基于属性加密的组合文档安全自毁方案

熊金波[1,2],姚志强[1,2],马建峰[1,2],李凤华[3],刘西蒙[2],李 琦[2]
(1.福建师范大学软件学院,福建福州 350108;2.西安电子科技大学计算机学院,陕西西安 710071;
3.中国科学院信息工程研究所信息安全国家重点实验室,北京 100093)

摘 要: 为保护组合文档全生命周期的隐私安全,提出了一种基于属性加密的组合文档安全自毁方案.该方案引入多级安全思想创建新的组合文档结构,采用访问密钥加密组合文档内容、基于属性的加密算法加密访问密钥,两者的密文经过一系列算法提取和变换后获得密文分量和封装自毁对象,分别存储在两个分布式哈希表网络和云服务器中.当组合文档过期后,该网络节点将自动丢弃所存密文分量,使得原始组合文档密文和访问密钥不可恢复,从而实现安全自毁.安全分析表明,该方案既能抵抗传统的密码分析或蛮力攻击,又能抵抗分布式哈希表网络的Sybil攻击.

关键词: 组合文档;隐私安全;基于属性的加密;安全自毁;分布式哈希表网络
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2014)02-0366-11
电子学报URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.02.024

A Secure Self-Destruction Scheme for Composite Documents with Attribute Based Encryption

XIONG Jin-bo[1,2], YAO Zhi-qiang[1,2], MA Jian-feng[1,2], LI Feng-hua[3], LIU Xi-meng[2], LI Qi[2]
(1. *Faculty of Software, Fujian Normal University, Fuzhou, Fujian* 350108, *China*; 2. *School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi* 710071, *China*; 3. *State Key Laboratory of Information Security, Institute of Information Engineering, Chinese Academy of Sciences, Beijing* 100093, *China*)

Abstract: In order to protect the confidentiality and privacy security of the composite documents within the whole life cycle, we leverage attribute-based encryption (ABE) algorithm to propose a secure self-destruction scheme for composite documents, referred to as SelfDoc. We firstly set up a new composite document structure by employing the idea of multilevel security, and then, use an access key to encrypt the composite document and the ABE algorithm to encrypt the access key. After a series of algorithms for extracting and transforming from the two ciphertexts, we obtain ciphertext shares and encapsulated self-destruction object, both of which are stored into two distributed hash table (DHT) networks and cloud servers respectively. Nodes in the DHT networks will self-discard the ciphertext shares periodically, so the original ciphertext and access key are unrecoverable after the expiration time, and the function of secure self-destruction is achieved. Compared with existing schemes, the security analyses indicate that SelfDoc scheme is able to resist the traditional cryptoanalysis or brute-force attack, and the Sybil attacks from the DHT networks.

Key words: composite document; privacy security; attribute-based encryption; secure self-destruction; DHT network

1 引言

云计算技术和新兴服务的迅速发展,促使文档的概念发生巨大转变.传统基于单个文件、单一格式的文档,如Web页面、PDF文档、图像文件、视/音频片断、Word文档、Excel表格和CAJ/KDH文档等,不再适应云服务对多样化格式、多种类型文档组合服务的新需求[1].

而组合文档是云服务环境中承载云服务和复杂工作流的数据载体.如国际项目联合申请、国际医疗联合会诊服务等,都需要创建和管理复杂的项目组合文档和医疗组合文档.不再像传统单一文件参与者仅属于单一的安全环境,这类组合文档以工作流的方式存在和处理,其参与者分布在全球范围,导致组合文档的处理需要跨越多个安全域且传递在不安全的信道中[2].此外,组合文档通常包含不同类型的隐私信息,如商业竞争分析、新产品定价战略、知识产权和个人电子健康记录等.

收稿日期:2013-07-15;修回日期:2013-11-09;责任编辑:覃怀银
基金项目:长江学者和创新团队发展计划(No.IRT1078);国家自然基金委员会-广东联合基金重点基金(No.U1135002);国家自然科学基金(No.61370078,No.61170251);国家科技重大专项(No.2011ZX03005-002);国家863高技术研究发展计划(No.2012AA013102)

这些隐私信息具有不同的安全级别,分属于组合文档的不同部分,在各个工作流步骤中被不同的参与者处理.可公开发布的组合文档(Publicly Posted Composite Documents,PPCD)[2,3]是近年提出的一种新型组合文档结构和机制,支持多工作流参与者的处理,能够跨越多个不同的安全域并传递在非安全信道中.文献[2]采用PKI方案和文献[3]采用基于身份的加密机制(Identity-Based Encryption,IBE)均在一定程度上实现了PPCD文档工作流处理过程中隐私信息的安全访问[4].以上特征使得PPCD适合于云服务环境中组合文档工作流的创建和处理.

然而,在PPCD工作流处理结束后,组合文档中隐私信息的安全保护往往被忽视,文献[2,3]均未论及这一问题.而隐私信息在过期后被泄漏能够给个人或单位带来严重后果.因此,保护组合文档过期后的隐私信息安全极其重要[5].最便捷的方式是直接删除这些包含隐私信息的组合文档.然而,从本地或Web页面中删除存储在云服务器中的组合文档并不能实现真正删除[6].研究人员证实,可以通过智能手机完全恢复出从Dropbox、Box和SugarSync中删除的图片、音频文件、PDF和Word文档等[7].

因此,安全删除云端数据极具挑战性,最先研究该问题的是文件确保删除[8,9],描述了一个能支持本地文件过期后确保删除的系统.Tang等人对其进行扩展,在已有云计算基础设施上构建了一个安全覆盖云存储系统[10,11],能支持灵活的文件访问策略且确保满足策略的文件安全删除.以上文献均属于集中式解决方案,需要额外指令或操作才能达到文件确保删除.

Geambasu等人提出了一种无需人工干预实现敏感数据自毁的Vanish系统[6],将加密敏感数据的对称密钥经过秘密共享处理后将密钥分量分发到分布式哈希表(Distributed Hash Table,DHT)网络中,利用该网络节点数据自更新功能丢弃所存密钥分量使得原始密钥不可恢复,从而敏感数据密文不能被解密和访问,实现该意义上的数据自毁.Wang等人改进Vanish系统并提出SSDD方案[12],将密钥和提取的部分密文一起分发到DHT网络中,从而提高方案的安全性.王等人在Vanish系统的基础上提出一种云环境中的数据确定性删除方法[13],采用密钥派生树组织并管理密钥,以有效提高密钥管理效率.

然而,Wolchok等人指出,Vanish系统采用的Vuze DHT网络存在Sybil攻击,敌手在过期之前能够获取到足够多的密钥分量以重构解密密钥[14].同样的,文献[6,12,13]等方案均易遭受这类攻击.针对该问题,Zeng等人提出SafeVanish方案[15],在密钥分量中增加随机值以扩展分量的长度,从而抵抗Vanish系统[6]中存在的Sybil跳跃攻击;同时,采用RSA加密对称密钥以抵抗Sybil嗅探攻击.然而,Vanish系统和SafeVanish方案均将完整的密文保存在云端,存在可能的蛮力攻击和密码分析攻击.Xiong等人在SSDD的基础上提出ISDS方案,利用IBE加密对称密钥后既能扩展密钥分量的长度以抵抗Sybil跳跃攻击,又能抵抗Sybil嗅探攻击,但存在身份隐私泄露的风险[5].综上所述,以上方案均存在不同程度的局限性:①均需要"封装对象在过期之前未受攻击"的理想假设;②大多数方案仅采用对称加密算法而面临复杂的密钥分发和密钥管理问题;③在隐私信息有效期内,不能提供多级安全和细粒度访问控制[16];④虽能达到敏感数据或文件的确保删除,但方案本身易遭受攻击.

为了解决以上问题,保护组合文档在生命周期内(组合文档工作流处理过程中)和过期后(使用价值结束)的隐私信息安全,本文创新组合文档的设计思路,结合基于属性的加密(Attribute-Based Encryption,ABE)和大规模分散的DHT网络提出基于属性加密的组合文档安全自毁方案,简称SelfDoc方案.其贡献主要体现在:

(1)融合多级安全思想,将组合文档的组件划分安全等级,相同安全等级的组件共享同一个对称密钥[16].相比原PPCD而言,减少了对称密钥的数量,简化了复杂的密钥管理;改进了原有PPCD结构,减小组合文档工作流在云端服务器的存储开销,节约成本.

(2)采用ABE算法,无需为工作流参与者颁发证书,能够缓解SafeVanish方案[15]中使用PKC而导致复杂的证书更新与密钥管理问题;在组合文档生命周期内,提供细粒度访问控制.

(3)不需要理想化假设的前提下,采用两个DHT网络分别存储不同的密文分量信息,确保当组合文档过期后实现自毁的同时达到系统安全,不仅能够抵抗过期后的传统密码分析和蛮力攻击,过期前的DHT网络跳跃和嗅探等Sybil攻击,还可任意时刻抵抗这二者的同时攻击.

2　预备知识

2.1　PPCD技术概述

云服务环境中,包含不同隐私信息的复杂组合文档工作流的处理需要PPCD技术,以支持组合文档的安全创建与管理[2,3].

PPCD的结构由3部分顺序组成:记录表(entry-table),密钥映射记录(key-map entries)和内容部分(content-parts)[3].

(1)内容部分.PPCD由多个内容部分组成,每个部分都可能由不同的参与者处理,对应不同的访问密钥.

(2)密钥映射记录.该记录是参与者访问密钥到内

容部分的映射,访问密钥是内容部分密钥(加密、解密、签名和验证4个密钥,其中加密和解密一般采用对称密钥)的子集.一旦获取记录中的访问密钥,参与者即可解密对应的内容部分.每条密钥映射记录均有记录名.

(3)记录表.每个参与者都有对应的记录条目,该条目用于快速识别和定位对应的密钥映射记录名.

PPCD工作流在工作步骤中,一旦被某个参与者接收,参与者便从记录表中查找并解密自己的条目,即可找到对应的密钥映射记录名,并从中获取文档名和访问密钥.解密对应文档并在处理完成后对文档加密并签名,然后传递给下一个流程的参与者.

SelfDoc方案在PPCD的基础上,简化了PPCD的结构以节约存储开销和成本,并减少密钥的数量以简化复杂的密钥管理问题.

2.2 ABE基础

ABE是一种典型的公钥密码方案,最早的ABE是由Sahai和Waters在2005年欧密会上提出的模糊身份加密方案[17].与IBE方案中的身份标识符为一个字符串不同,在ABE方案中,标识符为一组描述性属性的集合.系统中每个属性用Hash函数映射到$\mathbb{Z}_q^*$中,密文和用户密钥都与属性关联.ABE支持基于属性的门限策略,使用属性集合ω解密由属性集合ω^*加密消息的密文,当且仅当属性集合ω与ω^*中重叠的属性个数大于或等于某个门限阈值d时,能够成功解密.

授权中心Authority先发布系统公钥,系统公钥与属性集ω^*关联,公钥长度与ω^*中属性数目线性相关,公钥用于加密消息,并规定门限参数d.授权中心依据用户提供的属性集ω为用户创建私钥,属性集ω关联一个随机的$d-1$阶的多项式$q(x)$并规定$q(0)=y$.对于用户的解密过程,系统首先判断$|\omega\cap\omega^*|\geq d$是否成立,如果成立,则任选$|\omega\cap\omega^*|$中$d$个属性,即可由关联的多项式$q(x)$通过拉格朗日插值法恢复出加密密钥,从而解密出明文消息.

在SelfDoc方案中,利用ABE加密密钥映射记录名和访问密钥,经过相关处理产生密文分量并分发到DHT网络中,以抵抗Sybil攻击.

2.3 门限秘密共享

Shamir提出门限秘密共享方案,将需要共享的秘密S分解成n个分量:$S_1,S_2,\cdots,S_i,\cdots S_n$.从这些分量中提取任意大于等于$d$个$S_i$,能够恢复出原始秘密$S$;反之不能.该方案称之为$(d,n)$门限秘密共享[18].

在SelfDoc方案中,密钥映射记录名的密文属性分量被分发到DHT网络中;此外,访问密钥经过ABE加密后的密文属性分量和提取出的部分组合文档密文一起产生密文分量被分发到另一个DHT网络中.即使某个时刻DHT网络中的部分节点动态退出导致部分密文分量不可用,依据门限秘密共享方案的冗余性,只需要获取到任意大于等于d个密文分量就可以恢复出原始密文.

2.4 DHT网络概述

DHT网络是一种实现查询、存储、检索和管理数据的全球规模分布式对等网络[19,20].现有Internet中存在多种实用的DHT网络,典型代表有Vuze、CAN和OpenDHT等.以上DHT网络都具备如下特征,适合于构造SelfDoc方案:

(1)可用性.DHT网络支持可靠的分布式存储,节点中所存数据在一定期限内确保可用.可用性是DHT网络用于构造SelfDoc方案的基础.

(2)节点定时自动更新.每隔一定的时间期限,节点会自动丢弃原有数据以保存新数据.该功能使得节点中存储的密文分量在过期后能够自动丢弃而不可恢复,为SelfDoc安全自毁提供实现机制.

(3)大规模且全球分布.Falkner等指出在Vuze网络中同时在线的活动节点超过百万个,且全球分布超过190个国家[20].这种完全分散的DHT网络能够为SelfDoc方案提供健壮的抗攻击能力.

3 需求、假设和模型

为了实现SelfDoc方案,首先给出一个新的数据结构,称为组合文档自毁对象(Composite documents Self-destructing Object,CSO),CSO封装包含隐私信息的组合文档密文并阻止泄露给任意非授权第三方.

3.1 设计需求

SelfDoc方案应该满足如下设计需求:

(1)组合文档工作流生命周期内的可用性和可控性.在组合文档工作流各步骤处理过程中,CSO的隐私内容对工作流参与者是可用的;同时,需要实现不同参与者访问不同安全等级的文档组件,达到细粒度访问控制.

(2)过期后自毁.CSO必须在过期后能够自动销毁而无需人干预.SelfDoc方案提供前向安全,即便敌手在过期前获得完整的CSO副本和所有相关解密密钥,一旦过期CSO便对任何人不可读,包括工作流创建者和参与者,从而对组合文档机密性和隐私提供全生命周期安全保护.

(3)简单高效的密钥管理机制.现有方案仅采用对称加密算法,容易导致复杂的密钥管理问题,SelfDoc方案应具有简单高效的密钥管理机制.

(4)能够抵抗各种攻击.SelfDoc方案要求不仅能够抵抗传统的密码分析和蛮力攻击,以及DHT网络的Sybil攻击,还能够抵抗以上两种类型的同时攻击.

3.2 方案假设

为了满足 SelfDoc 方案的设计需求,在文献[6,12]的基础上做出如下安全假设:

(1)组合文档工作流具有时效性.SelfDoc 方案用于保护组合文档中隐私信息的安全,该隐私信息只有在有限的时间范围内对参与者有效和可用.

(2)网络连接.工作流创建者和参与者都可以连接到 Internet,以便能够与授权机构、云服务器和 DHT 网络进行交互,实现生命周期内相关密钥和密文的分发和提取,以及提交与获取 CSO.

(3)授权机构、工作流创建者和参与者可信.授权机构是可信服务器,是 ABE 的核心组件,负责产生系统公共参数和生成 ABE 公/私钥.工作流创建者和参与者是可信的,不会主动泄露或备份 CSO 或相关密钥数据.

(4)云服务器是不安全的.云服务器在提供存储服务的同时,可能存储多个 CSO 的备份,或者泄露给非授权实体.

3.3 系统模型

SelfDoc 方案的系统模型如图 1 所示,包含 6 个实体,分别为:组合文档工作流创建者、云服务器、授权机构、工作流参与者、DHT 网络和潜在敌手.

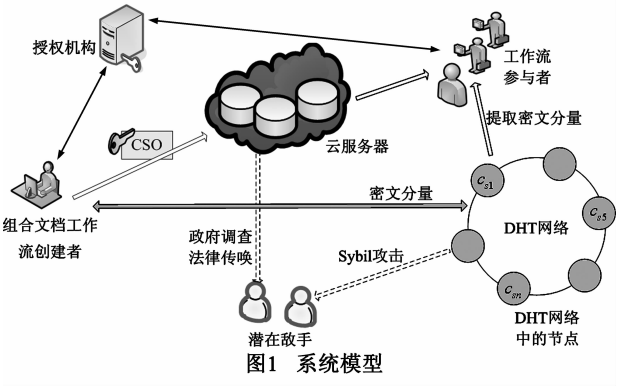

图1 系统模型

组合文档工作流创建者:创建者将包含隐私信息的组合文档封装成 CSO,并发送到云端服务器存储;并产生密文分量分发到 DHT 网络中.

云服务器:负责存储工作流创建者封装的 CSO,并确保只有经过认证的工作流参与者才能访问 CSO.

授权机构:负责系统所需的公共参数和支持 ABE 密钥的产生和处理.

工作流参与者:组合文档创建者指定的参与工作流处理的隐私信息共享者.不同的参与者参与不同的工作流步骤,拥有不同的访问权限,处理不同的文档组件.

DHT 网络:DHT 网络中的节点负责存储密文分量,并在过期后自动丢弃节点中的密文分量使得原始密文和解密密钥不可恢复.

潜在敌手:可能分别攻击 CSO 或 DHT 网络,或者同时发起攻击的非授权实体.

3.4 攻击模型

SelfDoc 方案的安全目标是确保经过预定义的时间段后,存储在云服务器中的 CSO 能够安全自毁.由假设(3)可知,在 SelfDoc 方案中,能够访问相同 CSO 的组合文档创建者和参与者之间相互信任.

在 SelfDoc 攻击模型中,潜在敌手可以分为三类:一类是在组合文档工作流过期后攻击 CSO,如云服务器能提供 CSO 以支持法院命令或传票、恶意攻击 CSO 或不小心泄露;一类是在组合文档工作流生命周期内攻击 DHT 网络本身,如 Sybil 攻击,收集 DHT 网络节点中保存的密文分量;一类是在任意时刻,同时攻击 CSO 和 DHT 网络,试图同时获得封装的密文和密文分量.

4 SelfDoc 方案构造

SelfDoc 方案将多级安全思想、ABE、DHT 网络与组合文档工作流相结合,提供组合文档使用期限内的细粒度访问控制和过期后的安全自毁,实现全生命周期保护组合文档隐私安全.

在新组合文档结构的基础上,首先概述 SelfDoc 方案,然后从算法层面和系统层面分别描述 SelfDoc 方案的具体构造.SelfDoc 方案中基本的符号及其描述如表 1 所示.

表 1 SelfDoc 方案符号及其描述

符号	描述
p	组合文档组件的明文
C	组合文档密文
A_k	访问密钥,包含对称密钥 k
N,d	密文分量个数及门限值
u,v	每次提取的 bit 数量及提取次数
ψ	系统公开参数
$\mathbb{G}_1,\mathbb{G}_2$	素数阶为 q 的双线性群
puk,msk,prk	系统公钥,主密钥和参与者私钥
α	密钥映射记录名
$\chi_{\text{part}},\chi_{\text{user}}$	文档组件属性集,参与者属性集
K_i	参与者 ABE 私钥分量
C_α,C_{KM}	α 和 A_k 经过 ABE 加密的密文
R,L	分发索引
E_i	从第 i 个文档组件中提取的值
$C_{\text{Extra}},C_{\text{CSO}}$	提取密文和封装密文
S_{CS}	密文分量
CSO	组合文档自毁对象

4.1 组合文档新结构

首先,将组合文档依据工作流的现实需要分割成多个组件,每个组件由共享相同访问控制策略的一个或多个单独的文档或片断组成.

然后,依据每个组件中所包含信息的隐私程度的不同而分为多个安全等级,如图 2 所示.在一个组合文档工作流中,具有相同安全等级的不同组件共享同一个对称密钥 k.与原有 PPCD 结构相比,新结构可以减少对称密钥的数量和管理开销.

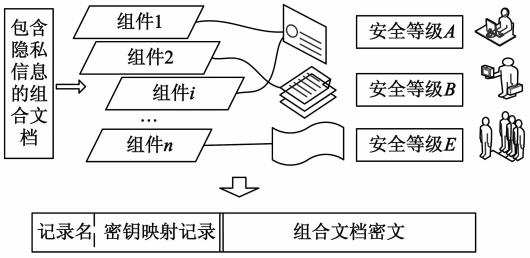

图2 组合文档新结构

在一个工作流步骤中,每个参与者可能对多个组件执行操作,需拥有对各组件操作的访问权限(只读或读写)所对应的访问密钥 A_k(包含加密密钥 k 和可能的签名/校验密钥).A_k 由工作流步骤对应文档组件的属性集 χ_{part} 加密并存储于密钥映射记录中,每个参与者都有一条对应步骤的密钥映射记录.然后,产生一个随机值作为该记录的记录名 α.最后添加随机的伪记录名以模糊化实际参与者数量.

由 A_k 加密各文档组件的组合文档密文 C 和密钥映射记录组成新的组合文档结构.与原有 PPCD 结构相比,新结构减少了记录表部分,从而减小了组合文档存储和管理开销.

4.2 SelfDoc 方案概述

SelfDoc 方案的工作原理如图 3 所示,主要分为加密(箭头向下)和解密(箭头向上)两个过程.

(1)加密过程

首先,依据组合文档新结构,使用 ABE 加密 α 得到 C_α,将 C_α 中的属性分量 $\{C_{\alpha i}=T_i^j\}_{i\in\chi_{part}}$ 结合分发索引 R 分发到第一个 DHT 网络中,封装密文 C'_α 和 R 构成元组 $<R,C'_\alpha>$ 存于授权中心.

然后,对 C 进行提取操作,将 C 变为 $C_{Extra}+C_{CSO}$;将 C_{KM} 中的 C'_{KM} 存于密钥映射记录中,将 C_{KM} 中的属性分量 $\{C_{KMi}=T_i^s\}_{i\in\chi_{part}}$ 和 C_{Extra} 构造拉格朗日多项式并产生密文分量 S_{CS},并结合分发索引 L 将 S_{CS} 分发到另外一个 DHT 网络中.

最后,将 C_{CSO} 和分发索引 L 一起封装为 CSO,将其存储到云服务器中.

(2)解密过程

该过程是工作流参与者在组合文档工作流生命周期内访问组合文档的过程,是加密过程的逆过程.

首先,参与者提交属性集 χ_{user},授权中心认证参与者之后判断如下条件是否成立:$|\chi_{part}\cap\chi_{user}|\geq d$.如果成立,参与者从授权中心获取到元组 $<R,C'_\alpha>$,提取出 R 并从第一个 DHT 网络中收集足够多的属性分量 $\{C_i=T_i^j\}_{i\in\chi_{part}}$;再从授权中心获得 ABE 私钥后即可解密 C'_α,得到记录名 α.

然后,参与者从云服务器中获取到 CSO,从中提取到 L 和 C_{CSO},依据 L 从第二个 DHT 网络中收集足够多的密文分量 S_{CS},利用拉格朗日插值法重构出 C_{Extra} 和 C_{KMi},将 C_{Extra} 和 C_{CSO} 还原为原始密文 C.

最后,利用 α 便可从密钥映射记录中找到对应的

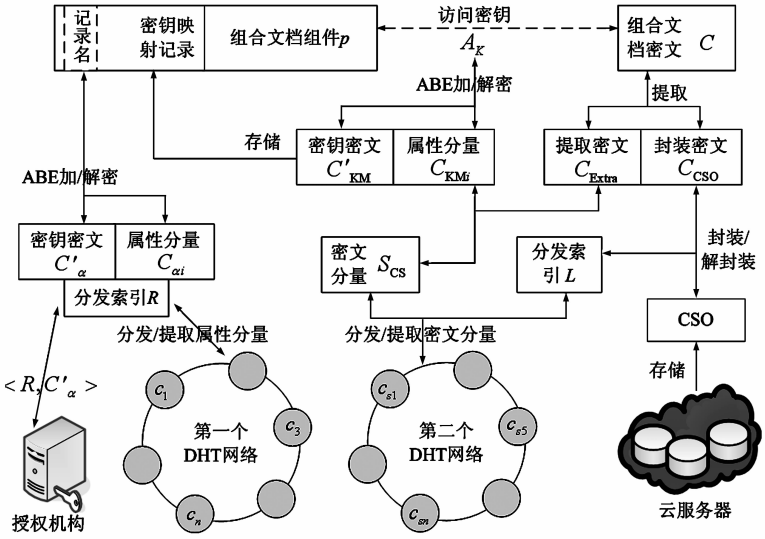

图3 SelfDoc方案工作原理

条目,将得到的属性分量 C_{KMi} 和密钥映射记录中保存的 C'_{KM} 合并为 C_{KM},再用 ABE 私钥解密 C_{KM} 得到记录中的访问密钥 A_k,从而获得其中的 k 解密原始密文 C,最终获得需要访问的组合文档明文.

组合文档工作流一旦过期,则 DHT 网络将分别自动删除节点中保存的属性分量和密文分量,因而无法重构出原始密文 C、C_{KM} 和 α,进而组合文档不可读,实现安全自毁.

4.3 SelfDoc 方案算法描述

算法层面主要完成由系统层面调用的底层算法的具体实现,主要包含以下多项式时间算法:

(1)Setup(κ)

系统建立时,给定安全参数 κ,算法产生密文分量个数 n,门限值 d,每次提取的 bit 数量 u,提取的次数 v 和一个对称加密算法 SE;此外,还产生 2 个加法循环群 $\mathbb{G}_1$ 和乘法循环群 $\mathbb{G}_2$,均为素数阶 q,以及 $e:\mathbb{G}_1\times\mathbb{G}_1\to\mathbb{G}_2$,其中 $\mathbb{G}_1$ 的生成元为 g,则系统公开参数为 $\psi=(n,k,u,v,\mathrm{SE},\mathbb{G}_1,\mathbb{G}_2,g,q,e)$. 系统定义拉格朗日系数 $\Delta_{i,s}$,对于 $i\in\mathbb{Z}_p$ 和 $\mathbb{Z}_p$ 上元素集合 S,有:$\Delta_{i,s}(x)=\prod_{j\in S,j\neq i}\frac{x-j}{i-j}$.

组合文档工作流创建者依据每个工作流步骤对应文档组件定义属性集 χ_{part},将其提交给授权机构,依据集合中属性个数 m,从 $\mathbb{Z}_p$ 中随机选择 $t_1,t_2,\cdots,t_m$ 和 y,授权机构向创建者发布系统公钥 puk 为 $(T_1=g^{t_1},\cdots,T_m=g^{t_m},Y=e(g,g)^y)$,保留主密钥 msk 为 $(t_1,t_2,\cdots,t_m,y)$.

(2)SE(ψ,k,p)$\to C_{part}$

给定系统参数 ψ,加密算法 SE 加密组合文档组件 p 并经过签名后变为密文 C_{part},所有 N 个组件加密完后汇总构成组合文档密文 C.

(3)Encrypt($\psi,\chi_{part},A_k/\alpha$)$\to C_{KM}/C_\alpha$

创建者为每个步骤的参与者定义相关组件的访问密钥 A_k,算法选取一个随机值 $\mathbb{Z}_p$,采用 χ_{part} 加密 A_k 为 $C_{KM}=(\chi_{part},C'_{KM}=A_k\cdot e(g,g)^{ys},\{C_{KMi}=T_i^s\}_{i\in\chi_{part}})$.

同时,选取另一个随机值 j,采用 χ_{part} 加密密钥映射记录名 α 为 $C_\alpha=(\chi_{part},C'_\alpha=\alpha\cdot e(g,g)^{yj},\{C_{\alpha i}=T_i^j\}_{i\in\chi_{part}})$.

(4)Extract(ψ,C)$\to(C_{Extra},C_{CSO})$

给定组合文档密文 C 和系统参数 ψ,对于 $i=1,2,\cdots,v(v<N)$,算法依次从第 1 个组件中提取数据 $E_1=[1,u\cdot d]$,从第 2 个组件中提取 $E_2=[u\cdot d+1,u\cdot d\cdot 2]$,从第 i 个组件中提取 $E_i=[u\cdot d\cdot(i-1)+1,u\cdot d\cdot i]$,从第 v 个组件中提取 $E_v=[u\cdot d\cdot(v-1)+1,u\cdot d\cdot v]$,则 $C_{Extra}=(E_1,E_2,\cdots,E_v)$,这里 $E_i=(E_{[i][0]},E_{[i][1]},\cdots,E_{[i][d-1]})$.

由于每次提取的 bit 数 u 是个变量,同一个组合文档工作流或不同工作流的不同步骤都可以选取不同的值,因此可以避免由于参与者不慎泄露 C_{Extra} 给敌手时,敌手在过期后能够获得完整的 C.

从密文 C 中提取出 C_{Extra} 后,剩下部分为封装密文 C_{CSO}.

(5)PolyGener(ψ,C_{Extra},C_{KMi})$\to Q_i(x)$

给定系统参数 ψ、密文 C_{Extra} 和属性分量 C_{KMi},首先将 C_{KMi} 等分成 d 块 $C_{KMi}=(c_0,c_1,\cdots,c_{d-1})$,算法联合 C_{Extra} 和 C_{KMi} 产生 $v+1$ 个 $d-1$ 阶的拉格朗日多项式如下:

$$Q_1(x)=E_{[1][d-1]}x^{d-1}+E_{[1][d-2]}x^{d-2}+\cdots+E_{[1][0]},$$
$$\cdots$$
$$Q_i(x)=E_{[i][d-1]}x^{d-1}+E_{[i][d-2]}x^{d-2}+\cdots+E_{[i][0]},$$
$$\cdots$$
$$Q_v(x)=E_{[v][d-1]}x^{d-1}+E_{[v][d-2]}x^{d-2}+\cdots+E_{[v][0]},$$
$$Q_{v+1}(x)=c_{d-1}x^{d-1}+c_{d-2}x^{d-2}+\cdots+c_1x+c_0.$$

(6)CipherSharesGener($\psi,Q_i(x)$)$\to S_{CS}$

给定系统参数 ψ 和以上 $v+1$ 个拉格朗日多项式,算法随机选择 n 个大于 1 的自然数 $x_1,\cdots,x_i,\cdots,x_n$,然后计算密文分量 $S_{CS}=(c_{s1},\cdots,c_{sn})$,其中 $c_{si}=(x_i,Q_1(x_i),Q_2(x_i),\cdots,Q_{v+1}(x_i))$.

(7)CipherSharesDistri(L,S_{CS})$\to<l_i,c_{si}>$

给定密文分量 S_{CS} 后,算法随机选择一个分发索引 L 作为安全伪随机数产生器的种子,生成 n 个分发索引 $l_1,l_2,\cdots,l_n$. 对于 $i=1,\cdots,n$,密文分量关联索引并产生 n 个元组 $<l_i,c_{si}>$,然后将所有元组分发到 l_i 对应的 DHT 网络节点保存.

(8)Encapsulate(ψ,L,C_{CSO})$\to$CSO

给定系统参数 ψ、分发索引 L 和密文 C_{CSO} 作为输入,算法将其封装成组合文档自毁对象 CSO,并将其保存到云服务器中.

(9)Decapsulate(CSO)$\to(L,C_{CSO})$

在组合文档工作流生命周期范围内,参与者经过身份认证后,从云服务器获得相应的 CSO,算法解封装并获得分发索引 L 和密文 C_{CSO}.

(10)ExtractShares(ψ,L)$\to(C_{Extra},C_{KMi})$

给定分发索引 L 作为安全伪随机数产生器的种子,产生并提取 $l_1,l_2,\cdots,l_n$ 中大于等于 k 个索引值,从第二个 DHT 网络中获得大于等于 d 个密文分量 c_{si},利用拉格朗日插值法重构 $v+1$ 个多项式,最终恢复出 C_{Extra} 和 C_{KMi}. 而从算法(9)中获得了密文 C_{CSO},组合 C_{Extra} 和 C_{CSO} 即可重构出组合文档原始密文 C.

(11)PriKeyGer(χ_{user})$\to$prk

在一个工作流步骤中,参与者向授权中心提交属

性集 χ_{user},授权中心随机选择一个 $d-1$ 阶的拉格朗日多项式 $q(x)$并规定 $q(0)=y$.则参与者私钥 prk 由私钥分量$(K_i)_{i\in\chi_{user}}$组成,其中$\forall\ i\in\chi_{user}, K_i=g^{q(i)/t_i}$.

(12)Decrypt$(\psi,\chi_{user},C_\alpha/C_{KM})\rightarrow\alpha/A_k$

在一个工作流步骤中,参与者提交属性集 χ_{user},授权中心认证参与者之后判断如下条件是否成立:$|\chi_{part}\cap\chi_{user}|\geq d$.如果成立,则任选集合 $D=|\chi_{part}\cap\chi_{user}|$中的 d 个属性,再由 C_α 中的分量$\{C_i=T_i^j\}_{i\in\chi_{part}}$,关联 $d-1$ 阶的多项式 $q(x)$通过拉格朗日插值法可以重构出 $e(g,g)^{yj}$如下:

$$\begin{aligned}&\prod_{i\in D}(e(K_i,C_i))^{\Delta_{i,S(0)}}\\&=\prod_{i\in D}(e(g^{q(i)/t_i},g^{t_ij}))^{\Delta_{i,S(0)}}\\&=\prod_{i\in D}(e(g,g)^{q(i)j})^{\Delta_{i,S(0)}}\\&=e(g,g)^{q(0)j}\\&=e(g,g)^{yj}.\end{aligned}$$

因此,算法可以解密出密钥映射记录名为:

$$\begin{aligned}&C'_\alpha/\prod_{i\in D}(e(K_i,C_i))^{\Delta_{i,S(0)}}\\&=C'_\alpha/e(g,g)^{yj}\\&=\alpha\cdot e(g,g)^{yj}/e(g,g)^{yj}\\&=\alpha.\end{aligned}$$

同理,获得 C_{KMi}和密钥映射记录后,将 C_{KMi}合并到记录中的C'_{KM},获得完整密文 C_{KM},利用 ABE 解密算法可以解密 C_{KM},得到参与者的访问密钥 A_k.

(13)DE$(\psi,A_k,C_{part})\rightarrow p$

给定系统参数 ψ、访问密钥 A_k 和组合文档密文 C_{part},解密算法 DE 可以验证和解密 C_{part}为组合文档的明文 p,以供参与者处理.

在设计上述算法的基础上,下面描述 SelfDoc 方案系统层面的功能与操作.

4.4 SelfDoc 方案系统描述

系统层面通过调用算法层面的各多项式时间算法描述 SelfDoc 方案高级操作的功能与实现,主要包含以下几个阶段:

(1)系统建立

给定安全参数 κ,调用系统建立算法(Setup)产生系统所需的公共参数、随机对称密钥 k 和支持 ABE 所需的公钥 puk 和主密钥 msk.

(2)组合文档工作流建立

组合文档工作流创建者首先分割组合文档为若干组件,并划分安全等级,明确工作流各步骤及对应文档组件属性集 χ_{part},调用算法(SE)加密组件 p 获得密文 C_{part};然后,调用 ABE 加密算法(Encrypt)加密参与者访问密钥 A_k 获得 C_{KM}.分解 C_{KM}为 C'_{KM}和属性分量$\{C_{KMi}=T_i^s\}_{i\in\chi_{part}}$,将 C'_{KM}存于密钥映射记录中,系统产生随机值作为记录名 α 并添加一些伪记录名.创建者为每个步骤执行上述算法后即可依据新的组合文档结构创建完整的组合文档工作流.

(3)密钥映射记录名的处理

创建者在创建组合文档工作流时为每个参与者和对应的工作流步骤创建一条密钥映射记录,其记录名作如下处理:

首先,调用 ABE 加密算法(Encrypt)加密 α 得到 C_α,结合 DHT 网络的分发索引 R 调用密文分量分发算法(CipherSharesDistri)将 C_α 中的属性分量$\{C_{\alpha i}=T_i^j\}_{i\in\chi_{part}}$分发到第一个 DHT 网络中;然后,$C_\alpha$ 中封装密文 C'_α 和 R 构成元组 $<R,C'_\alpha>$存于授权中心.

(4)组合文档密文的处理

创建者获得组合文档密文 C,为 C_{part}的集合,需要对 C 作如下处理:

首先,调用提取算法(Extract)从 C 的每个 C_{part}中提取出部分密文信息,组成提取密文 C_{Extra},剩余部分为封装密文 C_{CSO}.

(5)密文分量的处理

在组合文档工作流创建阶段可以获得 C_{KMi},结合 C_{Extra}并调用多项式产生算法(PolyGener)构建 $v+1$ 个 $d-1$ 阶拉格朗日多项式 $Q_i(x)$.

再调用密文分量产生算法(CipherSharesGener)产生密文分量 S_{CS},该算法结合 DHT 网络分发索引 L 将 S_{CS}分解成元组 $<l_i,c_{si}>$并保存到另一个 DHT 网络的节点中.

(6)CSO 封装

在获得 C_{CSO}、分发索引 L 和公共参数 ψ 后,创建者可以调用封装算法(Encapsulate)将 C_{CSO}封装为 CSO,然后将该 CSO 存储到云服务器中.

值得注意的是,SelfDoc 方案中,实际存储到云服务器的组合文档工作流为 CSO,相比原 PPCD 而言,存储空间要小得多.因此,SelfDoc 能节省存储开销,节约成本.

(7)生命周期内的细粒度访问控制

在组合文档工作流生命周期内,参与者通过身份认证后,执行如下处理:

首先,将 χ_{user}提交给授权中心,授权中心判断条件 $|\chi_{part}\cap\chi_{user}|\geq d$ 是否成立,如果成立,则返回相应的元组 $<R,C'_\alpha>$给参与者,参与者依据 R 从第一个 DHT 网络中获取足够多的分量$\{C_i=T_i^j\}_{i\in\chi_{part}}$,结合 C'_α 可以获得完整密文 C_α.授权中心调用私钥产生算法(PriKeyGer)

获得参与者的 ABE 私钥 prk,调用解密 ABE 解密算法(Decrypt),最终得到密钥映射记录名 α,并获得记录中保存的 C'_{KM}.

然后,参与者从云服务器中获得 CSO,调用解封装算法(Decapsulate)解封装 CSO 得到分发索引 L 和密文 C_{CSO}.依据 L 和提取分量算法(ExtractShares)从第二个 DHT 网络中获得足够多的密文分量 S_{CS},采用拉格朗日插值法重构出 C_{KMi}和 C_{Extra}.此时,组合 C_{CSO}和 C_{Extra}可以恢复出原始密文 C,合并 C'_{KM}和 C_{KMi}构成完整密文 C_{KM}.

最后,结合记录名 α、C_{KM}和原始密文 C 即为原组合文档.利用获得的 ABE 私钥 prk 解密 C_{KM}获得访问密钥 A_k,从而调用解密算法 DE 解密出相应的组合文档组件的明文 p.通过以上方法不同参与者访问不同步骤中不同安全级别的组合文档组件,实现组合文档生命周期内的细粒度访问控制.

参与者处理完该步骤的组合文档后,加密相关组件然后签名.再封装成 CSO 后存储到云端服务器中,以便下一步骤的参与者访问.

(8)过期后的组合文档安全自毁

当组合文档工作流完成所有步骤的处理,并超过一定的预定义时间期限后,SelfDoc 将利用 DHT 网络节点数据的自动更新功能丢弃所存的密钥分量,从而无法获得 α、C_{KM}和原始密文 C,以实现组合文档安全自毁.

SelfDoc 的两个 DHT 网络若采用 Vuze DHT 网络[6],则节点数据的自动更新周期默认为 8 小时,能够满足一般组合文档工作流的处理需要.当组合文档工作流的处理时间较长时,可采用如下方法延长组合文档自毁时间期限.

当接近 DHT 网络节点默认更新周期时,创建者重新选取随机变量,构造新的密文分量 S'_{CS},再重新选择分发索引 L'和 R',将新的 S'_{CS}分发到新的 DHT 网络节点中,则组合文档自毁时间可以延长一个周期.如此循环,可以实现 DHT 网络默认周期整数倍的组合文档自毁时间期限设定.

5 SelfDoc 方案安全性分析

在方案[5,6,12,15]中主要考虑追溯性攻击,均假设敌手不是实时的,存储在服务器中的封装对象在生命周期内不受攻击.而在云服务环境中,云服务器中存储的 CSO 在过期前也可能存在攻击.因此,已有方案的假设过于理想化,在 SelfDoc 方案中,不需要该假设,主要考虑三种类型的攻击,其中前两者与已有方案相似:一种是在 CSO 的生命周期内,对 DHT 网络实施攻击,以试图获取到足够多的密文分量;一种是在 CSO 过期后,采用传统的攻击手段攻击 CSO;一种是在 CSO 封装后的任意时刻,同时攻击 CSO 和 DHT 网络并试图恢复出组合文档的隐私信息.因此,本节从 DHT 网络的安全性、算法安全性和系统安全性三个方面对以上三种攻击途径进行安全性分析.

5.1 DHT 网络安全性分析

Vanish[6]指出,DHT 网络的体系结构和特性使得敌手在过期之前从 DHT 网络中获取密钥分量具有挑战性,并且在 Vuze DHT 网络中做了相关实验证实了存储嗅探攻击、查询嗅探攻击和标准 DHT 攻击都不能从 DHT 网络中获得足够多的分量以恢复出原始密钥.以此为依据,文献[12]表示 SSDD 方案中敌手也不能从 DHT 网络中获得足够多的分量以恢复出密钥与部分密文,从而也能抵抗 DHT 网络的各种攻击.

然而,文献[14]和 SafeVanish[15]均通过分析和相关实验证实 Vanish 系统存在 DHT 网络的 Sybil 攻击,从而证明 Vanish 系统是不安全的.文献[14]的结论得到了 Vanish 系统团队的认可,由此可以推论出 SSDD[12]也是不安全的.

DHT 网络的 Sybil 攻击主要分为跳跃(hopping)攻击和嗅探(sniffing)攻击两种.

(1)跳跃攻击

Vanish[6]存在跳跃攻击,其分发到 DHT 网络的密钥分量一般为 16~51 字节,定长且较短,在隐私数据生命周期内很容易被 Sybil 敌手通过跳跃的方式获取到足够多的密钥分量并保存起来,在过期之后恢复出密钥以解密并获得隐私数据.

SafeVanish[15]通过在密钥分量中增加一些随机值以增加密钥分量的长度而有效抵抗跳跃攻击;SSDD[12]通过将部分密文和密钥一起产生密钥分量分发到 DHT 网络中,也可以增加密钥分量的长度在一定程度上抵抗跳跃攻击;ISDS[5]采用 IBE 加密对称密钥,其密钥密文结合部分文档密文一起产生混合密文分量分发到 DHT 网络中,能增加密文分量的长度,从而抵抗跳跃攻击.

SelfDoc 将 C_{KMi}结合 C_{Extra}一起分发到第二个 DHT 网络中,由于 C_{KMi}是 ABE 加密后的属性分量,其长度随属性个数的增加呈线性增长,相比 Vanish[6]和 SSDD[12]的对称密钥要长的多,从而显著增加了密文分量的长度.因此,SelfDoc 能有效抵抗跳跃攻击.

(2)嗅探攻击

在隐私数据的生命周期内,敌手可以从 Vanish[6]和 SSDD[12]的 DHT 网络中嗅探出足够多的密钥分量以重构出解密密钥,因此存在嗅探攻击.SafeVanish[15]和 ISDS[5]分别采用公钥密码技术的 RSA 和 IBE 加密对称密

钥后,产生密钥分量和混合密文分量并分发到 DHT 网络节点中.这两种方案中,即使敌手从 DHT 网络中嗅探到足够多的分量,也没有相应的 RSA 或 IBE 私钥,不能解密出原始对称密钥,因而能够抵抗 Sybil 嗅探攻击.然而,由于 SafeVanish 基于 PKC 方案,因此存在复杂的公/私钥和证书管理问题.

SelfDoc 采用 ABE 加密 A_k 得到密文 C_{KM},然后将 C_{KM}中的 C_{KMi}结合 C_{Extra}构造拉格朗日多项式并产生 S_{CS} 后再将其分发到第二个 DHT 网络中.即使敌手从 DHT 网络中获得 S_{CS},没有 ABE 私钥也不可能恢复出 C_{KM},从而无法获得 A_k.因此,SelfDoc 能够有效抵抗 Sybil 嗅探攻击.相比 SafeVanish、ISDS 和 SelfDoc 均不需要预先产生所有用户的公/私钥对,也不需要颁发用户的公钥证书,因而不存在复杂的证书和密钥管理问题.

此外,SelfDoc 中还使用了另一个 DHT 网络以保存密钥映射记录名的属性分量,只有同时从这两个 DHT 网络中获得正确的分量信息,才能恢复出原始的密文和相关密钥,才能正常访问相关组合文档.而要同时攻破两个 DHT 网络,将给敌手带来更大的挑战.因此,SelfDoc 能够有效抵抗 DHT 网络的 Sybil 攻击.

5.2 SelfDoc 方案算法安全性分析

在 SelfDoc 中,不仅组合文档各组件是加密的,其密文也是经过提取的,而且加密和签名验证的 A_k 也是经过 ABE 加密的,A_k 的密文分量 C_{KMi}和部分组合文档密文都是经过构造拉格朗日多项式并产生密文分量等处理的.因此,没有完整的密文信息是不可能重构出原始访问密钥和组合文档密文的,也是不可能恢复出原始组合文档明文的[21].

SelfDoc 算法安全性方面,主要体现在组合文档工作流过期之后存在对 CSO 的传统攻击,主要分为蛮力攻击和密码分析攻击.

(1)蛮力攻击

蛮力攻击的工作原理是,基于完整密文及密钥空间大小持续尝试所有可能的解密密钥.Vanish[6]采用对称密钥加密敏感数据,其密文保存在网络中或云服务器,由文献[12]分析可知,敌手可以获得完整的密文,且对称密钥的密钥空间较小,因此,在计算能力允许的情况下,有可能试探出解密密钥而存在蛮力攻击.SafeVanish[15]的设计着重在于增加密钥分量的长度及采用公钥系统加密对称密钥并分发到 DHT 网络以抵抗 DHT 网络的 Sybil 攻击.该方案在封装 VDO 时仍然采用原 Vanish 系统的方法,由上述分析可知,该方案也存在蛮力攻击.

然而,SSDD[12]和 ISDS[5]由于封装并存储在云端的 VDO 和 DDO 是经过处理之后的不完整密文,即使敌手从云端获得了 VDO 或 DDO,也不可能尝试出可能的解密密钥,从而可以抵抗蛮力攻击.同样的,SelfDoc 采用提取算法将组合文档原始密文提取成 C_{Extra}和 C_{CSO},并将密文 C_{KM}中的 C_{KMi}与 C_{Extra}一起经过变换产生密文分量并分发到 DHT 网络中,而 C_{CSO}被封装成 CSO 并存储在云服务器中.因此,要实现蛮力攻击,获得原始组合文档密文 C 的唯一方式是从 DHT 网络中获得完整的密文分量 C_{KMi}和 C_{Extra},攻击的难度与密钥空间有关.在 SelfDoc 中,密钥空间与 C_{KMi}和 C_{Extra}的长度相关,明显大于 SSDD 的密钥分量.另一方面,即使敌手从云服务器获得了 CSO,解封装后的 C_{CSO}也是不完整的密文.因此,SelfDoc 比 SSDD 更能有效抵抗蛮力攻击.

(2)密码分析攻击

传统的密码分析攻击的前提也需要获得完整的原始密文.有上述分析可知,只有在 Vanish 和 SafeVanish 中,敌手才能获得完整的密文.因此,他们可能存在密码分析攻击.而在方案 SSDD、ISDS 和 SelfDoc 中,封装并存储在云端的密文均不完整,可以抵抗密码分析攻击.

5.3 SelfDoc 方案系统安全性分析

上述分析可知,Vanish[6]、SafeVanish[15]、SSDD[12]和 ISDS[5]仅考虑前两种类型的攻击,SelfDoc 都能够抵抗.下面考虑一种强类型的攻击:即 CSO 封装后的任意时刻,同时遭受对 CSO 的传统攻击和对 DHT 网络的 Sybil 攻击.由 5.1 节分析可知,SelfDoc 中,敌手在 CSO 的生命周期内,能够抵抗 DHT 网络的 Sybil 跳跃和嗅探攻击,且同时攻破的两个 DHT 网络分别获得正确的密文分量是困难的;即使同时攻击云服务器并获得 CSO,也不能恢复出原始密文和相关密钥信息,最终无法获得组合文档的明文.因此,SelfDoc 能够同时抵抗这两种类型的攻击,达到系统整体安全.

以上各方案抵抗攻击的能力比较汇总于表 2 所示.

综上所述,Vanish[6]由于存在 DHT 网络的 Sybil 攻击、可能的蛮力攻击和密码分析攻击,是不安全的方案;SafeVanish[15]能够抵抗 DHT 网络的 Sybil 攻击,但可能存在蛮力攻击和密码分析攻击而变得不安全;SSDD[12]能够抵抗跳跃攻击、蛮力攻击和密码分析攻击,但可能存在嗅探攻击,从 DHT 网络中获得足够的密钥和密文分量而重构出原始密文,从而获得敏感数据原文,因此,该方案也不是足够安全的;ISDS[5]能够抵抗以上两种类型的攻击;然而,以上方案均基于“封装对象在过期之前未受攻击”的理想假设,且不能抵抗第三种类型的同时攻.SelfDoc 不需要该前提假设也能实现组合文档生命周期内的细粒度访问控制和过期后的安全自毁,并具有简单高效的密钥管理机制;且既能抵抗 DHT 网路的 Sybil 攻击,又能抵抗蛮力攻击和密码分析攻击,还能抵抗第三种类型的同时攻击,因而是安全的

方案.此外,在 ISDS[5]中,由于组合文档创建者需要知道所有参与者的身份信息,存在可能的身份隐私泄露问题,而 SelfDoc 利用属性集合实现对称密钥的加密,有效地保护了参与者的身份隐私,是比 ISDS 更优的解决方案.在 SelfDoc 中,还可以引入权重属性基加密机制[22],将组合文档属性集和参与者属性集中的属性赋予不同的权值,除了满足门限约束外,还需满足属性的权重条件才能从授权中心获得 ABE 私钥,以解密出密钥映射记录名和访问密钥,实现组合文档生命周期内更细粒度的访问控制策略.

表 2　各方案抵抗攻击能力比较

攻击类型 / 方案	DHT 网络攻击		传统攻击		同时攻击
	跳跃攻击	嗅探攻击	蛮力攻击	密码分析	
Vanish[6]	不能抵抗	不能抵抗	不能抵抗	不能抵抗	不能抵抗
SafeVanish[15]	能够抵抗	能够抵抗	不能抵抗	不能抵抗	不能抵抗
SSDD[12]	能够抵抗	不能抵抗	能够抵抗	能够抵抗	不能抵抗
ISDS[5]	能够抵抗	能够抵抗	能够抵抗	能够抵抗	不能抵抗
SelfDoc	能够抵抗	能够抵抗	能够抵抗	能够抵抗	能够抵抗

6　结论

在云服务环境中,组合文档工作流将发挥重要作用.包含隐私信息的组合文档工作流在处理结束后的隐私信息保护机制往往被研究者们忽视.本文提出了一种 SelfDoc 方案,引入多级安全思想创新组合文档结构,将基于属性的加密机制和 DHT 网络融合到组合文档工作流的设计和处理中,以实现组合文档生命周期内的细粒度访问控制,以及过期后的安全自毁,从而为组合文档提供全生命周期隐私保护.安全性分析表明,SelfDoc 方案不需要理想的前提假设,且能够抵抗三种类型的攻击,是安全的解决方案.

本文是组合文档过期后隐私保护的初步研究成果,是属性加密机制和隐私保护研究领域的扩展,为进一步的研究提供思路.下一步工作的重点为:(1)在云计算开源软件 OpenStack 的分布式存储 Swift 平台[23]上,引入主动存储对象,将 CSO 设计成主动对象存储在 Swift 中,开发一个云服务环境中组合文档安全自毁系统,实现 SelfDoc 方案中 CSO 在云端的安全删除;(2)基于 Pairing-Based Cryptography Library 平台[24],在密文策略属性基加密工具集[25]的基础上对 SelfDoc 方案中的多项式时间算法进行仿真实验,测试该方案的计算开销和执行效率,并与已有方案进行性能对比分析与评价.

致谢　感谢匿名评审专家给本文提出的改进意见.

参考文献

[1] BALINSKY H, SIMSKE S J. Secure document engineering [A]. Proceedings of the 11th ACM symposium on Document Engineering[C]. New York: ACM, 2011. 269 – 272.

[2] BALINSKY H, SIMSKE S J. Differential access for publicly-posted composite documents with multiple workflow participants[A]. Proceedings of the 10th ACM Symposium on Document Engineering[C]. New York: ACM, 2010. 115 – 124.

[3] BALINSKY H, CHEN L Q, SIMSKE S J. Publicly posted composite documents with identity based encryption[A]. Proceedings of the 11th ACM Symposium on Document Engineering [C]. New York: ACM, 2011. 239 – 248.

[4] 李凤华,苏 ▮,史国振,等.访问控制模型研究进展及发展趋势[J].电子学报,2012,40(4):805 – 813.
LI Feng-hua, SHU Mang, SHI Zheng-guo, et al. Research status and development trends of access control model[J]. Acta Electronic Sinica, 2012, 40(4): 805 – 813. (in Chinese)

[5] XIONG J B, YAO Z Q, MA J F, et al. A secure document self-destruction scheme with identity based encryption[A]. Proceedings of the 5th International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS'13 [C]. Los Alamitos, CA: IEEE CS, 2013. 239 – 243.

[6] GEAMBASU R, KOHNO T, LEVY A, et al. Vanish: Increasing data privacy with self-destructing data[A]. Proceedings of the 18th USENIX Security Symposium [C]. Berkeley, CA: USENIX, 2009. 299 – 315.

[7] SAMSON T. Deleted Cloud Files Can be Recovered from Smartphones, Researchers Find [EB/OL]. http://www.infoworld.com/t/mobile-security/deleted-cloud-files-can-be-recovered-smartphones-researchers-find-214779, 2013 – 04.

[8] PERLMAN R. File system design with assured delete[A]. Proceedings of the Third IEEE International Security in Storage Workshop[C]. Los Alamitos, CA: IEEE CS, 2005. 83 – 88.

[9] PERLMAN R. The ephemerizer: making data disappear[J]. Journal of Information Systems Security, 2005, 1(1): 21 – 32.

[10] TANG Y, LEE P P, LUI J C, et al. FADE: Secure overlay cloud storage with file assured deletion[A]. Proceedings of the Security and Privacy in Communication Networks[C]. Berlin: Springer, 2010. 380 – 397.

[11] TANG Y, LEE P P, LUI J C, et al. Secure overlay cloud storage with access control and assured deletion[J]. IEEE Transactions on Dependable and Secure Computing, 2012, 9(6): 903 – 916.

[12] WANG G J, YUE F S, LIU Q. A secure self-destructing scheme for electronic data[J]. Journal of Computer and System Sciences, 2013, 79(2): 279 – 290.

[13] 王丽娜,任正伟,余荣威,等.一种适于云存储的数据确定性删除方法[J].电子学报,2012,40(2):266 – 272.
WANG Li-na, REN Zheng-wei, YU Rong-wei. A data assured deletion approach adapted for cloud storage[J]. Acta Electronic Sinica, 2012, 40(2): 266 – 272. (in Chinese)

[14] WOLCHOK S, HOFMANN O S, HENINGER N, et al. Defeating vanish with low-cost sybil attacks against large DHTs [A]. Proceedings of the 17th Annual Network & Distributed System Security Conference [C]. San Diego, CA: ISOC, 2010.1 – 15.

[15] ZENG L F, SHI Z, XU S, et al. SafeVanish: An improved data self-destruction for protecting data privacy[A]. Proceedings of the Second International Conference on Cloud Computing Technology and Science[C]. Los Alamitos, CA: IEEE CS, 2010.521 – 528.

[16] 熊金波,姚志强,马建峰,等.基于行为的结构化文档多级访问控制[J].计算机研究与发展,2013,50(7):1399 – 1408.
XIONG Jin-bo, YAO Zhi-qiang, MA Jian-feng, et al. Action-based multilevel access control for structured document[J]. Journal of Computer Research and Development, 2013, 50(7):1399 – 1408. (in Chinese)

[17] SAHAI A, WATERS B. Fuzzy identity-based encryption[A]. Proceedings of the Advances in Cryptology[C]. Berlin Heidelberg: Springer, 2005.457 – 473.

[18] SHAMIR A. How to share a secret[J]. Communications of the ACM, 1979, 22(11):612 – 613.

[19] RHEA S, GODFREY B, KARP B, et al. OpenDHT: A public DHT service and its uses[A]. Proceedings of the 5th ACM SIGCOMM[C]. New York: ACM, 2005.73 – 84.

[20] FALKNER J, PIATEK M, JOHN J, et al. Profiling a million user DHT[A]. Proceedings of the 7th ACM SIGCOMM Conference on Internet Measurement [C]. New York: ACM, 2007.29 – 134.

[21] XIONG J B, YAO Z Q, MA J F, et al. A secure document self – destruction scheme: an ABE approach[A]. Proceedings of the 15th International Conference on High Performance Computing and Communications, IEEE HPCC'13[C]. Los Alamitos, CA: IEEE CS, 2013.59 – 64.

[22] 刘西蒙,马建峰,熊金波,等.密文策略的权重属性基加密方案[J].西安交通大学学报,2013,47(8):44 – 48.
LIU Xi-meng, MA Jian-feng, XIONG Jin-bo, et al. Ciphertext policy weighted attribute based encryption scheme[J]. Journal of Xi'an Jiaotong University, 2013, 47(8):44 – 48. (in Chinese)

[23] Swift's Documentation [EB/OL]. http://docs.openstack.org/developer/swift/, 2013-07.

[24] The Pairing-Based Cryptography Library [EB/OL]. http://crypto.stanford.edu/pbc/, 2013-07.

[25] Ciphertext-Policy Attribute-Based Encryption Toolkit [EB/OL]. http://acsc.cs.utexas.edu/cpabe/index.html, 2013-07.

作者简介

熊金波 男,1981 年 2 月出生于湖南省益阳市.西安电子科技大学博士生,福建师范大学讲师,主要研究方向为隐私保护技术与文档安全.
E-mail: jinbo810@163.com

姚志强 男,1967 年 5 月出生于福建省莆田市.西安电子科技大学博士生,福建师范大学教授,主要研究方向为信息安全.
E-mail: yzq@fjnu.edu.cn

马建峰(通信作者) 男,1963 年 10 月出生于陕西省西安市.博士,西安电子科技大学计算机学院院长、教授、博士生导师,主要研究方向为密码学、计算机网络与信息安全.
E-mail: jfma@mail.xidian.edu.cn

李凤华 男,1966 年 3 月出生于湖北省浠水县.博士,中国科学院信息工程研究所研究员,博士生导师,主要研究方向为网络安全与可信计算.
E-mail: lfh@iie.ac.cn

刘西蒙 男,1988 年 4 月出生于陕西省西安市.博士生,主要研究方向为基于属性的密码学.
E-mail: snbnix@gmail.com

李 琦 男,1989 年 4 月出生于江苏省淮安市.博士生,主要研究方向为基于属性的密码学.
E-mail: qilijs@gmail.com