https://crad.ict.ac.cn
63(10): 2481−2496, 2026

云边存储场景下面向动态异构网络的跨用户数据加密去重

王荣熙[1,2,3] 哈冠雄[1,2,3] 贾春福[1,2,3] 陈柯言[1,2,3]

[1](南开大学密码与网络空间安全学院 天津 300350)

[2](天津市网络与数据安全技术重点实验室(南开大学) 天津 300350)

[3](数据与智能系统安全教育部重点实验室(南开大学) 天津 300350)

(wrx@mail.nankai.edu.cn)

Cross-User Encrypted Data Deduplication for Dynamic Heterogeneous Networks in Cloud-Edge Storage Scenario

Wang Rongxi[1,2,3], Ha Guanxiong[1,2,3], Jia Chunfu[1,2,3], and Chen Keyan[1,2,3]

[1](*College of Cryptology and Cyber Science*, *Nankai University*, *Tianjin* 300350)

[2](*Tianjin Key Laboratory of Network and Data Security Technology* (*Nankai University*), *Tianjin* 300350)

[3](*Key Laboratory of Data and Intelligent System Security* (*Nankai University*), *Ministry of Education*, *Tianjin* 300350)

Abstract Dynamic heterogeneous networks have become the mainstream paradigm for next-generation networks. The explosive integration of massive heterogeneous smart terminals imposes stringent requirements on encrypted data storage and privacy preservation within the cloud-edge collaborative architecture. Encrypted deduplication technology can identify and eliminate duplicate encrypted data, effectively reducing storage costs for cloud-edge environments while preserving data privacy, thus holding high practical value. However, existing encrypted deduplication schemes suffer from limitations such as vulnerability to duplicate faking attacks, high computational overhead, and heavy storage burdens on edge nodes, which fail to meet practical application demands. To address these issues, we propose SEDedup, a cross-user encrypted data deduplication scheme designed for cloud-edge storage scenarios. First, a server-aided encryption method suitable for cloud-edge storage is designed to achieve efficient and secure cross-user encrypted deduplication while effectively defending against duplicate faking attacks. Second, leveraging the temporal locality of data from heterogeneous terminals, an epoch-based duplicate data detection method is designed to achieve high-performance duplicate detection while minimizing the storage overhead on edge nodes. Finally, by incorporating generalized deduplication technology, lossless encrypted deduplication for similar heterogeneous terminal data is realized to maximize storage efficiency. Security analysis and performance evaluation demonstrate that SEDedup outperforms existing schemes in terms of performance while effectively preserving data privacy.

Key words privacy preservation; dynamic heterogeneous networks; cloud-edge storage; encrypted deduplication; server-aided encryption

摘 要 动态异构网络已成为新一代网络主流形态,海量异质智能终端的爆发式接入,对云边协同架构下的数据加密存储与数据隐私保护提出了严苛要求。加密去重技术可识别并删除重复的加密数据内容,能

收稿日期: 2026-06-06; **修回日期**: 2026-07-15

基金项目: 京津冀自然科学基金合作专项(25JJJC0005); 天津市科技重大专项(25ZXSFSN00070); 中央高校基本科研业务费专项资金资助(63261082)

This work was supported by the Beijing-Tianjin-Hebei Natural Science Cooperation Project (25JJJC0005), the Tianjin Science and Technology Major Special Project (25ZXSFSN00070), and the Fundamental Research Funds for the Central Universities (63261082).

通信作者: 贾春福(cfjia@nankai.edu.cn)

够在保护数据隐私的同时有效为云边存储节省存储成本,具有很高的实用价值。但现有加密去重方案存在易受到副本伪造攻击、计算开销高、边缘节点存储负担重等局限,无法满足实际应用需求。为此,提出一种云边存储场景下的跨用户加密数据去重方案SEDedup。首先,设计一种适用于云边存储的服务器辅助加密方法,实现高效安全的跨用户数据加密去重,同时可有效防御副本伪造攻击;其次,利用异质终端数据的时间局部性,设计一种基于时段的重复数据检测方法,在降低边缘节点存储成本的同时实现高性能重复检测;最后,融合广义去重技术,实现相似异质终端数据的无损加密去重,最大化存储效率。安全分析与性能评估表明,SEDedup在有效保护数据隐私的同时,性能方面优于现有方案。

关键词 隐私保护;动态异构网络;云边存储;加密去重;服务器辅助加密

中图法分类号 TP311.13; TP309

DOI: 10.7544/issn1000-1239.202660423 **CSTR:** 32373.14.issn1000-1239.202660423

近年来,动态异构网络成为物联网、云边端协同计算的核心组网形态,海量异质智能终端,如智能工控终端、智联汽车、个人智能物联网(Internet of thing, IoT)设备等,持续接入网络,产生了体量庞大、类型异构的终端数据。云边协同存储作为动态异构网络中异质终端数据的核心承载架构,凭借低延迟、位置感知的优势[1-4],成为海量异质终端数据管理的关键技术支撑。

在实际应用中,异质终端采集的数据存在大量冗余[4]。例如,在智能交通领域,车辆可能重复上传相同街道的相同或相似的图像或视频。在电子医疗系统中,同一科室就诊患者会生成大量的重复电子病历记录[5]。大量的重复数据会导致系统的存储空间被严重浪费。因此,可应用去重技术来识别并删除重复数据,仅保留重复数据的一个物理副本以提升系统的存储效率[6-9]。另一方面,数据隐私保护是动态异构网络下异质终端数据安全防护的核心需求,也是国家网络安全战略的重要组成部分。

为兼顾数据高性能存储与隐私保护,国内外学者提出了一系列加密去重方案[10-12],通过对加密数据执行去重,在保护数据隐私的同时节省存储空间。Bellare等人[10]提出消息锁加密(message-locked encryption, MLE),基于数据内容生成加密密钥,使得相同明文可被加密为相同密文,实现加密数据去重。然而,MLE无法抵御蛮力攻击(brute-force attack, BFA)[13-14]。为此,服务器辅助加密[13-16]被提出,其通过引入第三方密钥服务器来抵御BFA。随后,Liu等人[11]基于口令认证密钥交换(password authenticated key exchange, PAKE)协议实现了无需第三方服务器的加密去重。

然而,现有大多数加密去重方案[17-23]仅适用于由云服务器和客户端组成的2层云存储架构,难以应用于由云服务器、边缘节点和异质终端组成的3层的云边存储架构,无法满足海量异质智能终端构成的动态异构网络的业务需求。相较于云存储场景下的加密去重,云边协同存储需要考虑跨域去重[24-26]和边缘节点存储资源受限等问题[27]。近年来,一些云边存储中的加密去重方案被提出[2-4,28-32]。Fu等人[28]利用云边存储实现了医疗记录的加密去重。Xie等人[31]提出云边存储场景中可提供多级别安全保护的加密去重方案。然而,这些方案[28,31]均要求IoT设备持续地与远端的密钥服务器交互,导致系统通信代价过高。Zhang等人[3]提出云边存储中抗BFA且支持跨域数据去重的加密去重方案。但该方案不支持跨用户数据去重,用户间的大量重复数据会显著增加系统存储负担[2]。Gao等人[32]提出云边存储中基于物联网设备间关系的加密去重方案。然而,该方案无法抵BFA、侧信道攻击(side-channel attack, SCA)[33-34]和副本伪造攻击(duplicate faking attacks, DFA)[10]。Song等人[2]提出的FCDedup实现了云边存储中的加密去重,该方案无需终端设备与任何远端的第三方服务器交互,并且实现了跨用户、跨域数据的加密去重。然而,该方案仍存在2点局限:

1)安全。FCDedup无法抵抗SCA和DFA。终端设备可识别云服务器中是否存在特定数据,并且可通过上传不一致的标签和密文来破坏系统的数据完整性。

2)性能。FCDedup使用了大量计算昂贵的幂运算和配对操作,导致计算成本较高。此外,在FCDedup中,边缘节点无法执行跨用户去重,跨用户的重复数据需要由边缘节点传输至云服务器,导致系统产生了巨大的通信开销。

此外,现有方案均未考虑动态异构网络中边缘节点存储资源受限这一问题。用于重复检测的大量

数据标签会为节点带来很大的存储负担。尽管 Shin 等人[27] 基于 PAKE 协议降低了加密去重中节点的存储负担，但频繁执行的 PAKE 协议导致系统计算效率极低。

本文提出面向动态异构网络的云边存储架构下的跨用户加密数据去重方案 SEDedup。该方案设计了一种适用于云边存储场景的服务器辅助加密，使边缘节点可执行跨用户数据去重，避免了重复数据由边缘节点到云服务器的传输，提升了系统通信效率，同时可有效防御 BFA、SCA 和 DFA 等安全威胁。相较于现有的服务器辅助加密技术[13,35-37]，SEDedup 做出 2 点改进。1）密钥服务器仅需参与系统的初始化阶段，初始化完成后即可离线。数据的加密上传过程无需密钥服务器参与，避免了终端设备与密钥服务器间的频繁交互。2）终端设备无需与密钥服务器直接进行交互。设备仅需通过边缘节点与密钥服务器交互，减少了设备与远端实体间直接通信所带来的通信开销。

另外，本文通过分析多个真实世界数据集发现，终端数据具有显著的空间局部性和时间局部性，即同一区域内数据重复率高，同一时段内数据重复率高。空间局部性表明，节点执行跨用户去重可显著降低系统通信成本。此外，基于终端数据的时间局部性，本文提出一种基于时段的重复检测方法，以降低边缘节点的存储开销，同时实现高性能的数据重复检测。在该方法中，边缘节点仅需维护 1 个记录当前时段数据标签的临时索引和 1 个记录所有时段标签的布谷鸟过滤器（Cuckoo filter, CF）。利用数据的时间局部性，节点可利用临时索引检测到大部分重复数据，剩余重复数据也可基于CF进行检测。因此，节点可在仅存储少量数据的情况下实现高效重复检测。此外，通过结合广义去重技术，SEDedup 支持对相似数据的无损加密去重，显著提升系统存储效率。

综上，本文的主要贡献如下：

1）提出面向异构网络的云边存储架构下的跨用户加密去重方案 SEDedup，其通过设计一种适用于云边存储场景的新型服务器辅助加密方法，实现了异质终端数据的安全去重。

2）基于终端数据的时间局部性，提出基于时段的重复检测方法，通过维护临时索引和布谷鸟过滤器，边缘节点仅需存储少量数据即可实现高效重复检测，降低了节点的存储负担。通过结合广义去重技术，SEDedup 可实现对相似数据的无损加密去重，提升系统存储效率。

3）安全分析与性能评估表明，与现有最先进的加密去重方案相比，SEDedup 可提供更强的安全保护，且性能更优。

1 背景与动机

本节介绍传统云存储中的加密去重方案以及现有云边存储中的加密去重方案。现有方案的局限性是本文工作的主要动机。

1.1 云存储中的加密去重

去重技术通过识别并删除重复数据以节省存储开销。加密去重将加密算法与去重技术相结合，对加密数据进行去重，以实现高效存储与隐私保护。MLE[10] 通过基于数据内容生成密钥以实现高效的加密去重，但该方法无法防御 BFA[13]。Keelveedhi 等人[13] 提出服务器辅助加密以防御 BFA，该加密方法中的密钥基于数据内容和密钥服务器管理的主密钥共同生成。只要主密钥安全，该方法就可以防御离线 BFA。此外，服务器辅助加密通过对密钥生成请求实施速率限制，可有效缓解在线 BFA。随后，学术界提出了多种基于服务器辅助加密的加密去重方案[24,35,38-39]，但这些方案仅适用于传统的云存储架构，难以应用于云边存储。

1.2 云边存储中的加密去重

近年来，一些云边存储中的加密去重方案被提出。Xie 等人[31] 提出了云边存储中可防御频率分析攻击的安全去重方案。Fu 等人[28] 基于雾-多云存储架构实现医疗数据的安全去重。但这些方案[28,31] 需要设备与远端的密钥服务器持续交互以生成密钥，通信成本较高。Shin 等人[24] 和 Yang 等人[26] 实现了跨域数据的加密去重。但上述方案[24,26] 均要求同一域内的所有数据拥有者共享相同的参数。如文献 [2] 所述，这一做法在实际应用中难以实现。Zhang 等人[3] 同时实现了跨域去重和相似数据去重，但其方案不支持跨用户数据去重。Gao 等人[32] 基于终端设备间的联系提出了云边存储场景下的加密去重方案。但该方案无法防御 BFA、SCA 和 DFA。Song 等人[2] 提出 FCDedup，该方案支持跨用户数据去重，且性能优于文献 [3]，但仍存在以下局限。

1）安全。FCDedup 无法抵御 SCA 和 DFA。FCDedup 对重复数据和非重复数据采用不一致的处理流程，敌手可通过观察方案的处理差异来推断特定数据是否存在于云服务器上。此外，若首次上传

数据的终端设备向云服务器上传不一致的数据标签与密文以发起 DFA, 云服务器会依据这些异常数据标签执行去重操作, 并删除后续上传的数据。最终云服务器仅留存该异常密文, 致使后续上传方无法恢复原始数据, 破坏数据完整性。

2)性能。FCDedup 使用了大量的双线性配对和幂运算, 造成其计算开销较高。并且, FCDedup 需存储大量数据标签, 给边缘节点带来了极大的存储负担。同时, FCDedup 忽略了边缘节点存在的跨用户数据冗余问题, 仅在云服务器端执行跨用户数据去重。该设计会产生较高的通信开销: 同一区域内的大量跨用户重复数据仍需由边缘节点上传至云服务器。本文将在 1.3 节对此展开验证分析。

如表 1 所示, 与现有加密去重方案相比, SEDedup 首次同时实现了云边存储中的相似数据去重、跨域去重、SCA 和 DFA 的防御, 以及边缘节点高效存储。

Table 1 Comparative Analysis of SEDedup and Existing Encrypted Deduplication Schemes

表 1 SEDedup 与现有加密去重方案的对比分析

方案	相似数据去重	跨域去重	防御 BFA、SCA 和 DFA	节点高效存储
TSC'23[31]	×	×	√	×
TSC'20[24]	×	√	×	×
TDSC'21[28]	×	√	√	×
TPDS'22[3]	√	√	×	×
TPDS'23[2]	×	√	×	×
TPDS'25[32]	√	√	×	×
SEDedup	√	√	√	√

1.3 终端数据的局部性

本文基于多个真实世界数据集分析终端收集的 IoT 数据的时间局部性和空间局部性。数据集情况具体如下:

1)CMU-MMAC[40]。该数据集由卡内基梅隆大学的动作捕捉实验室收集, 包含了受试者执行烹饪和食物准备相关任务时的人类活动多模态测量数据。

2)Thermal solar plant dataset(TSPD)[41]。该数据集提供从 2016 年 12 月 28 日至 2018 年 10 月 10 日记录的实时太阳能热发电厂数据。

3) ECA&D[42]。该数据集包含来自欧洲及地中海地区多个气象站的每日观测数据。

4) ASHRAE[43]。该数据集包含过去 20 年间全球热舒适度实地研究的原始数据。

本文采用广义去重技术[3,44](细节见 2.3 节)将数据集中的数据分为基和偏差。相似但不同的数据可以产生相同的基以及不同的偏差。由于基中存在更多重复, 而偏差中重复较少, 本文仅对基进行去重。本文计算压缩率 $cr = 1 - D_{v_a}/D_{v_b}$, 其中 D_{v_b} 和 D_{v_a} 分别表示去重前后的数据量。压缩率越高, 表明去重带来的存储效益越高。

1.3.1 时间局部性

CMU-MMAC 和 TSPD 数据集中的数据携带时间信息。因此, 本文基于这 2 个数据集评估时间局部性。为 CMU-MMAC 和 TSPD 设定的时段长度分别为 5 min 和 1 d, 随后对同一时段内的数据进行去重并计算压缩率。如图 1 所示, CMU-MMAC 连续 10 个时段的压缩率超过 80%, 而 TSPD 连续 10 个时段的压缩率超过 90%。由此可见, 终端数据呈现出强时间局部性, 也就是说, 同一时段内存在大量的重复数据内容。基于数据的时间局部性, 可设计基于时段的重复检测方法, 以节省边缘节点的存储开销。

1.3.2 空间局部性

ECA&D 和 ASHRAE 数据集中的数据携带空间信息。因此, 本文基于这 2 个数据集评估空间局部性。本文分别从 ECA&D 和 ASHRAE 数据集中选取 10 个区域的数据进行评估, 对每个区域内的数据进行去重并计算压缩率。如图 1 所示, ECA&D 和 ASHRAE 中 10 个区域的平均压缩率分别为 80% 和 75%。由此可见, 终端数据具有强空间局部性, 即同一区域内存在大量重复的数据内容。基于空间局部性, 可通过在边缘节点对同一区域内的数据执行跨用户去重以减少系统通信成本。

2 预备知识

2.1 CF

CF[45] 是一种空间高效的概率型数据结构, 用于检查某个数据项是否属于一个数据集合。CF可能出现假阳性匹配, 但不会出现假阴性。CF支持数据项插入、常数时间成员查询和数据项删除。

1) $CF.Insert(x)$。以数据项 x 为输入, 输出插入 x 后的CF。

2) $CF.Check(x)$。以数据项 x 为输入, 输出标识符 $f \in \{0,1\}$, 表示 x 是否存在于CF中。若 $f=1$, 则表明CF可能包含 x; 如果 $f=0$, 则表明CF不包含 x。

3) $CF.Delete(x)$。以数据项 x 为输入, 输出删除 x 后的CF。

2.2 数字签名

数字签名可用于验证数字消息的真实性和完整

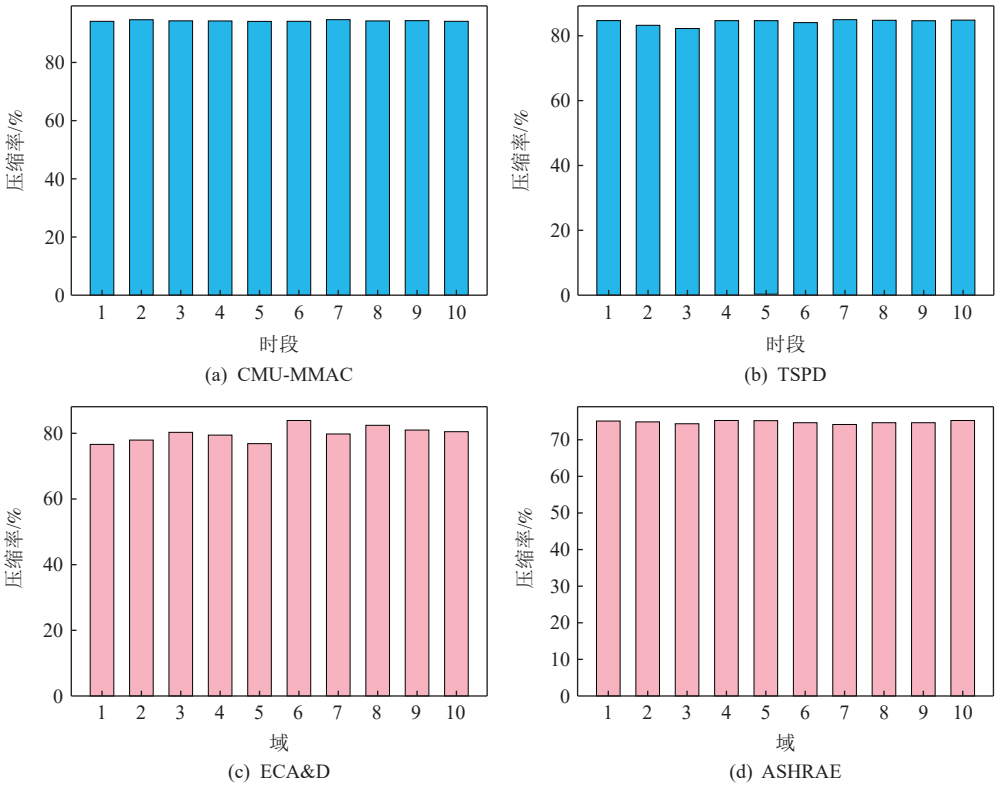


Fig. 1 Compression ratios for real-world IoT datasets

图 1 真实世界物联网数据集的压缩率

性，其包含以下 3 个概率多项式时间算法。

1）$Gen_S(\lambda)$：密钥生成算法以安全参数 λ 为输入，输出签名密钥对 (pk_S, sk_S)。

2）$Sign_S(sk_S, M)$：签名算法以私钥 sk_S 和消息 $M \in \{0,1\}^*$ 为输入，输出签名 σ。

3）$Vrfy_S(pk_S, M, \sigma)$：验证算法以公钥 pk_S、消息 M 和签名 σ 为输入，输出标识符 $f \in \{0,1\}$。若 $f=1$，则 σ 有效；否则，σ 无效。

2.3 广义去重

广义去重[3,44]是一种适用于相似数据的无损压缩技术。它使用转换函数 φ 将数据 M 拆分为基 b 以及偏差 d。φ 可以将相似但不相同的数据转换为相同的基和不同的偏差。本文采用文献 [45] 中的方法实现广义去重。假设消息空间为 F_{2^n}（即长度为 n 的比特向量集合）。φ 将数据 $M \in F_{2^n}$ 转换为 $\varphi(M) \triangleq (b,d) = \left(M_0^{k-1}, M_k^{n-1}\right)$。将 M 的前 k 个比特作为基 b，剩余 $n-k$ 个比特作为偏差 d。逆函数 φ^{-1} 拼接 b 和 d 来恢复 M：$M \leftarrow \varphi^{-1}(b,d) \triangleq \left[b_0, b_1, \cdots, b_{l_b-1}, d_0, d_1, \cdots, d_{l_d-1}\right]$，其中 l_b 和 l_d 分别表示 b 和 d 的长度。为以高概率将相似数据映射到相同基以提升存储效率，需构建置换函数 π，将置换集合 $I=\{i_1, i_2, \cdots, i_{n-k}\}$ 中的索引重新定位到 M 的末端。例如，利用 π 将 M 转换为 $\bar{M} \leftarrow \pi_I(M) \triangleq [\cdots, M_{i_1-1}, M_{i_1+1}, \cdots, M_{i_2-1}, M_{i_2+1}, \cdots, M_{i_{n-k}-1}, M_{i_{n-k}+1}]$。$M$ 的 b 和 d 分别为 $\bar{M}_0^{k-1}$ 和 $\bar{M}_k^{n-1}$。恢复 M 时，首先拼接 b 和 d 以重构 $\bar{M}$，然后根据 I 将 $\bar{M}$ 末端的 $n-k$ 个元素移回到原始位置以恢复 M。置换集合 I 对广义去重的存储效率至关重要。该集合中的索引选择由数据集特性决定[44]，通常通过预抽样训练，首先评估数据比特位之间的相关性，然后选择变异率（某个特定比特位在大量数据样本中发生状态翻转的频繁程度）最低的比特位索引。此外，基长度的设定可依靠启发式方法，依据设定的启发式条件计算满足性能要求的最小基长度。本文采用文献 [44] 中的方法确定基长度等参数。

3 方案概述

本节介绍 SEDedup 的系统架构和威胁模型。

3.1 系统架构

图 2 给出了 SEDedup 的系统架构，其中包括数据拥有者（D_O）、异质终端设备、边缘节点（edge node, EN）、云服务提供商（cloud service provider, CSP）和密钥服务器（key server, KS）。

1）D_O。指部署多个异质终端设备以采集数据（如气象、医疗或交通信息）的个人或机构，其利用

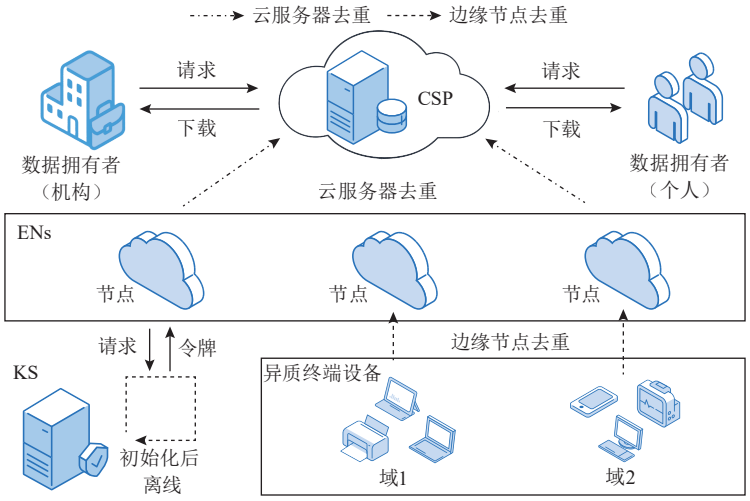


Fig. 2 System architecture

图 2 系统架构

CSP来管理收集到的 IoT 数据。

2)异质终端设备。其由 D_O 部署，负责采集 IoT 数据，通过边缘节点将加密的 IoT 数据上传至CSP。

3)边缘节点。EN部署于网络边缘，充当异质终端设备与CSP之间的中间层。边缘节点接收其域内异质终端设备上传的数据，执行跨拥有者的数据去重(即跨用户去重)。节点仅将非重复的数据上传至CSP以降低通信开销。

4)云服务提供商。CSP为 D_O 提供数据存储服务，其接收来自不同EN上传的数据，执行跨节点的数据去重以降低存储开销。

5)密钥服务器。KS负责管理系统级别的主密钥。在初始化异质终端设备时，KS利用主密钥协助EN为设备生成令牌。该令牌用于实现安全的跨用户加密去重。初始化完成后，KS可转为离线状态。

3.2 威胁模型

与现有服务器辅助加密方案[13,35,38]相同，本文假设KS是半可信的，其会诚实地执行既定协议，安全地存储主密钥，且不会与其他实体合谋。如果主密钥被敌手攻破，SEDedup 的安全性会降至收敛安全[10]。本文考虑的主要安全威胁如下：

1)诚实但好奇的CSP与EN。CSP和EN会诚实地执行协议，但也会试图获取异质终端设备上传的数据的信息。它们可访问自身存储的数据内容，并试图发起 BFA 以获取明文数据信息。

2)恶意的异质终端设备。与现有方案[2,3]不同，本文考虑来自恶意异质终端设备的威胁。它们可能发起 SCA 以推断数据在CSP的存储状态。具体而言，恶意异质终端设备向EN上传数据标签，并根据EN返回的重复检测结果来判断该数据是否已经被其他 D_O 的设备上传过。此外，恶意异质终端设备可发起 DFA 以破坏后续上传者的数据完整性。具体而言，其向CSP上传不一致的标签和密文。

本文假定实体间的通信是安全的(可基于 TLS 协议实现)。此外，与现有方案[2-3]相同，本文假定CSP不会与 D_O 或EN合谋。如文献 [46] 中所述，此类合谋会严重损害CSP的声誉，并导致其承担民事责任或面临刑事调查[25]。然而，EN容易被敌手攻破，因此本文假设EN可能与 D_O 合谋。

4 方案设计

本节首先给出 SEDedup 的设计挑战及解决方案，随后描述方案的详细流程。本文使用的主要符号总结见表 2。

4.1 设计挑战与解决方案

方案的设计挑战 C 和解决方法如下。

4.1.1 如何在抵御 BFA、SCA 和 DFA 的同时实现高效跨用户数据的加密去重？

现有方案[13,35,39]主要基于数据拥有者之间的密钥传输[2,47]或服务器辅助加密[13]这 2 种方法来实现安全的跨用户数据加密去重。前者计算成本高，且需要数据拥有者保持在线，实用性较差。此外，由于异质终端设备与远端密钥服务器间频繁交互带来显著通信开销，现有服务器辅助加密也难以应用于云边存储。

为此，本文设计一种适用于云边存储的服务器辅助加密方案，在实现高效跨用户数据加密去重的

Table 2 Major Notations Used in Our Paper

表 2 本文使用的主要符号

符号	描述
msk	主密钥
pk_{KS}, sk_{KS}	KS的 RSA 密钥对
spk, ssk	D_O的签名密钥对
dsk_i/k_{D_O}	D_i/D_O的秘密密钥
Δ_i	D_i的令牌
β_{D_O}	D_O的秘密值
Enc_{SE}/Dec_{SE}	对称加密/解密算法
M_i	IoT 数据
b_i/d_i	M_i的基/偏差
T_{b_i}	基的标签
I_{TE}, I_{FU}	临时索引，完整索引
$k_{b_i}, C_{k_{b_i}}$	基的密钥，加密的基的密钥
C_{b_i}, C_{d_i}	密文基，密文偏差
h_{b_i}	基b_i的哈希值

同时防御 BFA、SCA 和 DFA。SEDedup 部署一个管理系统主密钥msk的KS。在初始化新异质终端设备D时，其所在区域的边缘节点EN与D、KS和CSP交互，为D生成令牌$\Delta \leftarrow \frac{msk \cdot \beta_{D_O}}{dsk}$，其中$dsk$是$D$的密钥，$\beta_{D_O}$是CSP为$D_O$生成的秘密值。在密钥生成阶段，$D$与EN和CSP交互。EN和CSP辅助$D$基于令牌$\Delta$生成加密密钥。加密密钥基于$msk$和数据内容生成，因此不同$D_O$部署的设备在加密相同明文数据时会生成相同的加密密钥和相同的密文。同时，服务器辅助加密使方案中的EN可以执行跨用户数据去重。如 1.3 节所述，真实世界数据呈现显著的空间局部性，因此同一域内存在大量重复数据。在EN处执行跨用户去重可显著降低EN与CSP间的通信开销，从而提升系统的通信效率。

此外，为防御 DFA，EN基于密文数据生成数据标签，从而确保密文与标签的一致性。为防御 SCA，SEDedup 实现两阶段去重[48]，也就是说，对于跨用户重复数据，SEDedup 执行基于目标的去重，需要终端设备上传完整的数据内容。对于D_O内的重复数据（即用户内重复数据），SEDedup 执行基于源的去重，设备无需上传重复数据。因此，敌手不能通过EN返回的重复响应来判断其他所有者的数据存在性信息。并且，由于引入了主密钥msk，EN和CSP均无法通过发起 BFA 来恢复数据信息。

SEDedup 中的服务器辅助加密方案仅在初始化新终端设备时需要KS参与交互，密钥生成阶段无需KS介入，避免了异质终端设备或节点与KS间的频繁交互。此外，SEDedup 不需要终端设备直接与KS通信。终端设备、KS和CSP在EN的协助下共同生成令牌，有效消除了终端设备与远端实体间的复杂通信。

4.1.2 如何在实现高效重复检测的同时降低EN的存储开销？

受真实世界数据时间局部性（1.3 节）的启发，本文提出一种基于时段的重复检测方法。SEDedup 采用基于时段的设计方法。在时段e_i中，EN仅存储包含当前时段数据标签的临时索引I_{TE}，以及存储先前所有时段标签的CF。CSP存储一个记录所有时段数据标签的完整索引I_{FU}和一个记录所有标签信息的CF。在接收到异质终端设备上传的数据标签T后，EN首先在I_{TE}中查找T。由于 IoT 数据具有时间局部性，同一时段内的重复数据量大，EN基于I_{TE}找到重复数据的概率很高。若未在I_{TE}中找到T，EN将检测CF中是否包含T。若包含，EN将T上传至CSP，CSP基于完整索引I_{FU}进行重复检测；若不包含，表明该数据并非重复数据，需要设备上传完整的数据内容。在数据上传后，EN和CSP分别更新I_{TE}和I_{FU}，便于后续的重复检测。在每个时段结束后，EN删除I_{TE}和CF。CSP将这一时段中新出现的标签插入到CF中，生成一个更新后的CF′，将CF′发送到EN。基于I_{TE}和CF，EN能够在仅存储少量数据的情况下实现高效率的重复检测。

4.1.3 如何去重高度相似但并不完全相同的 IoT 数据？

SEDedup 通过结合广义去重技术[3,44]来对相似的 IoT 数据进行加密去重。具体而言，终端设备D利用广义去重技术将 IoT 数据转换为基和偏差。相似数据会被拆分为相同的基和不同的偏差。为简化说明，本文假设所有 IoT 数据均为同一类型。在处理多种类型的 IoT 数据时，可以针对每种数据类型分别进行广义去重，从而获取基和偏差[3]。基包含数据的大部分信息且重复率高，偏差体量较小且重复率低。因此，SEDedup 对基执行服务器辅助加密，实现对基的跨用户加密去重，降低存储开销。同时，本文对偏差采用随机加密，且不对偏差进行去重。因此，本文实现了针对相似 IoT 数据的无损加密去重。

4.2 方案构造

SEDedup 的具体构造如下。

1）初始化。KS输入安全参数λ，生成一个生成元为g、阶为素数q的乘法循环群$\mathbb{G}$。KS生成主密钥

$msk \in \mathbb{Z}_q$和RSA密钥对(pk_{KS}, sk_{KS})。KS选择哈希函数: $H_1: \{0,1\}^* \to \{0,1\}^\lambda$和$H_2: \{0,1\}^* \to \mathbb{G}$。KS公开系统参数$pp \leftarrow \{q, \mathbb{G}, g, H_1, H_2\}$和公钥$pk_{KS}$。KS安全存储主密钥$msk$和RSA私钥$sk_{KS}$。在向CSP注册后，数据拥有者$D_O$部署异质终端设备$D_i$以收集IoT数据。随后CSP为$D_O$生成秘密值$\beta_{D_O} \in \mathbb{Z}_q$，并在CSP中安全存储。$D_O$生成签名密钥对$(spk, ssk) \leftarrow Gen_S(\lambda)$和对称密钥$k_{D_O} \leftarrow \{0,1\}^\lambda$。另外，$D_O$为其设备$D_i$生成一个密钥$dsk_i \in \mathbb{Z}_q$，将$(ssk, k_{D_O}, dsk_i)$输入至$D_i$。$D_O$安全管理$(k_{D_O}, ssk)$并公开$spk$。当初始化域内新设备$D_i$时，边缘节点EN与$D_i$、CSP和KS交互，为$D_i$生成令牌$\Delta_i$。令牌$\Delta_i$生成过程如图3所示。

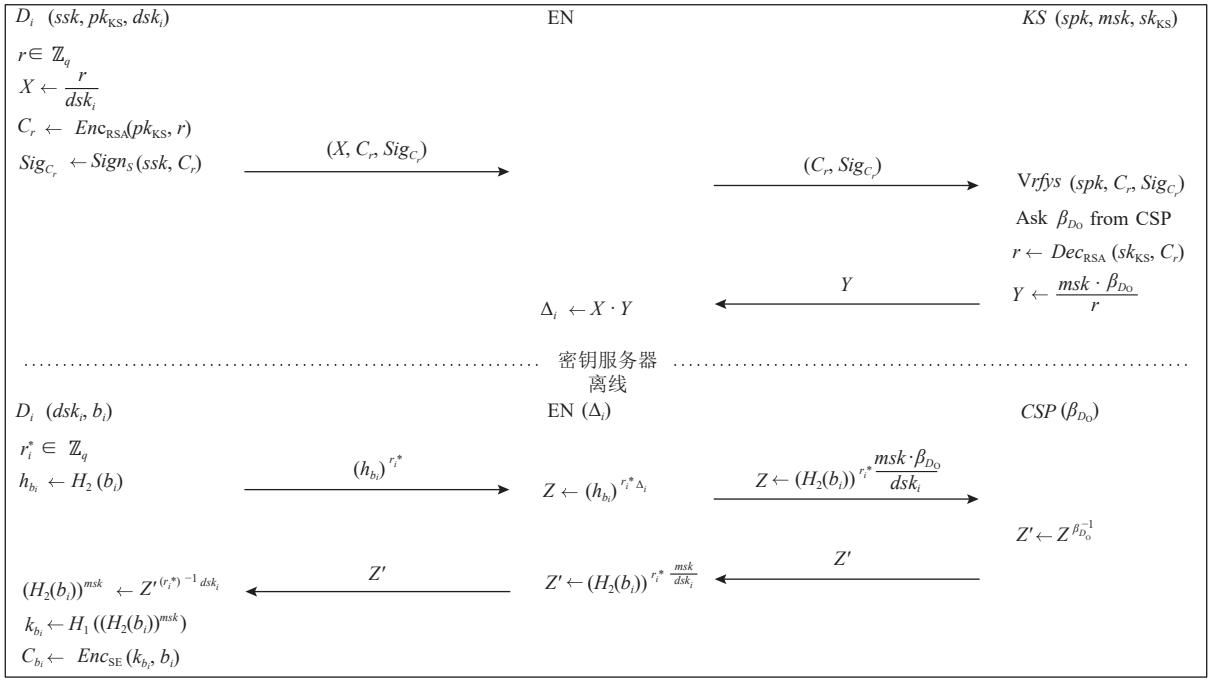


Fig. 3 Initialization, key generation, and data encryption

图3 初始化，密钥生成与数据加密

D_i生成随机值$r \in \mathbb{Z}_q$，计算$X \leftarrow \frac{r}{dsk_i}$，$C_r \leftarrow Enc_{RSA} \times (pk_{KS}, r)$和签名$Sig_{C_r} \leftarrow Sign_S(ssk, C_r)$。接着，$D_i$发送$(X, C_r, Sig_{C_r})$至节点EN。

EN转发(C_r, Sig_{C_r})到KS，KS计算$Vrfy_S(spk, C_r, Sig_{C_r})$。若$f=0$，则签名无效，KS终止交互。否则，KS通知CSP有新设备注册。CSP检查D_O是否注册，若未注册，则该请求无效；否则，CSP将秘密值β_{D_O}发送给KS。KS恢复$r \leftarrow Dec_{RSA}(sk_{KS}, C_r)$，返回$Y \leftarrow \frac{msk \cdot \beta_{D_O}}{r}$至EN。

EN计算D_i的令牌$\Delta_i \leftarrow X \cdot Y = \frac{msk \cdot \beta_{D_O}}{dsk_i}$。

在初始化阶段，终端设备D_i通过EN、CSP与KS交互来生成令牌Δ_i。D_i不需要直接与远端的KS交互，降低了系统的通信开销。此外，从安全角度出发，由于D_i为C_r生成了签名，EN无法使用伪造的C_r'来获得主密钥msk的信息。

2)密钥生成与数据加密。如图3所示，假设M_i是终端设备D_i收集的IoT数据。D_i首先利用广义去重将M_i转换为基b_i和偏差d_i。D_i分别使用服务器辅助加密和标准对称加密来对b_i和d_i进行加密。D_i使用密钥k_{D_O}对d_i执行对称加密得到密文偏差$C_{d_i} \leftarrow Enc_{SE}(k_{D_O}, d_i)$，然后计算基的哈希值$h_{b_i} \leftarrow H_2(b_i)$。接着，$D_i$和EN运行如下的密钥生成协议以得到基的密钥$k_{b_i}$。

D_i生成随机值$r^* \in \mathbb{Z}_q$，计算盲化后的哈希值$(h_{b_i})^{r^*}$，发送$(h_{b_i})^{r^*}$到EN。

EN计算$Z \leftarrow ((h_{b_i})^{r^*})^{\Delta_i} = (H_2(b_i))^{r^* \frac{msk \cdot \beta_{D_O}}{dsk_i}}$，返回$Z$到CSP。

CSP计算$Z' \leftarrow (Z)^{\beta_{D_O}^{-1}} = ((h_{b_i})^{r^*})^{\Delta_i \cdot \beta_{D_O}^{-1}} = (H_2(b_i))^{r^* \frac{msk}{dsk_i}}$，经由EN返回$Z'$到$D_i$。

D_i计算$(H_2(h_{b_i}))^{msk} \leftarrow Z^{(r_i^*)^{-1} dsk_i}$和基密钥$k_{b_i} \leftarrow H_1((H_2(b_i))^{msk})$。在得到$k_{b_i}$后，$D_i$计算密文基$C_{b_i} \leftarrow Enc_{SE}(k_{b_i}, b_i)$和加密的密钥$C_{k_{b_i}} \leftarrow Enc_{SE}(k_{D_O}, k_{b_i})$。由于基密钥$k_{b_i}$是基于数据内容和主密钥生成的，因此当不同数据拥有者上传相同的基时，他们能够生成相同的基密钥和密文基，系统可实现跨用户数据的加密去重。若恶意终端设备频繁向边缘节点发起密钥生成请求

以进行在线 BFA, 边缘节点可通过执行速率限制[13]进行防御。

3)重复检测与数据上传。D_i计算数据标签$T_{b_i} \leftarrow H_1(C_{b_i})$, 上传$(id_{D_O}, T_{b_i}, C_{d_i}, C_{k_{b_i}})$到EN, 其中$id_{D_O}$表示$D_O$的ID。EN基于$T_{b_i}$执行如下的重复检测过程。

EN首先在临时索引I_{TE}中检索T_{b_i}, 该索引记录了当前时段内所有数据拥有者的数据标签以及他们的ID。基于数据的时间局部性, 节点可基于I_{TE}检测到大部分的重复数据。若EN在I_{TE}中索引到$\langle T_{b_i}, id_{D_O}\rangle$, 表示基$b_i$已经被$D_O$上传, 该数据为$D_O$的用户内重复数据。这时EN将重复检测结果标记为$dr \leftarrow 1_I$。若$I_{TE}$中仅索引到$T_{b_i}$, 表明$b_i$已被其他$D_O$上传, 为跨用户重复数据。EN设定$dr \leftarrow 1_C$。若在$I_{TE}$中未索引到$T_{b_i}$, EN执行CF的元素查询算法得到$f \leftarrow CF.Check\times(T_{b_i})$。若$f=0$, 表明$b_i$是不重复的, EN设定$dr \leftarrow 0$; 否则, EN将$T_{b_i}$发送至CSP进行进一步的重复检测。

收到T_{b_i}后, CSP在完整索引I_{FU}中检索T_{b_i}。I_{FU}记录了此前所有时段的数据标签和标签对应的数据拥有者的ID。CSP的存储结构如图 4 所示。若在I_{FU}中索引到$\langle T_{b_i}, id_{D_O}\rangle$, CSP标记$dr \leftarrow 1_I$。若在$I_{FU}$中仅索引到$T_{b_i}$, CSP标记$dr \leftarrow 1_C$; 否则, 表明CF出现假阳性情况, CSP标记$dr \leftarrow 0$。CSP将$dr$返回至EN。

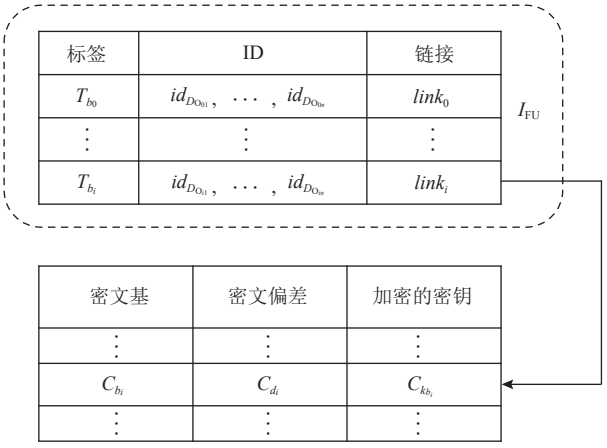


Fig. 4 Storage structure of CSP

图 4 CSP的存储结构

完成上述重复检测过程后, EN得到重复检测结果dr。若$dr=1_I$, 则C_{b_i}是D_O内部的重复数据, EN仅需上传密文偏差C_{d_i}至CSP。若$dr=1_C$或$dr=0$, EN要求D_i上传密文基C_{b_i}。若$dr=0$, EN计算标签$T_{b_i} \leftarrow H_1(C_{b_i})$, 在$I_{TE}$中记录$(T_{b_i}, id_{D_O})$, 上传$(T_{b_i}, C_{b_i}, C_{d_i}, id_{D_O}, C_{k_{b_i}})$至CSP。若$dr=1_C$, EN在$I_{TE}$中记录$(T_{b_i}, id_{D_O})$, 并上传$(id_{D_O}, C_{k_{b_i}}, C_{d_i})$至CSP。

4)数据访问。当D_O需访问其终端设备D_i上传的数据M_i时, 其向CSP发起请求。CSP检索密文基C_{b_i}、密文偏差C_{d_i}和加密的密钥$C_{k_{b_i}}$, 随后将它们返回给D_O。D_O使用密钥k_{D_O}解密$C_{k_{b_i}}$和C_{d_i}以得到密钥k_{b_i}和偏差d_i, 然后使用k_{b_i}解密C_{b_i}得到b_i。最后, D_O基于b_i和d_i恢复原始数据M_i。

4.3 讨 论

本节给出 SEDedup 的一些相关讨论。

1)存储优势。这里从系统整体存储开销的角度分析 SEDedup 的存储优势。与现有方案相比, SEDedup 实际上并未显著降低系统整体的存储成本, 而是将体量较大的完整数据索引存储在云服务器端, 从而显著减少边缘节点的存储开销。这一设计充分利用了云服务器的强大存储资源, 同时缓解了资源受限的边缘节点的存储负担, 可以有效规避边缘节点存储能力有限的这一问题, 避免边缘节点成为系统的性能瓶颈。因此, SEDedup 可以很好地提升云边协同存储场景下加密去重系统在实际部署中的可用性与系统整体的可扩展性。

2)广义去重。这里讨论广义去重技术如何服务于 SEDedup 的主体方案。一方面, SEDedup 利用广义去重实现了相似数据的无损去重, 显著提升了系统存储效率。另一方面, SEDedup 基于广义去重实现了高性能的数据差异化保护。SEDedup 分别使用服务器辅助加密和概率加密保护基和偏差, 对重复率高的基进行去重提升存储性能, 对敏感度高的偏差则提供高强度的安全保护以防止敏感信息泄露。SEDedup 通过结合广义去重与密码学技术实现了安全性与存储性能间的合理折中。

3)令牌安全性。这里讨论令牌泄露、被重放、被复制或需要更新时, 系统应如何保证安全性。首先, 为对抗令牌泄露以及防止针对令牌的重放攻击, 系统可在令牌生成阶段引入时间戳。当令牌泄露或被重放时, 系统可通过验证令牌时间戳的有效性进行防御。其次, 为防止敌手复制令牌后盗用, 系统可在EN部署可信执行环境, 将令牌的存储与运算封装在EN的可信执行环境中, 确保令牌仅在可信执行环境中调用且不可导出, 从而避免令牌被复制。最后, 若主密钥或设备密钥更新, 系统可通过让设备、EN和KS重新执行令牌生成过程进行更新。

5 安全性分析

本节基于 3.2 节中的威胁模型分析 SEDedup 的安全性。

5.1 令牌生成的安全性分析

本节基于数字签名的不可伪造性和RSA加密算法的语义安全性来证明方案中令牌生成的安全性。

定理 1. 若数字签名具有不可伪造性且RSA加密是语义安全的,那么EN在令牌生成期间无法得到主密钥msk、秘密值β_{D_0}或终端设备密钥dsk_i的任何信息。

证明. 在令牌生成阶段,EN接收到的数据包括$X \leftarrow \frac{r}{dsk_i}$, $C_r \leftarrow Enc_{\mathrm{RSA}}(pk_{\mathrm{KS}}, r)$, $Sig_{C_r} \leftarrow Sign_{\mathrm{S}}(ssk, C_r)$和$Y \leftarrow \frac{msk \cdot \beta_{D_0}}{r}$。若RSA加密是语义安全的,则EN无法从$C_r$中恢复$r$,因此也无法利用$X$或$Y$来计算$dsk_i$, msk和β_{D_0}。EN可能会试图生成随机值$r' \in Z_q$,同时伪造RSA密文$C'_r = Enc_{\mathrm{RSA}}(pk_{\mathrm{KS}}, r')$。但由于EN没有签名密钥$ssk$,其无法伪造出有效的签名$Sig_{C'_r} = Sign_{\mathrm{S}}(ssk, C'_r)$。综上,EN在令牌生成阶段无法得到$msk$, dsk_i或β_{D_0}的信息。 证毕.

5.2 数据机密性

本节证明SEDedup的数据机密性。安全模型由安全游戏$\mathcal{G}_{\mathcal{A}}$定义,其中包含敌手$\mathcal{A}$和挑战者$C$。$\mathcal{G}_{\mathcal{A}}$的具体描述如下。

1)初始化阶段。挑战者C输入安全参数λ,生成一个生成元为g、阶为素数q的乘法循环群$\mathbb{G}$、系统主密钥$msk \in \mathbb{Z}_q$和一个均匀随机的对称密钥$k_{D_0} \leftarrow \{0,1\}^{\lambda}$。另外,$C$选择哈希函数: $H_1: \{0,1\}^* \rightarrow \{0,1\}^{\lambda}$和$H_2: \{0,1\}^* \rightarrow \mathbb{G}$,公开系统参数$pp \leftarrow \{q, \mathbb{G}, g, H_1, H_2\}$。

2)查询阶段。敌手$\mathcal{A}$适应性地选择明文消息发起查询。对于接收到的查询消息m_i,挑战者C使用广义去重将其拆分为基b_i和偏差d_i,计算$k_{b_i} \leftarrow H_1((H_2(b_i))^{msk})$,使用对称加密算法计算密文偏差$C_{d_i} \leftarrow Enc_{\mathrm{SE}}(k_{D_0}, d_i)$、密文基$C_{b_i} \leftarrow Enc_{\mathrm{SE}}(k_{b_i}, b_i)$和加密的密钥$C_{k_{b_i}} \leftarrow Enc_{\mathrm{SE}}(k_{D_0}, k_{b_i})$。$C$返回$(C_{d_i}, C_{b_i}, C_{k_{b_i}})$给$\mathcal{A}$。

3)挑战阶段。$\mathcal{A}$输出2个等长的明文消息m_0和m_1,C随机选择$b \in \{0,1\}$,生成对应的密文$(C_d^b, C_b^b, C_{k_b}^b)$,并返回给$\mathcal{A}$。

4)猜测阶段。$\mathcal{A}$输出$b' \in \{0,1\}$。

定义 1. 若对于任何的概率多项式时间敌手$\mathcal{A}$,均存在一个可忽略的函数$negl(\lambda)$,满足$Adv_{\mathcal{A}} = |Pr[b = b'] - 1/2| \leqslant negl(\lambda)$,则SEDedup可提供数据机密性。

定理 2. 若对称加密算法是语义安全的,则SEDedup可提供数据机密性。

证明. 这里通过定义一系列不可区分的安全游戏来进行证明。

$\mathcal{G}_{\mathcal{A}}^0$: 该游戏与$\mathcal{G}_{\mathcal{A}}$相同。

$\mathcal{G}_{\mathcal{A}}^1$: 挑战者$C$将$k_{b_i}$替换为均匀随机的对称密钥$k_b^r \leftarrow \{0,1\}^{\lambda}$。$\mathcal{G}_{\mathcal{A}}^1$与$\mathcal{G}_{\mathcal{A}}^0$的不可区分性源自$msk$的保密性。由于$msk$对于敌手而言是保密的,$(H_2(b_i))^{msk}$与随机的群元素$R$具有不可区分性。因此,从敌手的视角来看,$k_{b_i} = H_1((H_2(b_i))^{msk})$与随机密钥$k_b^r$不可区分。

$\mathcal{G}_{\mathcal{A}}^2$: 挑战者$C$将对称加密的密文$C_d^b$、$C_b^b$和$C_{k_b}^b$分别替换为与它们长度相等的随机比特串$R^{(1)}$, $R^{(2)}$和$R^{(3)}$。$\mathcal{G}_{\mathcal{A}}^2$与$\mathcal{G}_{\mathcal{A}}^1$的不可区分性源自对称加密的语义安全性。由于对称密钥k_b^r与k_{D_0}对于敌手是保密的,基于对称加密的语义安全性,$\mathcal{G}_{\mathcal{A}}^2$与$\mathcal{G}_{\mathcal{A}}^1$具有不可区分性。

在$\mathcal{G}_{\mathcal{A}}^2$中,敌手$\mathcal{A}$得到的内容均为均匀随机的。因此,其只能够随机地输出b',其概率优势是可忽略的。由于$\mathcal{G}_{\mathcal{A}}^2$与$\mathcal{G}_{\mathcal{A}}^0$具有不可区分性,$\mathcal{A}$在$\mathcal{G}_{\mathcal{A}}^0$中的概率优势同样可忽略。综上,定理2得证,SEDedup可提供数据机密性。 证毕.

5.3 攻击抵抗性

本节证明SEDedup能够抵抗敌手发起的各类攻击。

1)SCA。SEDedup通过两阶段去重防御SCA。由于EN对用户间重复数据执行基于目标的去重,敌手基于重复检测的结果无法确定其他用户的数据存在性信息,因此可以高效防御SCA。

2)DFA。SEDedup对基的密文执行跨用户去重。敌手可能会发起DFA,试图诱导系统错误地执行去重操作,破坏合法用户的数据完整性。

定理 3. 若哈希函数H_1是抗碰撞的,那么SEDedup可以抵抗DFA。

证明. EN使用数据标签$T_{b_i} \leftarrow H_1(C_{b_i})$进行重复检测,可确保标签与密文间的一致性,以防御DFA。若敌手上传了不一致的密文基和标签,成功发起了DFA,那么其需要构造密文对(C'_{b_i}, C_{b_i})满足$C'_{b_i} \neq C_{b_i}$且$H_1(C'_{b_i}) = H_1(C_{b_i})$,这破坏了$H_1$的抗碰撞性。若敌手上传的密文与标签是一致的,但密文数据是恶意构造的,由于合法用户的数据内容不会与敌手恶意构造的密文重复,因此云服务器不会执行去重,合法用户的数据完整性不会被破坏。综上,SEDedup可有效抵御DFA。 证毕.

3)BFA。基的密钥k_{b_i}基于数据内容和主密钥msk生成。由于msk是保密的,EN或CSP都不能通过发起BFA来得到可能的基的密钥。若敌手试图发起大量请求来获得密钥,EN和CSP可执行速率限制机制进行防御。偏差的密文C_{d_i}和密钥的密文$C_{k_{b_i}}$均由对称密钥k_{D_0}加密。由于k_{D_0}是保密的,基于对称加密

算法的语义安全性，敌手无法对C_{d_i}或$C_{k_{b_i}}$发起 BFA。

4）合谋攻击（collusion attack, CA）。敌手可能同时攻破终端设备和其域内的EN，因此可得到dsk和$\Delta=\frac{msk\cdot\beta_{D_0}}{dsk}$。敌手试图使用$dsk$和$\Delta$获取数据哈希$h'$，通过$h'$发起 BFA 以恢复数据信息。具体而言，敌手基于$dsk$和$\Delta$计算$msk\cdot\beta_{D_0}\leftarrow dsk\cdot\Delta$，随后试图使用$msk\cdot\beta_{D_0}$从$(h')^{msk}$中获得$h'$。然而，由于敌手无法获取秘密值$\beta_{D_0}$，因此无法从$msk\cdot\beta_{D_0}$获取$msk$。此外，基于离散对数难题，敌手无法从$(h')^{msk}$中恢复$h'$。因此，SEDedup 能够防御EN和恶意终端设备发起的 CA。

6 评　　估

本文基于 OpenSSL[49] 和 GMP[50] 等工具库实现了方案 SEDedup 的原型。原型由异质终端设备、EN、KS和CSP组成。原型中使用的哈希函数和对称加密算法分别为 SHA-256 和 AES-256。签名算法基于 RSA 实现。RSA 加密的密钥长度为 2 048 比特。此外，本文还实现了当前云边协同场景下最先进的加密去重方案 FCDedup[2] 和 IoT-Dedup[32] 以进行对比实验。本文的实验在配备 2.5 GHz Intel Core i5-13400 四核处理器、64 位 Ubuntu 20.04.6 操作系统和 8 GB 内存的机器上运行。本节评估的结果均为 100 次以上运行结果的平均值。

6.1 复杂度分析

本节给出 SEDedup 的复杂度分析，包括终端设备计算复杂度、跨用户去重的通信复杂度和边缘节点的存储复杂度。复杂度分析使用的符号与分析结果如表 3 和表 4 所示。

Table 3 Notations Used in Complexity Analysis

表 3 复杂度分析中使用的符号

符号	描述
N_{total}	系统产生的数据标签总数
L_M	数据字节长度
V	物联网设备关系图中的设备节点总数
N_{epoch}	单个时段内数据标签总数
L_d	偏差的长度

在终端设备计算复杂度方面，3 个方案的渐进复杂度均为$O(L_M)$。但是 SEDedup 处理单条数据仅需执行 2 次哈希与指数运算。相比之下，FCDedup 需要执行多次群元素的幂运算和乘法运算，IoT-Dedup 需执行成本更高的特征词提取和模糊提取器运算。在跨用户去重的通信复杂度方面，SEDedup 结合广义去重技术，在边缘节点去重相似数据的基，仅向云端传输较小的密文偏差，将通信复杂度降低为$O(L_d)$。在边缘节点的存储复杂度方面，SEDedup 利用终端数据的时间局部性，仅维护当前时段的临时索引和 CF，将存储复杂度降低为$O(N_{\text{epoch}})$。

Table 4 Complexity Analysis

表 4 复杂度分析

方案	终端设备计算复杂度	通信复杂度（跨用户去重）	边缘节点存储复杂度
FCDedup	$O(L_M)$	$O(L_M)$	$O(N_{\text{total}})$
IoT-Dedup	$O(L_M)$	$O(L_M)$	$O(N_{\text{total}}+\lvert V\rvert^2)$
SEDedup	$O(L_M)$	$O(L_d)$	$O(N_{\text{epoch}})$

6.2 基于合成数据的实验评估

本节使用合成数据来评估方案的计算和通信开销，并评估不同参数下 CF 对系统通信开销的影响。在 FCDedup 中，匹配同一短哈希的标签数量n会影响系统性能。因此本文评估了当n分别为 5、10 和 20 时的 FCDedup 的开销。测试文件的大小设置为 10 MB。数据上传可分为 3 种情况：非重复数据上传（$dr=0$）、用户内重复数据上传（$dr=1_I$）和跨用户重复数据上传（$dr=1_C$）。虽然 IoT-Dedup 同时支持模糊去重和精确去重，但本节的评估仅关注其精确去重的性能，以确保实验评估的公平性。

6.2.1 非重复数据的上传

图 5(a)和图 5(b)给出了上传非重复数据时的计算和通信开销。相较于 FCDedup 和 IoT-Dedup，SEDedup 的计算开销明显更低。这是因为 FCDedup 在数据上传过程中需要执行多次计算昂贵的幂运算，而 SEDedup 则是基于高效的服务器辅助加密来保护数据隐私。此外，FCDedup 需要执行计算昂贵的配对运算，其原型需基于低效的 PBC 代码库[51] 实现。相比之下，SEDedup 仅需要执行幂运算，并且原型基于更高效的 GMP 代码库实现，因此 SEDedup 的计算开销更低。随着n的减小，FCDedup 的开销会不断降低。但即使当$n=5$时，SEDedup 的计算速度仍是 FCDedup 的 400 倍。此外，与 FCDedup 相比，SEDedup 显著降低了通信开销。原因在于 FCDedup 需要基于短哈希来检索候选标签以进行重复检测，这些候选标签的传输会引入额外的通信开销。SEDedup 的通信开销高于 IoT-Dedup，这是因为 IoT-Dedup 仅传输标签。然而，仅传输标签使得 IoT-Dedup 易受到 BFA、SCA 和 DFA。

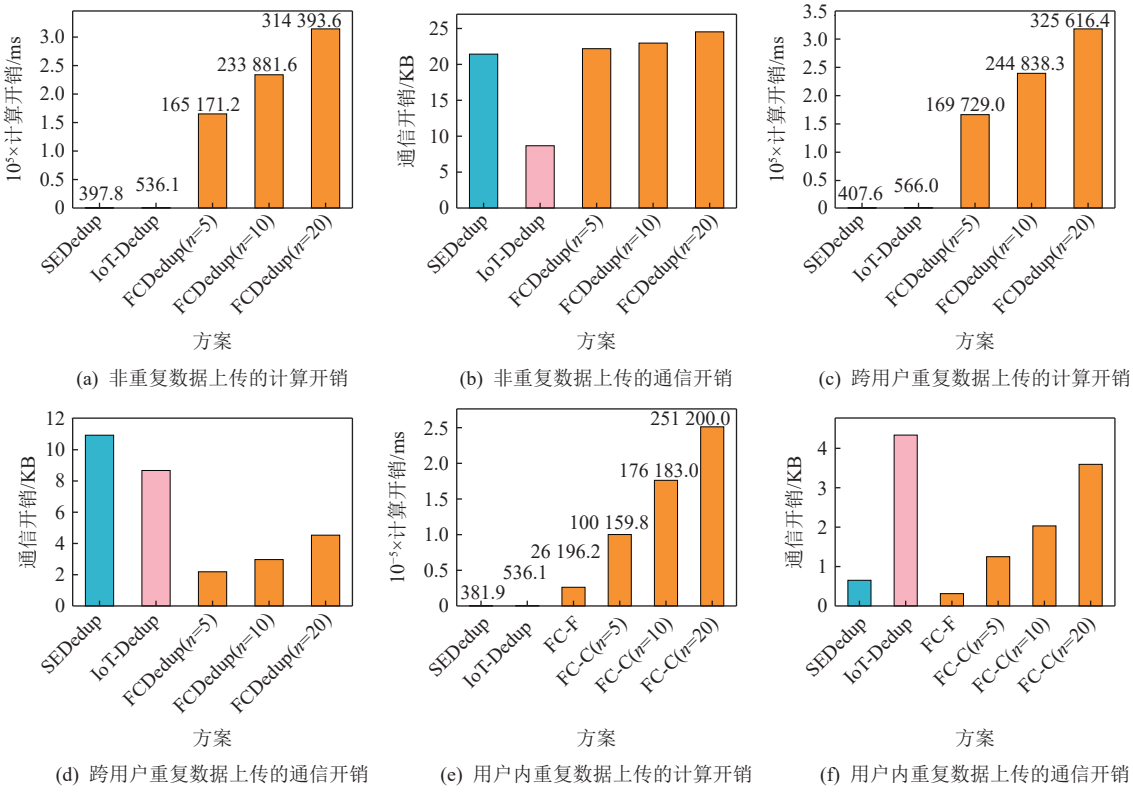


Fig. 5 Evaluation on synthetic data

图 5 合成数据评估

6.2.2 跨用户和用户内重复数据上传

图 5(c)和图 5(d)给出了跨用户重复数据上传的开销。SEDedup 的计算速度明显快于 FCDedup。这同样得益于 SEDedup 采用改进后的服务器辅助加密来保护数据，减少了配对运算和幂运算的执行次数。由于需要上传跨用户重复数据以防御 DFA 和 SCA，SEDedup 的通信成本高于 FCDedup。在 IoT-Dedup 中，跨用户和用户内重复数据上传 2 种情况的通信开销相同，因为这 2 种情况下设备都需要将完整的数据内容上传至CSP。

图 5(e)和图 5(f)给出了用户内重复数据上传的开销。在 FCDedup 中，用户内去重可能由EN或CSP执行。由于开销不同，本文分别用 FC-F 和 FC-C 来表示EN和CSP执行用户内去重的开销。当上传用户内重复数据时，得益于高效的服务器辅助加密，与 FC-F 相比，SEDedup 显著降低了计算开销。FC-C 的计算开销明显更高，因为它需要将所有候选标签转换为数据标签以进行重复检测，且这一开销会随着n的增加而增加。由于需要基于服务器辅助加密来生成密钥，SEDedup 的通信成本较 FC-F 相比增加了 25%。由于 IoT-Dedup 需从EN获取额外的元数据，因此其通信开销最高。

6.2.3 数据访问

本节使用不同大小的测试文件评估了数据访问的性能。如图 6 所示，与 FCDedup 相比，SEDedup 的计算速度提升了 31.5%~65.7%。这是因为 SEDedup 中的密钥恢复过程更高效。与 SEDedup 相比，IoT-Dedup 仅减少了 1 次密钥解密操作，因此两者在数据访问阶段的总计算开销相近。

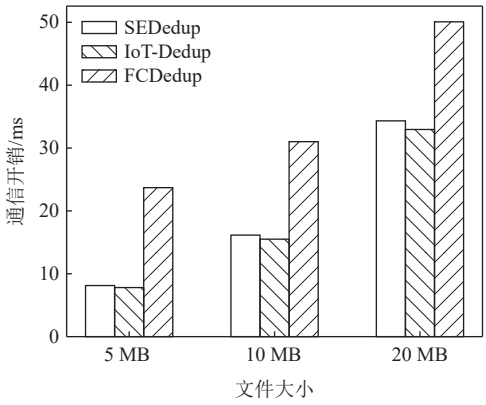


Fig. 6 Overhead of data access

图 6 数据访问开销

6.2.4 CF 对方案性能的影响

本节评估 CF 的容量、负载率和误判率对系统通信开销的影响。CF 的误判率会给系统带来一定的额

外通信成本。如表 5 所示，在过滤器容量固定的情况下，随着负载率升高，过滤器的误判率随之上升，导致系统产生的通信开销增加。另外，在相同负载率下，采用更大容量的 CF 能够有效降低误判率，进而降低通信开销。这表明，在实际部署中，适当增大过滤器的容量并控制其负载率，可以有效降低因过滤器误判而引起的通信开销。

Table 5 Impact of CF on Performance

表 5 CF 对性能的影响

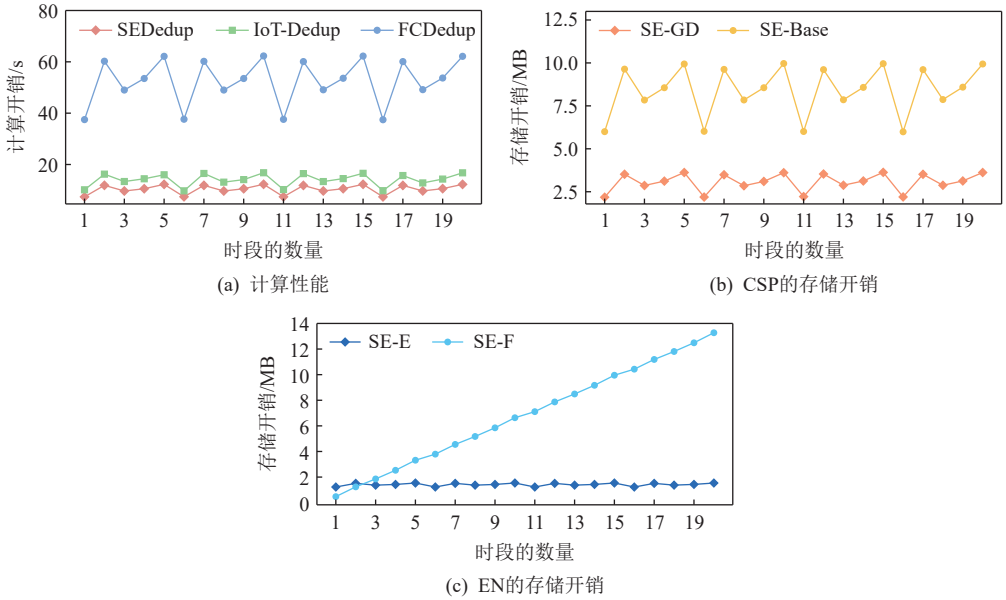
容量/项	负载率/%	误判率/%	通信开销/KB
100 000	50	1.17	18.88
100 000	80	1.97	31.74
100 000	95	2.28	36.67
500 000	50	0.83	13.31
500 000	80	1.17	18.85
500 000	95	1.45	23.36
1 000 000	50	0.79	12.70
1 000 000	80	1.25	20.14
1 000 000	95	1.40	22.62

6.3 基于真实世界数据的实验评估

1)计算开销。为模拟真实世界场景，本文假定系统中存在 2 个域，每个域内有 2 个 D_O，每个 D_O 部署 2 台异质终端设备。将数据集中同一时段内的数据均匀分给系统中的所有终端设备，这些终端设备上传数据至CSP。图 7(a)给出了连续上传 20 个时段内数据时的计算开销。与合成数据的评估结果一致，由于减少了高计算复杂度的幂运算和配对操作的执行，SEDedup 相较于 FCDedup 具有更加出色的计算性能。

2)存储效率。图 7(b)给出了 SEDedup 采用广义去重前后的存储开销变化，其中 SE-Base 和 SE-GD 分别表示使用广义去重前后的方案。由于真实数据集中存在大量高度相似但不相同的 IoT 数据，因此利用广义去重技术可显著提升存储效率。使用广义去重后，系统的存储开销可降低 63.1%~63.8%。

此外，本文评估了使用基于时段的重复检测方法前后EN的存储开销。这里使用 SE-F 来表示EN存储所有数据标签时的存储开销，使用 SE-E 来表示EN仅存储当前时段标签的存储开销。如图 7(c)所示，开始时 SE-E 更高，这是由于节点需要额外维护一个 CF，引入了一些额外的存储开销。随着时段的推进，SE-F 呈线性增长，而 SE-E 保持稳定，这验证了基于时段的重复检测方法能够有效缓解EN的存储压力。



(a) 计算性能 (b) CSP的存储开销

(c) EN的存储开销

Fig. 7 Evaluation on real-world data

图 7 真实数据评估

7 总 结

本文提出了面向云边存储中的跨用户数据加密去重方案 SEDedup。该方案提出了适用于云边存储场景改进的服务器辅助加密技术，基于该技术实现了高性能的跨用户数据加密去重，并且可抵抗 BFA、SCA 和 DFA 等安全威胁。此外，基于 IoT 数据的时

间局部性,本文提出了基于时段的重复检测方法,在实现高效重复检测的同时显著降低了EN的存储成本。并且,通过结合广义去重技术,SEDedup实现了相似IoT数据的无损加密去重,提升了系统的存储效率。安全分析和实验评估验证了方案SEDedup的安全性和高效性。相比于当前云边协同存储场景下最先进的方案FCDedup和IoT-Dedup,SEDedup实现了更好的性能。

作者贡献声明:王荣熙提出了论文方案,撰写论文;哈冠雄负责实验,修改论文;贾春福负责论文审核,指导了论文修改;陈柯言参与了论文实验,绘制了论文图表。

参 考 文 献

[1] Chen Yuping, Liu Bo, Lin Weiwei, et al. Survey of cloud-edge collaboration[J]. Computer Science, 2021, 48(3): 259−268 (in Chinese)
(陈玉平, 刘波, 林伟伟, 等. 云边协同综述[J]. 计算机科学, 2021, 48(3): 259−268)

[2] Song Mingyang, Hua Zhongyun, Zheng Yifeng, et al. FCDedup: A two-level deduplication system for encrypted data in fog computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(10): 2642−2656

[3] Zhang Chen, Miao Yinbin, Xie Qingyuan, et al. Privacy-preserving deduplication of sensor compressed data in distributed fog computing[J]. IEEE Transactions on Parallel and Distributed Systems, 2022, 33(12): 4176−4191

[4] Gao Yuan, Chen Liquan, Han Jinguang, et al. Similarity-based secure deduplication for IIoT cloud management system[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 21(4): 2242−2256

[5] Zhang Yuan, Xu Chunxiang, Li Hongwei, et al. HealthDEP: An efficient and secure deduplication scheme for cloud-assisted ehealth systems[J]. IEEE Transactions on Industrial Informatics, 2018, 14(9): 4101−4112

[6] Youngjoo S, Dongyoung K, Junbeom H. A survey of secure data deduplication schemes for cloud storage systems[J]. ACM computing surveys, 2017, 49(4): 1−38

[7] Prajapati P, Shah P. A review on secure data deduplication: Cloud storage security issue[J]. Journal of King Saud University-Computer and Information Sciences, 2022, 34(7): 3996−4007

[8] Xia Wen, Jiang Hong, Feng Dan, et al. A comprehensive study of the past, present, and future of data deduplication[J]. Proceedings of the IEEE, 2016, 104(9): 1681−1710

[9] Huang Qinlong, Zhang Zhicheng, Yang Yixian. Privacy-preserving media sharing with scalable access control and secure deduplication in mobile cloud computing[J]. IEEE Transactions on Mobile Computing, 2020, 20(5): 1951−1964

[10] Bellare M, Keelveedhi S, Ristenpart T. Message-locked encryption and secure deduplication[C]//Proc of Annual Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2013: 296−312

[11] Liu Jian, Asokan N, Pinkas B. Secure deduplication of encrypted data without additional independent servers[C]//Proc of the 22nd ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2015: 874−885

[12] Yang Zuoru, Li Jingwei, Lee P P C. Secure and lightweight deduplicated storage via shielded deduplication-before-encryption[C]//Proc of 2022 USENIX Annual Technical Conf (USENIX ATC 22). Berkeley, CA, USENIX Association, 2022: 37−52

[13] Keelveedhi S, Bellare M, Ristenpart T. DupLESS: Server-aided encryption for deduplicated storage[C]//Proc of the 22nd USENIX Security Symp (USENIX Security 13). Berkeley, CA: USENIX Association, 2013: 179−194

[14] Ha Guanxiong, Jia Chunfu, Ge Xiaowei, et al. Similarity-aware defense scheme for online brute-force attacks in encrypted deduplication[J]. IEEE Transactions on Dependable and Secure Computing, 2026, 23: 590−603

[15] Ha Guanxiong, Ge Xiaowei, Jia Chunfu, et al. Revisiting SGX-based encrypted deduplication via pow-before-encryption and eliminating redundant computations[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 22(3): 2037−2053

[16] Ha Guanxiong, Chen Yuchen, Jia Chunfu, et al. Scalable encrypted deduplication based on location-hiding secret sharing of data keys[J]. IEEE Transactions on Computers, 2025, 74: 3710−3721

[17] Liu Hongyan, Xian Hequn, Lu Xiuqing, et al. Verifiable secure data deduplication based on user-defined security requirements[J]. Journal of Computer Research and Development, 2018, 55(10): 2134−2148 (in Chinese)
(刘红燕, 咸鹤群, 鲁秀青, 等. 基于用户定义安全条件的可验证重复数据删除方法[J]. 计算机研究与发展, 2018, 55(10): 2134−2148)

[18] He Simeng, Yang Chao, Jiang Qi, et al. Deduplication on encrypted data based on zero-knowledge proof and key transmission[J]. Journal of Computer Research and Development, 2018, 55(6): 1222−1235 (in Chinese)
(何司蒙, 杨超, 姜奇, 等. 基于零知识验证的密文去重与密钥传递方法[J]. 计算机研究与发展, 2018, 55(6): 1222−1235)

[19] Yang Chao, Zhang Junwei, Dong Xuewen, et al. Providng method of ownership of encrypted files in cloud de-duplication deletion[J]. Journal of Computer Research and Development, 2015, 52(1): 248−258 (in Chinese)
(杨超, 张俊伟, 董学文, 等. 云存储加密数据去重删除所有权证明方法[J]. 计算机研究与发展, 2015, 52(1): 248−258)

[20] Zhang Xinyu, Xian Hequn, Lu Qian, et al. Secure data deduplication scheme based on Intel SGX[J]. Journal of Cryptologic Research, 2022, 9(2): 341−352 (in Chinese)
(张新宇, 咸鹤群, 卢倩, 等. 基于 Intel SGX 的安全数据去重方

法[J]. 密码学报, 2022, 9(2): 341−352)

[21] Ha Guanxiong, Jia Qiaowen, Chen Hang, et al. Data popularity-based encrypted deduplication scheme without third-party servers[J]. Journal on Communications, 2022, 43(8): 17−29 (in Chinese)
(哈冠雄, 贾巧雯, 陈杭, 等. 无第三方服务器的基于数据流行度的加密去重方案[J]. 通信学报, 2022, 43(8): 17−29)

[22] Gao Wenjing, Xian Hequn, Cheng Runhui. A cloud data deduplication method based on double-layered encryption and key sharing[J]. Chinese Journal of Computers, 2021, 44(11): 2203−2215 (in Chinese)
(高文静, 咸鹤群, 程润辉. 基于双层加密和密钥共享的云数据去重方法[J]. 计算机学报, 2021, 44(11): 2203−2215)

[23] Tang Xin, Zhou Linna, Shan Weijie, et al. Threshold re-encryption based secure deduplication method for cloud data with resistance against side channel attack[J]. Journal on Communications, 2020, 41(6): 98−111 (in Chinese)
(唐鑫, 周琳娜, 单伟杰, 等. 基于阈值重加密的抗边信道攻击云数据安全去重方法[J]. 通信学报, 2020, 41(6): 98−111)

[24] Shin T, Koo D, Yun J, et al. Decentralized server-aided encryption for secure deduplication in cloud storage[J]. IEEE Transactions on Services Computing, 2017, 13(6): 1021−1033

[25] Yang Xue, Lu Rongxing, Choo K K R, et al. Achieving efficient and privacy-preserving cross-domain big data deduplication in cloud[J]. IEEE Transactions on Big Data, 2017, 8(1): 73−84

[26] Yang Xue, Lu Rongxing, Shao Jun, et al. Achieving efficient and privacy-preserving multi-domain big data deduplication in cloud[J]. IEEE Transactions on Services Computing, 2018, 14(5): 1292−1305

[27] Shin H, Koo D, Hur J. Secure and efficient hybrid data deduplication in edge computing[J]. ACM Transactions on Internet Technology, 2022, 22(3): 1−25

[28] Fu Yinjin, Xiao Nong, Chen Tao, et al. Fog-to-multicloud cooperative eHealth data management with application-aware secure deduplication[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(5): 3136−3148

[29] Jiang Shunrong, Liu Jianqing, Zhou Yong, et al. FVC-Dedup: A secure report deduplication scheme in a fog-assisted vehicular crowdsensing system[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(4): 2727−2740

[30] Ni Jianbing, Zhang Kuan, Yu Yong, et al. Providing task allocation and secure deduplication for mobile crowdsensing via fog computing[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 17(3): 581−594

[31] Xie Qingyuan, Zhang Chen, Jia Xiaohua. Security-aware and efficient data deduplication for edge-assisted cloud storage systems[J]. IEEE Transactions on Services Computing, 2022, 16(3): 2191−2202

[32] Gao Yuan, Chen Liquan, Lai Jianchang, et al. IoT-Dedup: Device relationship-based IoT data deduplication scheme[J]. IEEE Transactions on Parallel and Distributed Systems, 2025, 36(5): 847−860

[33] Harnik D, Pinkas B, Shulman-Peleg A. Side channels in cloud services: Deduplication in cloud storage[J]. IEEE Security & Privacy, 2010, 8(6): 40−47

[34] Ha Guanxiong, Chen Yuchen, Cai Zhipeng, et al. Random coding responses for resisting side-channel attacks in client-side deduplicated cloud storage[J]. IEEE Transactions on Services Computing, 2025, 18: 1697−1710

[35] Qin Chuan, Li Jingwei, Patrick P C L. The design and implementation of a rekeying-aware encrypted deduplication storage system[J]. ACM Transactions on Storage, 2017, 13(1): 1−30

[36] Qi Saiyu, Wei Wei, Wang Jianfeng, et al. Secure data deduplication with dynamic access control for mobile cloud storage[J]. IEEE Transactions on Mobile Computing, 2023, 23(4): 2566−2582

[37] Ren Yanjing, Li Jingwei, Yang Zuoru, et al. Accelerating encrypted deduplication via SGX[C]//Proc of 2021 USENIX Annual Technical Conf (USENIX ATC 21). Berkeley, CA: USENIX Association, 2021: 957−971

[38] Ha Guanxiong, Jia Chunfu, Chen Yuchen, et al. A secure client-side deduplication scheme based on updatable server-aided encryption[J]. IEEE Transactions on Cloud Computing, 2023, 11(4): 3672−3684

[39] Zhang Yuan, Xu Chunxiang, Cheng Nan, et al. Secure password-protected encryption key for deduplicated cloud storage systems[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 19(4): 2789−2806

[40] Da la Torre F, Hodgins J, Montano S, et al. Carnegie mellon university multimodal activity (CMU-MMAC) database[EB/OL]. (2010)[2026-05-25]. http://kitchen.cs.cmu.edu

[41] Strittmatter S. Thermal solar plant dataset[EB/OL]. (2021)[2026-05-25]. https://github.com/stritti/thermal-solar-plant-dataset

[42] Van der Schrier G, Van der Besselaar E J M, Verver K T G. The European Climate Assessment & Dataset project[EB/OL]. (2010)[2026-05-25]. https://www.ecad.eu.

[43] Thomas P, Federico T, Veronika F L. ASHRAE global database of thermal comfort field measurements[EB/OL]. (2018)[2026-05-25]. https://doi.org/10.6078/D1F671

[44] Vestergaard R, Lucani D E, Zhang Qi. A randomly accessible lossless compression scheme for time-series data[C]//Proc of IEEE Conf on Computer Communications (IEEE INFOCOM 2020). Piscataway, NJ: IEEE, 2020: 2145−2154

[45] Fan Bin, Andersen D G, Kaminsky M, et al. Cuckoo filter: Practically better than Bloom[C]//Proc of the 10th ACM Int on Conf on Emerging Networking Experiments and Technologies. New York: ACM, 2014: 75−88

[46] Yan Zheng, Ding Wenxiu, Yu Xixun, et al. Deduplication on encrypted big data in cloud[J]. IEEE Transactions on Big Data, 2016, 2(2): 138−150

[47] Michael B S, Michael M. Encrypted key exchange: Password-based protocols secure against dictionary attacks[C]// Proc of IEEE Computer Society Symp on Research in Security and Privacy. Piscataway, NJ: IEEE, 1992: 72−84

[48] Li Mingqiang, Qin Chuan, Lee P P C. CDStore: Toward reliable, secure, and cost-efficient cloud storage via convergent dispersal[C]// Proc of 2015 USENIX Annual Technical Conf (USENIX ATC 15). Berkeley, CA: USENIX Association, 2015: 111−124

[49] The OpenSSL Project. OpenSSL[EB/OL]. (2025)[2026-5-25]. https://www.openssl.org/

[50] Torbjöm Granlund and GMP Development Team. GMP[EB/OL]. (2023)[2026-5-25]. http://gmplib.org/

[51] Lynn B. Pbc[EB/OL]. (2016)[2026-5-25]. http://crypto.stanford.edu/pbc/



Wang Rongxi, born in 2003. PhD candidate. His main research interests include security and privacy issues in cloud computing, storage security, and applied cryptography.

王荣熙, 2003 年生。博士研究生。主要研究方向为云计算中的安全与隐私问题、存储安全、应用密码学。



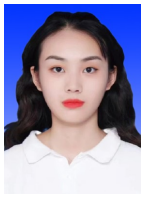
Ha Guanxiong, born in 1995. PhD. His main research interests include encrypted deduplication, cloud data security, and applied cryptography.

哈冠雄, 1995 年生。博士。主要研究方向为加密去重、云数据安全、应用密码学。



Jia Chunfu, born in 1966. PhD, professor, PhD supervisor. His main research interests include network and system security, applied cryptography, and malware analysis.

贾春福, 1966 年生。博士，教授，博士生导师。主要研究方向为网络与系统安全、应用密码学、恶意代码分析。



Chen Keyan, born in 2002. Master candidate. Her main research interests include applied cryptography.

陈柯言, 2002 年生。硕士研究生。主要研究方向为应用密码学。