软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2018,29(7):1909−1921 [doi: 10.13328/j.cnki.jos.005359]

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

基于离线密钥分发的加密数据重复删除方法*

张曙光[1,2], 咸鹤群[1,2,3], 王雅哲[3], 刘红燕[1,2], 侯瑞涛[1,2]

[1](青岛大学 计算机科学技术学院,山东 青岛 266071)

[2](中国科学院 网络测评技术重点实验室(中国科学院 信息工程研究所),北京 100093)

[3](信息安全国家重点实验室(中国科学院 信息工程研究所),北京 100093)

通讯作者: 咸鹤群, E-mail: xianhq@qdu.edu.cn

摘 要: 重复数据删除技术受到工业界和学术界的广泛关注.研究者致力于将云服务器中的冗余数据安全地删除,明文数据的重复删除方法较为简单.而用户为了保护隐私,会使用各自的密钥将数据加密后上传至云服务器,形成不同的加密数据.在保证安全性的前提下,加密数据的重复删除较难实现.目前已有的方案较多依赖于在线的可信第三方.提出一种基于离线密钥分发的加密数据重复删除方案,通过构造双线性映射,在不泄露数据隐私的前提下,验证加密数据是否源自同一明文.利用广播加密技术实现加密密钥的安全存储与传递.任意数据的初始上传者能够借助云服务器,以离线方式验证后继上传者的合法性并传递数据加密密钥.无需可信第三方在线参与,实现了云服务器对加密数据的重复删除.分析并证明了方案的安全性.仿真实验验证了方案的可行性与高效性.

关键词: 重复数据删除;双线性映射;隐私保护;数据的流行度

中图法分类号: TP309

中文引用格式: 张曙光,咸鹤群,王雅哲,刘红燕,侯瑞涛.基于离线密钥分发的加密数据重复删除方法.软件学报,2018,29(7):1909−1921. http://www.jos.org.cn/1000-9825/5359.htm

英文引用格式: Zhang SG, Xian HQ, Wang YZ, Liu HY, Hou RT. Secure encrypted data deduplication method based on offline key distribution. Ruan Jian Xue Bao/Journal of Software, 2018,29(7):1909−1921 (in Chinese). http://www.jos.org.cn/1000-9825/5359.htm

Secure Encrypted Data Deduplication Method Based on Offline Key Distribution

ZHANG Shu-Guang[1,2], XIAN He-Qun[1,2,3], WANG Ya-Zhe[3], LIU Hong-Yan[1,2], HOU Rui-Tao[1,2]

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

[2](Key Laboratory of Network Assessment Technology, CAS (Institute of Information Engineering, The Chinese Academy of Sciences), Beijing 100093, China)

[3](State Key Laboratory of Information Security (Institute of Information Engineering, The Chinese Academy of Sciences), Beijing 100093, China)

Abstract: Secure data deduplication has received great attention from both academic and industrial societies. It is highly motivated for cloud service providers to delete duplicated data from their storage. Plaintext data deduplication is a simple problem, but users tend to encrypt their data with their own keys before uploading them to the cloud, which makes it difficult to perform cross user deduplication. Most current solutions to the problem rely on trusted third parties. In this study, an encrypted data deduplication scheme is presented based on an offline key distribution protocol. A bilinear mapping is constructed to verify whether different encrypted data originate from

* 基金项目: 国家自然科学基金(61303197); 中国科学院网络测评技术重点实验室开放课题

Foundation item: National Natural Science Foundation of China (61303197); Open Project Program of the Key Laboratory of Network Assessment Technology, CAS

本文由“面向隐私保护的新型技术与密码算法”专题特约编辑薛锐研究员推荐.

收稿时间: 2017-05-29; 修改时间: 2017-07-13; 采用时间: 2017-08-22; jos 在线出版时间: 2017-10-17

CNKI 网络优先出版: 2017-10-17 13:38:03, http://kns.cnki.net/kcms/detail/11.2560.TP.20171017.1338.006.html

the same plaintext. Secure key storage and key delivery is achieved by using the broadcast encryption technique. An original uploading user of some data can validate successive uploading users via the cloud service provider, and the data encryption key can be distributed in an offline manner. The cloud service provider can accomplish encrypted data deduplication with no online interaction with any trusted third party. The security of the proposed scheme is analyzed and proven. Simulation experiments show that the scheme is efficient and applicable.

Key words: deduplication; bilinear mapping; privacy preservation; data popularity

随着信息技术的快速发展,数据量呈指数级增长.本地存储资源的逐渐紧张导致云存储服务的需求不断增加.云服务提供商为用户提供数据收集、压缩、加密和传输等在线存储服务[1].云存储技术的发展给用户带来便利的同时,也给云服务提供商带来了新的难题.重复存储的数据给云服务器造成了大量冗余.因此,云服务提供商积极寻找实现数据冗余最小化的技术,从而节约成本.目前已被广泛采用的技术是跨用户重复数据删除,即每个数据副本(文件或块)只存储 1 次,并给拥有此数据所有权的用户创建访问链接[2,3],该技术可以节省大量存储空间和网络带宽.研究表明,重复数据删除可以有效地降低备份应用程序的存储需求,甚至高达 90%~95%[4].类似地,在标准文件系统中的存储需求可降低 68%[5].

信息安全问题的日益凸显使用户对数据隐私的重视程度不断提高.越来越多的用户不再直接将数据上传至云服务器,而是将加密后的密文数据上传并存储在云服务器上.针对明文数据的跨用户重复数据删除较为简单.当数据为密文时,由于相同的数据被不同用户的密钥加密后,得到的密文不同,因此云服务器无法识别并删除冗余数据.为了解决此难题,最原始的方案是在上传数据之前采用某个全局密钥对其加密.全局密钥存储在云服务器,若云服务器不可信,方案的安全性则难以保证.在此基础上,研究者提出收敛加密方案,首先将数据进行散列计算,使用散列结果作为加密密钥对数据进行对称加密.由于散列函数具有确定性的结果,因此,相同数据加密之后得到的密文必然相同,故云服务器能够完成重复数据删除[6].然而,收敛加密容易遭受离线穷举攻击,无法达到语义安全的要求.当数据隐私度很高时不能采用该方法[7].为了提高重复数据删除的执行效率,有研究者提出,当数据的隐私度较低时,用户可以使用收敛加密保护数据.但当数据的隐私度较高时,用户必须采用自己独有的密钥对数据进行加密.研究者将数据划分为两类:非流行数据与流行数据.非流行数据是隐私度较高的数据,拥有此数据副本的用户数量小于设定阈值.流行数据是隐私度较低的数据,其拥有者数量等于或大于设定阈值[8,9].

为了解决非流行数据加密后的重复删除问题,有研究者提出,持有相同数据的用户之间可以进行密钥传递.即如果用户 A 将某数据加密并首次上传至云服务器后,拥有相同数据的用户 B 也希望将数据上传,则 A 可以将数据加密密钥通过某种方式传递给 B,B 使用该密钥对数据加密会得到相同的密文.然而,如何安全、高效地进行密钥传递是较难解决的问题.

本文的贡献如下:(1) 构造了一个基于双线性映射的方案,在不泄漏任何明文信息的前提下,使云服务器能够验证加密数据是否源自同一明文,充分保护用户数据隐私.(2) 使用广播加密,构造了一种安全的密钥传递方案,使云服务器能够对非流行的加密数据执行重复删除.对传统收敛加密算法加以改进,提高了流行数据的存储安全性.(3) 首次提出无需在线可信第三方且不要求初始上传者在线的加密数据重复删除方案,用户只需与云服务器交互即可实现数据上传和重复数据删除.

本文第 1 节介绍关于重复数据删除的相关研究工作.第 2 节介绍系统模型与设计目标.定义和预备知识在第 3 节中给出.第 4 节详细叙述方案的流程.第 5 节与第 6 节分别是安全分析和实验.第 7 节是总结与展望.

1 相关工作

传统的加密方案不适用于对重复数据的删除.因此,收敛加密成为重复数据删除的首选方案[6,10−12].虽然收敛加密简便且高效,但因其密钥是由数据明文经过散列计算得到的,数据的信息熵较低时,容易遭受离线穷举攻击[13].Perttula 等人提出在加密密钥中添加一个私有元素 X,克服了收敛加密的弱点,但执行效率较低[12].Bellare 等人提出密码学原语 message-locked encryption(MLE)[14],密钥计算方法的本质与收敛加密相同,故方案无法达

到语义安全.Puzio 等人提出了 ClouDedup 方案[7],将数据加密和解密过程外包给可信第三方 IS.该方案无法抵御云服务器在线穷举攻击,同时无法防御云服务器与 IS 的合谋攻击.Bellare 等人提出 DupLess[15],用户与可信第三方 IS 通过运行 oblivious pseudorandom function(OPRF)协议产生数据的加密密钥,方案通过限制每个周期内某个用户发往 IS 请求的次数,有效防止了云服务器的在线穷举攻击.然而,如果云服务器与可信第三方合谋,数据的安全性将遭到严重威胁.Stanek 等人提出,若数据的隐私度不同,则需要的安全保护程度应有所不同.因此,研究者将数据划分为高隐私度和低隐私度两种类型,即非流行数据和流行数据[9].非流行数据需要双层加密,内层为收敛加密,外层是门限密码加密.当云服务器中数据副本的数量达到设定阈值时,流行度发生转化,云服务器可自行解除外层加密,并实现重复数据删除.此外,研究者提出使用第四方服务器 identity server 防止女巫攻击[16].但该方案易遭受 IS 的离线穷举攻击,且无法对非流行数据进行重复数据删除,给云服务器带来额外的带宽开销与存储冗余.Puzio 提出 perfectDedup 方案[8],通过完全散列函数与可信第三方 IS 查询数据的流行度,使用语义安全的对称加密算法保护非流行数据.当数据的流行度发生转换时,删除非流行加密数据,使用收敛加密对流行数据加以保护.与之前的多个方案类似,该方案必须使用可信第三方,在现实应用中较难实现,实用性不强.该方案对非流行数据依然无法进行重复数据删除.Liu 等人提出无需可信第三方即可完成安全重复数据删除的方案[17].拥有相同数据副本的用户可通过 password authenticated key exchange(PAKE)协议进行密钥传递[17−19],使用相同的密钥对数据加密,得到相同的加密数据.虽然此方案安全性较高,但是由于每个用户在上传数据之前都需要与其他用户执行 PAKE 协议,且无论数据的隐私度如何,都需进行密钥传递,这将给云服务器带来额外的计算开销与通信开销.除此之外,此方案要求参与用户同时在线,显著降低了重复数据删除的执行效率.Cui 等人提出了一种基于属性的存储系统,借助混合云实现安全重复数据删除功能.私有云负责重复检测,公共云用于数据存储[20].方案的计算开销较大,且混合云在实际应用中较难部署.

基于上述方案待解决的问题,本文提出了一种基于离线密钥分发的加密数据重复删除方案,在无需可信第三方在线参与的条件下,实现对加密数据的重复删除.

2 系统模型与设计目标

2.1 系统模型

如图 1 所示,本文给出的系统模型包含 3 种实体:用户群、广播中心(BC)和云服务器.系统建立时,BC 为用户群或云服务器提供广播信息.云服务器为用户群提供加密数据存储与共享服务.用户可以与云服务器交互.非流行数据的初始上传者能够借助云服务器以离线的方式验证后继上传者的合法性,并将数据加密密钥安全地传递至后继上传者.流行数据的加密密钥可由用户自行计算.当数据相同时,不同用户计算出的密钥相同,加密得到的密文也相同.云服务器检测到冗余的加密数据,执行重复数据删除,并为拥有此数据的用户创建访问链接,记录用户数量.

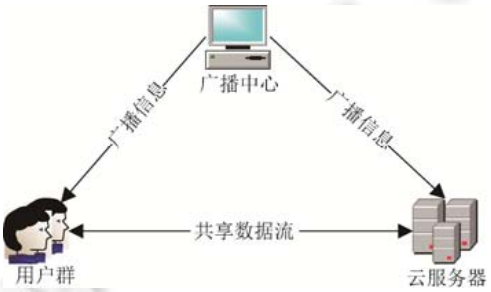

Fig.1 System model

图 1 系统模型

用户群包括数据的先前上传者与当前上传者,其中,当前上传者可分为初始上传者和后继上传者.如图 2 所示,先前上传者是已将数据 m_i 存储在云服务器的用户.当前上传者 U_j 是尝试上传某数据 m_j 的用户.若 m_j=m_i,则

U_j为m_i的后继上传者;若云服务器中不存在m_j,则U_j为m_j的初始上传者.

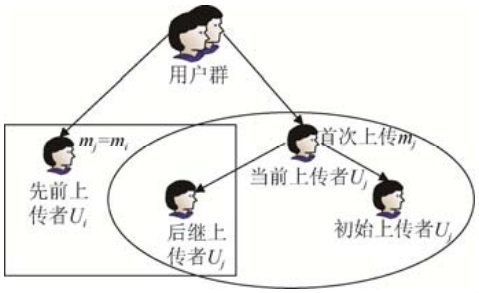

Fig.2 User category

图 2 用户分类

2.2 设计目标

本方案的设计目标是云服务器能够安全地完成加密数据重复删除,因此,方案应满足以下性质.

(1) 用户在查询数据的流行度时,所使用的查询标签不会泄露数据的任何明文信息.

(2) 非流行数据的加密密钥能够安全地传递至后继上传者.

(3) 确保流行加密数据的存储安全.

(4) 云服务器能够安全删除冗余的非流行加密数据与流行加密数据.

3 定义和预备知识

3.1 基于离线密钥分发的加密数据重复删除方案

一个基于离线密钥分发的加密数据重复删除方案包含以下 4 种操作:GetKey、PopularityQuery、UnpopularDedup 与 PopularDedup.

(1) GetKey:授权用户获取辅助密钥.广播中心将$M=(X_1,X_2)$安全发送至授权用户.

(2) PopularityQuery:数据的流行度查询.通过构造双线性映射查询标签,在不泄露数据信息隐私的情况下,完成数据的流行度查询.

(3) UnpopularDedup:非流行数据重复删除.持有相同数据的用户能够获取相同的加密密钥.云服务器检测到冗余加密数据,执行重复数据删除.对于首次上传的数据,用户获得一个全新密钥,并将加密数据存储在云服务器.

(4) PopularDedup:流行数据重复删除.① 拥有此数据的用户数量等于流行度阈值,使用改进后的收敛加密算法确保数据的安全性.② 当拥有此数据的用户数量超过流行度阈值后,执行客户端重复数据删除.

3.2 双线性映射

设$(G_1,+),(G_1,\cdot)$是阶为大素数 P 的加法循环群和乘法循环群.Z_P为模 P 的剩余类整环,Z_P^*是 Z_P的可逆元集合.定义双线性映射 $e:G_1\times G_1\to G_2$,并满足以下 3 种性质[21,22].

双线性:任意$M,N\in G_1$,且$\alpha,\beta\in_R Z_P^*$,都有$e(M^\alpha,N^\beta)=e(M,N)^{\alpha\beta}$;

可计算性:任意$M,N\in G_1$,存在有效算法可以计算$e(M,N)$;

非退化性:存在$M,N\in G_1$,使得$e(M,N)\neq\varepsilon$(其中,ε是G_1的幺元).

3.3 基于身份广播加密IBBE

在 IBBE 方案中,权威的广播中心 BC 为每个拥有授权身份的用户分发私钥sk_{IDi},用户可使用sk_{IDi}解密广播消息[23,24].

Setup(λ,n):系统初始设置.输入安全参数λ和一次加密中接收消息的最大用户数量 n,输出密钥对(PK,MSK).

主密钥 MSK 由 BC 保存.

$Extract(MSK,ID_i)$:生成用户私钥.输入 MSK 和身份 ID_i,输出 ID_i 对应的私钥 sk_{IDi},将 sk_{IDi} 安全发送至相应用户.

$Encrypt(S,PK)$:加密消息并广播消息密文.输入 PK 和用户身份集合 $S=\{ID_i\}i\in(1,2,...,n-1)$, 输出$(Hdr,K)$,其中 Hdr 是报头,K 是加密密钥.当BC需要给 S 中的授权用户发送消息 $M\in\{0,1\}^*$ 时,BC生成 $(Hdr,K)\leftarrow Encrypt(S,PK)$, 使用 K 对 M 加密得到 C_M,广播 (Hdr,S,C_M).

$Decrypt(S,ID_i,sk_{ID_i},Hdr,PK)$:用户解密广播消息密文.输入子集 S、身份 ID_i 和其私钥 sk_{IDi}、报头 Hdr 和公钥 PK.如果 $ID\in S$,用户即可使用 sk_{IDi} 对 Hdr 解密得到 K,使用 K 对 C_M 解密,恢复消息 M[25,26].

4 基于离线密钥分发的加密数据重复删除方案

4.1 符号说明

$(G_1,+),(G_1,\cdot)$表示阶为大素数 P 的加法循环群和乘法循环群,g 表示 G_1 的一个生成元.定义双线性映射 $e:G_1\times G_1\rightarrow G_2$.$H$ 为密码散列函数, $H:Z_P\rightarrow G_1$.SH 表示能够抵抗穷举攻击的短散列函数.数据 $Data$ 被分割成 v 个数据块,即 $Data=(m_1,m_2,m_3,...,m_v)$.$T$ 为流行度阈值.E 与 D 代表对称加密与解密.BC 为广播中心,S 是授权用户身份集合,即 $S=\{ID_1,...,ID_{n-1}\}$, n 为用户和服务器的总数量.

4.2 GetKey

如图 3 所示,系统建立时,用户会得到 4 个辅助密钥 $X_1,X_2,X_3=X_1+X_2$ 与 PK_{CSP},其中,X_1 与 X_2 来自广播中心 BC.PK_{CSP} 来自云服务器.云服务器从广播中心 BC 获得一个密钥池 $KP=\{\tau_i=E(X_3,r_i)\},i\in(1,2,...,n-1)$, 其中, $\{r_i\},i\in(1,2,...,n-1)$ 表示伪随机数生成器生成的 n–1 个固定长度的随机数.

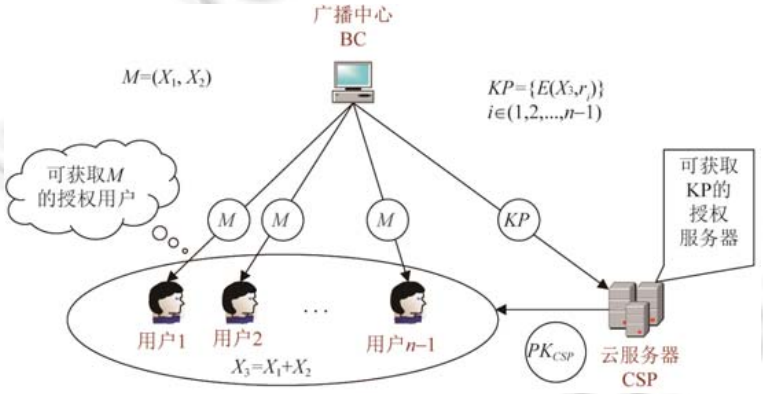

Fig.3 Get auxiliary keys

图 3 获取辅助密钥

4.3 PopularityQuery

由于数据块被划分为非流行数据块 $unpopular$ 与流行数据块 $popular$ 两种类型,且每种类型的加密算法是不同的.因此,在上传数据块之前,当前上传者 U_j 需要查询数据块的流行度.文献[8,9]提出,通过可信且实时在线的第三方查询数据块的流行度.然而,此类方案在实际应用中较难实现.因此,我们提出一种无需可信且实时在线第三方的流行度查询协议,并将其命名为 PopularityQuery.如图 4 所示,在执行 PopularityQuery 协议之前,初始上传者 U_i 已将 $sh_i=SH(m_i),Y_i=g^{y_i},L_1=e(g^{X_1},\&_i)$ 存储在云服务器上,其中,y_i 表示 U_i 独有的私钥, $\&_i=H(m_i)^{y_i}$ 表示 U_i 对数据块 m_i 的签名.

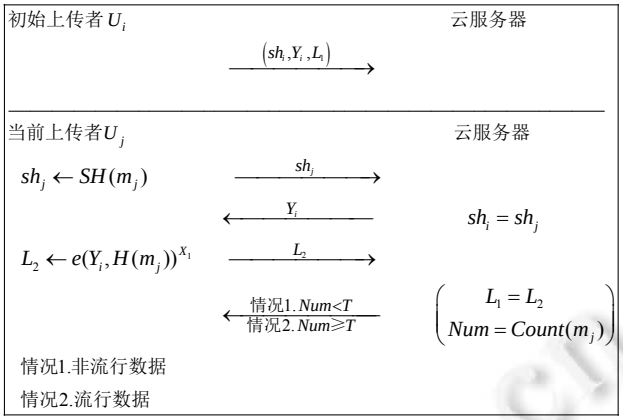

Fig.4 PopularityQuery

图 4 流行度查询

(1) 当前上传者 U_j 将短散列值 $sh_j=SH(m_j)$上传至云服务器.

若存在 $sh_j=sh_j$,云服务器将 U_i 的公钥 Y_i 发送至 U_j.U_j 使用辅助密钥 X_1,通过双线性映射计算得到查询标签 $e(Y_i,H(m_j))^{X_1}$,并将其上传至云服务器.

(2) 云服务器判断是否存在以下等式:

$$e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}.$$

① 若存在,则 $m_i=m_j$.云服务器统计出能够访问此数据块的用户数量 $Count(m_j)$,并与设定阈值 T 对比.若 $Count(m_j)<T$,则 m_j 为非流行数据块.反之,为流行数据块.

② 若不存在,说明 m_j 为非流行数据块.

4.4 UnpopularDedup

初始上传者 U_i 已将数据块的短散列值 $sh_i=SH(m_i)$、查询标签 $e(g^{X_1},\&_i)$、加密数据块 $E(K_{m_i}, m_i)$和加密密钥密文 $E(X_3,K_{m_i}-H(m_i))$ 存储在云服务器,其中, K_{m_i} 是用户使用伪随机数生成器生成的固定长度的随机数.

当前上传者 U_j 通过算法 PopularityQuery 查询数据块 m_j 的流行度.若 m_j 为非流行数据,根据不同条件分为以下两种情况.

(1) 若满足以下条件:

$$(\exists e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}) \wedge (0<Count(e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1})<T),$$

则 m_j 为非流行数据块,且 m_j 已存储在云服务器,即 U_j 为 m_j 的后继上传者.

① 云服务器将 $E(X_3,K_{m_i}-H(m_i))$ 发送至 U_j.

② U_j 使用 X_3 对 $E(X_3,K_{m_i}-H(m_i))$ 解密得到 $K_{m_i}-H(m_i)$.

③ U_j 使用 $K_{m_i}=K_{m_i}-H(m_i)+H(m_j)$ 对 m_j 加密得到 $E(K_{m_i}, m_j)$, 并将其上传至云服务器.

④ 由于 $E(K_{m_i}, m_j)=E(K_{m_i}, m_i) \leftarrow m_j=m_i$, 故云服务器删除冗余的加密数据 $E(K_{m_i}, m_j)$.

(2) 若云服务器无法找到相同查询标签,证明 m_j 为非流行数据块,且 m_j 不存在于云服务器中,即 U_j 为 m_j 的初始上传者.

① 云服务器在密钥池中随机选择 $\tau_z=E(X_3,r_z)(z=1,2,...,n-1)$, 并发送至 U_j.

② U_j 使用 X_3 对 $\tau_z=E(X_3,r_z)$ 解密得到 r_z.

③ U_j 使用 $K_{m_j}=r_z+H(m_j)$ 对 m_j 加密得到 $E(K_{m_j}, m_j)$, 并将密文存储在云服务器.

无论上传者 U_j 是初始上传者还是后继上传者,在上传加密数据块的同时,都需将查询标签 $e(g^{X_1},\&_i)$上传至云服务器.

上述算法如图 5 所示.

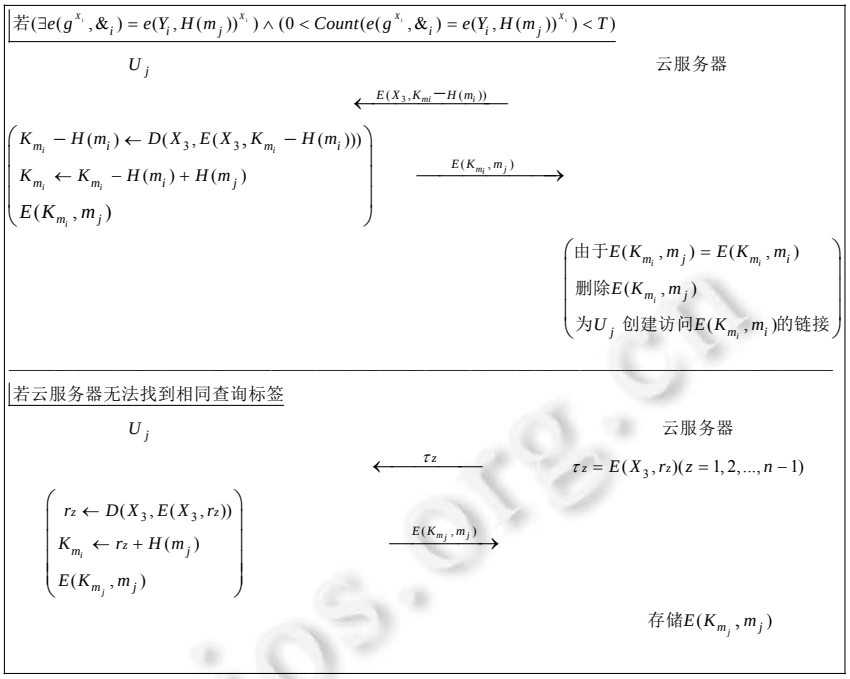

Fig.5 UnpopularDedup

图 5 非流行数据重复删除

4.5 PopularDedup

由于流行数据块的隐私度较低,因此可采用改进后的收敛加密算法对其加密.U_j 查询得知 m_j 为流行数据块,云服务器计算拥有 m_j 的用户数量 $Count(m_j)$.

如图 6 所示,其中,a 为当 $Count(m_j)=T$ 时,U_j 使用 $X_j=H(m_j)+X_2$ 对 m_j 加密得到 $C=E(X_j,m_j)$,将密文 C 上传至云服务器.b 为若 $Count(m_j)>T$,则改用效率更高的客户端重复数据删除(client-side deduplication),U_j 无需上传密文,云服务器为 U_j 创建访问此加密数据块的链接.

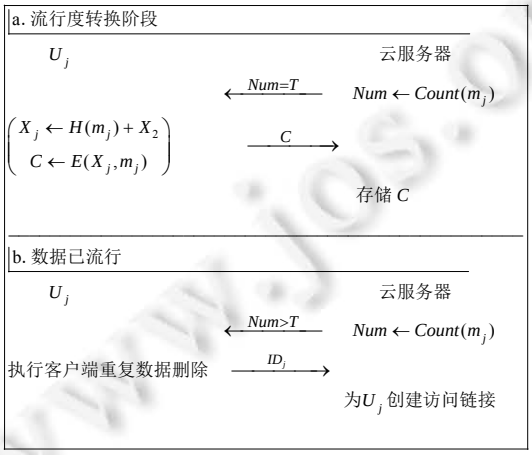

Fig.6 PopularUpload

图 6 流行数据块上传

5 安全性分析与证明

本节从以下 4 个方面详细分析该方案的安全性.

(1) 数据块验证的安全性证明

通过直接比较散列值判断数据块是否相同的方法,容易遭受云服务器的离线穷举攻击.本方案可有效避免此类威胁,并且能够将云服务器中存储的加密数据块与初始上传者的身份绑定.安全性证明如下文所述.

① 数据块验证结果的唯一性

本方案的安全性建立在特殊散列函数 H 安全性的基础上,G_1 表示阶为大素数 P 的加法循环群,g 表示 G_1 的生成元.由 H 的安全性假设,可得以下引理.

引理 1. 对于安全的特殊散列函数 H:$\{0,1\}^*\rightarrow G_1$,不同的数据块 m_i 与 m_j 拥有相同散列值的概率是可忽略的.我们采用ε表示可忽略值.

$$Prob\left[H(m_i)=H(m_j)\mid m_i\neq m_j\right]<\varepsilon.$$

定理 1. 在验证数据块是否相同时,初始上传者 U_i 的查询标签为 $e(g^{X_1},\&_i)$,当前上传者 U_j 的查询标签为 $e(Y_i,H(m_j))^{X_1}$.当 $m_i\neq m_j$ 时,$e(g^{X_1},\&_i)$与 $e(Y_i,H(m_j))^{X_1}$ 相同的概率是可忽略的.

$$Prob\left[e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}\mid m_i\neq m_j\right]<\varepsilon.$$

证明:不失一般性,由双线性映射的性质可得以下推论:

$$e(Y_i,H(m_j))^{X_1}=e(g^{y_i},H(m_j))^{X_1}=e(g,H(m_j))^{y_i\cdot X_1}=e(g^{X_1},H(m_j)^{y_i}).$$

由引理 1 可得,若 $m_i\neq m_j$, 则 $H(m_i)\neq H(m_j)$, 故 $e(g^{X_1},\&_i)=e(g^{X_1},H(m_i)^{y_i})\neq e(g^{X_1},H(m_j)^{y_i})$, 即:

$$Prob\left[e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}\mid m_i\neq m_j\right]<\varepsilon.$$

换言之,当且仅当 m_i=m_j 时,$e(g^{X_1},\&_i)$=$e(Y_i,H(m_j))^{X_1}$ 才会成立.因此,数据块的验证结果是唯一的. □

② 数据块验证结果的正确性

定理 2. 若等式 $e(g^{X_1},\&_i)$=$e(Y_i,H(m_j))^{X_1}$ 成立,m_i 与 m_j 不同的概率是可忽略的.

$$Prob\left[m_i\neq m_j\mid e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}\right]<\varepsilon.$$

证明:不失一般性,假设 U_i 的查询标签为 $e(g^{X_1},\&_i)$,由双线性映射性质可得以下等式:

$$e(g^{X_1},\&_i)=e(g^{X_1},H(m_i)^{y_i})=e(g,H(m_i))^{X_1\cdot y_i}=e(g^{y_i},H(m_i))^{X_1}=e(Y_i,H(m_i))^{X_1}.$$

由引理 1 可得: $m_i=m_j\leftarrow e(Y_i,H(m_i))^{X_1}=e(Y_i,H(m_j))^{X_1}\leftarrow e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}$.

因此: $Prob\left[m_i\neq m_j\mid e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}\right]<\varepsilon.$ □

(2) 防止查询标签泄露隐私数据块的明文信息

引理 2. G_1 表示阶为大素数 P 的乘法循环群,给定 $g, g^a, h\in G_1$, 其中, $a\in Z_P^*$, 计算 $h^a\in G_1$ 是困难的(计算 Diffie-Hellman 问题的困难性).

定理 3. 在用户不与云服务器合谋的情况下,云服务器无法以离线穷举攻击的方式从查询标签中获取数据块的任何明文信息.

证明:云服务器对用户发来的查询标签 $e(Y_i,H(m_j))^{X_1}$ 进行离线穷举攻击.

穷举大量数据块 $\{m_r\}, r\in(1,2,3,...,n)$, 试图找到 m_r=m_j.为了验证 m_r 与 m_j 是否相等,云服务器需要计算 $e(Y_i,H(m_j))^{X_1}$ 并将其与 $e(Y_i,H(m_j))^{X_1}$ 进行比较.云服务器容易计算 $e(Y_i,H(m_r))$,但由引理 2 可知,即使持有 $e(Y_i,H(m_j))$ 和 $e(Y_i,H(m_j))^{X_1}$,计算 $e(Y_i,H(m_r))^{X_1}$ 也是困难的.因此,云服务器无法以离线穷举攻击的方式从查询标签中获取数据块明文的任何信息. □

(3) 防止用户进行在线穷举攻击

定理 4. 在本方案中,用户 U_D 在持有辅助密钥 X_1 的情况下,无法对存储在云服务器中的非流行数据块进行

在线穷举攻击.

证明:

① U_D 穷举数据块 $\{m_r\}, r \in (1,2,3,...,n)$, 构造集合 $\{e(Y_i, H(m_r))^{X_1}\}$.

② U_D 将集合中的元素逐一发送至云服务器.

③ 云服务器根据是否存在等式 $e(g^{X_1},\&_i)=e(Y_i,H(m_j))^{X_1}$,回复 U_D 相应的信息.

④ U_D 根据回复的信息判断哪些数据块存储在云服务器.

由算法 UnpopularDedup 可知:

情况(a).当 $m_i=m_r$ 时,云服务器将 $E(X_3, K_{m_i} - H(m_i))$ 回复给 U_D.

情况(b).若云服务器中不存在 m_r,云服务器在密钥池中随机选择 $\tau = E(X_3, r_z)(z = 1,2,...,n-1)$, 并回复给 U_D.

由于两种情况下 U_D 获得的 $K_{m_i} - H(m_i)$ 和 r_z 是由相同方法得到伪随机数,U_D 无法区分情况(a)和情况(b),故无法对存储在云服务器的非流行数据块进行在线穷举攻击. □

(4) 防止恶意用户截取信息

由于恶意用户 U_D 是广播中心 BC 的授权用户,因此 U_D 拥有广播信息 $M=(X_1,X_2)$.假设 U_D 截获了用户 U_i 上传的查询标签 $e(g^{X_1},\&_i)$,并对 $e(g^{X_1},\&_i)$采取离线穷举攻击.

① 穷举数据块 $\{m_r\}, r \in (1,2,3,...,n)$.

② 构造集合 $\{e(Y_i, H(m_r))^{X_1}\}, r \in (1,2,3,...,n)$.

③ 查看是否存在以下等式 $e(g^{X_1},\&_i)=e(Y_i,H(m_r))^{X_1}$.

若 $e(g^{X_1},\&_i)=e(Y_i,H(m_r))^{X_1}$,则 $m_i=m_r$,数据块 m_i 的明文信息便遭到泄露.

解决方法:

① 系统初期,云服务器随机生成密钥对$\langle PK_{CSP}, SK_{CSP}\rangle$.

② 云服务器将 PK_{CSP} 发送至 U_i.

③ U_i 使用 PK_{CSP} 加密 $e(g^{X_1},\&_i)$得到密文 $Enc(PK_{CSP}, e(g^{X_1},\&_i))$,并将密文发送至云服务器.

如此,即使 U_D 截取查询标签也无法对其造成任何安全威胁.

6 仿真与实验分析

实验采用 PBC[27]、GMP[28]、PBC_bce[29]和 OPENSSL[30]函数库,使用 C++语言编程实现了客户端与服务器软件.选用腾讯云的云服务器,其配置为 4GB 内存,4 核 CPU,1Mbps 带宽,1T 存储盘.设定大小为 512bit 的基域,其中每个元素 $element \in Z_P^*$ 的大小为$|P|$=160bit.为了模拟真实情景,我们在云服务器中存储了超过 2 000 个不同的文件,且随机设定拥有每个文件的用户数量 $Count_F$.建立用户数量表 $CountTab$,记录 $Count_F$.设定流行度阈值 T=7,使非流行数据与流行数据的比例大致为 2:3.

实验共分 3 部分:首先,上传一个大小为 30MB 的文件 F_A,记录方案中各阶段所需的时间开销.然后,上传大小为 10MB 的文件 F_B,计算方案所需的总时间开销,并与 perfectDedup 方案对比.最后,以上传 100MB、500MB 的文件为例,分别计算在本方案、perfectDedup 方案与不进行重复数据删除方案(NoDedup)中的存储开销,以此验证本方案在重复数据删除中的高效性.每部分操作重复进行 10 次,取平均值作为最终结果.

(1) 各阶段所需时间开销

由于广播加密仅在系统建立初期执行一次,因此,其所需时间开销不进行计算.非流行数据上传实验结果如图 7 所示.由于方案将大部分的计算外包给云服务器,用户端数据分块、标签生成与对称加密所需时间开销非常小.相对而言,发生在云服务器端的流行度识别与密文上传所需时间开销较大.流行数据上传实验结果如图 8 所示,各阶段所需时间开销与非流行数据上传大致相同.如图 9 所示,当 $Count_{FA}>T$ 时,只需要进行客户端重复数据删除,不再上传收敛加密密文,因此,显著减少了计算开销并节约了网络带宽.

(2) 更少的总时间开销

本文的方案与 perfectDedup 方案的对比结果如图 10~图 12 所示.在数据的流行度查询阶段(包括标签生成

与流行度识别),本方案所需时间开销明显低于 perfectDedup.此外,本方案摆脱了实时在线第三方 IS.因此,本方案在总时间开销上具有较明显的优势.

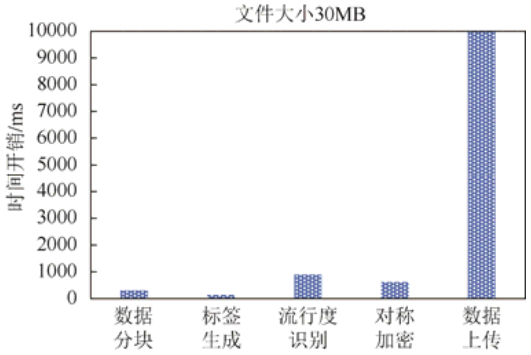

Fig.7 The time span for each phase of experiment ($Count_{FA}<T$)

图 7 实验中各阶段所需时间开销($Count_{FA}<T$)

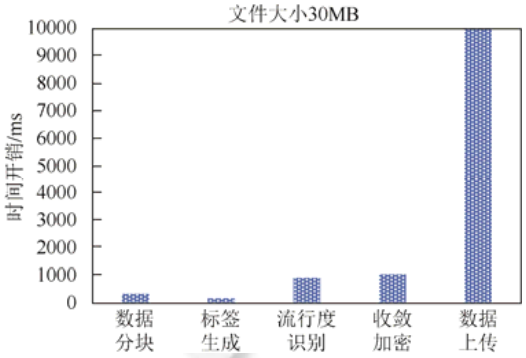

Fig.8 The time span for each phase of experiment ($Count_{FA}=T$)

图 8 实验中各阶段所需时间开销($Count_{FA}=T$)

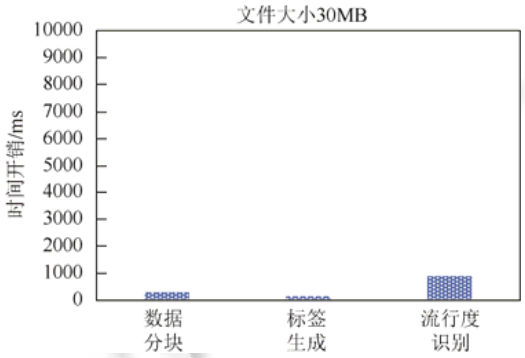

Fig.9 The time span for each phase of experiment ($Count_{FA}>T$)

图 9 实验中各阶段所需时间开销($Count_{FA}>T$)

Fig.10 Comparison of the time span between our scheme and the perfectDedup scheme ($Count_{FB}<T$)

图 10 本文方案与 perfectDedup 方案时间开销对比($Count_{FB}<T$)

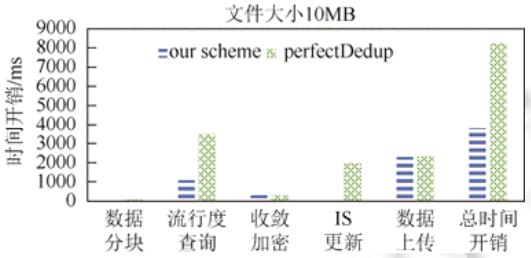

Fig.11 Comparison of the time span between our scheme and the perfectDedup scheme ($Count_{FB}=T$)

图 11 本文方案与 perfectDedup 方案时间开销对比($Count_{FB}=T$)

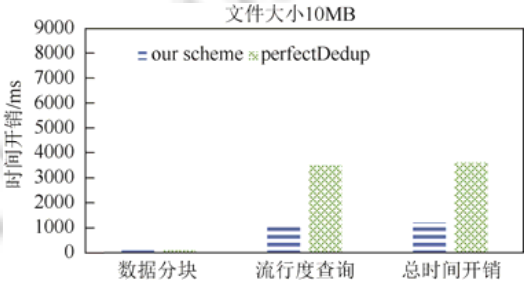

Fig.12 Comparison of the time span between our scheme and the perfectDedup scheme ($Count_{FB}>T$)

图 12 本文方案与 perfectDedup 方案时间开销对比($Count_{FB}>T$)

(3) 更少的存储空间开销

如图 13、图 14 所示,NoDedup 方案不执行重复数据删除,perfectDedup 方案无法删除非流行加密数据.本文方案的存储空间开销与持有数据的用户数量无关.且文件越大,本文方案的优势越明显.

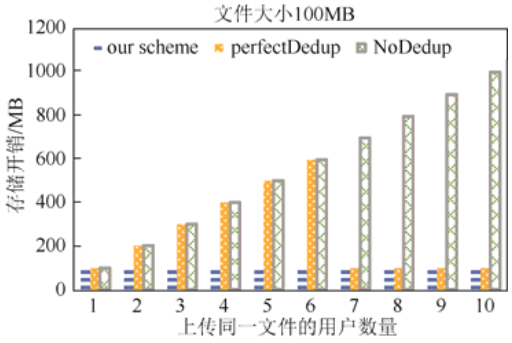

Fig.13 Cloud server storage costs of different schemes (Data size 100M)

图 13 3 种方案中云服务器存储开销对比(每个文件 100M)

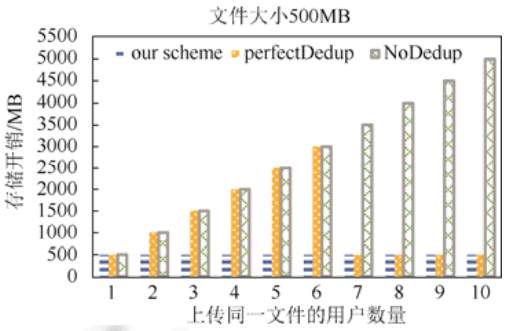

Fig.14 Cloud server storage costs of different schemes (Data size 500M)

图 14 3 种方案中云服务器存储开销对比(每个文件 500M)

(4) 性能分析比较

由以上实验结果可知,划分数据流行度、摆脱实时在线可信第三方能够显著提升重复数据删除方案的执行效率.表 1 给出了本文方案与其他代表性方案是否具有以上两个优点的分析和比较.

Table 1 Comparison of schemes characteristics

表 1 方案特点对比

方案	[7]	[8]	[9]	[15]	[17]	Our
划分数据流行度	×	√	√	×	×	√
摆脱实时在线可信第三方	×	×	×	×	√	√

7 总结与展望

本文研究了云存储环境下加密数据的重复删除问题,提出了一种基于离线密钥分发的加密数据重复删除方案.此方案通过构造语义安全的双线性映射,能够实现在不泄露数据任何明文信息的情况下完成流行度查询.通过广播加密为授权用户生成辅助密钥,保证非流行数据加密密钥的存储与传递的安全.持有相同非流行数据的不同用户能够获取相同的加密密钥,得到相同的加密数据,进而使云服务器能够对非流行数据进行重复数据删除.采用改进后的收敛加密算法保护隐私度较低的流行数据,用户能够自行生成加密密钥,进一步提高了方案的执行效率.通过安全分析与仿真实验,证明本方案具有较高的安全性与实用性.

如何摆脱广播中心,实现只有用户与云服务器两方交互的重复数据删除方案,是下一步的研究重点.

References:

[1] Fu YX, Luo SM, Shu JW. Survey of secure cloud storage system and key technologies. Journal of Computer Research and Development, 2013,50(1):136−145 (in Chinese with English abstract).

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

[3] Ao L, Shu JW, Li MQ. Data deduplication techniques. Ruan Jian Xue Bao/Journal of Software, 2010,21(5):916−929 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3761.htm [doi: 10. 3724/SP.J.1001.2010.03761]

[4] Jeramiah B. Opendedup: Open-Source deduplication put to the test. Belltown Media, 2013. http://opendedup.org/

[5] Meyer DT, Bolosky WJ. A study of practical deduplication. ACM Trans. on Storage (TOS), 2012,7(4):14.

[6] Douceur JR, Adya A, Bolosky WJ, *et al*. Reclaiming space from duplicate files in aserverless distributed file system. In: Proc. of the ICDCS. IEEE, 2002. 617–624.

[7] Puzio P, Molva R, Onen M. Cloudedup: Secure deduplication with encrypted data for cloud storage. In: Proc. of the CloudCom. IEEE Computer Society, 2013. 363–370.

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

[9] Stanek J, Sorniotti A, Androulak E, *et al*. A secure data deduplication scheme for cloud storage. In: Christin N, Safavi-Naini R, eds. LNCS 8437. Springer-Verlag, 2014. 99–118.

[10] Xu J, Chang E C, Zhou J. Weak leakage-resilient client-side deduplication of encrypted data in cloud storage. In: Proc. of the ACM SIGSAC Symp. on Information, Computer and Communications Security. ACM, 2013. 195–206.

[11] Adya A, Bolosky WJ, Castro M, *et al*. Farsite: Federated, available, and reliable storage for an incompletely trusted environment. ACM SIGOPS Operating Systems Review, 2002,36(SI):1–14.

[12] Hur J, Koo D, Shin Y, *et al*. Secure data deduplication with dynamic ownership management in cloud storage. IEEE Trans. on Knowledge and Data Engineering, 2016,28(11):1.

[13] Perttula. Attacks on convergent encryption. 2008. https://tahoe-lafs.org/hacktahoelafs/drew_perttula.html

[14] Bellare M, Keelveedhi S, Ristenpart T. Message-Locked encryption and secure deduplication. In: Proc. of the EUROCRYPT. LNCS 7881, Springer-Verlag, 2013. 296–312.

[15] Mihir B, Keelveedhi S, Ristenpart T. DupLESS: Server-Aided encryption for deduplicated storage. In: Proc. of the 22nd USENIX Conf. on Security. USENIX Association, 2013. 179–194.

[16] Douceur JR. The Sybil attack. In: Proc. of the Peer-to-Peer Systems. Springer-Verlag, 2002. 251–260.

[17] Liu J, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional servers. Technical Report, 455, ePrint archive, 2015. https://eprint.iacr.org/2015/455

[18] Li L, Xue R, Zhang HG, Feng DG, Wang L. Security analysis of authenticated key exchange protocol based on password. ACTA ELECTRONICA SINICA, 2005,33(1):166–170 (in Chinese with English abstract).

[19] Hu XX, Zhang ZF, Liu WF. Universal composable password authenticated key exchange protocol in the standard model. Ruan Jian Xue Bao/Journal of Software, 2011,22(11):2820–2832 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3910.htm [doi: 10.3724/SP.J.1001.2011.03910]

[20] Cui H, Deng RH, Li Y. Attribute-Based storage supporting secure deduplication of encrypted data in cloud. IEEE Trans. on Big Data, 2016, 1–13.

[21] Zhang XS. The construction and calculation of bilinear pairs in cryptography [Ph.D. Thesis]. Beijing: The Chinese Academy of Sciences, 2012 (in Chinese with English abstract).

[22] Chen YM, Cheng XG, Wang S. Pairing certificateless signature scheme based on information network security. Netinfo Security, 2017,(3):53–58 (in Chinese with English abstract).

[23] Sakai R, Furukawa J. Identity-Based broadcast encryption. Journal of Electronics & Information Technology, 2007,33(4): 1047–1050.

[24] Delerablée C. Identity-Based broadcast encryption with constant size ciphertexts and private keys. In: Proc. of the Advances in Crypotology, Int'l Conf. on Theory and Application of Cryptology and Information Security. Springer-Verlag, 2007. 200–215.

[25] Tan ZW, Liu ZJ, Xiao HG. A fully public key tracing and revocation scheme provably secure against adaptive adversary. Ruan Jian Xue Bao/Journal of Software, 2005,16(7):1333–1343 (in Chinese with English abstract). http://www.jos.org.cn/jos/ch/reader/create_pdf.aspx?file_no=20050716&journal_id=jos [doi: 10.1360/jos161333]

[26] Pang LJ, Li HX, Jiao LC. Design and analysis of a provable secure multi-recipient public key encryption scheme. Ruan Jian Xue Bao/Journal of Software, 2009,20(10):2907–2914 (in Chinese with English abstract). http://www.jos.org.cn/1000-9825/3552.htm [doi: 10.3724/SP.J.1001.2009.03552]

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

[28] Loukides M, Oram A. Programming with GNU SoftWare. O'Reilly & Associates, 1997,86(3):350–359.

[29] Steiner M. The PBC_bce broadcast encryption library. 2006. https://crypto.stanford.edu/pbc/bce/

[30] Hu XT, Qin ZP, Zhang H, Hao GS. Research and improved implementation of AES algorithm in OpenSSL. Control & Automation, 2009,25(12):83–85.

附中文参考文献:

[1] 傅颖勋,罗圣美,舒继武.安全云存储系统与关键技术综述.计算机研究与发展,2013,50(1):136–145.

[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.

[3] 敖莉,舒继武,李明强.重复数据删除技术.软件学报,2010,21(5):916–929. http://www.jos.org.cn/1000-9825/3761.htm [doi: 10.3724/SP.J.1001.2010.03761]

[18] 李莉,薛锐,张焕国,冯登国,王丽娜.基于口令认证的密钥交换协议的安全性分析.电子学报,2005,33(1):166–170.

[19] 胡学先,张振峰,刘文芬.标准模型下通用可组合的口令认证密钥交换协议.软件学报,2011,22(11):2820–2832. http://www.jos.org.cn/1000-9825/3910.htm [doi: 10.3724/SP.J.1001.2011.03910]

[21] 张旭升,林东岱.密码学中双线性对的构造与计算[博士学位论文].北京:中国科学院大学,2012.

[22] 陈亚萌,程相国,王硕.基于双线性对的无证书群签名方案研究.信息网络安全,2017,(3):53–58.

[25] 谭作文,刘卓军,肖红光.一个安全公钥广播加密方案.软件学报,2005,16(7):1333–1343. http://www.jos.org.cn/jos/ch/reader/create_pdf.aspx?file_no=20050716&journal_id=jos [doi: 10.1360/jos161333]

[26] 庞辽军,李慧贤,焦李成,王育民.可证明安全的多接收者公钥加密方案设计与分析.软件学报,2009,20(10):2907–2914. http://www.jos.org.cn/1000-9825/3552.htm [doi: 10.3724/SP.J.1001.2009.03552]

张曙光(1991－),男,山东曲阜人,硕士,主要研究领域为密码学,云计算安全.

刘红燕(1994－),女,硕士,主要研究领域为云中重复数据删除.

咸鹤群(1979－),男,博士,副教授,CCF 高级会员,主要研究领域为密码学,云计算安全,系统安全.

侯瑞涛(1993－),男,学士,主要研究领域为数据库数字水印.

王雅哲(1979－),男,博士,副研究员,主要研究领域为物联网安全,智能信息设备安全.