(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107819578 A
(43)申请公布日 2018.03.20

(21)申请号 201711322300.1

(22)申请日 2017.12.12

(71)申请人 电子科技大学
地址 611731 四川省成都市高新区(西区)西源大道2006号

(72)发明人 邓伏虎 王亚丽 熊虎 耿技

(74)专利代理机构 成都正华专利代理事务所(普通合伙) 51229
代理人 何凡

(51)Int.Cl.
H04L 9/08(2006.01)
H04L 9/32(2006.01)

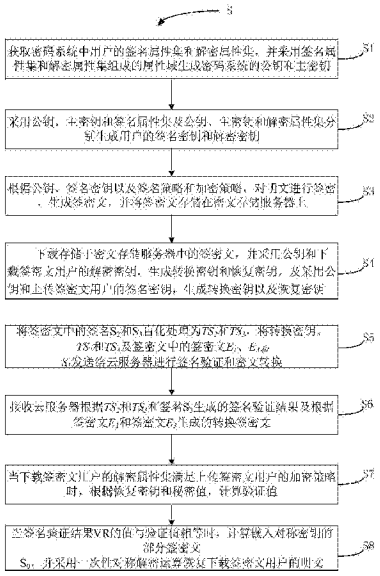
权利要求书5页 说明书7页 附图2页

(54)发明名称

基于属性的可验证外包解签密方法及其系统

(57)摘要

本发明公开了一种基于属性的可验证外包解签密方法,其包括采用签名属性集和解密属性集组成的属性域生成密码系统的公钥和主密钥,之后生成用户的签名密钥和解密密钥;对明文进行签密,生成签密文,并将签密文存储在密文存储服务器上;下载签密文,并生成转换密钥和恢复密钥;对签密文中的签名进行盲化处理,之后将转换密钥、部分签密文和盲化后的签名发送给云服务器;接收云服务器的签名验证结果及转换签密文;当下载签密文用户的解密属性集满足上传签密文用户的加密策略时,根据恢复密钥和秘密值,计算验证值;当签名验证结果的值与验证值相等时,计算嵌入对称密钥的部分签密文,并采用一次性对称解密运算恢复下载签密文用户的明文。



CN 107819578 A

1.基于属性的可验证外包解签密方法，其特征在于，包括：

S1、获取密码系统中用户的签名属性集θ_s和解密属性集θ_d，并采用签名属性集θ_s和解密属性集θ_d组成的属性域A生成密码系统的公钥PK和主密钥MSK；

S2、采用公钥PK、主密钥MSK和签名属性集θ_s及公钥PK、主密钥MSK和解密属性集θ_d分别生成用户的签名密钥SK_{θ_s}和解密密钥SK_{θ_d}；

S3、根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文$SCT_{xe}=(x_e,E_1,E_2,E_3,E_4,S_0,S_1,S_2,S_3,tt)$，并将签密文$SCT_{xe}$存储在密文存储服务器上；

S4、下载存储于密文存储服务器中的签密文SCT_{xe}，并采用公钥PK和下载签密文SCT_{xe}用户的解密密钥SK_{θ_d}，生成转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}，及采用公钥PK和上传签密文SCT_{xe}用户的签名密钥SK_{θ_s}，生成转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}；

S5、将签密文SCT_{xe}中的签名S_2和S_3盲化处理为TS_2和TS_3，之后将转换密钥TSK_{θ_d}和TSK_{θ_s}，盲化后的签名TS_2和TS_3及签密文SCT_{xe}中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换；

S6、接收云服务器根据TS_2、TS_3和签名S_3生成的签名验证结果$VR=Y^s\cdot e(g,g)^{t'}$及根据签密文E_1和签密文E_3生成的转换签密文$TCT=Y^{t\cdot\varepsilon}$；

S7、当下载签密文SCT_{xe}用户的解密属性集θ_d满足上传签密文SCT_{xe}用户的加密策略x_e时，根据恢复密钥RSK_{θ_d}、恢复密钥RSK_{θ_s}和秘密值t′，采用$Y^s\cdot e(g,g)^{t'}$计算验证值；

S8、当签名验证结果VR的值与验证值相等时，计算嵌入对称密钥的部分签密文$S_0=u'^{H(SSK)}v'^{H(d)}$，并采用一次性对称解密运算$SDec(SEK,E_2)$=M恢复下载签密文SCT_{xe}用户的明文M。

2.根据权利要求1所述的基于属性的可验证外包解签密方法，其特征在于，所述公钥PK和主密钥MSK的生成方法包括：

S11、选取两个阶为q的乘法循环群$\mathbb{G}_1$和乘法循环群$\mathbb{G}_2$，构建双线性映射函数$e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$；

S12、构造三个抗碰撞的哈希函数：

$$H_1:\{0,1\}^*\to\{0,1\}^l;\quad H_2:\mathbb{G}_1\to\mathbb{Z}_q^*;\quad H_3:\{0,1\}^*\to\mathbb{Z}_q^*$$

其中，H_1为将任意长度的比特串通过哈希函数映射到长度为l的比特串上；H_2为将乘法循环群$\mathbb{G}_1$上的元素映射到阶为q的整数乘法群$\mathbb{Z}_q^*$上；H_3为将任意长度的比特串通过哈希函数映射到阶为q的整数乘法群$\mathbb{Z}_q^*$上；

S13、采用多个随机数来自于乘法循环群$\mathbb{G}_1$构建公钥PK和主密钥MSK：$PK=(D,Y,v,u',v',\eta_1,\eta_2,\mu_0,\{\mu_i\}_{i\in[i]},\{h_\omega\}_{\omega\in A},H_1,H_2,H_3,\Pi_{SE},KDF,S,A)$，$MSK=g^a$

其中，h_ω、v、u'、v'、η_1、η_2、μ_0、μ_1、μ_2、...μ_ι为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；$D=(q,\mathbb{G}_1,\mathbb{G}_2,e)$，其为双线性群；$Y=e(g,g)^a$，$e(g,g)$为双线性对运算，g表示生成元，a为从乘法整数乘法群$\mathbb{Z}_q^*$中选取随机数；$\Pi_{SE}=(SEnc,SDec)$为一次性对称加密运算；$S=\{0,1\}^*$，其为明文空间；KDF为长度是ι的密钥获取函数；A为属性域。

3.根据权利要求2所述的基于属性的可验证外包解签密方法，其特征在于，所述解密密钥SK_{θ_d}的生成方法包括：

S21、从整数乘法群$\mathbb{Z}_q^*$中选取随机数β；

S22、采用随机数γ及公钥PK和主密钥MSK中的参数计算解密密钥SK_{θ_d}的组成元素K_d，K'_d和$K_{d,\omega}$：

$$K_d = g^{\alpha}v^{\beta}, K'_d = g^{\beta}, K_{d,\omega} = h_{\omega}^{\beta}, \forall \omega \in \theta_d$$

S23、采用K_d，K'_d，$K_{d,\omega}$和θ_d构成解密密钥$SK_{\theta_d} = (\theta_d, K_d, K'_d, \{K_{d,\omega}\}_{\omega \in \theta_d})$；

所述签名密钥SK_{θ_s}的生成方法包括：

S24、从乘法整数乘法群$\mathbb{Z}_q^*$中选取随机数γ；

S25、采用随机数γ及公钥PK和主密钥MSK中的参数计算签名密钥SK_{θ_s}的组成元素K_s，K'_s和$K_{s,\omega}$：

$$K_s = g^{\alpha}v^{\gamma}, K'_s = g^{\gamma}, K_{s,\omega} = h_{\omega}^{\gamma}, \forall \omega \in \theta_s$$

S26、采用K_s，K'_s，$K_{s,\omega}$和θ_s构成解密密钥$SK_{\theta_s} = (\theta_s, K_s, K'_s, \{K_{s,\omega}\}_{\omega \in \theta_s})$。

4.根据权利要求3所述的基于属性的可验证外包解签密方法，其特征在于，所述根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文SCT_{x_e}=(x_e,E_1,E_2,E_3,E_4,S_0,S_1,S_2,S_3,tt)，进一步包括：

S41、获取数据拥有者的签名属性集θ_s、签名密钥SK_{θ_s}、签名策略x_s和加密策略x_e；

S42、当签名属性集θ_s满足签名策略x_s时，则对明文M进行签名，并记录当前时间为tt；

S43、从整数乘法群$\mathbb{Z}_q^*$中选取随机数a′，对签名密钥SK_{θ_s}中的K_s，K'_s和$K_{s,\omega}$进行盲化处理：

$$K_s^R = K_s \cdot v^{\alpha'}, K_s^{R'} = K'_s \cdot g^{\alpha'} K_{s,\omega}^R = K_{s,\omega} \cdot h_{\omega}^{\alpha'}, \forall \omega \in \theta_s$$

S44、计算签密文SCT_{xe}的组成元素E_1，E_2，E_3，E_4，S_0，S_1，S_2和S_3：

$E_1 = g^{\varepsilon}$，其中，g为生成元，ε为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；

E_2=SEnc(SEK||d,M)，E_2为采用一次对称加密算法SEnc对明文M进行加密；SEK=Y^{ε}||S_1||tt为对称封装密钥，d为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；

$E_3 = \{E_3^{(i)} = v^{\alpha - \vec{\alpha}\vec{M}_e^{(i)}} h_{\varphi_e(i)}^{\varepsilon}\}_{i \in [1,l_e]}$，其中，$\vec{\alpha} = (\varepsilon, \delta_2, \ldots, \delta_{n_e})$，其组元素为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；$\vec{M}_e^{(i)}$为矩阵$M_e$的第i行，ε为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数，$\varphi_e(i)$为行标签函数的第i行，$l_e$为矩阵$M_e$的行数；

$E_4 = (\eta_1 \eta_2^{\varrho})^{\varepsilon}$，其中，$\varrho = H_2(E_1)$，其为对$E_1$作哈希映射；

S_0=$u'^{H(SEK)}v'^{H(d)}$，其中，H(•)为密码单向哈希函数；

$S_1 = g^{\omega_s}$；$S_2 = \{S_2^{(i)} = g^{\omega_i}(K_s^{R'})^{c_i}\}_{i \in [1,l_s]}$，其中，$d_i$、$c_i$均为从整数乘法群$\mathbb{Z}_q^{l_s}$中选取的向量；$l_s$为矩阵$M_s$的行数；$M_s$为$l_s$行$n_s$列的矩阵；

$S_3 = K_s^R(\prod_{i \in [l_s]}(K_{s,\varphi_s(i)}^R)^{\omega_i} \cdot h_{\varphi_s(i)}^{d_i})(\mu_0 \prod_{i \in [l]} \mu_i^{f_i})^{\varepsilon} E_1^{\psi\delta}$，其中，ψ=$H_3$(•)，其为耐碰撞哈希函数；($f_1$,...,$f_l$)∈$\{0,1\}^l$=$H_1$(•)表示耐碰撞哈希函数；

S45、采用加密访问策略xe，组成元素E_1，E_2，E_3，E_4，S_0，S_1，S_2和S_3及当前时间为tt构成签密文$SCT_{xe}=(xe,E_1,E_2,E_3,E_4,S_0,S_1,S_2,S_3,tt)$。

5.根据权利要求2所述的基于属性的可验证外包解签密方法，其特征在于，所述转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}的生成方法包括：

S41、从q阶的整数乘法群$\mathbb{Z}_q^*$中选取随机数t，并采用随机数t构成恢复密钥$RSK_{\theta_d}=t$；

S42、根据公钥PK和主密钥MSK中的参数，计算转换密钥

$TSK_{\theta_d}=(TK_d,TK'_d,\{TK_{d,\omega}\}_{\omega\in\theta_d})$：$TK_d=g^{\alpha}v^{at}$，$TK'_d=g^{at}$，$TK_{d,\omega}=h_{\omega}^{at}$，$\forall\omega\in\theta_d$；

所述转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}的生成方法包括：

S43、从q阶的整数乘法群$\mathbb{Z}_q^*$中选取随机数s，并采用随机数s构成恢复密钥$RSK_{\theta_s}=s$；

S44、根据公钥PK和主密钥MSK中的参数，计算转换密钥

$TSK_{\theta_s}=(\theta_s,TK_s,TK'_s,\{TK_{s,\omega}\}_{\omega\in\theta_s})$：$TK_s=g^{\alpha}v^{as}$，$TK'_s=g^{as}$，$TK_{s,\omega}=h_{\omega}^{as}$，$\forall\omega\in\theta_s$。

6.根据权利要求4所述的基于属性的可验证外包解签密方法，其特征在于，所述将签密文SCT_{xe}中的签名S_2和S_3盲化处理为TS_2和TS_3，进一步包括：

S51、采用恢复密钥RSK_{θ_s}重新随机化转换密钥TSK_{θ_s}：

$TK_s^R=TK_s\cdot v^{a'}$，$TK_s'^R=TK'_s\cdot g^{a'}$，$TK_{s,\omega}^R=TK_{s,\omega}\cdot h_{\omega}^{a'}$，$\forall\omega\in\theta_s$；

S52、从$\mathbb{Z}_q^*$中选取随机数t′，并采用恢复密钥RSK_{θ_s}=s将签名S_3盲化为$TS_3=S_3^s\cdot g^{t'}$，将签名S_2盲化为$TS_2=TS_2^{(i)}=g^{d_i+T\cdot c_i}$。

7.根据权利要求5所述的基于属性的可验证外包解签密方法，其特征在于，所述云服务器采用以下式子计算签名验证结果VR：

$$\frac{e(TS_3,g)}{(\prod_{i\in[l_s]}e(v^{m_i}\cdot h_{\varphi_s(i)},TS_2^{(i)}))\cdot e(\mu_0\prod_{i\in[l]}\mu_i^{f_i},E_1)e((\eta_1\eta_2^{\theta})^{\psi},S_1)}$$

采用以下式子进行密文转换：

$$\frac{e(TK_d\cdot\prod_{i\in[l_e]}TK_{i,\varphi_e(i)}^{k'_i},E_1)}{e(TK'_d,\prod_{i\in[l_e]}(E_3^{(i)})^{k'_i})}$$。

8.根据权利要求4所述的基于属性的可验证外包解签密方法，其特征在于，当下载签密文SCT_{xe}用户的解密属性集θ_d满足加密策略xe时，还包括下载签密文SCT_{xe}用户采用本地服务器进行解签密处理：

当当前解签密时间tt′与签密时间tt差异的绝对值小于等于解签密时间时，通过如下等式验证签名S_3是否有效：

$$Y\overset{?}{=}\frac{e(S_3,g)}{(\prod_{i\in[l_s]}e(v^{m_i}\cdot h_{\varphi_s(i)},S_2^{(i)}))\cdot e(\mu_0\prod_{i\in[l]}\mu_i^{f_i},E_1)\cdot e((\eta_1\eta_2^{\theta})^{\psi},S_1)}$$

其中，向量$\vec{w}=(1,x_2,\ldots,x_{n_s})\cdot\vec{M}_s^{(i)},\forall i\in[l_s]$，随机数$x_2,\ldots,x_{n_s}$为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；$f_i=(f_1,\ldots,f_l)\in\{0,1\}^l=H_1(S_2,tt,xs,xe)$；令$T=\gamma+d'$，$\sum_{i\in[l_s]}(T\cdot c_i+d_i)\cdot\vec{w}_i=(T,Tx_2,\ldots,Tx_{n_s})\cdot(1,0,\ldots,0)+(1,x_2,\ldots,x_{n_s})\cdot(0,0,\ldots,0)=T$；

若签名S_3有效，且当下载签密文SCT_{x_e}用户的解密属性集θ_d满足上传签密文SCT_{xe}用户的加密策略xe时，恢复部分对称密钥Y^{ε}：

$$\frac{e(K_d\cdot\prod_{i\in[l_e]}K'^{k'_i}_{d\varphi_e(i)},\varepsilon_1)}{e(K'_d,\prod_{i\in[l_e]}(\varepsilon_3^{(i)})^{k'_i})}=Y^{\varepsilon}$$

其中，$\vec{k'}=(k'_1,k'_2,\ldots,k'_{l_e})$为从$\mathbb{Z}_q^{l_e}$中选取向量；

$\sum_{i\in[l_e]}k'_i\cdot(\vec{\alpha}\cdot\vec{M}_e^{(i)})=(\varepsilon,\delta_2,\ldots,\delta_{n_e})\cdot(1,0,\ldots,0)=\varepsilon$，

$\vec{k'}\cdot\mathbf{M}_e=\vec{1}_{n_e}$，$\sum_{i\in[l_e]}\vec{k}'_i\cdot\vec{M}_e^{(i)}=\vec{1}_{n_e}$，$\mathbb{Z}_q^{l_e}$为长度为$l_e$的整数群；$M_e$为$l_e$行$n_e$列的矩阵；$\vec{M}_e^{(i)}$为矩阵$M_e$的第i行；对于任意的i，当$\varphi_{e(i)}\notin\theta_d$时，$k'_i=0$，$\varphi_e$为行标签函数；

采用一次性对称解密运算$SDec(SEK,E_2)=M$恢复下载签密文SCT_{xe}用户的明文M。

9.一种用于权利要求1-8任一所述的基于属性的可验证外包解签密方法的解签密系统，其特征在于，包括：

授权中心，用于获取密码系统中用户的签名属性集θ_s和解密属性集θ_d，并采用签名属性集θ_s和解密属性集θ_d组成的属性域A生成密码系统的公钥PK和主密钥MSK；

采用公钥PK、主密钥MSK和签名属性集θ_s及公钥PK、主密钥MSK和解密属性集θ_d分别生成用户的签名密钥SK_{θ_s}和解密密钥SK_{θ_d}；

上传签密文的用户服务器，用于根据公钥PK、签名密钥SK_{θ_s}以及签名策略xs和加密策略xe，对明文M进行签密，生成签密文$SCT_{xe}=(xe,E_1,E_2,E_3,E_4,S_0,S_1,S_2,S_3,tt)$，并将签密文$SCT_{xe}$存储在密文存储服务器上；

下载签密文的用户服务器，下载存储于密文存储服务器中的签密文SCT_{xe}，并采用公钥PK和下载签密文SCT_{xe}用户的解密密钥SK_{θ_d}，生成转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}，及采用公钥PK和下载签密文SCT_{xe}用户的签名密钥SK_{θ_s}，生成转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}；

将签密文SCT_{xe}中的签名S_2和S_3盲化处理为TS_2和TS_3，之后将转换密钥TSK_{θ_d}和TSK_{θ_s}，盲化后的签名TS_2和TS_3及签密文SCT_{xe}中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换；

接收云服务器根据TS_2、TS_3和签名S_3产生的签名验证结果$VR=Y^s\cdot e(g,g)^{t'}$及根据签密文E_1和签密文E_3产生的转换签密文$TCT=Y^{t\cdot\varepsilon}$；

当下载签密文SCT_{xe}用户的解密属性集θ_d满足上传签密文SCT_{xe}用户的加密策略xe时，根据恢复密钥RSK_{θ_d}、恢复密钥RSK_{θ_s}和秘密值t'，采用$Y^s\cdot e(g,g)^{t'}$计算验证值；

当签名验证结果VR的值与验证值相等时，计算签名$S_0=u'^{H(SSK)}v'^{H(d)}$，并采用一次性对

称解密运算SDec(SEK,E_2)＝M恢复下载签密文SCT_{xe}用户的明文M;

云服务器,用于接收下载签密文的用户服务器上传的转换密钥TSK_{θ_a}、TSK_{θ_x},盲化后的签名TS_2和TS_3及签密文SCT_{xe}中的签密文E_1、E_3和S_1,并根据接收的数据进行签名验证和密文转换。

基于属性的可验证外包解签密方法及其系统

技术领域

[0001] 本发明涉及云环境下的外包服务领域及网络安全的数据加密和隐私保护领域，具体涉及一种基于属性的可验证外包解签密方法及其系统。

背景技术

[0002] 随着云计算的迅速发展，许多公司和个人都选择利用云服务器存储和共享数据。由于云服务器具有较强的计算能力，他们便考虑将大量复杂的计算外包给云服务器。以在线社交网络为例，由于建立和维护社交网络数据的成本高昂，许多社交服务被外包给第三方提供者。但是在外包过程中，最关键的问题是如何保障社交信息的安全性和隐私性。云服务器既可访问用户上传的数据，也可能泄露和篡改社交信息。这对于利用云服务存储、分享社交信息是一个极大的挑战。对此，现有技术中提出了一种基于云的PHR共享系统的密文策略的属性基签密方案，该方案虽然能够保障PHR信息的安全性和隐私性，但是在验证签名和解密阶段的计算成本增加了PHR用户的计算开销和通信开销。

发明内容

[0003] 针对现有技术中的上述不足，本发明提供了一种将签密文中的签名验证和解密过程中复杂的计算外包给云服务器处理的基于属性的可验证外包解签密方法及其系统。

[0004] 为了达到上述发明目的，本发明采用的技术方案为：

[0005] 第一方面，提供基于属性的可验证外包解签密方法，其包括：

[0006] S1、获取密码系统中用户的签名属性集θ_s和解密属性集θ_d，并采用签名属性集θ_s和解密属性集θ_d组成的属性域A生成密码系统的公钥PK和主密钥MSK；

[0007] S2、采用公钥PK、主密钥MSK和签名属性集θ_s及公钥PK、主密钥MSK和解密属性集θ_d分别生成用户的签名密钥SK_{θ_s}和解密密钥SK_{θ_d}；

[0008] S3、根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文$SCT_{x_e}=(\chi_e,E_1,E_2,E_3,E_4,S_0,S_1,S_2,S_3,t)$，并将签密文$SCT_{x_e}$存储在密文存储服务器上；

[0009] S4、下载存储于密文存储服务器中的签密文SCT_{x_e}，并采用公钥PK和下载签密文SCT_{x_e}用户的解密密钥SK_{θ_d}，生成转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}，及采用公钥PK和上传签密文SCT_{x_e}用户的签名密钥SK_{θ_s}，生成转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}；

[0010] S5、将签密文SCT_{x_e}中的签名S_2和S_3盲化处理为TS_2和TS_3，之后将转换密钥TSK_{θ_d}和TSK_{θ_s}，盲化后的签名TS_2和TS_3及签密文SCT_{x_e}中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换；

[0011] S6、接收云服务器根据TS_2、TS_3和签名S_3生成的签名验证结果$VR=Y^s\cdot e(g,g)^{t'}$及根据签密文E_1和签密文E_3生成的转换签密文$TCT=Y^{t\cdot\varepsilon}$；

[0012] S7、当下载签密文SCT_{x_e}用户的解密属性集θ_d满足上传签密文SCT_{x_e}用户的加

密策略x_e时，根据恢复密钥RSK_{θ_d}、恢复密钥RSK_{θ_s}和秘密值t′，采用$Y^s \cdot e(g,g)^{t'}$计算验证值；

[0013] S8、当签名验证结果VR的值与验证值相等时，计算签密文$S_0 = u'^{H(SSK)}v'^{H(d)}$，并采用一次性对称解密运算SDec(SEK,$E_2$)＝M恢复下载签密文$SCT_{x_e}$用户的明文M。

[0014] 第二方面，提供一种基于属性的可验证外包解签密系统，其包括：

[0015] 授权中心，用于获取密码系统中用户的签名属性集θ_s和解密属性集θ_d，并采用签名属性集θ_s和解密属性集θ_d组成的属性域A生成密码系统的公钥PK和主密钥MSK；

[0016] 采用公钥PK、主密钥MSK和签名属性集θ_s及公钥PK、主密钥MSK和解密属性集θ_d分别生成用户的签名密钥SK_{θ_s}和解密密钥SK_{θ_d}；

[0017] 上传签密文的用户服务器，用于根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文$SCT_{x_e} = (\chi_e, E_1, E_2, E_3, E_4, S_0, S_1, S_2, S_3, tt)$并将签密文$SCT_{x_e}$存储在密文存储服务器上；

[0018] 下载签密文的用户服务器，下载存储于密文存储服务器中的签密文SCT_{x_e}，并采用公钥PK和下载签密文SCT_{x_e}用户的解密密钥SK_{θ_d}，生成转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}，及采用公钥PK和下载签密文SCT_{x_e}用户的签名密钥SK_{θ_s}，生成转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}；

[0019] 将签密文SCT_{x_e}中的签名S_2和S_3盲化处理为TS_2和TS_3，之后将转换密钥TSK_{θ_d}和TSK_{θ_s}，盲化后的签名TS_2和TS_3及签密文SCT_{x_e}中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换；

[0020] 接收云服务器根据TS_2、TS_3和签名S_3生成的签名验证结果VR＝$Y^s \cdot e(g,g)^{t'}$及根据签密文E_1和签密文E_3生成的转换签密文TCT＝$Y^{t \cdot \varepsilon}$；

[0021] 当下载签密文SCT_{x_e}用户的解密属性集θ_d满足上传签密文SCT_{x_e}用户的加密策略x_e时，根据恢复密钥RSK_{θ_d}、恢复密钥RSK_{θ_s}和秘密值t′，采用$Y^s \cdot e(g,g)^{t'}$计算验证值；

[0022] 当签名验证结果VR的值与验证值相等时，计算嵌入对称密钥的部分签密文S_0＝$u'^{H(SSK)}v'^{H(d)}$并采用一次性对称解密运算SDec(SEK,E_2)＝M恢复下载签密文SCT_{x_e}用户的明文M；

[0023] 云服务器，用于接收下载签密文的用户服务器上传的转换密钥TSK_{θ_d}和TSK_{θ_s}，盲化后的签名TS_2和TS_3及签密文SCT_{x_e}中的签密文E_1、E_3和S_1，并根据接收的数据进行签名验证和密文转换。

[0024] 本发明的有益效果为：本方案将属性基签密技术与外包技术相结合，在将签密文中的签名S_2和S_3盲化处理后，通过这种方式在保证信息的安全性和隐私性的同时，可以有效地防止用户诬陷云服务器提供错误的解签密服务；

[0025] 本方案由于采用外包方式进行解签密操作，不仅降低了用户端的解签密的计算成本，同时也保证了云服务器帮助用户解密和验证签名过程中部分签密文信息的正确性；另外，本方案尤其适用于带宽、资源受限的移动设备的使用。

附图说明

[0026] 图1为基于属性的可验证外包解签密方法一个实施例的流程图。

[0027] 图2为基于属性的可验证外包解签密系统的原理框图。

具体实施方式

[0028] 下面对本发明的具体实施方式进行描述，以便于本技术领域的技术人员理解本发明，但应该清楚，本发明不限于具体实施方式的范围，对本技术领域的普通技术人员来讲，只要各种变化在所附的权利要求限定和确定的本发明的精神和范围内，这些变化是显而易见的，一切利用本发明构思的发明创造均在保护之列。

[0029] 参考图1，图1示出了基于属性的可验证外包解签密方法一个实施例的流程图；如图1所示，该方法S包括步骤S1至步骤S7。

[0030] 在步骤S1中，获取密码系统中用户的签名属性集θ_s和解密属性集θ_d，并采用签名属性集θ_s和解密属性集θ_d组成的属性域A生成密码系统的公钥PK和主密钥MSK。

[0031] 实施时，本方案优选所述公钥PK和主密钥MSK的生成方法包括：

[0032] S11、选取两个阶为q的乘法循环群$\mathbb{G}_1$和乘法循环群$\mathbb{G}_2$，构建双线性映射函数$e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$；

[0033] S12、构造三个抗碰撞的哈希函数：

[0034] $H_1:\{0,1\}^*\to\{0,1\}^l;\ H_2:\mathbb{G}_1\to\mathbb{Z}_q^*;\ H_3:\{0,1\}^*\to\mathbb{Z}_q^*$

[0035] 其中，H_1为将任意长度的比特串通过哈希函数映射到长度为l的比特串上；H_2为将乘法循环群$\mathbb{G}_1$上的元素映射到阶为q的整数乘法群$\mathbb{Z}_q^*$上；H_3为将任意长度的比特串通过哈希函数映射到阶为q的整数乘法群$\mathbb{Z}_q^*$上；

[0036] S13、采用多个随机数来自于乘法循环群$\mathbb{G}_1$构建公钥PK和主密钥MSK：

$PK=(D,Y,\nu,u',v',\eta_1,\eta_2,\mu_0,\{\mu_i\}_{i\in[l]},\{h_\omega\}_{\omega\in A},H_1,H_2,H_3,\Pi_{SE},KDF,S,A),\ MSK=g^\alpha$

[0037] 其中，$h_\omega,\nu,u',v',\eta_1,\eta_2,\mu_0,\mu_1,\mu_2,\ldots\mu_l$为从整数乘法群$\mathbb{Z}_q^*$中的选取的随机数；$D=(q,\mathbb{G}_1,\mathbb{G}_2,e)$，其为双线性群；$Y=e(g,g)^a$，$e(g,g)$为双线性对运算，g表示生成元，a为从乘法整数乘法群$\mathbb{Z}_q^*$中选取随机数；$\Pi_{SE}=(SEnc,SDec)$为一次性对称加密运算；$S=\{0,1\}^*$，其为明文空间；KDF为长度是l的密钥获取函数；A为属性域。

[0038] 在步骤S2中，采用公钥PK、主密钥MSK和签名属性集θ_s及公钥PK、主密钥MSK和解密属性集θ_d分别生成用户的签名密钥SK_{θ_s}和解密密钥SK_{θ_d}；

[0039] 其中，所述解密密钥SK_{θ_d}的生成方法包括：

[0040] S21、从乘法整数乘法群$\mathbb{Z}_q^*$中选取随机数β；

[0041] S22、采用随机数γ及公钥PK和主密钥MSK中的参数计算解密密钥SK_{θ_d}的组成元素K_d，K'_d和$K_{d,\omega}$：

[0042] $K_d=g^\alpha\nu^\beta,K'_d=g^\beta,K_{d,\omega}=h_\omega^\beta,\forall\omega\in\theta_d$

[0043] S23、采用K_d，K'_d，$K_{d,\omega}$和θ_d构成解密密钥$SK_{\theta_d}=(\theta_d,K_d,K'_d,\{K_{d,\omega}\}_{\omega\in\theta_d})$；

[0044] 所述签名密钥SK_{θ_s}的生成方法包括：

[0045] S24、从乘法整数乘法群$\mathbb{Z}_q^*$中选取随机数γ；

[0046] S25、采用随机数γ及公钥PK和主密钥MSK中的参数计算签名密钥SK_{θ_s}的组成元素K_s,K'_s和$K_{s,\omega}$：

[0047] $K_s = g^{\alpha}\nu^{\gamma}, K'_s = g^{\gamma}, K_{s,\omega} = h_{\omega}^{\gamma}, \forall \omega \in \theta_s$

[0048] S26、采用K_s,K'_s,$K_{s,\omega}$和θ_s构成解密密钥$SK_{\theta_s} = (\theta_s, K_s, K'_s, \{K_{s,\omega}\}_{\omega \in \theta_s})$。

[0049] 在步骤S3中，根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文$SCT_{x_e} = (\chi_e, E_1, E_2, E_3, E_4, S_0, S_1, S_2, S_3, tt)$，并将签密文$SCT_{x_e}$存储在密文存储服务器上。

[0050] 在本发明的一个实施例中，所述根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文$SCT_{x_e} = (\chi_e, E_1, E_2, E_3, E_4, S_0, S_1, S_2, S_3, tt)$进一步包括：

[0051] S41、获取数据拥有者的签名属性集θ_s、签名密钥SK_{θ_s}、签名策略x_s和加密策略x_e；

[0052] S42、当签名属性集θ_s满足签名策略x_s时，则对明文M进行签名，并记录当前时间为tt；

[0053] S43、从整数乘法群$\mathbb{Z}_q^*$中选取随机数a′，对签名密钥SK_{θ_s}中的K_s,K'_s和$K_{s,\omega}$进行盲化处理：

[0054] $K_s^R = K_s \cdot \nu^{a'}, K_s^{R'} = K'_s \cdot g^{a'} K_{s,\omega}^R = K_{s,\omega} \cdot h_{\omega}^{a'}, \forall \omega \in \theta_s$

[0055] S44、计算签密文SCT_{x_e}的组成元素E_1,E_2,E_3,E_4,S_0,S_1,S_2和S_3：

[0056] $E_1 = g^{\varepsilon}$，其中，g为生成元，ε为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；

[0057] $E_2 = SEnc(SEK||d, M)$，E_2为采用一次对称加密算法SEnc对明文M进行加密；$SEK = Y^{\varepsilon}||S_1||tt$为对称封装密钥，d为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；

[0058] $E_3 = \{E_3^{(i)} = \nu^{\varepsilon \cdot \vec{\alpha} \cdot \vec{M}_e^{(i)}} h_{\varphi_e(i)}^{\varepsilon}\}_{i \in [l_e]}$，其中，$\vec{\alpha} = (\varepsilon, \delta_2, \ldots, \delta_{n_e})$，其组成元素为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；$\vec{M}_e^{(i)}$为矩阵$M_e$的第i行，ε为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数，$\varphi_e(i)$为行标签函数的第i行，$l_e$为矩阵$M_e$的行数；

[0059] $E_4 = (\eta_1 \eta_2^{\varepsilon})^{\varepsilon}$，其中，$\varrho = H_2(E_1)$，其为对$E_1$作哈希映射；

[0060] $S_0 = u^{\varepsilon \cdot H(SEK)} v^{\varepsilon \cdot H(d)}$，其中，H(•)为密码单向哈希函数；

[0061] $S_1 = g^{\varepsilon \cdot \delta}$； $S_2 = \{S_2^{(i)} = g^{\delta_i} (K_s^{R'})^{c_i}\}_{i \in [l_s]}$，其中，$d_i$、$c_i$均为从整数乘法群$\mathbb{Z}_q^{l_s}$中选取的向量；$l_s$为矩阵$M_s$的行数；$M_s$为$l_s$行$n_s$列的矩阵；

[0062] $S_3 = K_s^R (\prod_{i \in [l_s]} (K_{s,\varphi_s(i)}^R)^{c_i} \cdot h_{\varphi_s(i)}^{d_i}) (\mu_0 \prod_{i \in [l]} \mu_i^{f_i})^{\varepsilon} E_4^{\psi \cdot \delta}$，其中，$\psi = H_3(•)$，其为耐碰撞哈希函数；$(f_1, \ldots, f_l) \in \{0,1\}^l = H_1(•)$表示耐碰撞哈希函数；

[0063] S45、采用加密访问策略x_e，组成元素E_1,E_2,E_3,E_4,S_0,S_1,S_2和S_3及当前时间为tt构成签密文$SCT_{x_e} = (\chi_e, E_1, E_2, E_3, E_4, S_0, S_1, S_2, S_3, tt)$。

[0064] 在步骤S4中，下载存储于密文存储服务器中的签密文SCT_{x_e}，并采用公钥PK和下

载签密文SCT_{χ_e}用户的解密密钥SK_{θ_d},生成转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d},及采用公钥PK和下载签密文SCT_{χ_e}用户的签名密钥SK_{θ_s},生成转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}。

[0065] 其中,所述转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}的生成方法包括:

[0066] S41、从q阶的整数乘法群$\mathbb{Z}_q^*$中选取随机数t,并采用随机数t构成恢复密钥$RSK_{\theta_d}=t$;

[0067] S42、根据公钥PK和主密钥MSK中的参数,计算转换密钥$TSK_{\theta_d}=(TK_d,TK_d',\{TK_{d,\omega}\}_{\omega\in\theta_d})$:$TK_d=g^{\alpha t}\nu^{\beta t},TK_d'=g^{\beta t},TK_{d,\omega}=h_\omega^{\beta t},\forall\omega\in\theta_d$;

[0068] 所述转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}的生成方法包括:

[0069] S43、从q阶的整数乘法群$\mathbb{Z}_q^*$中选取随机数s,并采用随机数s构成恢复密钥$RSK_{\theta_s}=s$;

[0070] S44、根据公钥PK和主密钥MSK中的参数,计算转换密钥$TSK_{\theta_s}=(\theta_s,TK_s,TK_s',\{TK_{s,\omega}\}_{\omega\in\theta_s})$:$TK_s=g^{\alpha s}\nu^{\gamma s},TK_s'=g^{\gamma s},TK_{s,\omega}=h_\omega^{\gamma s},\forall\omega\in\theta_s$。

[0071] 在步骤S5中,将签密文SCT_{χ_e}中的签名S_2和S_3盲化处理为TS_2和TS_3,之后将转换密钥TSK_{θ_d}和TSK_{θ_s},盲化后的签名TS_2和TS_3及签密文SCT_{χ_e}中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换;

[0072] 在本发明的一个实施例中,所述将签密文SCT_{χ_e}中的签名S_2和S_3盲化处理为TS_2和TS_3进一步包括:

[0073] S51、采用恢复密钥RSK_{θ_s}重新随机化转换密钥TSK_{θ_s}:

[0074] $TK_s^R=TK_s\cdot\nu^{t'},TK_s'^R=TK_s'\cdot g^{t'},TK_{s,\omega}^R=TK_{s,\omega}\cdot h_\omega^{t'},\ \forall\omega\in\theta_s$;

[0075] S52、从$\mathbb{Z}_q^*$中选取随机数t',并采用恢复密钥$RSK_{\theta_s}=s$将签名S_3盲化为$TS_3=S_3^s\cdot g^{t'}$,将签名S_2盲化为$TS_2=TS_2^{(i)}=g^{d_i+T\cdot a_i}$。

[0076] 在步骤S6中,接收云服务器根据TS_2、TS_3和签名S_3产生的签名验证结果$VR=Y^s\cdot e(g,g)^{t'}$及根据签密文E_1和签密文E_3产生的转换签密文$TCT=Y^{t\cdot\varepsilon}$。

[0077] 实施时,本方案优选云服务器采用以下式子计算签名验证结果VR:

[0078] $$\frac{e(S_3,g)}{(\prod_{i\in[l_e]}e(\nu^{m_i}\cdot h_{\varphi_e(i)},S_2^{(i)}))\cdot e(\mu_0\prod_{i\in[l]}\mu_i^{I_i},E_1)\cdot e((\eta_1\eta_2^g)^\varepsilon,S_1)}$$

[0079] 采用以下式子进行密文转换:

[0080] $$\frac{e(TK_d\cdot\prod_{i\in[l_e]}TK_{d,\varphi_e(i)}^{k_i'},E_1)}{e(TK_d',\prod_{i\in[l_e]}(E_3^{(i)})^{k_i'})}$$。

[0081] 在步骤S7中,当下载签密文SCT_{χ_e}用户的解密属性集θ_d满足上传签密文$SCT_{\chi_e'}$用户的加密策略x_e时,根据恢复密钥RSK_{θ_d}、恢复密钥RSK_{θ_s}和秘密值t',采用$Y^s\cdot e(g,$

$g)^{t'}$计算验证值。

[0082] 此处需要说明的是签名验证结果和验证值虽然是采用的相同公式进行计算的，由于其中一个是采用转换密钥计算的，一个是采用恢复密钥计算的，其计算得到的值可能会存在差异，存在差异，则表明两者不相等，则令嵌入对称密钥的部分签密文S_0=TCT，并停止解签密操作。

[0083] 在步骤S8中，当签名验证结果VR的值与验证值相等时，计算部分签密文$S_0 = u^{H(SSK)}v^{H(d)}$，并采用一次性对称解密运算SDec(SEK，E_2)=M恢复下载签密文SCT_{χ_e}用户的明文M。

[0084] 由于下载签密文SCT_{χ_e}用户在本地对签密文SCT_{χ_s}进行解签密操作时，需要大量的双线性对运算，给用户带来巨大的计算成本，本方案在步骤S4至S5对签密文SCT_{χ_s}中的部分签名和部分签密文进行盲化处理后，外包给云服务器进行签名验证和解密操作，之后再采用步骤S6至步骤S8对云服务器的签名验证进行正确性判断，之后再在本地进行一次性对称解密操作，通过这种方式可以大幅度降低用户解签密的计算开销。

[0085] 对于不受带宽和资源限制的移动设备，其也可以在下载签密文SCT_{χ_e}用户的服务器(移动设备)上进行解签密操作，其实现方案为：

[0086] 当下载签密文SCT_{χ_e}用户的解密属性集θ_d满足加密策略x_e时，还包括下载签密文SCT_{χ_e}用户采用本地服务器进行解签密处理：

[0087] 当当前解签密时间tt′与签密时间tt差异的绝对值小于等于解签密时间时，通过如下等式验证签名S_3是否有效：

[0088] $$Y \stackrel{?}{=} \frac{e(S_3,g)}{(\prod_{i\in[l_s]} e(v^{\varpi_i}\cdot h_{\rho_s(i)}, S_2^{(i)}))\cdot e(\mu_0\prod_{i\in[l]}\mu_i^{f_i}, E_1)\cdot e((\eta_1\eta_2^{\theta})^{\psi}, S_1)}$$

[0089] 其中，向量$\vec{\varpi} = (1, x_2, \ldots, x_{n_s})\cdot \vec{M}_s^{(i)}, \forall i \in [l_s]$，随机数$x_2, \ldots, x_{n_s}$为从整数乘法群$\mathbb{Z}_q^*$中选取的随机数；$f_i = (f_1, \ldots, f_l) \in \{0,1\}^l = H_1(S_2, tt, \chi_s, \chi_e)$；令T＝γ+a′，$\sum_{i\in[l_s]}(T\cdot c_i + d_i)\cdot \varpi_i = (T, Tx_2, \ldots, Tx_{n_s})\cdot(1,0,\ldots,0) + (1, x_2, \ldots, x_{n_s})\cdot(0,0,\ldots,0) = T$；

[0090] 若签名S_3有效，且当下载签密文SCT_{χ_e}用户的解密属性集θ_d满足上传签密文SCT_{χ_s}用户的加密策略x_e时，恢复部分对称密钥Y^e：

[0091] $$\frac{e(K_d\cdot\prod_{i\in[l_e]} K_{u\rho_e(i)}^{k_i'}, E_1)}{e(K_u', \prod_{i\in[l_e]}(E_3^{(i)})^{k_i'})} = Y^e$$

[0092] 其中，$\vec{k'} = (k_1', k_2', \ldots, k_{l_e}')$为从$\mathbb{Z}_q^{l_e}$中选取向量；

[0093] $\sum_{i\in[l_e]} k_i'\cdot(\vec{\alpha}\cdot\vec{M}_e^{(i)}) = (\varepsilon, \delta_2, \ldots, \delta_{n_e})\cdot(1,0,\ldots,0) = \varepsilon$，

[0094] $\vec{k'}\cdot\mathbf{M}_e = \vec{1}_{n_e}$，$\sum_{i\in[l_e]} k_i'\cdot\vec{M}_e^{(i)} = \vec{1}_{n_e}$，$\mathbb{Z}_q^{l_e}$为长度为$l_e$的整数群；$M_e$为$l_e$行$n_e$列

的矩阵；$M_e^{(i)}$为矩阵M_e的第i行；对于任意的i，当$\varphi_{e(i)} \notin \theta_d$时，$k'_i=0$，$\varphi_e$为行标签函数；

[0095] 采用一次性对称解密运算SDec(SEK,E_2)＝M恢复下载签密文SCT_{x_e}用户的明文M。

[0096] 参考图2，图2示出了基于属性的可验证外包解签密系统的原理框图；如图2所示，该解签密系统包括：

[0097] 授权中心，用于获取密码系统中用户的签名属性集θ_s和解密属性集θ_d，并采用签名属性集θ_s和解密属性集θ_d组成的属性域A生成密码系统的公钥PK和主密钥MSK；

[0098] 采用公钥PK、主密钥MSK和签名属性集θ_s及公钥PK、主密钥MSK和解密属性集θ_d分别生成用户的签名密钥SK_{θ_s}和解密密钥SK_{θ_d}；

[0099] 上传签密文的用户服务器，用于根据公钥PK、签名密钥SK_{θ_s}以及签名策略x_s和加密策略x_e，对明文M进行签密，生成签密文$SCT_{x_e}=(x_e,E_1,E_2,E_3,E_4,S_0,S_1,S_2,S_3,tt)$，并将签密文$SCT_{x_e}$存储在密文存储服务器上；

[0100] 下载签密文的用户服务器，下载存储于密文存储服务器中的签密文SCT_{x_e}，并采用公钥PK和下载签密文SCT_{x_e}用户的解密密钥SK_{θ_d}，生成转换密钥TSK_{θ_d}和恢复密钥RSK_{θ_d}，及采用公钥PK和上传签密文SCT_{x_e}用户的签名密钥SK_{θ_s}，生成转换密钥TSK_{θ_s}以及恢复密钥RSK_{θ_s}；

[0101] 将签密文SCT_{x_e}中的签名S_2和S_3盲化处理为TS_2和TS_3，之后将转换密钥TSK_{θ_d}和TSK_{θ_s}、盲化后的签名TS_2和TS_3及签密文SCT_{x_e}中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换；

[0102] 接收云服务器根据TS_2、TS_3和签密文S_1和E_1产生的签名验证结果$VR=Y^s \cdot e(g,g)^{t'}$及根据签密文E_1和签密文E_3产生的转换签密文$TCT=Y^{t \cdot \varepsilon}$；

[0103] 当下载签密文SCT_{x_e}用户的解密属性集θ_d满足上传签密文SCT_{x_e}用户的加密策略x_e时，根据恢复密钥RSK_{θ_d}、恢复密钥RSK_{θ_s}和秘密值t'，采用$Y^s \cdot e(g,g)^{t'}$计算验证值；

[0104] 当签名验证结果VR的值与验证值相等时，计算签名$S_0=u'^{H(SSK)}v'^{H(d)}$，并采用一次性对称解密运算SDec(SEK,E_2)＝M恢复下载签密文SCT_{x_e}用户的明文M；

[0105] 云服务器，用于接收下载签密文的用户服务器上传的转换密钥TSK_{θ_d}和TSK_{θ_s}，TS_2和TS_3及签密文SCT_{x_e}中的签密文E_1、E_3和S_1，并根据接收的数据进行签名验证和密文转换。

[0106] 综上所述，本方案将验证签名和解密过程中复杂的计算外包给云服务器，

[0107] 不仅降低了用户端的解签密的计算成本，同时也保证了云服务器帮助用户解密和验证签名过程中部分签密文信息的正确性。

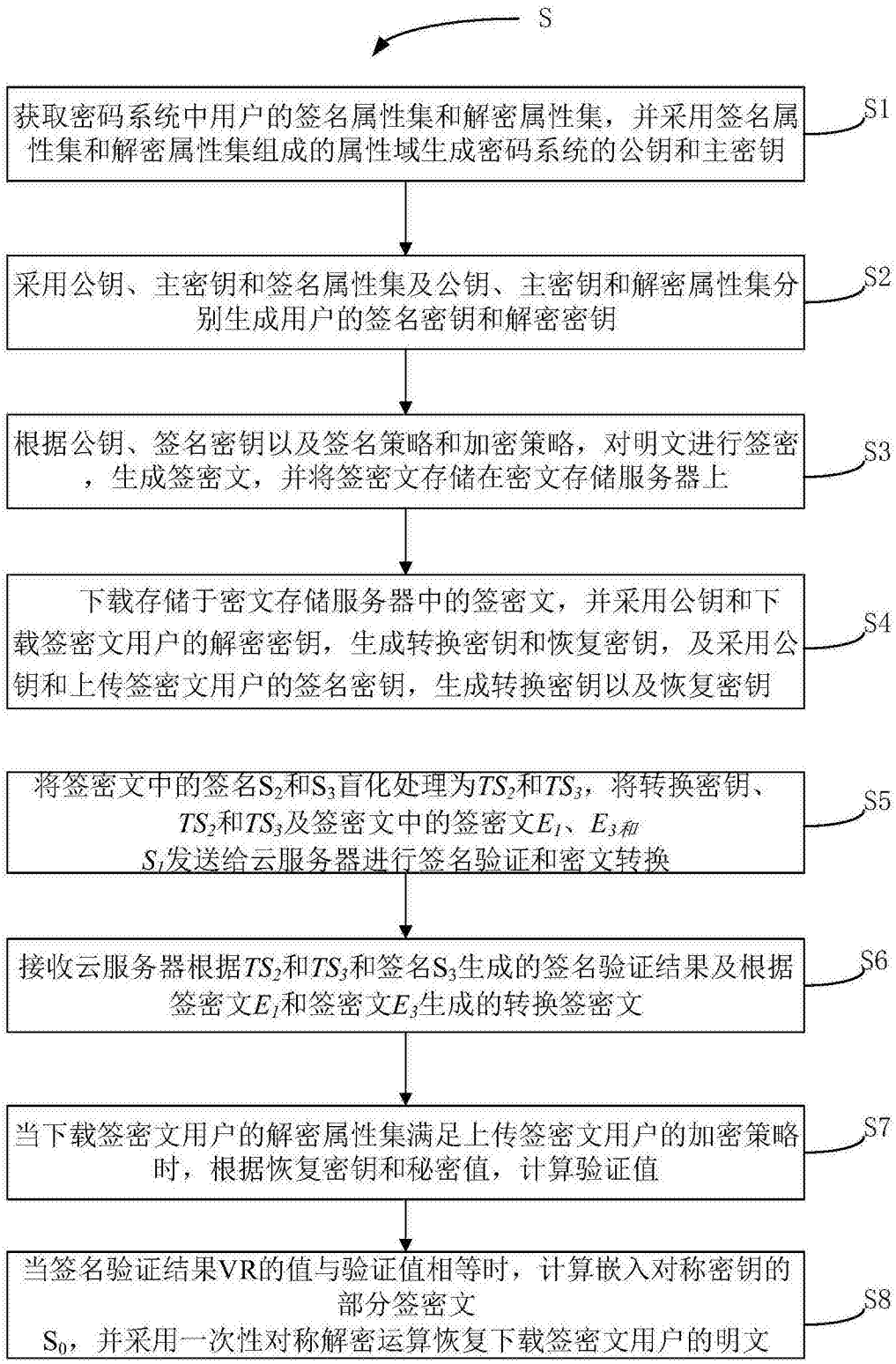

S
获取密码系统中用户的签名属性集和解密属性集，并采用签名属性集和解密属性集组成的属性域生成密码系统的公钥和主密钥
S1
采用公钥、主密钥和签名属性集及公钥、主密钥和解密属性集分别生成用户的签名密钥和解密密钥
S2
根据公钥、签名密钥以及签名策略和加密策略，对明文进行签密，生成签密文，并将签密文存储在密文存储服务器上
S3
下载存储于密文存储服务器中的签密文，并采用公钥和下载签密文用户的解密密钥，生成转换密钥和恢复密钥，及采用公钥和上传签密文用户的签名密钥，生成转换密钥以及恢复密钥
S4
将签密文中的签名S_2和S_3盲化处理为TS_2和TS_3，将转换密钥、TS_2和TS_3及签密文中的签密文E_1、E_3和S_1发送给云服务器进行签名验证和密文转换
S5
接收云服务器根据TS_2和TS_3和签名S_3生成的签名验证结果及根据签密文E_1和签密文E_3生成的转换签密文
S6
当下载签密文用户的解密属性集满足上传签密文用户的加密策略时，根据恢复密钥和秘密值，计算验证值
S7
当签名验证结果VR的值与验证值相等时，计算嵌入对称密钥的部分签密文S_0，并采用一次性对称解密运算恢复下载签密文用户的明文
S8

图1

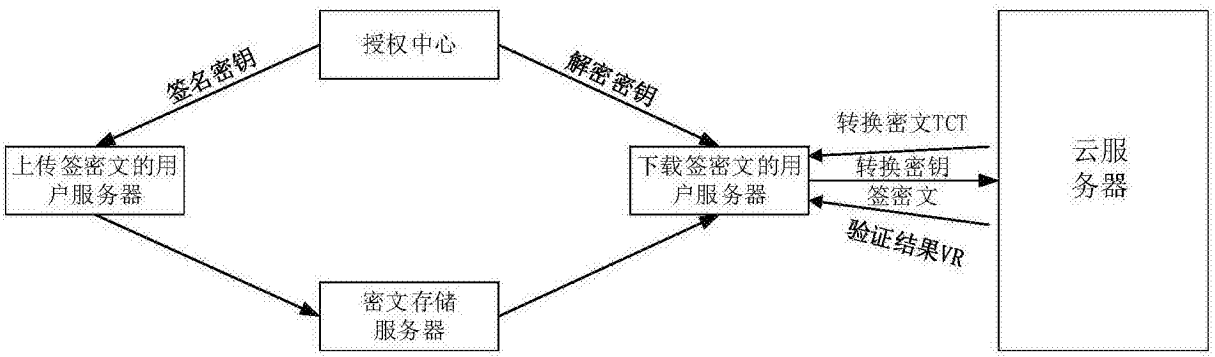

授权中心
签名密钥
解密密钥
上传签密文的用户服务器
下载签密文的用户服务器
转换密文TCT
转换密钥
签密文
验证结果VR
云服务器
密文存储服务器

图2