软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2019,30(12):3815−3828 [doi: 10.13328/j.cnki.jos.005610]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

云加密数据安全重复删除方法*

张曙光[1,2,3], 咸鹤群[1,2], 王利明[3], 刘红燕[1]

[1](青岛大学 计算机科学技术学院,山东 青岛 266071)

[2](广西密码学与信息安全重点实验室(桂林电子科技大学),广西 桂林 541004)

[3](中国科学院 信息工程研究所 第五研究室,北京 100093)

通讯作者: 咸鹤群, E-mail: xianhq@126.com

摘 要: 在云环境存储模式中,采用用户端数据加密虽然能够有效降低数据的存储安全风险,但同时会使云服务商丧失重复数据鉴别能力,导致存储开销随数据量增大而不断攀升.加密数据重复删除技术是解决该问题的方法之一,现有方案通常基于可信第三方设计,安全性假设过强,执行效率较低.基于椭圆曲线与密文策略属性加密两种高安全密码学原语,构造了重复加密数据识别与离线密钥共享两种安全算法,进而实现一种无需初始数据上传用户与可信第三方实时在线的加密数据重复删除方法.详细的安全性与仿真实验分析,证明该方法不仅实现数据的语义安全,同时能够保证系统的高效率运行.

关键词: 加密数据重复删除;椭圆曲线;密文策略属性加密;数据流行度

中图法分类号: TP309

中文引用格式: 张曙光,咸鹤群,王利明,刘红燕.云加密数据安全重复删除方法.软件学报,2019,30(12):3815−3828. http://www.jos.org.cn/1000-9825/5610.htm

英文引用格式: Zhang SG, Xian HQ, Wang LM, Liu HY. Secure cloud encrypted data deduplication method. Ruan Jian Xue Bao/Journal of Software, 2019,30(12):3815−3828 (in Chinese). http://www.jos.org.cn/1000-9825/5610.htm

Secure Cloud Encrypted Data Deduplication Method

ZHANG Shu-Guang[1,2,3], XIAN He-Qun[1,2], WANG Li-Ming[3], LIU Hong-Yan[1]

[1](College of Computer Science and Technology, Qingdao University, Qingdao 266071, China)

[2](Guangxi Key Laboratory of Cryptography and Information Security (Guilin University of Electronic Technology), Guilin 541004, China)

[3](The Fifth Research Laboratory, Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China)

Abstract: Deduplication states that only one copy of the same data is stored in the cloud server. In order to protect data privacy, users usually encrypt their data before uploading them. When encrypted with different keys, the same data may have different ciphertext results. It is difficult for the cloud server to identify and eliminate the duplicate copies. Most current solutions to the problem rely heavily on online trusted third parties, resulting in unsatisfying efficiency and security. A secure cloud encrypted data deduplication scheme is proposed, which supports offline key deliver. By constructing a duplicate check tag, it can be verified whether encrypted data originate from the same plaintext data. The ciphertext policy attribute based encryption is used to ensure the check tag is securely generated. The initial uploader of some specific data is able to deliver the encryption key to the subsequent uploaders via the cloud server in an offline

* 基金项目: 国家自然科学基金(61702294); 山东省自然科学基金(ZR2019MF058); 广西密码学与信息安全重点实验室研究课题(GCIS201722); 赛尔网络下一代互联网技术创新项目(NGII20170414)

Foundation item: National Natural Science Foundation of China (61702294); Shandong Provincial Natural Science Foundation (ZR2019MF058); Guangxi Key Laboratory of Cryptography and Information Security (GCIS201722); CERNET Innovation Project (NGII20170414)

收稿时间: 2018-02-06; 修改时间: 2018-05-22; 采用时间: 2018-06-04

manner. Deduplication can be completed without online participation of any trusted third party. Security analysis and proving are presented. The feasibility and efficiency of the scheme are verified via simulation experiments.

Key words: encryption data deduplication; elliptic curve; ciphertext policy attribute based encryption (cp-abe); data popularity

随着大数据时代的到来,作为基础设施的云存储服务变得愈加重要.在云服务持续高速度发展的背景下,服务提供商不再局限于一味地堆积硬件,而是逐步通过尽可能提高存储效率的方式,达到“无形”增加存储空间并换取经济效益的目的.目前,提高存储效率的技术主要包括数据压缩和重复数据删除.数据压缩技术虽然能够通过对整体数据重新编码,实现存储空间的更少占用,但由于压缩后的数据需要在解码后才可正常使用,这无疑增加了系统的计算负担.重复数据删除技术的思想是通过摒除数据的重复存储,进而减少存储冗余[1,2].生而逢时,在如火如荼发展的云计算和大数据应用场景中,同一数据副本时常被不同用户重复存储,造成巨量存储空间浪费,重复数据删除技术恰成为解决该问题的最佳方法.经最新研究表明:重复数据删除技术可以在备份应用系统中减少高达 90%的存储需求,在标准文件系统中使存储需求降低约 70%[3].

良好的云存储系统应能够为用户提供安全的数据存储环境,然而在实际应用中,云服务提供商并非完全可信.例如,Facebook 在 2013 年泄露了用户的联系信息[4],iCloud 在 2014 年泄露了用户的私密照片[5].数据加密是解决此类风险的良好选择,然而由于数据的加密密钥由用户在本地独立生成,密钥的多样性导致相同数据副本被加密为不同密文,使得云服务提供商无法识别数据是否重复,造成大量存储冗余.如何对加密后的数据执行重复安全删除,是云存储安全领域的研究热点之一.

起初,研究者提出由云服务商提供唯一密钥并执行加密操作,如此,数据控制权依然驻留在云服务商中,虽然能够抵抗外部敌手攻击,但无法防止数据由服务商内部泄露.Douceur 等研究者提出客户端收敛加密(convergent encryption,简称 CE)方法[6].计算数据副本的哈希值并将其作为加密密钥,此时输入同一数据副本即可得到相同数据密文[7,8].收敛加密虽拥有较高的执行效率,却未实现语义安全,容易遭受离线暴力破解攻击[9,10].Bellare 等研究者提出信息锁加密方案(message-locked encryption,简称 MLE)[11],虽复杂化了密钥计算与加密方式,但与 CE 相比,其核心思想无变化,因此同样无法实现语义安全[12,13].Bellare 等研究者提出了 DupLESS[14],相同数据的不同属主与可信第三方运行茫然伪随机函数计算协议(oblivious pseudorandom function,简称 OPF),用以输出相同加密密钥.Duan 等研究者对 DupLESS 进行扩展与改进,对可信第三方的任务进行分解,将密钥生成过程的参与方扩展为多个用户[15].文献[14,15]中的方案无法抵抗云服务器在线穷举攻击.Puzio 等研究者提出首个基于双层加密的重复加密数据删除方案 ClouDedup[7],内层是高效的收敛加密,外层加密与解密工作外包给可信第三方.除了安全性的提高,双层加密带来的还有高额的计算开销与通信开销.与文献[14,15]相似,ClouDedup 无法防止云服务商与第三方的合谋攻击.Stanek 等人提出:用户在上传数据之前需要确定数据的类型,若数据属主数量低于预定义流行度阈值,则该数据副本将被定义为非流行数据;反之,则将其标记为流行数据[17].非流行数据采用双层加密.随着数据副本数量不断增加,当等于阈值后,云服务商便进行外层解密,进而借助内层收敛加密的特性,执行重复数据删除.同时,为了抵抗敌手进行女巫攻击[16,18],引入身份服务器.与文献[7]中的方案类似,多方服务器的引入带来高额的计算与通信开销.Puzio 等研究者基于完美哈希函数(PHF)设计了数据流行度查询算法,依赖第三方的协助,查询数据副本流行度,并根据查询结果执行相应的加密算法[19].该方案无法解决非流行加密数据重复删除的问题[3],且与文献[14,15,17]类似,可信第三方实体必须实时在线参与,然而在实际应用中,部署完全可信的第三方比较困难.Liu 等研究者设计首个无可信第三方参与的加密数据重复删除方案,使用口令认证密钥交换协议(password authenticated key exchange,简称 PAKE)传递密钥,相同数据副本属主能够计算得到同一加密密钥[9].方案的不足点在于,参与方必须实时在线,导致系统的可行性与实用性较低.

本文贡献:

本文在划分数据类型的基础上,提出一种无需初始数据上传用户与可信第三方实时在线的加密数据重复删除方案.

1) 基于椭圆曲线构造流行度查询标签,在语义安全的前提下,使用该标签验证加密副本是否产生于同一

明文,并判断其流行度.借助密文策略属性加密,保证查询标签生成协议的安全实现;

2) 设计安全的密钥共享协议,确保同一数据副本的初始属主能够借助云服务商,将加密密钥安全离线共享至后继属主,实现非流行数据重复删除.构造新的流行数据加密算法,增强流行数据的存储安全;

3) 总结常见的敌手模型,通过安全分析证明本方案可抵御敌手模型中的恶意攻击.

1 系统设计与敌手模型

1.1 系统模型

如图 1 所示,本系统共包含 3 类实体:密钥生成中心(KDC)、用户群(users)与云服务器(CSP).系统建立初期,KDC 为用户生成密钥对集合,并将随机值密文参数集合部署在云服务器,然后转入离线状态.云服务器为用户提供数据的在线存储与共享服务,且具有删除重复加密数据的功能.

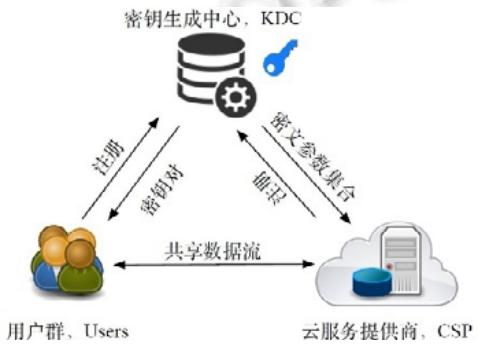

Fig.1 System model

图 1 系统模型

1.2 设计目标

本方案需要满足以下性质.

1) 有效性

a) 云服务器能够识别重复的加密数据,并判断数据类型(非流行数据或流行数据),根据数据类型采取相应加密算法;

b) 数据初始上传者能够将加密密钥通过云服务器,以离线的方式传递给后继上传者;

c) 云服务器能够执行加密数据重复删除.

2) 安全性

a) 使用椭圆曲线生成的查询标签识别数据冗余度与流行度,识别过程不泄漏数据的任何明文信息;

b) 初始上传者将加密密钥以密文形式存储在云服务器,但云服务器无法对其解密;

c) 客户端加密数据重复删除与云服务器端重复数据删除混合使用,防止侧信道攻击.

3) 高效性

a) 保证流行度查询标签生成算法和密钥传递算法的高效性;

b) 针对不同流行度数据,采用不同加密算法,在确保安全性的前提下,提高系统执行效率.

1.3 敌手模型

在数据安全需求方面,用户假定云服务提供商是不可信的;用户在系统效率方面的要求与云服务提供商的存储成本存在一定矛盾.因此,本文不考虑用户与云服务器合谋攻击.由于在重复数据删除方案中,侧信道攻击主要针对客户端重复数据删除(穷举并上传文件,观察是否发生重复数据删除),而本方案只对隐私度比较低的流行数据使用客户端重复数据删除,因此侧信道攻击问题不是本文的研究重点.

本文的敌手有以下两类.

1) 云服务提供商

云服务提供商能够按照系统所设计的协议与用户执行所有的交互,可以访问或复制用户存储在云服务器上的加密数据、查询标签等所有信息,因此可以对查询标签与加密数据执行离线穷举攻击,其攻击方式为:猜测穷举某数据内容的所有可能,构造查询标签集合并与用户的查询标签进行比较,验证猜测正确性,最终获得数据内容.

2) 用户群中的恶意成员(恶意用户)

恶意用户拥有与合法用户完全相同的访问能力和权限,掌握 KDC 分配的密钥对.其可能的攻击方式如下.

a) 劫持受害者与云服务器的通信信道,假冒云服务器,与受害者执行方案中的所有交互协议,对受害者的查询标签执行离线穷举攻击,即:穷举猜测某数据内容的所有可能,构造查询标签集合并与用户的查询标签进行比较,验证猜测正确性,获得数据内容;

b) 执行在线穷举攻击,穷举某数据内容的所有可能,逐一构造查询标签并发送至云服务器,根据云服务器的回复消息判断该数据是否已被存储在云服务器.

2 定义与预备知识

2.1 具有离线密钥传递的云加密数据安全重复删除方案

本方案共包含以下 4 种算法.

a) *SystemSet*:系统初始设置算法.KDC 为用户生成属性密钥对,并为云服务器部署密文参数;

b) *PopularityCheck*:流行度查询算法.由用户与云服务器共同完成.持有相同数据的用户,可以在不泄露任何数据内容的情况下获得相同的查询标签,进而查询数据流行度;

c) *UnPopularDedup*:非流行加密数据重复删除算法.由用户与云服务器共同完成.云服务器存储首次上传的加密数据;若云服务器检测到冗余数据被上传,则将其删除,并为当前用户创建数据的访问链接;

d) *PopularUpload*:流行加密数据重复删除算法.由用户与云服务器共同完成.若拥有某数据的用户数量等于流行度阈值,则用户上传收敛加密密文;若大于流行度阈值,则执行客户端重复数据删除,即:用户无需实际上传加密数据,云服务器会为其创建数据的访问链接.

2.2 有限域上的椭圆曲线

定义有限域 $GF(P)$,其特征 $P\neq 2,3$,参数 $a,b\in GF(P)$满足 $4a^3+27b^2\neq 0$.

定义满足等式 $y^2=x^3+ax+b$ 的点$(x,y)\in GF(P)\times GF(P)$与无穷远点 O 构成的集合为椭圆曲线 $E_{(a,b)}(GF(P))$[20−23].

在下面定义的加法运算下,这些点可构成 Abelian 群:O 是恒等元,假设 M,N 为 $E_{(a,b)}(GF(P))$上的两个点,若 $M=O$,则$-M=O,M+N=N+M=N$;设定 $M=(x_1,y_1),N=(x_2,y_2)$,则$-M=(-x_1,-y_1)$,且 $M+N=O$;若 $M=-N$,则 $M+N=(x_3,y_3)$,其中,

$$x_3=\mu^2-x_1-x_2,\ y_3=\mu(x_1-x_3)-y_1,\ \mu=\begin{cases}\dfrac{3x_1^2+a}{2y_1}, & M=N\\ \dfrac{y_2-y_1}{x_2-x_1}, & M\neq N\end{cases}.$$

2.3 密文策略属性加密

安全的基于密文策略属性加密(CP-ABE)方案通常包含以下算法[24−26].

a) $Setup(\lambda)\rightarrow\langle PK,MS\rangle$:系统初始化.输入安全参数$\lambda$,输出密钥对$\langle PK,MS\rangle$;

b) $Encrypt(PK,F,S)\rightarrow C_S$:加密算法.输入公钥 PK,消息 F,访问结构 S,输出密文 C_S;

c) $KeyGen(MS,PK,AT_i)\rightarrow SK_{AT_i}$:私钥生成算法.输入主密钥 MS,公钥 PK,用户的属性集合 AT_i,输出用户私钥 SK_{AT_i};

d) $Decrypt(PK,SK,C_S)\rightarrow F$:解密算法.输入公钥 PK,用户私钥 SK,密文 C_S,其中,访问策略隐含在 C_S 中.当且仅当 $AT_i\in S$,才能解密得到消息 F[27,28].

3 具有离线密钥传递的云加密数据安全重复删除方案

3.1 方案概述

系统建立初始,KDC通过*SystemSet*算法为每个注册用户生成属性加密算法的公私钥对,并将密文参数集合部署到云服务器.在*PopularityCheck*算法中,用户发送数据短哈希值至云服务器,以获取生成查询标签所需要的参数值,并使用椭圆曲线计算流行度查询标签,用以查询数据的流行度.在此之后,云服务器将查询结果回传至用户,并与用户执行*UnPopularDedup*或*PopularUpload*,其中,*UnPopularDedup*表示非流行加密数据重复删除算法,*PopularUpload*表示流行加密数据重复删除算法.

3.2 *SystemSet*

1) KDC执行以下算法生成密钥对$\langle PK,MS\rangle$.
   a) 生成公共元素$\{q,G_1,G_2,g,e\}$,其中,G_1与G_2表示两个乘法循环群,q与g分别表示G_1的阶与某一生成元,e:$G_1\times G_1\to G_2$表示双线性映射;
   b) 选择哈希函数H:$\{0,1\}^*\to\{0,1\}^m$,并随机选择$T\in Z_q^{2\times m}$,其中,$T=\begin{pmatrix}t_{0,1}\cdots t_{0,m}\\ t_{1,1}\cdots t_{1,m}\end{pmatrix}$, $m\in N$;
   c) 选取$y\in Z_q$,$g_1\in G_1$,计算$Y=(g,g_1)^y$与g^T,其中,$g^T=\begin{pmatrix}g^{t_{0,1}}\cdots g^{t_{0,m}}\\ g^{t_{1,1}}\cdots g^{t_{1,m}}\end{pmatrix}$;
   d) 定义公钥$PK=\{q,G_1,G_2,g,e,g_1,Y,g^T,H\}$,主秘密$MS=\{y,T\}$.
2) 用户群$\{U_i\}_{i\in[1,Num_U]}$通过以下算法1获取各自私钥,其中,Num_U表示用户数量.

算法 1. 私钥生成算法.

Input:KDC主秘密*MS*,用户$\{U_i\}_{i\in[1,Num_U]}$的属性集合$S=\{AT_i\}_{i\in[1,Num_U]}$;

Output:$\{U_i\}_{i\in[1,Num_U]}$的私钥集合$\{SK_{AT_i}\}_{i\in[1,Num_U]}$.

1: **For** i=1 to Num_U **do**
2: 用户U_i发送属性集合AT_i至KDC;
3: KDC计算$h=H(at_1\|at_2\|\ldots\|at_n)$,$at_j\in AT_i$; //$H$表示密码哈希函数,$j\in[1,n]$,$n$表示$U_i$的属性个数;
4: KDC随机选择$z\in Z_q$,生成解密密钥$SK_{AT_i}=\{SK_{i1},SK_{i2}\}=\left\{g_1^y\left(\prod_{i\in[1,m]}g^{t_{h_i,i}}\right)^z,g^z\right\}$;
5: **Return** $\{SK_{AT_i}\}_{i\in[1,Num_U]}$;

3) KDC通过以下方式得到密文参数集合,并将其部署在云服务器.
   a) 生成随机数向量集合:$\{N_r=\langle\lambda_r,\mu_r\rangle\}_{r\in[1,Num_U]}$与$\{R_r=\langle\theta_r,\omega_r\rangle\}_{r\in[1,Num_U]}$;
   b) 计算$\{N_r-R_r\}_{r\in[1,Num_U]}$;
   c) 通过算法2加密$(\{N_r-R_r\},\{R_r\})_{r\in[1,Num_U]}$,得到$\begin{pmatrix}\{X_{r1}=Encrypt(PK,S,N_r-R_r)\}\\ \{X_{r2}=Encrypt(PK,S,R_r)\}\end{pmatrix}_{r\in[1,Num_U]}$,
   其中,$Encrypt(\cdot)$表示公钥加密算法.

算法 2. 属性加密算法.

Input:随机数向量集合$(\{N_r-R_r\},\{R_r\})_{i\in[1,Num]}$,公钥*PK*,访问结构*S*;

Output:随机值密文集合$(\{X_{r1}=Encrypt(PK,S,N_r-R_r)\},\{X_{r2}=Encrypt(PK,S,R_r)\})_{i\in[1,Num]}$.

1: **For** r = 1 to Num_U **do**
2: 计算$h=H(s_1\|s_2\|\ldots\|s_n)$,$s_i\in S$;

3: 随机选择$\varepsilon\in Z_q,\partial\in Z_q$,计算$\begin{pmatrix} C_{r1,1}=(N_r-R_r)\cdot Y^{\varepsilon}, C_{r1,2}=g^{\varepsilon}, C_{r1,3}=\left(\prod_{i\in[1,m]} g^{t_{h_i},i}\right)^{\varepsilon} \\ C_{r2,1}=R_r\cdot Y^{\partial}, C_{r2,2}=g^{\partial}, C_{r2,3}=\left(\prod_{i\in[1,m]} g^{t_{h_i},i}\right)^{\partial} \end{pmatrix}$;

//其中,h_i表示 h 的第 i 比特,$h_i\in\{0,1\}$;

4: **Return** $(\{X_{r1}=(S,C_{r1,1},C_{r1,2},C_{r1,3}),X_{r2}=(S,C_{r2,1},C_{r2,2},C_{r2,3}))_{i\in[1,Num]}$;

3.3 *PopularityCheck*

3.3.1 获取生成查询标签所需随机数

U_i选取短哈希函数 SH,计算数据 F_i的短哈希值 $sh_i=SH(F_i)$,并发送 sh_i至云服务器(短哈希函数具有较高的碰撞率,相同数据的短哈希值必定相同,不同数据的短哈希值可能相同).

1) 若云服务器中存在与 sh_i相同的短哈希值 $sh_i'=SH(F_i')$,则 U_i可能为数据 $F_i'(F_i)$的后继上传者,其中,F_i'表示 U_i'上传的数据,执行以下操作.

a) 云服务器将与 F_i'关联的信息发送至 U_i.该信息包含 η_i'与$\begin{pmatrix} X_{i1}'=Encrypt(PK,AT,N_i'-R_i') \\ X_{i2}'=Encrypt(PK,AT,R_i') \end{pmatrix}$,其中,$\eta_i'$表示云服务器为 U_i'选定的随机数; $X_{i1}'\in\{X_{r1}\},X_{i2}'\in\{X_{r2}\}$表示云服务器为 U_i'选取的密文参数(如前文所述,密文参数集合来自 KDC,云服务器无法获得明文信息);

b) U_i设定 $\eta_i=\eta_i'$,通过以下方法解密 X_{i1}'与 X_{i2}',其中,$Decrypt(\cdot)$表示公钥加密中的解密算法:

$$\langle\lambda_i'-\theta_i',\mu_i'-\omega_i'\rangle=Decrypt(PK,SK_i,X_{i1}')=\frac{C_{i1}'\cdot e(C_{i1,3}',SK_{i2})}{e(C_{i1,2}',SK_{i1})},$$

$$\langle\theta_i',\omega_i'\rangle=Decrypt(PK,SK_i,X_{i2}')=\frac{C_{r2,3}'\cdot e(C_{r2,3}',SK_{i2})}{e(C_{r2,3}',SK_{i1})};$$

c) U_i计算$\langle\lambda_i'-\theta_i',\mu_i'-\omega_i'\rangle+\langle\theta_i',\omega_i'\rangle$得到随机数向量$\langle\lambda_i',\mu_i'\rangle$, 并设定$\langle\lambda_i,\mu_i\rangle=\langle\lambda_i',\mu_i'\rangle$.

注意:若存在 $N_{\max}$个短哈希值与 sh_i相同,则以上操作执行 $N_{\max}$次,即,设定$\{\langle\eta_{i,j},\lambda_{i,j},\mu_{i,j}\rangle\}_{j\in[1,N_{\max}]}$为生成流行度查询标签所需参数.

2) 若云服务器无法查找出相同的短哈希值,则 U_i是数据 F_i的初始上传者.

a) 云服务器从密文参数集合$\{X_{r1}\},\{X_{r2}\}(r\in[1,n])$中随机选择 $X_{a1}\in\{X_{r1}\}$与 $X_{b2}\in\{X_{r2}\}(a\neq b)$,另外生成随机数$\eta_i$,一起发送至 U_i,并将 X_{a1},X_{b2}与 U_i关联;

b) U_i解密 X_{a1}与 X_{b2}得到:

$$\langle\lambda_a-\theta_a,\mu_a-\omega_a\rangle=Decrypt(PK,SK_i,X_{a1})=\frac{C_{a1,1}\cdot e(C_{a1,3},SK_{i2})}{e(C_{a1,2},SK_{i1})},$$

$$\langle\theta_b,\omega_b\rangle=Decrypt(PK,SK_i,X_{b2})=\frac{C_{b2,1}\cdot e(C_{b2,3},SK_{i2})}{e(C_{b2,2},SK_{i1})};$$

c) U_i计算$\langle\lambda_a-\theta_a,\mu_a-\omega_a\rangle+\langle\theta_b,\omega_b\rangle$得到随机数向量$\langle\lambda_i,\mu_i\rangle=\langle\lambda_a-\theta_a+\theta_b,\mu_a-\omega_a+\omega_b\rangle$,设定$\langle\eta_{i,j},\lambda_{i,j},\mu_{i,j}\rangle_{j=0}$为生成流行度查询标签所需参数,由于云服务器未找到相同的短哈希值,因此将 j 设定为固定值 0.

3.3.2 流行度查询

U_i使用随机数向量集合$\{\langle\eta_{i,j},\lambda_{i,j},\mu_{i,j}\rangle\}_{j\in[0,N_{\max}]}$(第 3.3.1 节中,两种情况下 j 的取值分别为 $j\in[1,N_{\max}]$与 $j=0$,将二者合并得到 $j\in[0,N_{\max}]$)与云服务器执行算法 3,以查询数据的流行度.

算法 3. 流行度查询算法.

Input:U_i持有的随机值$\{\langle\eta_{i,j},\lambda_{i,j},\mu_{i,j}\rangle\}_{j\in[0,N_{\max}]}$;

Output:流行数据,非流行数据.

1: **For** j=0 to $N_{\max}$ **do**

2: U_i 计算$(x_{i,j},y_{i,j})=\eta_{i,j}\cdot A+\lambda_{i,j}\cdot A+\mu_{i,j}\cdot B$; //$A$ 与 B 代表椭圆曲线上两个点;

3: U_i 计算盲化因子 $l_{i,j}=x_{i,j} \bmod n$,并计算密文 $C_{i,j}=H(F_i+l_{i,j})$;

4: U_i 计算 $\overline{C}_{i,j}=C_{i,j}-\mu_{i,j}$,并将其发送至云服务器;

5: **For** j=0 to $N_{\max}$ **do**

6: 云服务器计算 $\sigma_{i,j}=\eta_{i,j}-SK_{CSP}\cdot\overline{C}_{i,j}$; //$SK_{CSP}$ 表示云服务器私钥;

7: **If** 云服务器中存有与$\sigma_{i,j}$ 相同的值 $\sigma'_{i,j}$

8: 计算 $\sigma'_{i,j}(\sigma_{i,j})$ 的数量,并将其记作 $Count_{\sigma'_{i,j}}$;

9: 设定 $Num_{\sigma_{i,j}}=Count_{\sigma_{i,j}}$;

9: **Else**

10: 设定 $Num_{\sigma_{i,j}}=0$;

11: **If** $Num_{\sigma_{i,j}}<T$ //其中,T 表示系统设定流行度阈值;

13: **Return** 非流行数据; //云服务器与 U_i 执行非流行加密数据重复删除算法 *UpopularDedup*.

14: **Else**

15: **Return** 流行数据; //云服务器与 U_i 执行流行加密数据重复删除算法 *PopularDedup*.

3.3.3 UnpopularDedup

1) 假设云服务器中存在 $\sigma_{i,j}=\sigma'_{i,j}$,即 $F_i=F_i'$,则 U_i 是 F_i 的后继上传者(假设 U_i' 为 F_i' 的初始上传者).

a) 云服务器将 $L_i'=E(k'_{i,j},KF_i'-H(F_i'))$ 发送至 U_i,其中,$E(\cdot)$为对称加密算法,$k'_{i,j}=y'_{i,j} \bmod n$ 表示 U_i' 为保护数据 F_i' 的加密密钥 K'_{F_i} 而选取的密钥,$y'_{i,j}$ 由 U_i' 在执行 *PopularityCheck* 时计算得出;

b) 由 *PopularityCheck* 协议可知 $k_{i,j}=k'_{i,j}\leftarrow y_{i,j}=y'_{i,j}\leftarrow\sigma_{i,j}=\sigma'_{i,j}$.$U_i$ 使用 $k_{i,j}$ 解密 L_i' 得 $K'_{F_i}-H(F_i')$.由于 $H(F_i)=H(F_i')\leftarrow\sigma_{i,j}=\sigma'_{i,j}$,因此 $K_{F_i}=K'_{F_i}-H(F_i')+H(F_i)$;

c) U_i 使用 K_{F_i} 对 F_i 加密得到 $E(K_{F_i},F_i)$.由于 $K_{F_i}=K'_{F_i}$ 且 $F_i=F_i'$,因此 $E(K_{F_i},F_i)=E(K'_{F_i},F_i')$.故云服务器删除 $E(K_{F_i},F_i)$.

2) 若云服务器中不存在任何查询标签与$\sigma_{i,j}$ 相同,则执行以下协议.

a) 云服务器随机选择用户 U_z 的 $L_z=E(k_z,K_{F_z}-H(F_z))_{z\in[1,Num_U]}$,并将其发送至 U_i,其中,F_z(U_z 上传的数据)与 F_i 的短哈希值相同,但长哈希值不同,即 $sh_z=sh_i\wedge H(F_z)\neq H(F_i)$;

b) 由 *PopularityCheck* 可知 $y_{z,j}=y_{i,j}\leftarrow sh_z=sh_i$.因此,$U_i$ 使用 $k_{i,j}=y_{i,j} \bmod n$ 对 L_z 解密得到 $K_{F_z}-H(F_z)$;

c) U_i 使用 $K_{F_i}=K_{F_z}-H(F_z)+H(F_i)$(由于 $H(F_z)\neq H(F_i)$,因此 $K_{F_i}=K_{F_z}-H(F_z)+H(F_i)\neq K_{F_z}$)对 F_i 加密得到密文 $E(K_{F_i},F_i)$;

d) U_i 将 $E(K_{F_i},F_i)$ 存储在云服务器.

3.3.4 PopularDedup

1) 若 $Num_{\sigma_{i,j}}=T$,则 F_i 正由非流行数据向流行数据转换.

a) U_i 计算数据 F_i 的哈希值 $H(F_i)$;

b) U_i 设定 F_i 的加密密钥为 $K_{F_i}=H(F_i)+y_{i,j} \bmod n$;

c) U_i 使用 K_{F_i} 加密数据 F_i 得到 $E(K_{F_i},F_i)$,并将其上传至云服务器.

2) 若 $Num_{\sigma_{i,j}}>T$,表示 F_i 已是流行数据.由于 $E(K_{F_i},F_i)$ 已被存储在云服务器,用户不再执行上传操作,即采用效率更高的客户端加密数据重复删除(client-side deduplication).

4 安全分析与证明

结合前文所述的敌手模型,本节从以下 4 个方面分析方案的安全性.

4.1 密文参数与查询标签安全性

1) 密文参数安全.

定理 1. 若 $AT_{CSP}\not\models S$,则属性加密方案中的解密算法(*Decrypt*)无法正常执行,其中,AT_{CSP} 表示云服务器属性集合,$\not\models$表示云服务器属性集合无法满足用户属性集合.

证明:由 *SystemSet* 可知:

- 密文: $X_{r1}=(S,C_{r1}=(N_r-R_r)\cdot Y^{\varepsilon},C_{r2}=g^{\varepsilon},C_{r3}=\left(\prod_{i\in[1,m]}g^{t_{h_i,i}}\right)^{\varepsilon}$;
- 解密密钥: $SK_{AT_i}=\{SK_{i1},SK_{i2}\}=\left\{g_1^{y}\left(\prod_{i\in[1,m]}g^{t_{h_i,i}}\right)^{z},g^{z}\right\}$.

由 *PopularityCheck* 可知 $Decrypt(PK,SK_i,X_{j1})=\dfrac{C_{j1}\cdot e(C_{j3},SK_{i2})}{e(C_{j2},SK_{i1})}=\dfrac{C_{j1}\cdot e(C_{j3},g^{z})}{e\left(C_{j2},g_1^{y}\left(\prod_{i\in[1,m]}g^{t_{h_i,i}}\right)^{z}\right)}$.

若 $AT_{CSP}\not\models S$,则 $h_{CSP}=H(at_{CSP_1}\|at_{CSP_2}\|...\|at_{CSP_n})\neq H(s_1\|s_2\|...\|s_n)=h$,即 $g_1^{y}\left(\prod_{i\in[1,m]}g^{t_{h_{CSP_i},i}}\right)^{z}\neq g_1^{y}\left(\prod_{i\in[1,m]}g^{t_{h_i,i}}\right)^{z}$,因此,云服务器无法解密 X_{r1}.

同理,云服务器无法解密 X_{r2}. □

2) 流行度查询标签安全.

云服务器虽持有$\sigma_{i,j}$与参数密文 X_{j1},X_{j2},然而,由定理 1 可知,云服务器无法解密 X_{j1},X_{j2},故只能通过以下方式穷举查询标签,以猜测加密数据的明文信息.

a) 穷举数据集合$\{F_r\}_{r\in[1,n]}$;
b) 穷举随机参数值集合$\{x_t\}_{t\in[1,n]}$;
c) 穷举随机参数值集合$\{\mu_z\}_{z\in[1,n]}$;
d) 计算标签集合$\{\sigma_{CSP}=\eta_i-SK_{CSP}\cdot(H(F_r+x_t \bmod n)-\mu_z)\}$;
e) 将得到的结果与 σ_i' 逐一比较,其中, $\sigma_i'=\eta_i-SK_{CSP}\cdot(H(F_i+x_i \bmod n)-\mu_i)$,观察是否存在相等值;
f) 若存在相等值,则表明 $F_r=F_i$.

然而,由以上可知,云服务器攻击的时间复杂度为 $O(n^3)$.由于 n 可视为无限大值,因此在实际应用中,实现第 e)步是极为困难的.

4.2 防止假冒云服务器的行为

以下为恶意用户 U_D 假冒云服务器与受害者 U_i 运行 *PopularityCheck* 算法的过程.

a) U_i 向云服务器发出执行 *PopularityCheck* 请求;
b) U_D 截获请求消息,发送 $\eta_D G,X_{D1},X_{D2}$ 至 $t(a)$;
c) U_i 计算 $C_D=H(M+s_D)$ 与 $C_D'=C_D-\mu_D$,并将 C_D' 发送至 U_D;
d) U_D 将查询标签$\sigma_D=\eta_D-d_D\cdot C'$发送至 U_i;
e) 由于 U_D 持有$\eta_D G,X_{D1},X_{D2}$,因此可以对 C_D 采取离线穷举攻击,即执行以下操作:
   ① 穷举数据$\{F_r\}_{r\in[1,n]}$;
   ② 计算密文集合$\{C_r\}=\{H(F_r+l_D)\}$;
   ③ 与 C_i 逐一对比.若 $C_r=C_i$,则 $F_r=F_i$.

解决方法:用户在与云服务器通信之前,需要借助公钥基础设施(PKI)获取并验证云服务器身份,借助 PK_{CSP} 协商会话密钥对通信内容加密.U_D 便无法仿冒云服务器身份获取有用信息.

4.3 防止用户进行在线穷举攻击

定理 3. 恶意用户 U_D 无法对云服务器中的非流行数据 F_i 执行在线穷举攻击.

证明:不失一般性,U_D 的攻击方式为:

a) U_D 穷举数据 $\{F_r\}_{r\in[1,n]}$;

b) U_D 将穷举结果逐一与云服务器运行 *PopularityCheck* 和 *UnpopularDedup*;

c) 云服务器根据是否存在等式 $\sigma'_{i,j}=\sigma'_r$ 回复 U_D 相应信息;

d) U_D 根据响应,判断攻击是否成功.

由 *UnpopularDedup* 可知:

- 情况 a:当 F_r=F_i 时,云服务器将 $L'_i=E(k'_{i,j},K'_{F_i}-H(F'_i))$ 回复给 U_D;
- 情况 b:若 F_r 为首次上传数据,云服务器随机选择用户 U_z 的 $L_z=E(k_z,K_{F_z}-H(F_z))_{z\in[1,Num_U]}$ 发送至 U_D.

由于两种情况下,U_D 获得的伪随机数 $K'_{F_i}-H(F'_i)$ 和 $K_{F_z}-H(F_z)$ 的计算方式相同,U_D 无法区分情况 a 与情况 b,故无法对存储在云服务器的非流行数据进行在线穷举攻击. □

4.4 标签唯一性与正确性证明

1) 唯一性证明

由安全哈希算法 H 的抗碰撞性得到引理 1.

引理 1. 对于安全的哈希算法 H,若 $F'_i=F_i$,则 $H(F'_i)=H(F_i)$ 的概率是可忽略的.我们采用ε表示可忽略值:

$$Prob[H(F'_i)\neq H(F_i)\,|\,F'_i=F_i]<\varepsilon.$$

定理 3. 若 $F'_i=F_i$,则 $\sigma'_{i,j}\neq\sigma_{i,j}$ 的概率是可忽略的: $Prob[\sigma'_{i,j}\neq\sigma_{i,j}\,|\,F'_i=F_i]<\varepsilon$.

证明:

根据 *PopularityCheck* 可知:若 $sh'_i=sh_i$,则 $\eta'_iG=\eta_iG\wedge X'_{i1}=X_{i1}\wedge X'_{i2}=X_{i2}$,故 $l'_{i,j}=l_{i,j}\leftarrow(x'_{i,j},y'_{i,j})=(x_{i,j},y_{i,j})$.

由引理 1 可得:

$$Prob[H(F'_i+l'_{i,j})\neq H(F_i+l_{i,j})\,|\,F'_i=F_i\wedge l'_{i,j}=l_{i,j}]<\varepsilon\ .$$

因此, $\overline{C'_i}=\overline{C_i}\leftarrow C'_i=C_i$,故 $\sigma'_{i,j}=\sigma_{i,j}$.换言之,持有相同数据的不同用户,生成流行度查询标签 $\sigma_{i,j}(\sigma'_{i,j})$ 是唯一的. □

2) 正确性证明

用户 U'_i 已将数据 F'_i 的流行度查询标签 $\sigma'_{i,j}$ 存储在云服务器.

当 U_i 与云服务器执行 *PopularityCheck* 时,云服务器生成数据 F_i 的流行度查询标签$\sigma_{i,j}$,并且判断 $\sigma'_{i,j}=\sigma_{i,j}$ 是否成立.

定理 4. 若 $\sigma'_{i,j}=\sigma_{i,j}$,则 $F'_i\neq F_i$ 的概率是可忽略的:

$$Prob[F'_i\neq F_i\,|\,\sigma'_{i,j}=\sigma_{i,j}]<\varepsilon.$$

证明:不失一般性,由 *PopularityCheck* 可知$\sigma_{i,j}$=k_i–SK_{CSP}·($H(F_i$+$x_{t,j}$ mod $n)$–$\mu_{i,j}$).

a) 若 $\sigma'_{i,j}=\sigma_{i,j}$,则 $\eta'_{i,j}-SK_{CSP}\cdot(H(F'_i+x'_{i,j}\bmod n)-\mu'_{i,j})=\eta_{i,j}-SK_{CSP}\cdot(H(F_i+x_{i,j}\bmod n)-\mu_{i,j})$;

b) 由于 $sh'_i=sh_i$,故 $\eta'_{i,j}=\eta_{i,j}\wedge x'_{i,j}=x_{i,j}\wedge\mu'_{i,j}=\mu_{i,j}$,因此, $H(F'_i)=H(F_i)$;

c) 根据引理 1 可得:若 $\sigma'_{i,j}=\sigma_{i,j}$,则 $F'_i=F_i$.

证毕. □

5 实验分析

实验采用 C++语言,借助 OPENSSL[29],GMP[30],PBC[31]和 CP-ABE[32]函数库实现了系统软件.以阿里云作为云服务提供商,租用虚拟机配置为 4Core CPU,8GB 内存,1Mbps 带宽,1T 存储空间.椭圆曲线基域大小设定为 512bit,域中元素大小为 160bit.随机选取了 2 500 个文件存储在云服务器中.随机设定拥有每个文件的用户数量.

设置流行度阈值为 T=8,非流行数据与流行数据的比例大致为 3:4.

通过以下 3 组实验,证明方案的高效性.

a) 上传大小为 80MB 的文件 F_A,计算本方案各阶段的时间开销;

b) 上传大小为 10MB 的文件 F_B,分别测试本方案与 *PerfectDedup* 方案[19]的总时间开销;

c) 上传大小相同的文件,比较本方案、文献[17]中的方案、文献[19]中的方案各自所需的存储开销.

实验中的每步操作重复执行 20 次,取平均值作为实验结果.

1) 系统每阶段的时间开销.

将文件设定为以下 3 种情况:非流行数据($Count_{F_A}<T$)、流行度转换数据($Count_{F_A}=T$)与流行数据($Count_{F_A}>T$).分别测量 3 种情况下文件分块、查询标签生成、流行度查询、加密与上传各自所需要的时间开销.实验结果如图 2 所示,发生在用户端的文件分块、查询标签生成、加密所需要的时间开销较小.上传与流行度查询操作在云服务器端执行,所需要的时间开销远远超过用户端.当 $Count_{F_A}>T$ 时,用户不再需要文件的加密与上传操作,大幅减少了计算开销,节省了网络带宽.

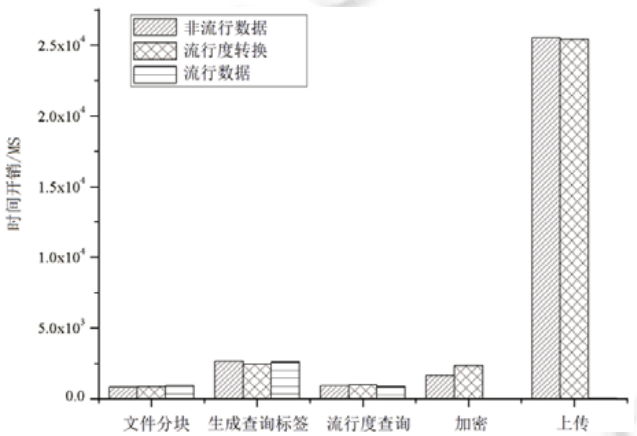

Fig.2 Time span on each stage of the system

图 2 系统每阶段的时间开销

如何高效且安全地识别冗余数据,是加密数据重复删除方案的基础.本文对较为优异的现有相关方案的生成查询标签算法进行了效率测试,并与本方案比较.实验结果如图 3 所示,本方案在生成查询标签方面,明显优于其他方案.

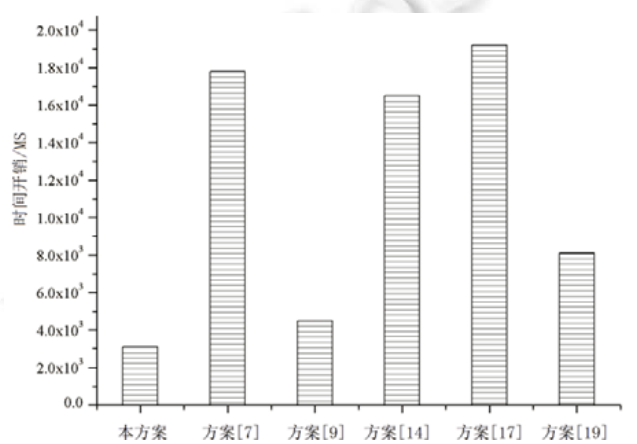

Fig.3 Time span on check tag generation of each scheme

图 3 各方案生成查询标签所需时间开销

为达到语义安全,本方案需要将初始上传属主的加密密钥安全共享至后继上传属主,这会使数据加密算法产生部分额外通信与计算开销.然而,由于密钥传递方式的设计较为高效,因此它对加密算法的性能影响较小.本方案与 CE[6]的比较结果在表 1 中给出,其中,$t(a)$表示本方案在执行加密算法时产生的时间开销,$t(b)$表示执行收敛加密时产生的时间开销,$t(b)-t(a)$表示执行两种加密算法时产生时间开销的差值,$\frac{t(b)-t(a)}{t(a)}$表示以上差值为系统带来影响的大小.实验结果表明:二者加密算法所需的时间开销相差甚小;且随着数据不断增大,差值渐成为可忽略值.

Table 1 Comparison of time span between our scheme and CE

表 1 本方案与 CE 方案的时间开销对比

数据大小/MB	5	10	20	45	120	400	1 100
$t(a)$/MS	210	321	644	1 311	3 643	12 012	32 024
$t(b)$/MS	1 005	1 212	1 450	2 098	4 550	12 731	32 799
$(t(b)-t(a))$/MS	795	891	806	787	907	719	775
$\left(\frac{t(b)-t(a)}{t(a)}\right)\Big/$MS	3.79	2.78	1.25	0.60	0.25	0.06	0.02

2) 较少的总时间开销.

本方案与 *PerfectDedup* 方案[19]所需要的总时间开销对比如图 4 所示.与 *PerfectDedup* 相比,由于本方案不需要进行第三方服务器的数据更新,流行度查询阶段需要的时间开销较小,因此在总时间开销方面,本方案具有较明显的优势.

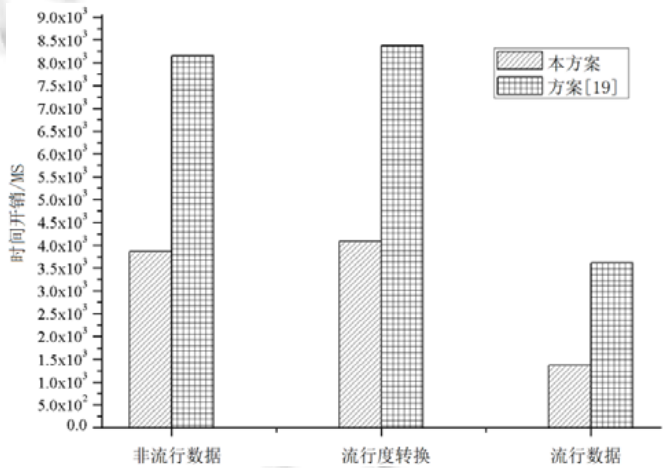

Fig.4 Comparison of total time span between our scheme and *PerfectDedup*

图 4 本方案与 *PerfectDedup* 方案的总时间开销对比

3) 占用更少的存储空间.

通过上传大小为 500MB 文件,测试本方案、文献[17]中的方案、文献[19]中的方案各自占用云服务器中的存储空间情况,实验结果如图 5 所示.由于本方案支持非流行数据重复删除,因此能够节省更多的存储空间.文件越大,优势越明显.

4) 方案特点比较.

由上述实验可知,摆脱实时在线第三方的依赖与划分数据流行度,是减少方案计算开销与通信开销的有效方法.表 2 分析了本方案与其他代表性方案是否具备上述两种方法的特点.

Table 2 Scheme characteristics comparison

表 2 方案特点比较

方案	Ref.[7]	Ref.[9]	Ref.[14]	Ref.[15]	Ref.[17]	Ref.[19]	本方案
无实时在线第三方	×	√	×	×	×	×	√
划分数据流行度	×	×	×	×	√	√	√

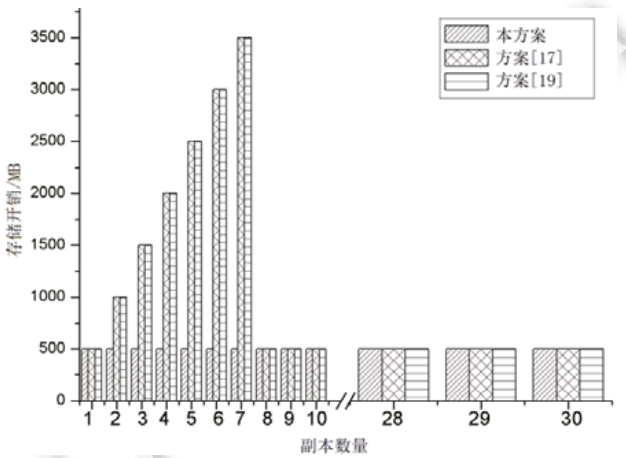

Fig.5 A comparison of three schemes of cloud server storage overhead (each file 500MB)

图 5 3 种方案中云服务器存储开销对比(每个文件 500MB)

6 总结与展望

本文提出一种无需初始数据上传用户和可信第三方实时在线参与的加密数据重复删除方法.基于椭圆曲线构造流行度查询标签,在语义安全的前提下,使用该标签识别数据冗余度与流行度.借助密文策略属性加密,保证查询标签生成协议与密钥共享协议的安全实现,同一数据副本的初始上传用户能够借助云服务商,将加密密钥安全离线共享至后继上传用户,实现非流行数据重复删除.改进后的收敛加密算法,能够使用户自行计算安全加密密钥,不仅保证了流行数据的存储安全,同时提高了云服务商消除流行重复加密数据的效率.本文最后进行了详细的安全性分析与效率评估,并与其他现有方案对比,证明本方案在满足语义安全的同时,进一步提高了加密数据重复删除系统的执行效率.

在本文基础上设计具有动态更新数据所有权的安全加密数据重复删除方案,是下一步的研究方向.

References:

[1] Lai J, Xiong J, Wang C, *et al*. A secure cloud backup system with deduplication and assured deletion. In: Proc. of the Int'l Conf. on Provable Security. Cham: Springer-Verlag, 2017. 74−83.

[2] Fu YJ, Xiao N, Liu F. Research and development on key techniques of data deduplication. Journal of Computer Research & Development, 2012,49(1):12−20 (in Chinese with English abstract).

[3] Shin Y, Koo D, Hur J. A Survey of Secure Data Deduplication Schemes for Cloud Storage Systems. ACM Press, 2017.

[4] Guarini D. Experts say Facebook leak of 6 million users' data might be bigger than we thought. 2013. http://www.huffingtonpost.com/entry/facebook-leak-data_n_3510100

[5] iCloud leaks of celebrity photos. 2014. https://en.wikipedia.org/wiki/ICloud leaks of celebrity photos

[6] Douceur JR, Adya A, Bolosky WJ, *et al*. Reclaiming space from duplicate files in a serverless distributed file system. In: Proc. of the Int'l Conf. on Distributed Computing Systems. IEEE, 2002. 617−624.

[7] Puzio P, Molva R, Onen M, *et al*. Cloudedup: Secure deduplication with encrypted data for cloud storage. In: Proc. of the 5th IEEE Int'l Conf. on Cloud Computing Technology and Science. IEEE, 2013. 363−370.

[8] Storer MW, Greenan K, Long DDE, *et al*. Secure data deduplication. In: Proc. of the 2008 ACM Workshop on Storage Security and Survivability. VA: ACM Press, 2008. 1−10.

[9] Jian L, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional independent servers. In: Proc. of the ACM Sigsac Conf. on Computer and Communications Security. ACM Press, 2015. 874−885.

[10] Liu XF, Sun WH, Lou WJ, *et al*. One-tag checker: Message-locked integrity auditing on encrypted cloud deduplication storage. In: Proc. of the IEEE Conf. on Computer Communications. IEEE, 2017. 1−9.

[11] Bellare M, Keelveedhi S, Ristenpart T. Message-locked encryption and secure deduplication. In: Proc. of the Int'l Conf. on the Theory and Applications of Cryptographic Techniques. Berlin, Heidelberg: Springer-Verlag, 2013.

[12] Abadi M, Dan B, Mironov I, *et al*. Message-locked encryption for lock-dependent messages. In: Proc. of the Advances in Cryptology (CRYPTO 2013). Berlin, Heidelberg: Springer-Verlag, 2013. 374−391.

[13] Bellare M, Keelveedhi S. Interactive message-locked encryption and secure deduplication. In: Proc. of the Public-key Cryptography (PKC 2015). Berlin, Heidelberg: Springer-Verlag, 2015. 296−312.

[14] Bellare M, Keelveedhi S, Ristenpart T. DupLESS: Server-aided encryption for deduplicated storage. In: Proc. of the Usenix Conf. on Security. USENIX Association, 2013. 179−194.

[15] Duan Y. Distributed Key Generation for Encrypted Deduplication: Achieving the Strongest Privacy. 2014. 57−68.

[16] Dinger J, Hartenstein H. Defending the Sybil attack in P2P networks: Taxonomy, challenges, and a proposal for self-registration. In: Proc. of the Int'l Conf. on Availability, Reliability and Security. IEEE, 2006.

[17] Stanek J, Sorniotti A, Androulaki E, *et al*. A secure data deduplication scheme for cloud storage. In: Proc. of the Int'l Conf. on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer-Verlag, 2014. 99−118.

[18] Douceur JR. The Sybil attack. In: Proc. of the Revised Papers from the 1st Int'l Workshop on Peer-to-Peer Systems. Springer-Verlag, 2002. 251−260.

[19] Puzio P, Molva R, Önen M, *et al*. PerfectDedup: Secure data deduplication. In: Proc. of the Int'l Workshop on Data Privacy Management. Springer Int'l Publishing, 2015. 150−166.

[20] Zhang FG, Wang CJ, Wang YM. Digital signature and blind signature based on elliptic curve. Journal of China Institute of Communications, 2001,22(8):22−28 (in Chinese with English abstract).

[21] Wang DQ, You L, DuanYC. Summarizing and comparison of the algorithms for the order of Jacobian group of elliptic curves over finite fields. NetInfor Security, 2014,8(7):41−47 (in Chinese with English abstract).

[22] Feng DG. Mathematical Methods and Techniques in Information Security. Beijing: Tsinghua University Press, 2009 (in Chinese).

[23] Hu L, Feng DG, Wen TH. Fast multiplication on a family of Koblitz elliptic curves. Ruan Jian Xue Bao/Journal of Software, 2003, 14(11):1907−1910 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/14/1907.htm

[24] Zhang K, Li H, Ma J, *et al*. Efficient large-universe multi-authority ciphertext-policy attribute-based encryption with white-box traceability. Science China Information Sciences, 2018.

[25] Bethencourt J, Sahai A, Waters B. Ciphertext-policy attribute-based encryption. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE Computer Society, 2007. 321−334.

[26] Cmalluhi QM, Trinh VC. A ciphertext-policy attribute-based encryption scheme with optimized ciphertext size and fast decryption. In: Proc. of the ACM on Asia Conf. on Computer and Communications Security. ACM Press, 2017. 230−240.

[27] Wang PP, Feng DG, Zhang LW. CP-ABE scheme supporting fully fine-grained attribute revocation. Ruan Jian Xue Bao/Journal of Software, 2012,23(10):2805−2816 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/4184.htm [doi: 10.3724/SP.J.1001.2012.04184]

[28] Liu ZB, Liu H, Huo YY. Data access control protocol for the cloud computing based on ciphertext-policy attribute based encryption (CP-ABE). NetInfor Security, 2014,13(7):57−60 (in Chinese with English abstract).

[29] Hu XT, Qin ZP, Zhang H, *et al*. Research and improved implementation of AES algorithm in OpenSSL. Microcomputer Information, 2009,25(12):83−85 (in Chinese with English abstract).

[30] Loukides M, Oram A. Programming with GNU Software. O'Reilly & Associates, 1997.

[31] Lynn B. The pairing-based cryptographic library. 2015. http://crypto.Stanford.edu/pbc/

[32] John B, Amit S. Brent W. Ciphertext-policy attribute-based encryption. 2006. http://acsc.cs.utexas.edu/cpabe/

附中文参考文献:

[2] 付印金,肖侬,刘芳.重复数据删除关键技术研究进展.计算机研究与发展,2012,49(1):12−20.

[20] 张方国,王常杰,王育民.基于椭圆曲线的数字签名与盲签名.通信学报,2001,22(8):22−28.

[21] 王冬勤,游林,段勖超.有限域上椭圆曲线 Jacobian 群求阶算法综述与比较.信息网络安全,2014,8(7):41−47.

[22] 冯登国.信息安全中的数学方法与技术.北京:清华大学出版社,2009.

[23] 胡磊,冯登国,文铁华.一类 Koblitz 椭圆曲线的快速点乘.软件学报,2003,14(11):1907−1910. http://www.jos.org.cn/1000-9825/14/1907.htm

[27] 王鹏翩,冯登国,张立武.一种支持完全细粒度属性撤销的 CP-ABE 方案.软件学报,2012,23(10):2805−2816. http://www.jos.org.cn/1000-9825/4184.htm [doi: 10.3724/SP.J.1001.2012.04184]

[28] 刘占斌,刘虹,火一莽.云计算中基于密文策略属性基加密的数据访问控制协议.信息网络安全,2014,13(7):57−60.

[29] 胡晓婷,覃中平,张红,等.OpenSSL 中 AES 算法的研究与优化.微计算机信息,2009,25(12):83−85.

张曙光(1991－),男,山东曲阜人,硕士,主要研究领域为云存储安全,区块链,隐私保护.

王利明(1978－),男,博士,正高级工程师,CCF 专业会员,主要研究领域为云存储安全,区块链,隐私保护,通信安全,5G 安全.

咸鹤群(1979－),男,博士,副教授,CCF 高级会员,主要研究领域为云存储安全,区块链,隐私保护.

刘红燕(1994－),女,硕士,主要研究领域为云存储安全,隐私保护.