

[0105] 最后的密文输出:CT=(C0,C1,C2),该密文是面向用户身份向量集合V∪{(IDn+1)}加密的。

[0106] 其中,在步骤5中所述的“加密方运行抗碰撞哈希函数H{·},计算得到$ID_{n+1} \leftarrow H(C_0, C_2) \in \mathbb{Z}_N$”,其计算方法如下:哈希函数的输入是密文C0、C2,输出为映射到ZN:{0,1,...,N-1}域中的元素。该哈希函数可以从Pairing-Based Cryptosystems函数包中的库函数中调用得到。

[0107] 模块三:私钥生成模块该模块由两部分参与:

[0108] (一)由PKG承担的私钥生成功能具体分为两步实现:

[0109] 步骤7:$r \overset{R}{\leftarrow} \mathbb{Z}_N$, $A_0, A_1 \overset{R}{\leftarrow} \mathbb{G}_{p_3}$, $U_j \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ ($j \in [1, n+1] \backslash \mathrm{I}$),

[0110] PKG运行随机数生成算法,随机选择$\mathbb{Z}_N$域中的一个元素r作为指数,随机选择两个$\mathbb{G}_{p_3}$群中的元素A0,A1。对于用户的身份向量ID,我们令I={i:IDi∈SID},并对所有不在集合I中的用户随机分配$\mathbb{G}_{p_3}$群中的一个元素Uj,j∈[1,n+1\I]。

[0111] 步骤8:PKG完成多项式次乘法和求幂运算,得到最后的用户私钥

[0112] $$SK_{\mathrm{ID}} \leftarrow \left(g^{\alpha} \left(h \cdot \prod_{i \in \mathrm{I}} u_i^{ID_i} \right)^{r} A_0, g^{r} A_1, \left\{ u_j^{r} U_j \right\}_{j \in [1, n+1] \backslash \mathrm{I}} \right)$$

[0113] 上式中的IDi是通过引入抗碰撞哈希函数,将用户身份IDi∈SID映射到$\mathbb{Z}_N$域中得到的元素,表示为$ID_i \in \mathbb{Z}_N$。如图3中所示的用户分层模型,每个用户的私钥构成如图4中所示。

[0114] (二)由上级用户承担的私钥生成功能同样分为两步实现:

[0115] 步骤7*:$t \overset{R}{\leftarrow} \mathbb{Z}_N$, $R_0, R_1 \overset{R}{\leftarrow} \mathbb{G}_{p_3}$, $T_j \overset{R}{\leftarrow} \mathbb{G}_{p_3}$ ($j \in [1, n+1] \backslash \mathrm{I}$),对于用户的身份向量ID,我们令I={i:IDi∈SID},

[0116] 上级用户首先运行随机数生成算法,随机选择$\mathbb{Z}_N$域中的一个元素t作为指数,随机选择两个$\mathbb{G}_{p_3}$群中的元素R0,R1,并对所有不在集合I中的用户随机分配$\mathbb{G}_{p_3}$群中的一个元素Tj,j∈[1,n+1]\I。

[0117] 步骤8*:令

[0118] $$a_0 = g^{\alpha} \left(h \cdot \prod_{i \in \mathrm{I}'} u_i^{ID_i} \right)^{r'} A_{0'}, \quad a_1 = g^{r'} A_{1'}, \quad \left\{ b_j \right\}_{j \in [1, n+1] \backslash \mathrm{I}'} = \left\{ u_j^{r'} U_{j'} \right\}_{j \in [1, n+1] \backslash \mathrm{I}'}$$

[0119] 上级用户由自身的私钥SKID′=(a0,a1,{bj}j∈[1,n+1]\I′)出发,经过多项式次乘法和求幂运算可以得到用户身份向量ID=(ID′,ID)对应的私钥SKID

[0120] $$SK_{\mathrm{ID}} = \left(a_0 \left(b_i^{ID} \right)_{i \in \mathrm{I} \backslash \mathrm{I}'} \left(h \cdot \prod_{i \in \mathrm{I}} u_i^{ID_i} \right)^{t} R_0, a_1 g^{t} R_1, \left\{ b_j u_j^{t} T_j \right\}_{j \in [1, n+1] \backslash \mathrm{I}} \right)$$

[0121] 通过隐式设置,令$r = r' + t \in \mathbb{Z}_N$, $A_0 = A_{0'} U_{i'} R_0 \in \mathbb{G}_{p_3}$ ($i \in \mathrm{I} \backslash \mathrm{I}'$), $A_1 = A_{1'} R_1 \in \mathbb{G}_{p_3}$和$U_j = U_{j'} T_j \in \mathbb{G}_{p_3}$ ($j \in [1, n] \backslash \mathrm{I}$),用户代理生成的私钥与PKG生成的私钥形式一致,表示为:

[0122] $$SK_{\mathbf{ID}} \leftarrow \left(g^{\alpha} \left(h \cdot \prod_{i \in \mathbf{I}} u_i^{ID_i} \right)^{r} A_0, g^{r} A_1, \left\{ u_j^{r} U_j \right\}_{j \in [1,n] \setminus \mathbf{I}} \right)$$

[0123] 上级用户至此代理完成了等同于PKG的私钥生成任务。

[0124] 如图3所示的特例，分层的用户访问结构中，用户ID_1，ID_3，ID_4，ID_6，ID_7处在集合V中，则他们的用户私钥各部分组成如图4所示。

[0125] 模块四：解密模块该模块功能的实现具体分为三步：

[0126] 步骤9：解密方首先验证密文的有效性，运行抗碰撞哈希函数：$ID_{n+1} \leftarrow H(C_0, C_2) \in \mathbb{Z}_N$，检测下列等式是否成立：

[0127] $$e(g, C_1) \stackrel{?}{=} e\left(C_0, \left(h \cdot u_{n+1}^{ID_{n+1}} \cdot \prod_{i \in \mathbb{I}} u_i^{ID_i} \right) \right)$$

[0128] 若等式成立，进行如下步骤10、11，若不成立，系统输出NULL。

[0129] 步骤10：对于满足$ID \in Pref(V)$条件的用户，由自身的私钥$SK_{ID} = (a_0, a_1, \{b_j\}_{j \in [1,n+1] \setminus \mathbf{I}})$可首先计算得到

[0130] $$K = a_0 \cdot \prod_{j \in \mathbb{I} \setminus \mathbf{I}} b_j^{ID_j} = g^{\alpha} \left(h \cdot \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{r} \cdot \left(A_0 \prod_{j \in \mathbb{I} \setminus \mathbf{I}} U_j^{ID_j} \right)$$

[0131] 其中，$\mathbf{I} = \{i : ID_i \in S_{ID}\}$，$\mathbb{I} = \{i : ID_i \in S_V\}$

[0132] 步骤11：解密用户根据上步得到的K，进行两次双线性对运算和一次乘法运算，计算输出明文消息M：

[0133] $$M = C_2 \frac{e(C_1, a_1)}{e(K, C_0)}$$

[0134] 基于双线性对的正交特性，所有$\mathbb{G}_{p_3}$群中的元素均可与$\mathbb{G}_{p_1}$群中的元素通过双线性映射消掉：

[0135] $$\frac{e(C_1, a_1)}{e(K, C_0)} = \frac{e\left(\left(h \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{\beta}, g^{r} A_1 \right)}{e\left(g^{\alpha} \left(h \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{r} \cdot \left(A_0 \prod_{j \in \mathbb{I} \setminus \mathbf{I}} U_j \right), g^{\beta} \right)} = \frac{e\left(\left(h \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{\beta}, g^{r} \right)}{e\left(g^{\alpha} \left(h \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{r}, g^{\beta} \right)}$$

[0136] $$= \frac{e\left(\left(h \prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{\beta}, g^{r} \right)}{e\left(g^{\alpha}, g^{\beta} \right) e\left(h \left(\prod_{i \in \mathbb{I}} u_i^{ID_i} \right)^{r}, g^{\beta} \right)} = \frac{1}{e(g, g)^{\alpha\beta}}$$

[0137] 即

[0138] $$M = C_2 \frac{e(C_1, a_1)}{e(K, C_0)} = e(g, g)^{\alpha\beta} M \frac{1}{e(g, g)^{\alpha\beta}} = M$$

[0139] 其中，步骤9中的“抗碰撞哈希函数”运行方法与步骤5中的相同。

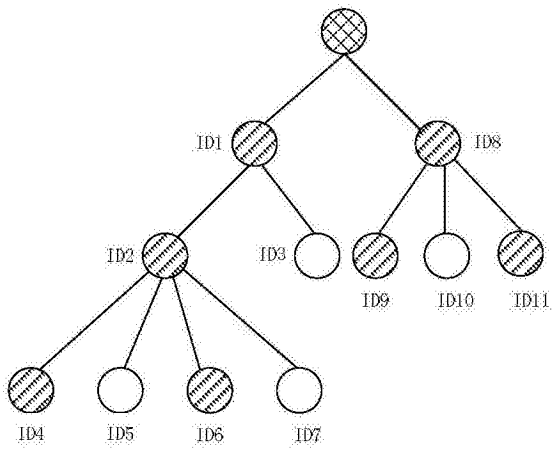

ID1
ID8
ID2
ID3
ID9
ID10
ID11
ID4
ID5
ID6
ID7

图1

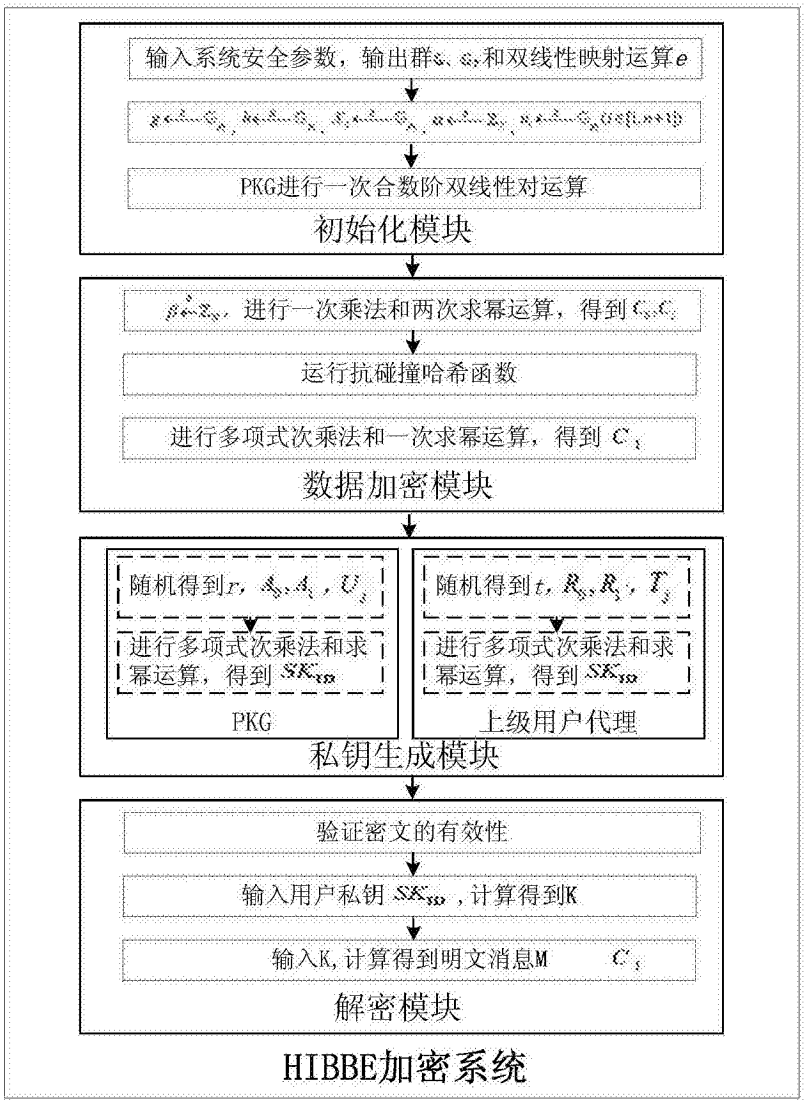

输入系统安全参数，输出群
和双线性映射运算
PKG进行一次合数阶双线性对运算
初始化模块
进行一次乘法和两次求幂运算，得到
运行抗碰撞哈希函数
进行多项式次乘法和一次求幂运算，得到
数据加密模块
随机得到
进行多项式次乘法和求幂运算，得到
PKG
随机得到
进行多项式次乘法和求幂运算，得到
上级用户代理
私钥生成模块
验证密文的有效性
输入用户私钥
，计算得到K
输入K,计算得到明文消息M
解密模块
HIBBE加密系统

图2

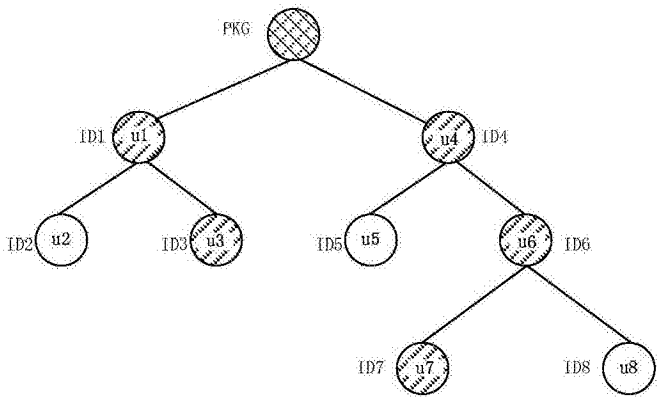


图3

$$\begin{pmatrix} & a_0 & a_1 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 & b_8 \\ ID_1: & g^{\alpha}\left(h\cdot u_1^{ID_1}\right)^{r_1} & g^{r_1} & \varnothing & u_2^{r_1} & u_3^{r_1} & u_4^{r_1} & u_5^{r_1} & u_6^{r_1} & u_7^{r_1} & u_8^{r_1} \\ ID_2: & g^{\alpha}\left(h\cdot u_1^{ID_1}u_2^{ID_2}\right)^{r_2} & g^{r_2} & \varnothing & \varnothing & u_3^{r_2} & u_4^{r_2} & u_5^{r_2} & u_6^{r_2} & u_7^{r_2} & u_8^{r_2} \\ ID_3: & g^{\alpha}\left(h\cdot u_1^{ID_1}u_3^{ID_3}\right)^{r_3} & g^{r_3} & \varnothing & u_2^{r_3} & \varnothing & u_4^{r_3} & u_5^{r_3} & u_6^{r_3} & u_7^{r_3} & u_8^{r_3} \\ ID_4: & g^{\alpha}\left(h\cdot u_4^{ID_4}\right)^{r_4} & g^{r_4} & u_1^{r_4} & u_2^{r_4} & u_3^{r_4} & \varnothing & u_5^{r_4} & u_6^{r_4} & u_7^{r_4} & u_8^{r_4} \\ ID_5: & g^{\alpha}\left(h\cdot u_4^{ID_4}u_5^{ID5}\right)^{r_5} & g^{r_5} & u_1^{r_5} & u_2^{r_5} & u_3^{r_5} & \varnothing & \varnothing & u_6^{r_5} & u_7^{r_5} & u_8^{r_5} \\ ID_6: & g^{\alpha}\left(h\cdot u_4^{ID_4}u_6^{ID6}\right)^{r_6} & g^{r_6} & u_1^{r_6} & u_2^{r_6} & u_3^{r_6} & \varnothing & u_5^{r_6} & \varnothing & u_7^{r_6} & u_8^{r_6} \\ ID_7: & g^{\alpha}\left(h\cdot u_4^{ID_4}u_6^{ID6}u_7^{ID_7}\right)^{r_7} & g^{r_7} & u_1^{r_7} & u_2^{r_7} & u_3^{r_7} & \varnothing & u_5^{r_7} & \varnothing & \varnothing & u_8^{r_7} \\ ID_8: & g^{\alpha}\left(h\cdot u_4^{ID_4}u_6^{ID6}u_8^{ID_8}\right)^{r_8} & g^{r_8} & u_1^{r_8} & u_2^{r_8} & u_3^{r_8} & \varnothing & u_5^{r_8} & \varnothing & u_7^{r_8} & \varnothing \end{pmatrix}$$

图4

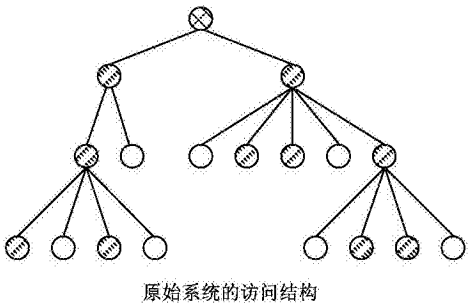


图5 (a)

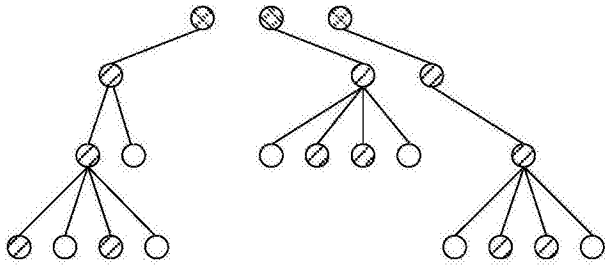


图5(b)