doi:10.3969/j.issn.1000-436x.2013.03.014

构造适合云的公平交换协议

蓝才会[1,2]，王彩芬[1]
（1. 西北师范大学 数学与信息科学学院，甘肃 兰州 730070；2. 兰州城市学院 信息工程学院，甘肃 兰州 730070）

摘 要：针对云计算中数据不是存储在本地而是以密文形式托管到“云端”，导致已有的公平交换协议不能很好适应云环境这一问题，构造了一个可在随机预言模型下证明安全的匿名条件的代理重加密方案，并在其基础上设计了一个能运用于云环境交换数据的公平交换协议。

关键词：云计算；公平交换协议；条件代理重加密；选择密文安全
中图分类号：TP309 **文献标识码**：A **文章编号**：1000-436X(2013)03-0111-08

Constructing fair-exchange protocols for cloud computing

LAN Cai-hui[1,2], WANG Cai-fen[1]
(1. College of Mathematics and Information Science, Northwest Normal University, Lanzhou 730070, China;
2. College of Information Project, Lanzhou City University, Lanzhou 730070, China)

Abstract: The existing fair-exchange protocols can't be run in cloud servers, because the user's data aren't stored in local computers, but be escrowed to 'clound'. An anonymous conditional proxy re-encryption scheme was constructed and its security was proved in random oracle model based on modified decisional bilinear Diffie-Hellman problem. At the same time, a fair-exchange protocol based on their scheme was designed, it can be employed when two users exchange their data information in cloud.

Key words: cloud computing; fair-exchange protocols; conditional proxy re-encryption; CCA security

1 引言

云计算作为一项兴起中的技术，以开放的标准和服务为基础，以互联网为中心，让互联网上的各种计算资源协同工作，共同组成数个庞大的数据中心和计算中心，为各类用户提供安全、快速、便捷的数据存储和网络计算等特定服务。它预示着存储信息和运行应用程序的方式将发生重大变化。程序和数据不再运行和存放在本地，相反，一切都托管到“云端”，这里的云端是指一个云状的、可通过 Internet 访问的、由个人计算机和服务器构成的集合。目前有些公司开已经通了云计算业务，如 Google、亚马逊和微软[1~3]等。为解决数据隐私的保护问题，数据往往采用密文形式存放，这就需要对加密数据进行检索，常见的有线性搜索[4]、基于关键字的公钥搜索[5]和安全索引[6]等。

随着云计算的不断发展成熟，在其环境下的电子商务、电子政务等活动必将展开。和基于 Internet 的现代经济活动一样，需要保证商务活动的保密性、完整性、不可否认性、公平性等安全特性。公平交换协议作为一类重要的安全协议，公平性尤为重要。非形式上来说，公平性是指在交易的任何阶段，交易的双方都处在平等地位，即要么双方都可以得到对方的信息，要么任何一方都得不到对方的信息。由于公平交换协议在密码理论和应用领域的重要性，研究人员在 Internet 环境下提出各种各样的方案[7~11]。遗憾的是这些方案不能很好适应云模式的商务活动。究其主要原因是：

收稿日期：2011-11-28；**修回日期**：2012-09-01
基金项目：国家自然科学基金资助项目（61163038，61063041）；甘肃省自然科学基金资助项目（3ZS051-A25-042）
Foundation Items: The National Natural Science Foundation of China (61163038, 61063041); The Natural Science Foundation of Gansu Province (3ZS051-A25-042)

在云计算中，用户的私有数据不存放在本地计算机上，而是以密文形式存放在“云端”。一种天真的想法是交易方从云端取到数据，再利用已有的公平交换协议进行交换，这种做法需要多次数据传输和多次加解密，严重影响其性能且不符合云服务模式；另一种不切实际的想法是用已有的公平交换协议来交换双方的解密钥，这虽然可以减少数据传输和加解密的次数，但另一个更为严重的问题是泄露了彼此的密钥，会造成双方数据无秘密可言，所以这种做法只适合双方无条件共享数据，但现实生活中这种情况几乎没有。

因此，相对于 Internet 环境下的公平交换协议的构造，云环境下的公平交换协议需要考虑到数据不是存储在本地，而是要以密文形式放在“云端”，并且只是交换其中部分数据，而不是所有。这就要求交换后得到的信息具有搜索功能，也就是能够判断出哪个密文是要被交换的，另外，交易双方能够使对方能且只能得到用于被交换的数据，对其他数据一无所知。这和条件代理重加密[12]的特点相似，在条件代理重加密方案中，一个半可信的 Proxy（代理人）能够利用额外信息（从授权人那里得到）将满足条件(关键字)的 Alice（授权人）的密文重加密成 Bob（受理人）的密文。在条件代理重加密中，也需要代理人利用额外信息能够判断出那个密文满足条件，以及授权人通过额外信息使得受理人能且只能解密满足条件的密文。

本文假设用户存放在云中的数据是采用基于关键字搜索的公钥加密算法加密后的密文，用条件代理重加密方案构造了一个能够运用在云环境下交换数据的公平交换协议。协议属于离线的半可信第三方公平交换协议[10]。协议由交换协议、恢复协议和终止协议组成。

2 模型

在本文的模型中，包括了 3 个实体：交易双方 i, j 和一个离线的半可信第三方（STTP）。但不像传统的公平交换协议，在云计算中数据是以密文形式存放在“云端”。所以，双方交换的不是数据本身，而是另外的信息 rk_{ij}, rk_{ji}。交易方 i 通过 rk_{ji} 能且只能得到 j 用于交换的数据，同样 j 通过 rk_{ij} 能且只能得到 i 用于交换的数据。在协议运行前，发起人 i 需要到 STTP 处注册获取证书 $Cert_i$，证书可以让响应方 j 相信 STTP 拥有 rk_{ij}。具体交易过程如下。

1) 发起人传递证书 $Cert_i$ 给响应方。

2) 响应方验证 $Cert_i$，若不成立，则协议自动停止，否则，用 STTP 的公钥加密 rk_{ji} 得到 C_j 并发送给发起人。要保证发起人能够验证 C_j 是 rk_{ji} 在 STTP 公钥下的密文。

3) 发起人验证 C_j，若不成立，则执行终止子协议，否则，把 rk_{ij} 发送给响应方。

4) 响应方验证 rk_{ij}，若不成立，则执行恢复子协议，否则，把 rk_{ji} 发送给发起人。这里的 rk_{ij} 的验证，可以通过随机选择消息 m，并在发起人的公钥下加密，再用 rk_{ij} 和响应方得私钥解密得到 m'，通过比较 m 和 m' 来验证。

5) 发起人验证 rk_{ji}，不成立，执行恢复子协议，否则，交易完成。rk_{ji} 的验证方法同上。

交易完成后，发起人把得到的 rk_{ji} 给云服务提供商，让他用 rk_{ji} 完成数据的部分处理，并把处理后的数据传输给发起人。响应方同样，与传统的公平交换协议不一样的地方是：交易完成后，双方所得到的信息可以让 STTP 和云服务商知道。所以，这里的数据隐私是指攻击者知道信息 rk_{ij}, rk_{ji}，也不会知道交易双方存储在云中的数据，进一步，对于发起人来说，就算攻击知道 rk_{ij}, rk_{ji} 并控制了响应方，也只能得到被用于交换的数据。

这里引进数据私有性的攻击模型，和条件代理重加密[12,13]很类似。具体就是在下面游戏中，不存在一个攻击者 A 可借助一个挑战者 D 以不可忽略的概率 $Adv = | pr\{d' = d) - \frac{1}{2} |$ 赢得游戏。

1) D 生成公共参数，并送给 A。

2) A 可以要求询问下列问题：

① 用户的公钥；

② 用户的私钥；

③ 两用户 (i, j) 的 rk_{ij}；

④ 云服务商利用 rk_{ij} 处理的结果；

⑤ 密文解密。

3) A 决定第一阶段询问结束，它输出挑战公钥 pk_* 和 2 个长度相同的明文 m_0, m_1。D 随机选择 $d \in \{0,1\}$，生成 m_d 的密文 C_*。

4) A 可以继续发起剩余询问，但要求不能询问 pk_* 的私钥，也不能对 C_* 和云服务商对 C_* 处理后的

结果进行解密询问，以及在 pk 的私钥被询问后，不能进行③和④询问。最后，A 输出 $d^{'} \in \{0,1\}$，若 $d^{'} = d$，则 A 赢得游戏。

公平性和传统的公平交换协议要求一致，简单地讲要么交易双方都能得到各自的数据，要么得不到任何数据。具体是：有一个攻击者，还有一个诚实方 B，和一个半可信方 T，在开始时，选择 2 组被交换的数据 M_A, M_B，以及对应 rk_{AB}, rk_{BA}。游戏如下。

1) 产生公共参数，A、B、T 的公私钥，以及用到验证算法 V，把公共参数，协议参与方的公钥和 A 的私钥给攻击者。

2) 攻击者可以进行如下活动：

① 和 B 的交互；

② 和 T 的交互；

③ 插入 B 和 T 的交互，但不能阻止。

3) 游戏结束后，以下概率

① $Adv = 1 - pr\{[V(rk_{AB}) = \text{accept} \rightleftarrows V(rk_{BA}) = \text{accep}]\}$ 可忽略，则公平交换了 rk_{AB}, rk_{BA}。根据协议模型，很显然，若 rk_{AB}, rk_{BA} 能公平交换，则数据也满足了公平交换。

② $Adv = 1 - pr\{[V(M_A) = \text{accept} \rightleftarrows V(M_B) = \text{accep}]\}$ 可忽略，则公平交换了数据。这里允许泄露部分 rk_{AB}, rk_{BA} 的信息，只要泄露的消息不能为对方提供任何有助于获取数据的帮助即可。

3　条件代理重加密

在 Asia CCS09 上，Weng 等人[12]首次提出了条件代理重加密系统的定义，并构造了一个条件代理重加密方案。同年，该作者又提出了一个更为高效的代理重加密方案[13]。在 2009 年中国密码学会上，周德华等人提出了一个基于身份的条件代理重加密方案[14]。但是这些方案有一个共同的缺陷是条件（关键字）是已知的，这不能应用于安全级别要求高的环境中，也不符合基于关键字的搜索公钥加密的要求。而一旦匿名关键字，构造的方案就必须满足：1) 授权人和受理人仅用自己的私钥可以解密，和关键字无关；2) 半可信的代理人利用额外信息能够判断出那个密文满足授权人给定的条件，这可以借助基于关键字的公钥加密思想来实现（设置一个用于搜索的陷门值）；3) 授权人生成的重加密钥可以使得受理人能且只能得到满足条件的信息。

3.1　单向的匿名条件的代理重加密

定义 1　一个单向的代理重加密方案包括如下几个算法。

$SetUp(\lambda)$：给定安全参数 λ，生成系统的全局参数 $param$；

$KeyGen(i)$：为用户 U_i 产生公私钥 (pk_i, sk_i)；

$RkeyGen(sk_i, pk_j, w^*)$：授权者 U_i 通过输入自己的私钥 sk_i、关键字 w^* 以及受理人 U_j 的 pk_j，产生代理重加密钥 $rk_{i \xrightarrow{w^*} j}$；

$Trapdoor(sk_i, w^*)$：授权者 U_i 利用自己的私钥 sk_i 和关键字 w^*，产生用于搜索的陷门 tk_{iw^*}；

$Encrypt(pk_i, m, w)$：给定公钥 pk_i、关键字 w 和消息 m，输出用户 U_i 的密文 CT_i；

$ReEncrypt(CT_i, tk_{iw^*}, rk_{i \xrightarrow{w^*} j})$：已知密文 CT_i、陷门 tk_{iw^*} 和代理重加密钥 $rk_{i \xrightarrow{w^*} j}$，代理人执行如下 2 个过程。

1) 用陷门判断密文 CT_i 是否包括关键字 w^*；

2) 如果不包括，输出 $\perp$，否则，把 CT_i 重加密成受理人 U_j 的密文 CT_j。

$Decrypte(CT, sk)$：输入私钥 sk 和密文 CT，输出消息 m 或 $\perp$。

一个单向条件代理重加密是正确的，意味着对任意的关键字 w^*、任意的 (m, w)、任意的 $(pk_i, sk_i) \leftarrow KeyGen(i)$、$(pk_j, sk_j) \leftarrow KeyGen(j)$ 以及 $CT_i \leftarrow Encrypt(pk_i, m, w)$，下面 2 个等式都成立：

1) $\Pr[Decrypte(CT_i, sk_i) = m] = 1$

2) 若 $w = w^*$，有

$$\begin{cases} \Pr[Test(CT_i, tk_{iw^*}) = 1] = 1 \\ \Pr[Decrypte(ReEncrypt(CT_i, RKeyGen(sk_i, w, pk_j)), sk_j) = m] = 1 \end{cases}$$

否则，

$$\begin{cases} \Pr[Test(CT_i, tk_{iw^*}) = 0] \geqslant 1 - neg(\lambda) \\ \Pr[Decrypte(ReEncrypt(CT_i, RKeyGen(sk_i, w, pk_j)), sk_j) = \perp] \geqslant 1 - neg(\lambda) \end{cases}$$

一个匿名的条件代理重加密的安全性需要考虑关键字的私有性和消息的私有性。但在现实生活中，相同的消息应该具有相同的关键字，反过来说 2 个消息的关键字不同，意味着他们是不同的消息。若关键字的语义安全不满足，则攻击者可以通过关

键字来判断消息，显然消息的私有性也不满足。也就是说在一般情况下，消息满足私有性，则关键字的私有性也是满足的，所有本文只对消息的私有性进行了分析。相对于关键字已知的条件代理重加密，需要询问陷门值。另外，攻击者不允许访问挑战消息所对应的关键字的陷门值。

3.2 本文的构造

在这一节，利用双线性对构造了一个匿名条件的代理重加密方案，具体如下。

SetUp(λ)：根据安全参数 λ 产生 (p,G_1,G_2,e)，G_1,G_2 是 2 个相同阶 p 的乘法群，$e:G_1\times G_1\to G_2$ 是一个双线性对；随机选择 $g\in G_1$，并令 $Z=e(g,g)$；最后选择 3 个散列函数：$H_1:\{0,1\}^*\to G_1$, $H_2:\{\{0,1\}^*, G_2\}\to z_p{}^*$, $H_3:G_2\to\{0,1\}^*$。

KeyGen(i)：选择随机数 $x_i\in z_p$，U_i 生成公私钥对 $(pk_i,sk_i)=(g^{x_i},x_i)$。

RkeyGen(sk_i,pk_j,w^*)：U_i 计算代理重加密钥 $rk_{i\xrightarrow{w^*}j}=(rk_{1ijw^*},rk_{2ijw^*})=(H_1(w^*)^{\frac{1}{x_i}},pk_j^{\frac{1}{x_i}}=g^{\frac{x_j}{x_i}})$。

Trapdoor(sk_i,w^*)：U_i 计算陷门 $tk_{iw^*}=H_1(w^*)^{\frac{1}{x_i}}$，连同代理重加密钥发送给代理人。

Encrypt(pk_i,m,w)：按下面 3 个步骤生成 U_i 的密文。

1) 选择一个强不可伪造的签名方案 *Sig*。假设签名和验证钥为 (ssk,svk)，并令 $A=svk$。

2) 随机选择 $r\in z_p$ 并计算

$$T=e(H_1(w),g),\quad R=H_2(A,Z^r),$$
$$B=pk_i^r,\quad C=H_1(w)^R,$$
$$D=H_3(T^{r+R}),\quad E=mZ^{rR}T^{rR}$$

3) 运行签名算法 $s=Sig(C,D,E)$，最后输出密文 $CT_i=(A,B,C,D,E,s)$。

ReEncrypt($CT_i,tk_{iw^*},rk_{i\to j}$)：按下面 3 个步骤把满足条件 U_i 的密文转换成 U_j 的密文。

1) 用验证钥 A 验证 s 是否为消息 (C,D,E) 的签名。

2) 用 $D=H_3[e(B,tk_{iw^*})e(C,g)]$ 检查 CT_i 是否包含关键字 w^*。

3) 若都成立，输出 $CT_j=(A,B',C,D',E,s)$，否则，输出 $\perp$。这里 $B'=e(B,rk_{2ijw^*})=e(g,g)^{x_jr}$，$D'=e(B,rk_{1ijw^*})=e(H_1(w^*),g)^r=e(H_1(w),g)^r$。

Decrypte(CT,sk)：按以下 2 种情况进行解密。

1) CT 是原始密文

① 用验证钥 A 验证 s 是否为消息 (C,D,E) 的签名。

② 计算 $T_1=B^{\frac{1}{x_i}}=g^r$，$T_2=e(T_1,g)=e(g,g)^r$，$R=H_2(A,T_2)$，$W=C^{R^{-1}}=H_1(w)$，$T_3=e(W,T_1)=e(H_1(w),g)^r$。

③ 验证 $D=H_3[T_3e(W,g)^R]$。

④ 若所有检查都成立输出 $m=\dfrac{E}{(T_2T_3)^R}=\dfrac{E}{e(g,g)^{rR}e(H_1(w),g)^{rR}}$，否则，输出 $\perp$。

2) CT 为重加密后的密文，即 U_j 解密过程如下。

① 计算 $T_1'=B'^{\frac{1}{x_j}}=e(g,g)^r$，$R=H_2(A,T_1')$，$W=C^{R^{-1}}=H_1(w)$，$T_2'=D'e(W,g)^R$.

② 用验证钥 A 验证 s 是否为消息 $(C,H_3(T_2'),E)$ 的签名。

③ 若所有检查都成立输出 $m=\dfrac{E}{(T_1'D')^R}=\dfrac{E}{e(g,g)^{rR}e(H_1(w),g)^{rR}}$，否则，输出 $\perp$。

3.3 安全性分析

定理 1 如果在 (G_1,G_2) 中，变形的判定性的双线性对 Diffie-Hellman 问题 (DBDHP)是困难的，那么方案在随机预言模型下是选择密文安全的。

证明 算法 D 收到一个随机实例 (g,g^a,g^b,g^c,Q)，它的目标是确定 $e(g,g)^{\frac{ac}{b}}$ 是否等于 Q。在游戏中 D 扮演挑战者，而 A 作为攻击者。D 需要维护 3 个初始化为空的散列表 H_1_list、H_2_list 和 H_3_list。游戏开始，D 把系统参数 $param=(g,G_1,G_2,e,H_1,H_2,H_3)$ 给 A。

Hash 询问：在任何时候 A 可以询问散列函数 H_1,H_2,H_3。

$H_1(w)$ 询问：若 (w,h_1,r) 已经在 H_1_list，则返回先前值 h_1。否则，D 随机选择 $r\in Z_p$，返回 $h_1=g^{br}$ 且把 (w,h_1,r) 写入 H_1_list。

$H_2(t,T^{(2)})$ 询问：若 $(t,T^{(2)},h_2)$ 已存在 H_2_list，则输出 h_2。否则，选择 $h_2\in Z_p{}^*$，把 $(t,T^{(2)},h_2)$ 加入 H_2_list 并返回 h_2。

$H_3(T^{(3)})$ 询问：若 $(T^{(3)},h_3)$ 已在 H_3_list，则返

回 h_3。否则，选择 $h_3 \in \{0,1\}^*$，把 $(T^{(3)}, h_3)$ 写入 H_3_list 并输出 h_3。

阶段 1　在这一阶段，A 可以发起一系列以下询问。

公钥产生询问：D 首先定义了一个有偏的随机数 $coin \in \{0,1\}$，令 $\Pr[coin=0]=\delta$。若 $(coin=0)$，D 选择随机数 $x_i \in Z_p^*$ 并计算 $pk_i = g^{x_i}$，否则计算 $pk_i = g^{bx_i}$。D 把 pk_i 给 A 并把 $(pk_i, x_i, coin_i)$ 增加到 k_list。

私钥产生询问：D 从 k_list 找 $(pk_i, x_i, coin_i)$，若 $(coin_i=1)$，返回⊥并终止，表示未腐化不能回答这次询问。否则，返回 x_i。

代理重加密询问 (pk_i, pk_j, w)：D 首先从 k_list 取得 $(pk_i, x_i, coin_i)$ 和 $(pk_j, x_j, coin_j)$，然后，根据以下情形为 A 计算 $rk_{i \xrightarrow{w} j}$。

1) 若 $(coin_i=0)$，表示知道 $sk_i = x_i$。从 H_1_list 取得 h_1 并计算 $rk_{i \xrightarrow{w} j}$。最后返回 $rk_{i \xrightarrow{w} j} = (rk_{1ijw} = h_1^{x_{0i}^{-1}}, rk_{2ijw} = pk_j^{x_i^{-1}})$ 并把 $(pk_i, pk_j, w, rk_{i \xrightarrow{w} j})$ 写到 rk_list。

2) 若 $(coin_i=1 \wedge coin_j=1)$，表示 $sk_i = bx_i$ 和 $sk_j = bx_j$。取得 h_1, r 从列表 H_1_list 中，返回 $rk_{i \xrightarrow{w} j} = (g^{\frac{r}{x_i}}, g^{\frac{x_j}{x_i}})$ 并把 $(pk_i, pk_j, w, rk_{i \xrightarrow{w} j})$ 写到 rk_list。

3) 若 $(coin_i=1 \wedge coin_j=0)$，表示 $sk_i = bx_i$ 和 $sk_j = x_j$。从 H_1_list 中获得 h_1, r，返回 $rk_{i \xrightarrow{w} j} = (g^{\frac{r}{x_i}}, \perp)$ 并把 $(pk_i, pk_j, w, rk_{i \xrightarrow{w} j})$ 加到 rk_list。

陷门询问 (pk_i, w)：D 从 k_list 获得 $(pk_i, x_i, coin_i)$ 并根据以下情形计算 tk_{iw}。

1) If $(coin_i=0)$，表示 $sk_i = x_i$.从 H_1_list 获得 h_1，计算 $tk_{iw} = h_1^{x_i^{-1}}$ 并返回。最后把 (pk_i, w, tk_{iw}) 写到 tk_list。

2) If $(coin_i=1)$，表示 $sk_i = bx_i$.从 H_1_list 获得 h_1 和 r，计算 $tk_{iw} = g^{\frac{r}{x_i}}$ 并返回。最后，把 (pk_i, w, tk_{iw}) 加到 tk_list。

解密询问 (pk_i, CT_i)：D 首先从 k_list 取得 $(pk_i, x_i, coin_i)$。若 $coin_i=0$，返回 $Decrypte(CT_i, x_i)$，否则，按下面 2 种情况处理。

1) CT_i 为原始密文，按下列步骤处理。

① 验证 s 是否为 (C, D, E) 的签名。若不是，输出⊥并终止。

② D 连续查找 H_3_list、H_1_list 和 H_2_list，看是够存在 $(T^{(3)}, h_3)$、(h_1, r) 和 $(T^{(2)}, h_2)$ 满足等式：$D_i = h_3$，$[T^{(3)} / e(C_i, g)]^{r^{-1}} = e(B_i, g)^{x_i^{-1}}$ 和 $C_i = h_1^{h_2}$。若不存在,输出⊥并终止，否则返回 $m = E/[T^{(2)} T^{(3)} / e(h_1, g)^{h_2}]^{h_2}$。

2) CT_j 为转换后的密文，按下列步骤处理。

①验证 s 是否 $(C, D = D^{'} e(C, g), E)$ 的签名。不是，输出⊥并终止。

②D 连续从 H_1_list 找 (h_1, r)，从 H_2_list 找 $(T^{(2)}, h_2)$，看是否满足等式：$B_j^{'\, x_j^{-1}} = e(C_j, g)^{r^{-1}}$，和 $C_j = h_1^{h_2}$。如果不存在，输出⊥并终止，否则返回 $m = E/[T^{(2)} D^{'}]^{h_2}$。

代理重加密询问 (pk_i, pk_j, CT_i)：D 首先从 k_list 取得 $coin_i$ 和 $coin_j$。若 $(coin_i=1 \wedge coin_j=0)$，输出⊥并终止，否则，按如下步骤处理。

1) 从 tk_list 找满足 $H_3(e(tk_{iw}, B_i) e(C_i, g)) = D_i$ 的 tk_{iw}，然后从 tk_list 取得。若找不到，可以通过解密询问 (pk_i, CT_i) 取得 w。

2) tk_{iw} 和 $rk_{i \xrightarrow{w} j}$ 可以通过查找 tk_list 和 rk_list，或者是通过陷门询问 (pk_i, w) 和重加密密钥询问 (pk_i, pk_j, w) 得到。

3) 返回 $CT_j = ReEncrypt(CT_i, rk_{i \xrightarrow{w} j})$。

挑战阶段：当 A 决定阶段 *1* 结束，它输出挑战公钥 pk_* 和 2 个长度相同的明文 $M_0 = (m_0, w_0)$，$M_1 = (m_1, w_1)$。D 从 k_list 获得 $(pk_*, x_*, coin_*)$。若 $coin_* = 0$，输出⊥并终止，否则，随机选择 $d \in \{0,1\}$ 和一个强的一次签名方 Sig。(ssk, svk) 为签名的签名钥和验证钥，然后返回如下数据。

$$T^* = e(g^{br}, g),\quad R^* = H_2(A, Q),\quad A^* = svk,$$

$$B^* = g^{ax^*} = (pk^*)^{\frac{a}{b}},\quad C^* = g^{brR^*},$$

$$D^* = H_3(e(g^{cr}, g^a) e(g^{br}, g)^{R^*}) = H_3(e(g^{br}, g)^{\frac{ac}{b}+R^*}),$$

$$E^* = m_d Q^{R^*} e(g^{cr}, g^a)^{R^*},\quad s^* = Sig(ssk, (C^*, D^*, E^*)),$$

$$CT^* = (A^*, B^*, C^*, D^*, E^*, s^*)。$$

这里的 $r = H_1(w_d)$，可以通过 H_1_list 得到。

阶段 2　A 像阶段 1 一样发起剩余的询问，但要满足文献[12]中安全游戏的限制，另外攻击者不允

许访问挑战消息所对应的关键字的陷门值。D 像阶段 1 一样回答这些询问。

猜测 最后，A 输出一个猜测值 $d' \in \{0,1\}$。若 $d' = d$，D 输出 1，否则，输出 0。

为了看清 D 攻破变形的判定性的双线性 Diffie-Hellman 问题。把 ac/b 堪称加密过程的随机数 r^*，并考虑 CT_* 为 m_d 在公钥 $pk^{*'} = g^{\frac{bx^*}{c}}$ 下的密文。很显然，若 $Q = e(g,g)^{r^*} = e(g,g)^{ac/b}$，$R^* = H_2(A^*, Q)$，则 CT_* 是关于 m_d 在 $pk^{*'} = g^{\frac{bx^*}{c}}$ 下的合法密文。这样，D 可以根据 A 回答来判断，即可以解决变形的判定性的双线性 Diffie-Hellman 问题。

为了完成证明，用 q_{H_1}, q_{H_2}, q_{H_3}, q_{pk}, q_{sk}, q_{rk}, q_{re} 和 q_{de} 分别表示 H_1 询问，H_2 询问, H_3 询问, 公钥产生询问，私钥产生询问，代理重加密钥询问, 重加密询问和解密询问的次数。e 表示自然对数的底数。

分析 D 可能失败的($\perp$)的情形：在私钥产生询问阶段，若 $coin = 1$，则 D 失败；在重加密密钥询问 (pk_i, pk_j, w) 阶段和重加密询问 (pk_i, pk_j, CT_i) 阶段，若 $(coin_i = 1 \wedge coin_j = 0)$，则 D 失败；在解密询问 (pk_i, CT_i)，若 A 攻破 Sig，则 D 失败，并令 $\beta = \Pr[A \text{breaks} Sig]$；在挑战阶段，若 $coin = 0$，则 D 失败。令 $q_{\max} = \max(q_{sk,} q_{rk,} q_{re})$。很容易知道，在模拟过程中，$D$ 没有失败的概率至少为

$$\delta^{q_{sk}}(1-(1-\delta)\delta)^{q_{rk}+q_{re}}(1-\beta)(1-\delta)$$
$$\geqslant \delta^{q_{\max}}(1-\delta)(1-(1-\delta)\delta)^{2q_{\max}}(1-\beta)$$

基于 Boneh D 等[15]的结果：$\delta^{q_{\max}}(1-\delta)$ 最大为 $\frac{q_{\max}}{1+q_{\max}}$，并有 $q_{\max}$ 充分大时，$(1-(1-\delta)\delta)^{2q_{\max}}$ 的值接近 $\frac{1}{e^2}$。因此，有 $\delta^{q_{sk}}(1-(1-\delta)\delta)^{q_{rk}+q_{re}}(1-\beta)(1-\delta) \geqslant \frac{1-\beta}{e^3(1+q_{\max})}$。

另外，在模拟过程中，在 A 询问 H_1 时，返回 $r^* = \frac{ac}{b}$ 概率至多有 $\frac{q_{H_1}}{p}A$ 公钥产生询问时询问了 $pk^{*'} = g^{\frac{bx^*}{c}}$ 概率至多有 $\frac{q_{pk}}{p}$，并且在私钥产生询问时返回 $\frac{bx^*}{c}$ 概率至多 $\frac{q_{sk}}{p}$。因此，D 的优势至少有 $\frac{\varepsilon[1-(q_{H_1}+q_{pk}+q_{sk})/p]}{e^3(1+q_{\max})}(1-\beta)$。证毕。

4 协议

基于上述的匿名条件的代理重加密方案，笔者将构造一个公平交换协议。协议属于离线的半可信第三方（STTP）公平交换协议，由交换协议、恢复协议和终止协议组成。在正常情况下，交易双方能够得到对方的条件代理重加密钥和搜索陷门。交易方得到对方信息后，可以把信息提供给云服务提供商，让提供商执行搜索和重加密，这样可以防止多次传输和加解密数据，提高了性能。同时，由条件重加密的特点可知，能够保证提供商得不到任何有关明文的信息，以及交易方只能得到满足条件的信息。

协议中使用的记号如下。

U_i，U_j 为交易双方，且它们钥交换的信息分别满足条件 w_i, w_j；STTP 为提供服务的半可信第三方；

$U_i \to U_j : X$ 表示 U_i 向 U_j 发送信息 X；

$pk_{\text{STTP}} = g^{x_s}$ 为 STTP 的公钥；

$Sign$ 表示安全的签名算法。

在协议执行前，发起者 U_i 需要到 STTP 处注册，获得证书 $Cert_i$，过程如下。

1) $U_i \to STTP : H_1(w_i), H_1(w_j), rk_{i \xrightarrow{w_i} j}, tk_{iw_i}$。

2) STTP 随机选择 $r \in Z_p^*$ 和消息 m，用方案中的加密算法计算出 CT_i^1，并用 tk_{iw_i} 去判断密文是否包含了关键字 w_i；接着用方案的重加密算法把 CT_i^1 转换成 CT_j^2；最后，计算签名 $\sigma = Sign(m, H_1(w_i), H_1(w_j), CT_j^2)$。

3) $STTP \to U_i : Cert_i = (m, H_1(w_i), H_1(w_j), CT_j^2, \sigma)$。

协议描述如下。

如果参与协议的每一方都是诚实的，则只要执行如下的交换子协议即可。

交换子协议如下。

1) $U_i \to U_j : Cert_i = (CA, CB, CC, CD, CE) = (m, H_1(w_i), H_1(w_j), CT_j^2, \sigma)$。

U_j 解密 CT_j^2 得到 m'，验证 $CA = m', CB = H_1(w_i), CC = H_1(w_j), \sigma$ 是否为 STTP 关于消息 (CA, CB, CC, CD)。若都成立，执行步骤 2，否则，协议自动终止。

2) $U_j \to U_i : (rk_{1 jiw_j} pk_{\text{STTP}}^t, g^t), rk_{2 jiw_j}$，这里的 t 是一个 Z_p^* 上的随机数。

U_i 验证等式 $e(rk_{1jiw_j} pk_{\mathrm{STTP}}^t, g^{x_i}) = e(H_1(w_j), rk_{2jiw_j}) e(g^{x_i t}, pk_{\mathrm{STTP}})$ 和 $e(pk_j, rk_{2jiw_j}) = e(pk_i, g)$ 是否成立。若成立，执行步骤 3，否则，执行终止协议。

3) $U_i \to U_j : rk_i \xrightarrow{w_j} j, tk_{iw_i}$。

U_j 随机选择消息 m_i，在关键字 w_i 下，用方案的加密算法生成 U_i 的密文，并用 tk_{iw_i} 去判断密文是否包含了 w_i。再用方案的重加密算法转换成 U_j 的密文，并用自己的私钥解密得到 $m_i^{'}$。比较 $m_i^{'}$ 是否等于 m_i，若是，执行步骤 4，否则，执行 U_j 恢复协议。

4) $U_j \to U_i : rk_j \xrightarrow{w_j} i, tk_{jw_j}$。

U_i 随机选择消息 m_j，在关键字 w_j 下，用方案的加密算法生成 U_j 的密文，并用 tk_{jw_j} 去判断密文是否包含了 w_j。再用方案的重加密算法转换成 U_i 的密文，并通过自己的私钥解密得到 $m_j^{'}$。比较 $m_j^{'}$ 是否等于 m_j，若是，交换完成，否则，执行 U_i 恢复协议。

U_i 恢复协议如下。

1) $U_i \to \mathrm{STTP} : (rk_{1ijw_j} pk_{\mathrm{STTP}}^t, g^t), rk_{2ijw_j}$。

STTP 先检查是否执行了终止协议和恢复协议，若已经执行，则终止。否则，验证 $e(rk_{1ijw_j} pk_{\mathrm{STTP}}^t, pk_i) = e(H_1(w_j), rk_{2ijw_j}) e(pk_i, g^{x_s t})$ 和 $e(pk_j, rk_{2ijw_j}) = e(pk_i, g)$ 是否成立，若成立，计算 $rk_{1ijw_j} = \dfrac{rk_{1ijw_j} pk_{\mathrm{STTP}}^t}{(g^t)^{x_s}}$ 并执行步骤 2。否则，拒绝做任何事。

2) $\mathrm{STTP} \to U_i : rk_j \xrightarrow{w_j} i = (rk_{1ijw_j}, rk_{2ijw_j}), tk_{jw_j} = rk_{1ijw_j}$。

同时 $\mathrm{STTP} \to U_j : rk_i \xrightarrow{w_j} j, tk_{iw_i}$。

U_j 恢复协议如下。

1) $U_j \to \mathrm{STTP} : Cert_i = (CA, CB, CC, CD, CE), rk_j \xrightarrow{w_j} i = (rk_{1ijw_j}, rk_{2ijw_j}), tk_{jw_j}$。

STTP 先检查是否执行了终止协议和恢复协议，若已经执行，则终止。否则，验证 CE 是否为 STTP 关于消息 (CA, CB, CC, CD) 签名，以及等式 $e(rk_{1ijw_j}, pk_i) = e(H_1(w_j), rk_{2ijw_j})$、$e(pk_j, rk_{2ijw_j}) = e(pk_i, g)$ 和 $tk_{jw_j} = rk_{1ijw_j}$。若都成立，执行步骤 2，否则，拒绝做任何事。

2) $\mathrm{STTP} \to U_j : rk_i \xrightarrow{w_j} j, tk_{iw_i}$，同时 $\mathrm{STTP} \to U_i : rk_j \xrightarrow{w_j} i = (rk_{1jiw_j}, rk_{2jiw_j}), tk_{jw_j} = rk_{1ijw_j}$。

终止协议（只是 U_i 执行）如下。

U_i 向 STTP 发送终止请求，STTP 检查是否执行了终止协议和恢复协议，若已经执行，则终止。否则，对终止标志签名，并发送给交易双方。

5　协议分析

本节证明协议满足最主要和最基本的性质：即保密性和公平性。

1) 保密性：协议的目的是在云环境下 2 个交易方交换特定的数据，但在执行协议的过程中没有交换数据，只是交换了各自的条件代理重加密钥和陷门。根据定理 1 可以知道，即使 STTP 或其他攻击者获取了条件代理重加密钥和陷门，没有交易方的合作，不可能解密。也就是，本文构造的公平交换协议满足保密性。

2) 公平性：所谓公平性，是指在协议执行的任何时刻，没有哪个参与方处于有利地位。

证明　协议中 U_i 条件代理重加密钥和搜索陷门的验证是通过 U_j 随机选择消息 m_i，在关键字 w_i 下，用方案的加密算法生成 U_i 的密文，并用 tk_{iw_i} 去判断密文是否包含了 w_i。再用方案的重加密算法转换成 U_j 的密文，并用自己的私钥解密得到 $m_i^{'}$。比较 $m_i^{'}$ 是否等于 m_i 来完成。让 $CT_i^1 = (A, B, C, D, E, s)$，而转换后的密文为 $CT_j^2 = (A, B^{'}, C, D^{'}, E, s)$。

根据搜索的判定等式 $D = H_3[e(B, tk_{iw_i}) e(C, g)]$ 和散列函数的单向性，一定有 $e(pk_i, tk_{iw_i}) = e(H_1(w_i), g)$，从而 $tk_{iw_i} = H_1(w_i)^{\frac{1}{x_i}}$。进一步可以确定 $rk_{1ijw_i} = tk_{iw_i}$。再根据加密、重加密和解密过程，有 $e(pk_i, rk_{2ijw_i})^{\frac{1}{x_j}} e(pk_i, rk_{1ijw_i}) = e(g, g) e(H_1(w_i), g)$，从而有 $rk_{2ijw_i} = g^{\frac{x_j}{x_i}}$。

同样，U_j 的条件代理重加密密钥和搜索陷门验证过程一样。

另外，U_j 也不可能发送 $(rk_{1jiw_j} pk_{\mathrm{STTP}}^t, g^t), rk_{2jiw_j}$ 来欺骗 U_i，因为要通过验证等式，就必须满足 $rk_{1ijw_i} = H_1(w_i)^{\frac{1}{x_i}}$，$rk_{2ijw_i} = g^{\frac{x_j}{x_i}}$。

由此得出，交易双方不可能通过发送错误的条件代理重加密密钥和搜索陷门来欺骗对方。

1) 如果所有的参与方都是诚实的，在成功执行完交换协议后，交易的双方都能收到对方所生产的条件代理重加密钥和搜索陷门。

2) U_i 试图欺骗，U_j 可以执行恢复协议；另外，U_i 得到 $(rk_{1\,jiw_j}\ pk^t_{\mathrm{STTP}}, g^t), rk_{2\,jiw_j}$，也具有公平性，因为需要 *STTP* 帮助才能得到 $rk_{1\,jiw_j}$。只有 $rk_{2\,jiw_j}$，U_i 根本求不出 $g^r = B^{\frac{1}{x_j}}$，从而无法计算 $e(H_1(w), g)^r$，即无法解密 U_j 的密文。

3) U_j 试图欺骗，U_i 可以执行终止协议和恢复协议；并且 U_j 执行恢复协议，需要提供自己的条件代理重加密和搜索陷门，STTP 恢复 U_i 的给 U_j，同时把 U_j 的也给 U_i。

综上所述，协议能满足公平性。

6 结束语

本文构造了一个匿名条件的代理重加密方案，并在随机预言模型下证明是安全的。由于现有的公平交换协议在云环境中不能很好应用，所以，作者在所构造的条件代理重加密的基础上设计了一个适合云环境的公平交换协议，协议属于离线的半可信第三方公平交换协议，由交换协议、恢复协议和终止协议组成。分析了协议的机密性和公平性。

参考文献：

[1] Google docs- online documents, spreadsheets, present[EB/OL]. http:// docs.google.com. 2009.

[2] Windows Azure platform[EB/OL]. http:// www.mirosoft.com/ windows sazure/, 2009.

[3] Amazon elastic compute cloud[EB/OL]. http:// aws.amzaon.con/ec2, 2009.

[4] SONG D, WAGNER D, PERRIG A. Practical techniques for searches on encrypted data[A]. Proceedings of the IEEE Symposium on Security and Privacy(S&P'00)[C]. Berkeley, CA,USA. 2000. 44-55.

[5] BONEH D, CRESCENZO G, OSTROVSKYET R, *et al*. Public key encryption with keyword search[A]. Proceedings of the 23rd Annual International Conference on the Theory and Applications Cryptographic Techniques[C]. Interlaken, Switzerland, 2004. 506-522.

[6] PARK D, KIM K, LEE P. Public key encryption with conjunctive field keyword search[A]. Proceedings of the 2004 Workshop on Information Security Applications (WISA'04)[C]. Wuhan, China. 2004. 73-86.

[7] ZHOU J, GOLLMANN D. A faire non-repudiation protocol[A]. Proc of the 1996 IEEE Symp on Security and Privacy[C]. Oakland: IEEE Computer Press, 1996. 55-61.

[8] FRANKLIN M K, REITER M K. Faire exchange with a semi-trusted third party[A]. Proc of the 4th ACM Conf on Computer and Communications Security[C]. Zurich, Switzerland, 1997. 1-5.

[9] 孙艳宾, 谷利泽, 孙燕等. 基于并发签名的公平交易协议的分析与改进[J]. 通信学报, 2010, 31(9):146-150.
SUN Y B, GU L Z, SUN Y, *et al*. Analysis and improvement of the CS-based fair exchange protocol[J]. Journal on Communications, 2010, 31(9):146-150.

[10] ASOKAN N, SCHUNTER M, WAIDNER M. Optimistic protocols for fair exchange[A]. Tsutomu Matsumoto, Editor, 4th ACM Conference on Computer and Communications security[C]. Switzerland, 1997. 6-17.

[11] 刘景伟, 孙蓉, 郭庆燮. 公平交换签名方案[J]. 中国科学: 信息科学, 2010, 40(6): 786-795.
LIU J W, SUN R, GUO Q X. Fair exchange signature scheme[J]. Science in China: Information Science, 2010, 40(6): 786-795.

[12] WENG J, DENG R H, DING X, *et al*. Conditional proxy re-encrytion secure against chosen-ciphertext attack[A]. Proc of ASIACCS'09[C]. Sydney, Australia, 2009. 322-332.

[13] WENG J, YANG Y, TANG Q, *et al*. Efficient conditional proxy re-encryption with chosen-ciphertext security[A]. Proc of ISC'09[C]. 2009. 151-166.

[14] ZHOU D H, CHEN K F, LIU S L. Identity-based conditional proxy re-encryption[A]. Proc of CHINACRYPT'09[C]. 2009. 85-97.

[15] BONEH D, FRANKLIN M. Identity-based encryption from the weil pairing[J]. SIAM Journal of Computing, 32(3):586-615.

作者简介：

蓝才会（1977-），男，畲族，江西永丰人，西北师范大学博士生，兰州城市学院讲师，主要研究方向为信息安全。

王彩芬（1963-），女，河北安国人，博士，西北师范大学教授、博士生导师，主要研究方向为信息安全、电子商务协议的分析和设计。