计算机系统应用 ISSN 1003-3254, CODEN CSAOBN
Computer Systems & Applications,2021,30(1):45−53 [doi: 10.15888/j.cnki.csa.007743]

E-mail: csa@iscas.ac.cn
http://www.c-s-a.org.cn
Tel: +86-10-62661041

云存储环境下基于时释性加密的 CP-ABE 方案①

张　戈, 华　蓓

(中国科学技术大学 计算机科学与技术学院, 合肥 230027)
通讯作者: 华　蓓, E-mail: bhua@ustc.edu.cn

摘　要: 云存储是未来存储业务的发展方向, 数据安全是云存储客户的首要关切. 密文策略属性加密 (CP-ABE) 算法允许数据拥有者将访问策略嵌入密文中, 并结合数据访问者的密钥实施访问控制, 特别适合云存储环境, 但 CP-ABE 不支持与时间相关的访问控制. 本文提出基于时释性加密的 CP-ABE 方案, 通过在 CP-ABE 中融入时释性加密 (TRE) 机制来实现带有时间控制的密文共享, 允许数据拥有者基于用户属性和访问时间制定更加灵活的访问策略. 论文通过安全分析表明, 该方案能够抵抗来自用户、云存储平台和授权机构的非法访问、非法用户的串谋攻击以及选择明文攻击.

关键词: 云存储; 访问控制; 密文策略属性加密; 时释性加密; 数据安全

引用格式: 张戈,华蓓.云存储环境下基于时释性加密的 CP-ABE 方案.计算机系统应用,2021,30(1):45–53. http://www.c-s-a.org.cn/1003-3254/7743.html

CP-ABE Solution Based on Time-Release Encryption in Cloud Storage Environment

ZHANG Ge, HUA Bei

(School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China)

Abstract: Cloud storage is the future development direction of the storage business, and data security is the primary concern of cloud storage customers. The Ciphertext-Policy Attribute-Based Encryption (CP-ABE) algorithm allows the data owner to embed the access policy in the ciphertext and implement access control in conjunction with the key of data accessor, which is particularly appropriate for cloud storage environments. However, CP-ABE does not support time-related access control. This study proposes a CP-ABE scheme based on Time-Release Encryption (TRE). By incorporating a TRE mechanism in CP-ABE to achieve ciphertext sharing with time control, this scheme allows data owners to formulate a more flexible access strategy based on user attributes and access time. And then, we conduct security analysis to verify that this scheme can resist illegal access from users, cloud storage platforms and authorized institutions, as well as collusion attacks of illegal users. In addition, this scheme can also resist chosen-plaintext attack.

Key words: cloud storage; access control; Ciphertext-Policy Attribute-Based Encryption (CP-ABE); Time-Release Encryption (TRE); data security

近年来, 科学计算和商业计算等众多应用领域产生了海量的数据, 且数据量仍在急剧增长, 给数据存储和管理造成了极大的压力. 云计算技术的兴起使得高效低成本的云存储得以实现. 用户可将数据托管到云数据中心, 并通过互联网进行访问和管理, 而数据中心提供数据备份、容灾等服务, 极大地降低了用户在数据存储方面的投资和管理成本[1].

然而有数据表明, 目前大约 80% 左右的企业不愿意将业务数据放在公有云上, 用户的主要担忧是数据安全性. 一方面云存储服务商大量涌现, 用户对于这些

① 收稿时间: 2020-06-01; 修改时间: 2020-06-23; 采用时间: 2020-06-30; csa 在线出版时间: 2020-12-31

云服务商缺乏信任; 另一方面云数据中心可能存在技术弱点, 数据不加密或者加密强度不够, 存在信息泄漏的风险. 目前云数据中心主要采用高强度加密标准对数据进行加密, 防止数据被窃取和篡改, 同时通过专业的安全防护措施防范各种安全攻击. 尽管数据机密性是云存储用户的一大关切, 但与此同时用户对数据的开放性也有期待, 希望不同类型的客户可以在各自的权限范围内自由访问数据. 云存储环境下的数据访问控制成为一个亟待解决的问题.

基于密文的访问控制技术可以解决数据的安全访问问题. 用户将数据加密后上传到云存储平台, 只有拥有合法解密密钥的用户才能解密数据. 然而在传统的公钥加密体制中, 用户的公钥是和身份信息绑定的, 进而生成对应的用户证书, 存储和管理用户证书的开销很大; 此外传统公钥体制也不能实现一对多的加解密模式, 不适应数据共享的云存储环境.

针对以上问题, Goyal 等人[2] 提出了基于属性的加密策略 (Attribute-Based Encryption, ABE). 在这种策略中, 每个用户都由一个属性集合表示, 用户的密钥与其属性相关联; 数据拥有者可以自行设定访问控制结构, 访问控制结构是属性的表达式, 只有属性集合满足访问控制结构的用户才能够解密成功. 在图 1 的例子中, 用户 A 的属性集为{网络专业, 硕士}, 用户 B 的属性集为{通信专业, 大四}, 用户 B 的属性集满足访问控制结构, 而用户 A 的属性集不满足, 所以只有用户 B 能够解密密文. ABE 将密钥与用户属性关联, 极大地降低了密钥存储和管理的开销. 由于密钥和访问控制结构都与属性关联, ABE 可以基于属性实施灵活的访问控制. 基于属性加密也可以实现一对多加解密模式, 满足云存储环境下数据安全共享的需求.

ABE 主要分为密钥策略属性加密 (KP-ABE) 和密文策略属性加密 (CP-ABE) 两大类. 在 KP-ABE (Key-Policy ABE) 中, 密钥与访问策略 (访问控制结构) 相关联, 密文和属性相关联; 而在 CP-ABE (Ciphertext-Policy ABE) 中, 密文和访问策略相关联 (访问控制结构作为加密函数的输入参数之一), 密钥和属性相关联. 两类方法的解密过程都是属性集与访问控制结构进行匹配的过程, 匹配成功则密文顺利解开. KP-ABE 以数据访问者为主导, 通过将私钥与访问策略关联选择要接收的数据, 适合云存储环境中对数据的查询. CP-ABE 以加密者 (数据拥有者) 为主导, 通过将密文和访问策略关联对数据访问者作出限制, 适合云存储环境中对数据的访问控制.

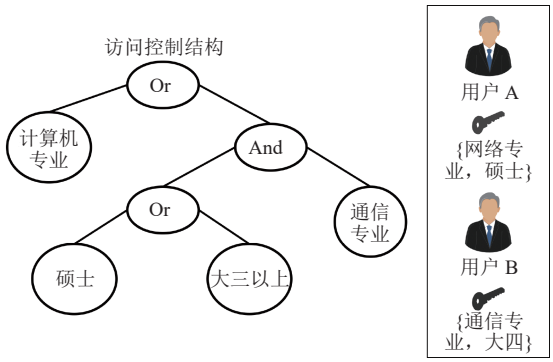

图 1 基于属性加密的访问控制

以上访问控制都没有考虑时间因素, 事实上许多数据访问控制是时间敏感的. 比如, 视频网络平台发布影片或视频有一个允许发布时间, 这个发布时间由制片方制定, 制片方提前将影视作品提交到平台, 但要求平台不得提前发布或泄露作品. 除限制发布时间外, 制片方也希望有办法阻止一些视频平台非法得到影视作品后进行出售. 因此, 结合用户属性和时间因素来制定访问控制策略具有非常现实的需要.

时释性加密 (Timed-Release Encryption, TRE)[3] 是在发送者和接收者之间传输密文的一种密码学原语. 它允许发送者提前将加密的消息发送给接收者并指定未来特定的解密时间, 接收者只有在该时间到来后才能解密消息. TRE 是时间控制加密技术的典型代表. TRE 依赖一个可信的时间服务器产生安全参数和公私钥对, 数据发送者利用时间公钥、接收者公钥和设定的发布时间加密明文, 将密文发送给接收者; 时间服务器在设定的时间到来时发布一个时间陷门; 数据接收者利用私钥和时间陷门解密数据. 近年来, TRE 在电子拍卖、电子投票、在线考试等领域发挥了重要的作用. 然而, TRE 只是基于时间进行访问控制, 并没有针对用户的访问控制能力.

将 TRE 引入 CP-ABE 可以解决云存储环境下带有时间因素的访问控制问题, 但相关研究目前尚未开展, 在技术上也存在一些挑战. 首先, 引入时间服务器之后, 如何降低访问控制对时间服务器的依赖; 其次, 如何把时间服务器的公钥对与 CP-ABE 的访问控制结构相结合, 把时间陷门与私钥和用户属性相结合, 来增强共享数据的机密性与可用性; 最后, 如何保证引入 TRE 之

后的 CP-ABE 能够抵抗各种非法用户的攻击行为.

本文针对以上问题提出云存储环境下基于时释性加密的 CP-ABE 方案, 允许基于属性和时间来制定和实施数据访问控制. 在该方案中, 用户的私钥组成部分来源于云存储平台、用户属性和时间服务器, 降低访问控制对单一第三方机构的依赖; 将用户的私人标记融入用户的私钥中, 抵抗非法用户的串谋攻击; 对共享数据进行时间与属性的双重加密, 提高共享数据的机密性与抗伪装攻击能力.

1 相关工作

ABE 可以有效解决细粒度访问控制和大批量用户动态扩展的问题, 是云存储环境下理想的访问控制方案. 但在早期的 CP-ABE 方案中, 私钥需要依赖一个可信的密钥托管中心, 即单个的授权机构, 不适合当前信任分散的、开放的云环境. Dong 等人[4] 和 Yang 等人[5] 引入多授权机构来解决 CP-ABE 的密钥托管问题, 将信任和负载分配到多个授权机构中. 然而在他们的方案中, 每个授权机构使用相同的主密钥来产生用户私钥, 若用户具有足够多的属性就可以重建主密钥, 增大了非授权用户串谋攻击的可能性. 关志涛等人[6] 针对授权方不完全可信的问题, 提出了一种多授权机构访问控制模型. 在模型中设计了最小化属性分组算法, 用户需要访问共享数据时按需获取密钥, 不仅减少了资源消耗, 还提升了分配效率.

TRE 允许发送者在加密消息时指定未来的解密时间, 接收者只有在预设的时间到来后才能解密消息. 1996 年 Rivest 等人[7] 对 TRE 进行了详细论证, 此后 TRE 的理论和实践均取得了非凡的进步. 2005 年 Chan 和 Blake[8] 第一次基于 TRE 提出可扩展的被动式服务器用户匿名方案, 大大降低了对时间服务器的依赖. Hwang 等人[9] 对 TRE 的预开放功能作了第一次尝试. 随后 Cheon 等人[10]于 2008 年正式提出了安全公钥 TRE 的概念, TRE的安全性得到了进一步提升. 2014 年袁科等人[11] 对 TRE 的研究历史和不同场景下的加密方案做了总结和分析, 具体阐述了 TRE 的安全目标与攻击模型.

2 基础知识

2.1 双线性映射

设存在一个大素数p以及两个循环群G和G_1, 它们的阶均为p, g是G的某个生成元. 那么, 从G到G_1存在一个映射$e: G\times G\to G_1$, 且具备如下性质[12].

(1) 双线性: 对于$\forall a,b\in Z_p^*$和$\forall u,v\in G$, 均有:

$$e\left(g^a,g^b\right)=e\left(g^b,g^a\right)=e(g,g)^{ab}$$

(2) 非退化性: 存在$a,b\in G$, 使得$e(a,b)\neq 1$, 这里 1 为群G_1的单位元.

(3) 可计算性: 对于$\forall u,v\in G$, $e(u,v)$都能被一个有效的算法计算出来.

则我们称上述映射e为一个双线性映射, 一般情况下, 循环群G为加法循环群, 循环群G_1是乘法循环群.

2.2 拉格朗日插值法

设存在一个次数为k的多项式函数, 如果给定此多项式函数上$k+1$不同取值点: $(x_0,y_0),\cdots,(x_k,y_k)$, 那么就存在唯一确定的一个拉格朗日插值多项式为:

$$L(x)=\sum_{j=0}^{k}y_j l_j(x)$$

其中, 每一个$l_j(x)$为拉格朗日基本多项式 (插值基函数), 其表达式为:

$$l_j(x)=\prod_{0\le m\le k,m\neq j}\frac{x-x_m}{x_j-x_m}=\frac{x-x_0}{x_j-x_0}\cdots\frac{x-x_{j-1}}{x_j-x_{j-1}}*\frac{x-x_{j+1}}{x_j-x_{j+1}}\cdots\frac{x-x_k}{x_j-x_k},0\le j\le k$$

2.3 DBDH 问题

判定双线性 Diffie-Hellman 问题 (Decisional Bilinear Diffie-Hellman Problem, DBDH 问题)[12]. 已知一个五元组为$\left(g,g^a,g^b,g^c,Z\right)$, 其中$g$是群$G$的某个生成元, $a,b,c\in Z_p^*$, $Z\in G_1$, 要想判定$Z=e(g,g)^{abc}$是否为真是困难的.

假设存在一个算法 A 在多项式时间内解决 DBDH 问题的概率为δ, 则δ可以被忽略. 因此, 不存在这样的算法可以用不可忽略的概率在多项式时间内解决 DBDH 问题.

因此, 本文设计的安全机制是以 DBDH 困难问题为基础构建的.

2.4 CP-ABE 算法

CP-ABE 加密算法涉及到属性集合、访问控制结构和访问控制树.

(1) 属性集合: 设每个参与者的身份信息由一个属性集合 A_i 表示, P 为所有参与者属性的集合, 即 A_i 是 P 的一个非空子集. 例如: $P=${学生, 大二, 计算机专业, 成绩 80 分}, $A_i=${学生, 大二, 成绩 80 分}.

(2) 访问控制结构: 访问控制结构是一组判断条件, 通常表示为Γ, 包含P中的若干属性元素和门限逻辑运算符 (如 OR、AND 等). 若某个属性集满足该判断条件, 称这个属性集为授权集, 反之为非授权集.

(3) 访问控制树: 访问控制树被用来描述访问控制结构, 每个叶子节点x代表一个属性λ_x, 每个父节点是一个门限逻辑运算符. 图 2 是与访问控制结构{((计算机学院)OR(网络工程))AND((年级<大四)AND(成绩≥70))}相对应的访问控制树.

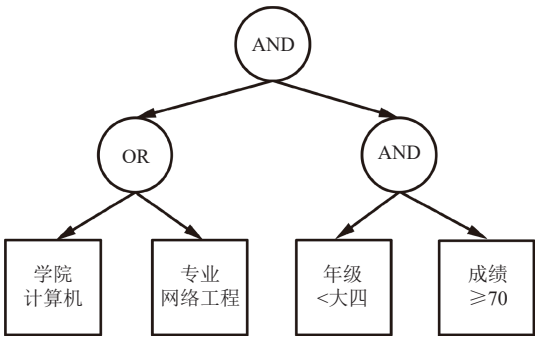

图 2 访问控制树

典型且常用的 CP-ABE 机制由下列 4 个模块构成:

(1) 初始化 (Setup): 产生一个公钥对$\{PK, MK\}$, 其中, MK是主密钥, PK是公开参数;

(2) 加密算法 (Encrypt): 输入集合为{明文M, 访问控制结构Γ, 公开参数PK}, 输出为密文C_Γ, C_Γ与访问控制结构相关联;

(3) 私钥生成算法 (KeyGen): 输入集合为{属性集S, 公开参数PK, 主密钥MK}, 输出SK, 即为用户的私钥;

(4) 解密算法 (Decrypt): 输入集合为{密文C_Γ, 公开参数PK, 私钥SK, 访问控制结构Γ}, 若属性集S满足访问控制结构Γ, 则可以还原出明文M.

2.5 TRE 技术

TRE 依赖一个可信的时间服务器, 在指定的时间到达时发布一个时间陷门, 接收者得到这个时间陷门后才能够解密消息. 时间服务器提供的是绝对的发布时间, 时间陷门就是它的自我签名, 在这个过程中不需要额外的签名服务.

TRE 方案包含 5 个模块, 分别为初始化 (Setup)、密钥生成算法 (KeyGen)、时间陷门发布算法 (TS-Release)、加密算法 (Encrypt) 和解密算法 (Decrypt). TRE 方案共有 3 类参与实体, 分别是时间服务器、发送者和接收者. 具体过程如下:

(1) 初始化 (Setup): 时间服务器产生安全参数以及公私钥对;

(2) 密钥生成算法 (KeyGen): 接收者根据时间服务器产生的安全参数及时间公钥计算自己的公私钥对;

(3) 时间陷门发布算法 (TS-Release): 时间服务器生成字符串$T \in \{0,1\}^l$用来表示发布时间t, 时间服务器也生成$s_T = H(T)^s$来作为时间陷门并发布出去, 其中s是时间服务器的私钥. 为了避免有恶意攻击者伪造时间陷门的风险, 在发布时间点T, 时间服务器给所有接收者发布在T时刻的签名, 这个签名就是时间陷门s_T;

(4) 加密算法 (Encrypt): 发送者利用时间公钥和接收者的公钥以及设定的发布时间t对明文数据M进行加密, 形成密文C, 将C发送给接收者;

(5) 解密算法 (Decrypt): 当接收者拿到密文C和时间陷门$s_T = H(T)^s$之后, 利用其私钥进行解密操作, 还原出明文数据M.

3 共享模型与安全威胁

以下是 TRE 技术运用到 CP-ABE 方案的一个应用场景: Alice 是计算机学院的一名老师, 计划在下周二对计算机学院网络工程专业的大二学生进行一次在线考试, 但不巧的是她当天必须参加一个重要会议. 在这种情况下, Alice 可以在 CP-ABE 方案中融入 TRE 来解决她的问题. Alice 可以预先发送以下消息:

$$\left(Enc\left(ID, ts_{\text{pub}}, r, M, T\right), T\right)$$

其中, M是考试试题, ID是M的目标用户组的属性集合{“计算机学院”, “网络工程专业”, “大二学生”}, ts_{pub}为时间服务器的公钥, r是随机新鲜因子, T是预定的发布时间. 所有学生都提前收到了这个消息, 但只有在预定的发布时间到来时, 满足属性集合ID的学生才能够利用时间服务器产生的时间陷门解密得到试题M.

3.1 共享模型

在开放云存储环境下, 基于时释性加密的 CP-ABE 方案的密文数据共享模型主要由以下 5 类参与实体构成: 数据拥有者、数据访问者、云存储平台、授权机构和时间服务器. 系统模型如图 3 所示.

(1) 数据拥有者: 提供共享数据, 设定访问结构以形成访问策略, 设置发布时间, 同时得到时间服务器发布的时间公钥, 将共享数据用时间公钥并按照访问结构加密后发送给云存储平台.

(2) 数据访问者: 可以从云存储平台获取加密后的共享数据, 等发布时间到来后, 得到时间服务器在发布时间产生的时间陷门, 形成时间私钥, 而后结合自己的密钥来解密共享数据.

(3) 云存储平台: 为数据共享提供一个方便稳定的平台, 并能够管理数据拥有者托管给它的共享数据. 云存储平台会和授权机构一起对共享系统中的属性集合以及用户的密钥进行有效的管理和分配.

(4) 授权机构: 与云存储平台一起对共享系统中的属性集合以及用户的密钥进行有效的管理和分配.

(5) 时间服务器: 完全可信的第三方机构, 会诚实地按照一定周期发布时间陷门, 且这个时间陷门不可伪造, 为云存储环境下的用户提供一个可靠准确的时间参照.

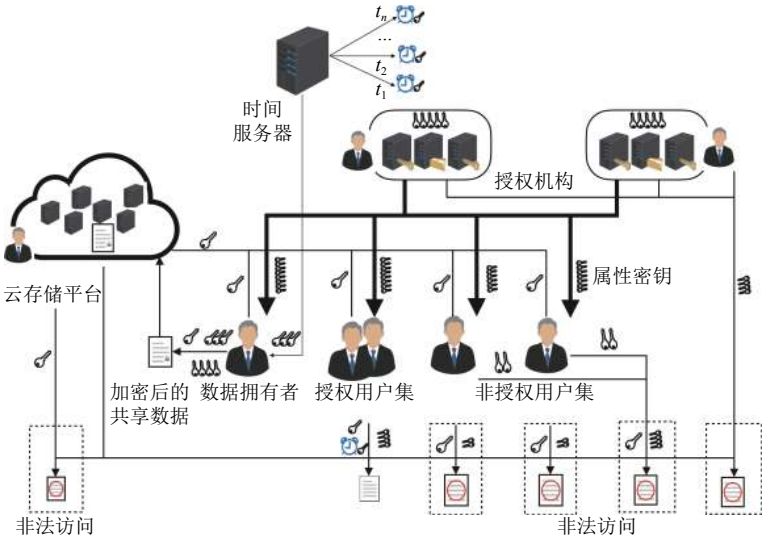

图 3 时释性加密的 CP-ABE 方案模型

3.2 安全威胁

在第 3.1 节介绍的共享模型中, 数据访问者可以是云存储环境下的任何参与实体, 可能存在想要对共享数据进行非法访问的用户; 云存储平台虽然可以按照共享机制正常执行任务, 但是出于利益考虑可能试图解密用户存储在上面的数据; 授权机构也可能为了牟利而利用用户的密钥试图解密共享数据; 在我们的安全假设中, 云存储平台和授权机构都是半可信的实体, 它们会诚实地按照规定执行每一步操作, 二者不存在合谋的可能. 在这种情况下, 基于时释性加密的 CP-ABE 方案可能存在以下 3 类安全威胁.

(1) 针对托管数据的攻击: 云存储平台和授权机构可能尝试解密共享数据, 导致数据的机密性及隐私性得不到保障.

(2) 串谋攻击: 数据拥有者加密共享数据的依据是用户的属性集合以及访问控制结构, 单个的非授权用户无法访问共享数据, 但是多个非授权用户联合各自的属性有可能解密共享数据.

(3) 选择明文攻击: 攻击者已经掌握了共享数据其中一些成对的小部分的明文与密文, 而且攻击者可以通过渗透或者冒充的方法能够让其设定好的任何明文得到共享机制的加密, 从而, 攻击者又可以得到相对应的密文, 也就是说, 攻击者拥有了共享机制中加密的权限. 这种攻击方式在云存储环境中很常见, 也是攻击者容易实现并能够获得较大优势的攻击类型.

4 基于时释性加密的 CP-ABE 方案

本文提出的基于时释性加密的 CP-ABE 方案由初始化、密钥构造、加密、解密 4 个算法组成. 时间服务器被动地产生时间公私钥对和时间陷门, 不与其它实体交互. 在密钥构造部分, 用户的私钥由两部分构成, 分别是云存储平台管理的对称密钥和授权机构产生的属性密钥. 在加密部分, 融合访问控制结构、发布时间与时间公钥对共享数据进行双重加密. 在解密部分, 在预定发布时间到达之前, 用户通过计算并采用嵌套的方式获取访问控制树根节点值, 在发布时间到达之后用户获取时间陷门并解密. 下面详细介绍方案的算法步骤.

4.1 初始化算法

初始化算法在 CP-ABE 方案中增加了时间服务器的初始化以及云存储平台对称密钥的生成.

(1) 在整个共享系统中, 需要由授权机构产生一个安全参数, 获得安全参数之后, 产生一个大素数p和随机生成元g, 接着产生阶为p的循环群G, $g \in G$, 而后构造双线性映射$e: G \times G \to G_1$.

(2) 选取以下散列函数$H: \{0,1\}^* \to G$.

(3) 随机选取$\alpha, \beta \in Z_p^*$, 构造主密钥(g^α, β)和公钥$PK(G, g, g^\beta, H, e(g,g)^\alpha)$.

(4) 共享系统中所有的属性集合由云存储平台产生, 为$\Omega = \{\lambda_1, \lambda_2, \cdots, \lambda_n\}$, 除此之外云存储平台选取$\varepsilon \in Z_p^*$作为对称密钥.

(5) 时间服务器选取$q \in Z_p^*$作为时间私钥ts_{priv}, 构造g^q作为时间公钥ts_{pub}.

4.2 密钥构造算法

密钥构造算法给 CP-ABE 方案中的用户私钥增加了云存储平台产生的对称密钥部分.

云存储平台为每一个注册进入系统的用户产生唯一的标记$g^{\varepsilon u_i}$, 根据用户的属性为其生成属性集合$S = \{\lambda_1, \cdots, \lambda_k, \cdots, \lambda_m\}$, 并分配给其对称密钥$\varepsilon$.

用户拿到属性集和标记以后, 将它们发送给授权机构, 向授权机构申请属性密钥.

授权机构根据得到的属性集合为其中的每个属性$\lambda_j \in S$随机分配一个参数$r_j \in Z_p^*$, r_j为Z_p^*中的随机值, 之后使用这个随机分配的参数与用户的标记$g^{\varepsilon u_i}$为其相对应的属性λ_j产生一个参数集合$\{g^{\varepsilon u_i} H(\lambda_j)^{r_j}, g^{r_j}\}$. 最后, 根据双线性映射计算出用户的属性密钥并发送给用户, 计算过程如式 (1) 所示:

$$\begin{aligned} SK_{u_i}^G &= \left(D = g^{\frac{\alpha}{\beta}} g^{\frac{u_i \varepsilon}{\beta}};\right. \\ &\quad \left. \forall \lambda_j \in S: D_k = g^{u_i \varepsilon} H(\lambda_j)^{r_j}, D_k' = g^{r_j}\right) \\ &= \left(D = g^{\frac{\alpha + u_i \varepsilon}{\beta}};\right. \\ &\quad \left. \forall \lambda_j \in S: D_k = g^{u_i \varepsilon} H(\lambda_j)^{r_j}, D_k' = g^{r_j}\right) \end{aligned} \tag{1}$$

最终, 用户结合云存储平台产生的对称密钥和授权机构产生的属性密钥, 形成了自己的私钥SK_{u_i}为:

$$\begin{aligned} SK_{u_i} = &\left(D = g^{\frac{\alpha + u_i \varepsilon}{\beta}}, \varepsilon;\right. \\ &\left. \forall \lambda_j \in S: D_k = g^{\varepsilon u_i} H(\lambda_j)^{r_j}, D_k' = g^{r_j}\right) \end{aligned}$$

4.3 加密算法

加密算法改变了原有 CP-ABE 方案中对共享数据的单次加密策略, 转变为由发布时间和时间公钥参加的二次加密. 数据拥有者在上传明文m到云存储平台之前, 需要制定访问控制结构并构造访问控制树, 同时设定发布时间T, 而后先使用时间公钥和发布时间T对明文m进行初次加密得到密文M, 接着再使用公钥PK、对称密钥和访问控制树对密文M进行二次加密得到密文C_T.

(1) 构造访问控制树

数据拥有者首先根据自己的需要制定访问控制结构, 而后构建访问控制树, 树中的叶子节点为访问控制结构中的属性, 树中的父节点为门限逻辑运算符.

访问控制树的根节点为 R, 数据拥有者设置 R 的节点值为随机数$s \in Z_p^*$, 并为其定义一个多项式$f_R(x)$, 令$f_R(0) = s$; 然后, 数据拥有者为其他父节点进行编码, 编码值为$index(x)$, 表明这个节点是其父节点$parent(x)$的第$index(x)$的左子节点; 接着为这些编码好的每个父节点也定义一个多项式f_x, 并令$f_x(0) = f_{parent(x)}(index(x))$.

(2) 加密共享数据

数据拥有者预先设定好发布时间T, 并得到时间服务器发布的公钥ts_{pub}, 利用散列函数H、双线性映射e、主密钥β、访问控制树根节点 R 的节点值s和ts_{pub}共同加密明文数据m, 得到密文 M. 加密过程如式 (2) 所示:

$$M = e\left(ts_{\text{pub}}, H(T)^{\beta s}\right) m = e\left(g^q, H(T)^{\beta s}\right) m \tag{2}$$

假设集合$\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$是数据拥有者设定好的访问控制树中的属性集合, 则数据拥有者对密文M进行再次加密得到密文C_T, 而后将C_T发送给云存储平台并存放在那里, 加密过程如式 (3) 所示:

$$\begin{aligned} C_T &= (Enc(M, \Gamma, PK, \varepsilon, s), T) \\ &= \left(\Gamma, Me(g,g)^{\alpha \varepsilon s}, g^{\beta s}, T;\right. \\ &\quad \left. \forall \lambda_k \in Leaf: C_k = g^{f_k(0)}, C_k' = H(\lambda_k)^{f_k(0)}\right) \end{aligned} \tag{3}$$

4.4 解密算法

解密算法在 CP-ABE 的最终解密过程中融入了 TRE 中时间陷门与时间私钥的解密思想.

当数据拥有者把密文C_T存放至云存储平台以后, 共享系统中所有的用户都可以向云存储平台进行查询并下载得到C_T, 而后试图去解密它. 整个解密过程分为两步:

(1) 数据访问者通过计算并采用嵌套的方法获取到访问控制树根节点 的节点值.

1) 数据访问者拿到访问控制树Γ, 找出其中的每个叶子节点x, 而后使用自己的私钥SK_{u_i}对其执行解密操作.

如果访问控制树的叶子结点所代表的属性λ_x在数据访问者自己的属性集中, 即$\lambda_k \in S$, 那么就计算T_x, 计算公式如式 (4) 所示; 否则, 令$T_x = 0$.

$$\begin{aligned} T_x &= \frac{e(D_x, C_x)}{e\left(D'_x, C'_x\right)} = \frac{e\left(g^{u_i\varepsilon} H(\lambda_x)^{r_x}, g^{f_x(0)}\right)}{e\left(g^{r_x}, H(\lambda_x)^{f_x(0)}\right)} \\ &= \frac{e\left(g^{u_i\varepsilon}, g^{f_x(0)}\right) e\left(H(\lambda_x)^{r_x}, g^{f_x(0)}\right)}{e(g, H(\lambda_x))^{r_x f_x(0)}} \\ &= e(g,g)^{u_i\varepsilon f_x(0)} \end{aligned} \tag{4}$$

2) 对于访问控制树剩下的父节点y, 数据访问者自底向上依次解密.

令N为父节点y的一切子节点集合, 若$(z \in N) \Lambda (T_z \neq 0)$, 且集合容量不小于$y$的门限值, 那么就计算$T_y$, 计算公式如式 (5) 所示; 否则, 令$T_y = 0$.

$$\begin{aligned} T_y &= \prod_{z \in N} T_z^{\frac{\prod_{j \in N, j \neq N} index(j)}{index(z) - index(j)}} \\ &= \prod_{z \in N} e(g,g)^{u_i\varepsilon f_{parent(z)}(index(z)) \prod_{j \in N, j \neq N} \frac{index(j)}{index(z) - index(j)}} \\ &= e(g,g)^{u_i\varepsilon \sum_{z \in N} \left(f_y(index(z)) \prod_{j \in N, j \neq N} \frac{index(j)}{index(z) - index(j)} \right)} \\ &= e(g,g)^{u_i\varepsilon f_y(0)} \end{aligned} \tag{5}$$

式 (5) 被数据访问者嵌套计算, 只有当访问者的私钥满足访问控制树, 即访问者属性的确符合数据拥有者指定的访问控制结构, 属于授权用户, 访问者才可以计算出$T_R = e(g,g)^{u_i\varepsilon s}$, T_R为访问控制树根节点的节点值; 否则$T_R = 0$.

接着等预定的发布时间T到达以后, 数据访问者获取时间服务器在T时刻发布的时间陷门$H(T)^q$, 进而解密得到明文数据m. 解密公式如式 (6) 所示:

$$\begin{aligned} m' &= \frac{Me(g,g)^{\alpha\varepsilon s}}{\left[\frac{e\left((g^{\beta})^s, g^{\frac{\alpha + u_i\varepsilon}{\beta}} \right)}{T_R} \right]^{\varepsilon} * e(g^{\beta s}, H(T)^q)} \\ &= \frac{Me(g,g)^{\alpha\varepsilon s}}{\left[\frac{e\left((g^{\beta})^s, g^{\frac{\alpha + u_i\varepsilon}{\beta}} \right)}{e(g,g)^{u_i\varepsilon s}} \right]^{\varepsilon} * e(g^{\beta s}, H(T)^q)} \\ &= m \end{aligned} \tag{6}$$

最终数据访问者在T时刻到来时, 成功解密得到明文数据m.

5 安全分析

5.1 基于时释性加密的 CP-ABE 方案能够抵抗多种不同来源的非法访问.

(1) 能够抵抗来自非法用户的非法访问.

非法用户即非授权用户, 这类用户若想要解密共享数据, 就需要充分利用共享系统中的相关消息. 通过分析整个共享机制可以看出, 非法用户能够利用的消息有: 云存储平台中的密文$g^{\beta s}$以及用户私钥SK_{u_i}中的D和$g^{u_i\varepsilon}$.

我们对以上消息构造双线性映射, 如式 (7) 所示:

$$\begin{aligned} e\left(D, g^{\beta s}\right) &= e\left(g^{\frac{\alpha + u_i\varepsilon}{\beta}}, g^{\beta s}\right) \\ &= e(g,g)^{\alpha s} e(g,g)^{u_i\varepsilon s} \end{aligned} \tag{7}$$

通过式 (7) 可以看出, 只有获取到s, 即根节点的节点值, 才能够解密共享数据. 非法用户因其属性不满足数据拥有者设定的访问控制结构而无法获取访问控制树根节点的节点值s, 没有办法执行解密步骤, 也就无法解密还原共享数据. 也就是说, 本文的方案可以使得非法用户对共享数据的解密无从下手, 使其攻击行为不能得逞.

(2) 能够抵抗来自云存储平台的非法访问.

数据拥有者加密共享数据时会使用云存储平台产生的对称密钥ε, ε虽然由云存储平台进行维护, 但是要想解密共享数据, 必须首先还原出访问控制树根节点的值s, 而仅仅通过ε不可能获取到s的值, 因为访问控制树是经过数据拥有者加过密之后上传至云存储平台的, 因此云存储平台没有办法执行解密步骤, 也就无法解密获得共享数据. 也就是说, 本文的方案可以有效阻止云存储平台访问共享数据.

(3) 能够抵抗来自授权机构的非法访问

授权机构能获取到属性私钥. 由于用户的私钥是由属性私钥和对称密钥ε共同组成, 要想进一步解密共享数据, 授权机构就必须首先拿到ε, 根据双线性映射理论, 授权机构无法从用户的标记中得到ε, 则它就必须和云存储平台合谋. 由于在我们的安全假设中, 二者属于半可信实体, 不存在合谋的可能. 因此, 授权机构也无法解密获得共享数据. 因此, 本文的方案能够有效地阻止授权机构非法访问共享数据.

综上所述, 本文提出的共享机制能够抵抗系统中不同参与实体的非法访问, 进而解决了托管数据的机密性问题.

5.2 基于时释性加密的 CP-ABE 方案能够抵抗不同用户的串谋攻击.

假设存在两个非法用户u_i和u_j想要解密共享数据, 他们想联合各自的属性私钥来直接执行解密程序, 在这之后, 需要根据式 (4) 解密叶子结点, 则他们能够得到的叶子结点值集合如式 (8) 所示:

$$\begin{aligned}&\left\{e(g,g)^{u_i f_x(1)},\cdots,e(g,g)^{u_i f_x(m)}\right.\\&\left.e(g,g)^{u_j f_x(n)},\cdots,e(g,g)^{u_j f_x(k_x)}\right\}\end{aligned}\tag{8}$$

可以看到, 叶子结点的值分别都具有用户u_i和u_j的标记, 因此无法利用式 (5) 继续进行父节点的还原, 也就无法获得访问控制树根节点的节点值, 进而也就无法执行解密步骤.

综上所述, 多个非授权用户联合各自的属性不可能解密还原共享数据, 本文提出的共享机制可以有效地抵抗串谋攻击.

5.3 基于时释性加密的 CP-ABE 方案能够抵抗选择明文攻击.

定义 1. 关于 CP-ABE 的选择明文攻击游戏[13]

攻击游戏是模拟敌手和挑战者交互的一个过程, 从而把现实中的攻击行为的全过程展现出来. 关于 CP-ABE 的选择明文攻击游戏, 挑战者和敌手的具体交互过程如下:

(1) 系统建立: 在模拟云存储环境中, 挑战者根据我们的密文共享机制进行云存储共享系统的构建, 产生公私密钥, 敌手可以拿到公钥.

(2) 初步查询: 敌手拿到挑战者给出的公钥之后, 可以向挑战者发出查询请求, 敌手想要查询与属性集关联的用户私钥.

(3) 进行质询: 敌手此时已经拿到存放在云存储平台的访问控制树Γ, 同时, 敌手拿到了部分明文, 于是敌手选取了 2 个相同长度的明文, 开始进行属性集比对, 发现上一步查询到的属性集均不满足访问控制树Γ. 此时, 将自己手中的 2 个明文分别命名为m_0和m_1, 将m_0和m_1以及访问控制树Γ一起给挑战者发送过去. 挑战者收到以后, 会生成随机值$b\in\{0,1\}$, 而后根据设定好的加密策略对m_b完成加密操作, 输出密文C并给敌手发送过去.

(4) 继续查询: 敌手继续向挑战者申请查询操作, 查询与属性集关联的用户私钥, 但是已经查询到的属性集仍然不满足访问控制树Γ.

(5) 进行猜测: 此时挑战者必须给出关于密文C的猜测, 即挑战者必须给出如下回答: $b'=0$或者$b'=1$.

如果挑战者猜测的$b'=b$, 那么此时敌手获得胜利, 敌手的优势可以如式 (9) 定义:

$$\left|\Pr[b'=b]-\frac{1}{2}\right|\tag{9}$$

由于我们的安全机制是建立在 DBDH 问题的基础之上的, 因此, 在任意概率的多项式时间内, 若敌手在 DBDH 游戏中的攻击优势可以被忽略, 那么就可以说明我们的密文共享机制可以抵抗选择明文攻击, 即是选择明文安全的. 证明过程如下:

假设存在一个攻击者 A, 要想攻破我们的密文共享机制, 则 A 为敌手, 同时需要构建 DBDH 的攻击者, 可以通过构建一个模拟器来充当 DBDH 的攻击者, 这个模拟器被命名为 B, 模拟器 B 攻破 DBDH 问题的优势为δ, 通过 DBDH 问题可以得出, δ可以被忽略.

(1) 系统建立: 在模拟云存储环境中, 模拟器 B 产生 3 个随机值, 分别为: $a,b,c\in Z_p^*$, 同时模拟器 B 还获取到了共享机制的公钥$PK\left(G,g,g^{\beta},H,e(g,g)^{\alpha}\right)$, 模拟器 B 进行如下计算: $ab+x=\alpha$, 由此得到了x的值.

(2) 初步查询: 模拟器B拿到授权机构给出的公钥PK之后, 攻击者A可以向模拟器B发出查询请求, 攻击者A把需要查询的属性集S提交给模拟器B, 同时提交的还有用户的标记$g^{u_i\varepsilon}$. 模拟器B拿到属性集S后, 会选取随机值r_j分配给属性集S中的每个属性元素, 之后, 模拟器B会计算相应的用户私钥并发送给攻击者A, 计算过程如式 (10) 所示:

$$\left\{g^{\frac{ab+x+u_i\varepsilon}{b}},\forall\lambda_j\in S:g^{u_i\varepsilon}H\left(\lambda_j\right)^{r_j},g^{r_j}\right\}\tag{10}$$

至此, 共享机制当中的授权机构成功被模拟器B模拟出来.

(3) 进行质询: 攻击者A此时已经拿到存放在云存储平台的访问控制树Γ, 同时, 攻击者A拿到了部分明文, 于是攻击者A选取了 2 个相同长度的明文, 开始进行属性集比对, 发现步骤 (2) 查询到的属性集均不满足访问控制树Γ. 此时, 攻击者A将自己手中的 2 个明文分别命名为m_0和m_1, 将m_0和m_1以及访问控制树Γ一起给模拟器B发送过去. 模拟器B收到以后, 会连同访问控制树根节点的值c一起发送给授权机构. 授权机构会生

成随机值$b \in \{0,1\}$, 而后根据设定好的加密策略对m_b完成加密操作, 输出密文为: $M_b e(g,g)^{\alpha s}$和$M_{\tilde{b}} e(g,g)^{zs}$, 其中, $z \in Z_p^*$是随机值.

(4) 继续查询: 攻击者A继续向模拟器B申请查询操作, 查询与属性集关联的用户私钥, 但是已经查询到的属性集仍然不满足访问控制树Γ.

(5) 进行猜测: 此时攻击者A必须给出关于密文的猜测, 即攻击者A会给出自己的猜测值b', 模拟器B利用攻击者A的猜测值也要做出自己的猜测, 猜测是否正确, 需要根据式 (11) 判断:

$$\begin{aligned} M_{b'} e(g,g)^{\alpha c} &= M_{b'} e(g,g)^{(ab+x)c} \\ &= M_{b'} e(g,g)^{abc} e(g^x, g^c) \end{aligned} \tag{11}$$

如此, 密文数据被模拟器B分析出来的概率为δ. 若是猜测结果不对, 密文数据就没有办法被模拟器B判断, 因为$z \in Z_p^*$是随机值.

综上, 共享系统的密文机制被攻击者攻破的概率为$\delta/2$, 根据 DBDH 问题, 这个概率可以被忽略, 即本文提出的密文安全共享机制可以有效抵抗选择明文攻击.

6 结束语

本文指出在云存储环境下基于用户属性和访问时间进行细粒度访问控制将成为一种迫切需求, 并提出了基于时释性加密的 CP-ABE 方案, 通过在 CP-ABE 中融入 TRE 机制来实现带有时间控制的密文共享机制. 该方案大大减少了对时间服务器和第三方机构的依赖, 能够有效抵抗来自用户、云存储平台和授权机构的非法访问、非法用户的串谋攻击以及选择明文攻击. 未来将重点研究用户密钥的撤销和更新问题, 保证共享数据不被注销后的用户解密获取.

参考文献

1 俞能海, 郝卓, 徐甲甲, 等. 云安全研究进展综述. 电子学报, 2013, 41(2): 371–381. [doi: 10.3969/j.issn.0372-2112.2013.02.026]

2 Goyal V, Pandey O, Sahai A, *et al*. Attribute-based encryption for fine-grained access control of encrypted data. Proceedings of the 13th ACM Conference on Computer and Communications Security. Alexandria, VA, USA. 2006. 89–98.

3 Cathalo J, Libert B, Quisquater JJ. Efficient and non-interactive timed-release encryption. Proceedings of the 7th International Conference on Information and Communications Securit. Beijing, China. 2005. 291–303.

4 Dong X, Yu JD, Luo Y, *et al*. Achieving an effective, scalable and privacy-preserving data sharing service in cloud computing. Computers & Security, 2014, 42: 151–164.

5 Yang K, Jia XH, Ren K, *et al*. DAC-MACS: Effective data access control for multiauthority cloud storage systems. IEEE Transactions on Information Forensics and Security, 2013, 8(11): 1790–1801. [doi: 10.1109/TIFS.2013.2279531]

6 关志涛, 杨亭亭, 徐茹枝, 等. 面向云存储的基于属性加密的多授权中心访问控制方案. 通信学报, 2015, 36(6): 116–126.

7 Rivest RL, Shamir A, Wagner DA. Time-lock puzzles and timed-release crypto. Cambridge, MA: MIT LCS Tech, 1996.

8 Chan ACF, Blake IF. Scalable, server-passive, user-anonymous timed release cryptography. 25th IEEE International Conference on Distributed Computing Systems. Columbus, OH, USA. 2005. 504–513.

9 Hwang YH, Yum DH, Lee PJ. Timed-release encryption with pre-open capability and its application to certified E-mail system. Proceedings of the 8th International Conference on Information Security. Singapore. 2005. 344–358.

10 Cheon JH, Hopper N, Kim Y, *et al*. Provably secure timed-release public key encryption. ACM Transactions on Information and System Security, 2008, 11(2): 4.

11 袁科, 刘哲理, 贾春福, 等. TRE 加密技术研究. 计算机研究与发展, 2014, 51(6): 1206–1220. [doi: 10.7544/issn1000-1239.2014.20130177]

12 谷利泽, 郑世慧, 杨义先. 现代密码学教程. 2 版. 北京: 北京邮电大学出版社, 2015.

13 Doshi N, Jinwala DC. Fully secure ciphertext policy attribute-based encryption with constant length ciphertext and faster decryption. Security and Communication Networks, 2014, 7(11): 1988–2002. [doi: 10.1002/sec.913]