基于 Reed-Solomon 编码的抗边信道攻击云数据安全去重方法

刘小梅, 唐　鑫, 杨舒婷, 陈　雄, 高语灿

国际关系学院网络空间安全学院 北京 中国 100091

摘要　跨用户数据去重技术, 通过在用户端减少重复数据上传来提高云端数据存储效率和用户的带宽使用效率。然而, 在数据上传过程中, 云服务商反馈给用户的确定性去重响应为攻击者建立了一个极具安全风险的边信道, 攻击者利用该边信道可推断出目标数据在云端的存在性隐私。现有的抗边信道攻击跨用户去重方法, 采用各种混淆策略试图扰乱攻击者的判断, 然而, 这些方法难以实现完全混淆, 攻击者仍然可通过字典攻击、附加块攻击等方式达到数据窃取的目的。目前, 如何防止攻击者利用边信道窃取数据的存在性隐私, 成为了跨用户数据去重技术亟待解决的重要问题。为应对这一挑战, 本文采用了一种基于广义去重的新型跨用户安全去重框架, 将原始数据从字节层面分解为基和偏移量, 对基进行跨用户去重, 并对偏移量进行云端去重。特别地, 本文采用 Reed-Solomon 纠删码编码思想实现基的提取, 使得从相似的数据中可以较高概率提取出相同的基。不仅可以实现对攻击者的混淆, 还可以有效节省通信开销和云端存储开销。此外, 为了进一步提高效率, 本文在偏移量上传前, 引入数据压缩算法, 减少偏移量间的冗余数据量。实验结果表明, 在实现有效抵抗边信道攻击的前提下, 本方法相比该领域最新工作在通信和存储效率等方面具有显著优势。

关键词　广义去重; 边信道攻击; 云存储; Reed-Solomon 编码

中图法分类号　TP302　DOI 号 10.19363/J.cnki.cn10-1380/tn.2022.11.05

Reed-Solomon Coding Based Secure Deduplication for Cloud Storage with Resistance Against Side Channel Attack

LIU Xiaomei, TANG Xin, YANG Shuting, CHEN Xiong, GAO Yucan

School of Cyber Science and Engineering, University of International Relations, Beijing 100091, China

Abstract　Cross-user data deduplication technology improves cloud data storage efficiency and user bandwidth usage efficiency by reducing repeated data uploads on the user side. However, during the data uploading process, the deduplication response fed back to the user by the cloud service provider a side channel with a very high security risk for the attacker, and the attacker can use this side channel to infer the existence of the target data in the cloud. The existing cross-user deduplication methods against side-channel attacks use various obfuscation strategies to try to disrupt the attacker's judgment. However, these methods are still difficult to achieve complete obfuscation, and attackers can still use dictionary attacks, additional block attacks, etc. to complete the attack. At present, how to prevent attackers from stealing the existential privacy of data by using side channels has become an important problem to be solved urgently in cross-user data deduplication technology. To address this challenge, this paper adopts a new cross-user security deduplication framework based on generalized deduplication. We decompose the original data into bases and offsets from the byte level, then we conduct cross-user deduplicates on the bases, and deduplicate the offsets in the cloud side. In particular, this paper adopts the idea of Reed-Solomon erasure coding to achieve basis extraction, so that the same bases can be extracted from similar data with a high probability. Not only can confuse attackers, but also effectively save communication bandwidth and cloud storage overhead. In addition, in order to further improve the efficiency, this paper introduces a data compression algorithm before uploading the deviation to reduce the amount of redundant data between the offsets. Under the premise of effectively resisting side-channel attacks, the experimental results show that this method has significant advantages in communication and storage efficiency compared with the latest work in this field.

Key words　generalized deduplication; side channel attack; cloud storage; Reed-Solomon coding

通讯作者: 唐鑫, 博士, 副教授, Email: xtang@uir.edu.cn。

本课题得到国家自然科学基金项目（No. 62102113）、国际关系学院国家安全高精尖学科建设科研专项基金资助项目（No. 2021GA08）、国际关系学院大学生学术支持计划项目（No. 3262022SYJ012）以及国际关系学院中央高校基本科研业务项目（No. 3262022T20）资助。

收稿日期: 2022-06-20; 修改日期: 2022-10-10; 定稿日期: 2012-10-10

1 引言

随着大数据时代的到来, 用户使用各式终端在互联网上产生了海量、高冗余的数据。云存储的出现, 为这些数据的高效存放提供了一种有效的解决方案。然而, 大量冗余重复的数据为云服务商带来了沉重的存储和管理开销。据统计, 当前在云端保存的数据重复率高达 60%左右[1]。为了将这些数据上传到云端, 用户也需要付出巨大的冗余通信开销。

跨用户数据去重技术[2]是解决以上问题的有效手段。在跨用户数据去重技术中, 用户仅需上传云端不存在的副本, 以提高通信、存储效率。具体地, 当用户上传文件时, 首先生成文件的哈希值[3]作为去重请求发送给云服务商。后者在接收到用户请求后, 检查所请求文件副本是否在云端存在。如果云端已保存该文件副本, 则会向用户端返回一个存在性响应来指示文件的存在, 并在云端文件副本中添加该用户的所有权信息。用户将不再需要上传文件。如果云端尚未保存该文件副本, 此时云服务商将通过响应要求用户上传整个文件。考虑到数据去重的粒度, 跨用户去重可分为块级去重和文件级去重。其中, 前者目前的应用较为普遍。在这个技术实施过程中, 用户将文件划分固定大小或可变大小的块, 上传块的哈希值给云服务商以确定块存在性。相较于文件级去重, 这种方式可以进一步提高去重效率。

尽管跨用户数据去重技术可以有效地节约存储空间和提高带宽利用率, 然而, 该技术返回的确定性响应为攻击者提供了一个边信道。后者可以通过发起边信道攻击[3], 观察云端响应来判断所请求数据在云端的存在性。具体地, 攻击者可以通过猜测目标数据的内容, 并向云端发起去重请求。如果云服务商不要求进一步上传数据, 攻击者即可知道所猜测的数据在云端存在, 存在性隐私暴露。这种隐私窃取的方法在现实场景中普遍可行。在真实攻击案例中, 攻击者可能为了非法获取目标对象的薪资信息, 生成带有已知目标对象姓名、猜测工资金额的数据, 以向云端发送去重请求的方式, 确定所请求文件的存在性, 以此判断目标对象的薪资情况。为了提高文件命中率, 攻击者可更近一步的以字典攻击等方式生成多个可能的数据文件, 重复向云端发送重复数据检查请求, 以达到目的。

为了解决边信道攻击问题, 当前, 国内外已开展了大量的研究工作。一类方法是云服务商通过设置去重响应阈值[4-6]的方式限制攻击者获取存在性隐私。当用户请求去重的数据在云端的副本数未达到阈值时, 云服务商要求用户上传数据, 从而使得攻击者无法探查到数据存在性情况; 反之, 当用户请求去重的数据在云端的副本数量达到阈值时, 云服务商返回存在性响应阻断数据的进一步上传。此时, 即使攻击者获知数据的存在也不会产生危害, 因为数据已经在云端成为流行数据。该类方法虽然在一定程度上可以实现数据存在性隐私保护, 但是在云端副本数量未达到阈值时, 仍要求用户上传大量冗余数据。一方面给云服务商带来了大量的存储开销, 另一方面也使得用户需付出额外的通信开销。另一类方法是采用混淆策略[7-11]来混淆攻击者。该类方法的基本思路为: 将文件分为若干块, 假定用户姓名、证件号等敏感信息存在于其中一个或多个块中。对云服务商来说, 无论敏感信息是否在云端存在, 其需要对请求者生成一个混淆的响应, 使得后者无法通过去重响应区分存在性状态。然而在真实场景中, 云服务商难以辨别敏感块的位置, 在敏感块命中的情况下, 仍然会出现隐私泄露的风险。尽管后续研究者们提出一些改进方法, 但这些方法仍然存在缺陷, 难以实现完全混淆。近年来, 有研究者提出广义去重方法[12-13], 并随之发展出一套安全去重框架。在该框架下, 原始数据被分解为基和偏移量。为了实现混淆, 只对基进行跨用户去重, 对偏移量开展云端去重。由于相似的数据可以一定概率提取出相同的基, 因此, 攻击者无法根据基的去重响应判断目标数据的云端存在性, 边信道攻击可被有效抵抗。然而, 如何以较高概率从相似数据提取出相同的基, 并且进一步降低偏移量上传的通信开销, 是该方法现阶段面临的重要挑战。

针对以上问题, 本文提出了一种基于 Reed-Solomon 编码的跨用户安全去重方法。具体地, 该方法采用基于广义去重的新型跨用户安全去重框架, 在基处理中引入 Reed-Solomon 纠删码编码思想, 以较高概率从相似的数据中提取出相同的基, 从而提高去重效率。除此之外, 本文对偏移量引入字典压缩算法以进一步降低通信开销。具体的, 本文的主要贡献如下:

(1) 本文提出了一种新型的基于 Reed-Solomon 纠删码的跨用户安全去重框架。将原始数据从字节层面分解为基和偏移量, 对基进行跨用户去重, 对偏移量开展云端去重。特别地, 该框架支持基于 Reed-Solomon 纠删码的基提取方案, 以及偏移量字典压缩方法。在所提框架下, 数据存在性和去重响应之间的确定性联系被打破, 去重效率可得到有效提升。

(2) 本文设计了一种通用的基提取方法和偏移量压缩算法。相较于目前主流算法对相似文件去重效率不高的问题, 所提方法利用 Reed-Solomon 纠删码编码思

想, 提升了所提取基的泛化能力。此外, 针对偏移量上传的开销问题, 本文将数据压缩算法引入偏移量计算中, 在上传前对偏移量进行压缩, 进一步提高了通信开销。

(3) 最后, 本文对所提方法开展了安全性分析和性能验证。通过在真实数据集中与当前该领域的最新工作进行对比, 验证了: 1)本文所提出的方法能够有效抵抗边信道攻击, 防止存在性隐私泄露; 2)本文所提出的基提取方法可以较高概率从相似文件中提取出相同的基, 降低云端存储开销; 3)本文所引入的偏移量压缩方法可以有效降低通信开销。

接下来, 我们将在第二节介绍云数据去重方面的相关工作, 并在第三节介绍系统模型与威胁模型等准备工作, 本文所提出的方法及相应的实验验证将在第四节和第五节展开介绍, 最后将在第六节进行总结和展望。

2 相关工作

2.1 数据去重方法及边信道攻击

云数据去重技术是解决冗余数据存储的一种有效手段, 去重技术的效率受去重发生的位置以及数据处理单元大小等因素的影响。数据去重技术按照数据处理单元大小可以分为文件级数据去重和块级数据去重。文件级数据去重是早期流行的一种服务类型, 其中每个文件只存储一个副本。如果两个或多个文件具有相同的哈希值, 则认为它们是相同的[13], 这种方法实现简单, 但去重粒度较粗, 性能较差。块级数据去重[14]是指系统将文件分割成固定大小的块或者可变大小的块进行操作, 每个块只存储一个副本。相比较于文件级数据去重, 块级数据去重的粒度更新, 所以能够实现更好的去重效率。

为了提高云端数据的存储效率, 云服务商往往会在云端进行数据去重工作, 我们称之为云端数据去重[15]。这种方法让用户无法察觉到数据去重的发生。云端数据去重提高了云端数据存储的空间利用率, 但是不能节省带宽, 用户仍需要将所有数据上传至云端。除此之外, 不仅同一用户需要向云端上传相同数据的多个副本, 不同用户存储的数据副本之间也可能存在冗余。跨用户去重[16]是解决这些问题的有效手段。具体地, 在上传前, 用户首先向云端发送数据去重请求, 例如将目标数据的哈希值发送给云服务商。若数据已在云端存在, 则用户无需进一步上传。反之, 若数据在云端不存在, 则要求用户上传数据。由此, 重复数据不用再通过网络传输, 从而达到提高存储和通信效率的目的。目前, 该类方法被广泛应用于云存储服务中。

然而, 跨用户去重技术面临边信道攻击的风险。边信道攻击区别于其他网络安全威胁, 其不依赖于硬件设备和软件环境, 主要通过与系统、服务交互, 观察系统、服务的响应来推测系统、服务内部的敏感信息存在性, 这种攻击方式有效性高于数学分析、暴力破解等传统攻击方式。在跨用户去重技术中, 攻击者可以利用数据确认机制, 通过发起字典攻击、附加块攻击等方式确认特定文件或目标块的存在性, 从而导致隐私的泄漏[17]。除此之外, 云存储中边信道也可用于隐蔽通信[4], 被攻击者操纵的云服务商通过与用户端按照正常协议的交互, 可将秘密信息编码传输出去。

2.2 抗边信道攻击的跨用户去重方法

在跨用户去重技术中, 云服务商针对特定去重请求返回的确定性响应为攻击者提供了一个边信道。因此, 研究者们针对如何替代确定性响应, 混淆攻击者上开展了一系列的研究。Harnik 等人[4]首先提出了一种基于门限去重的抗边信道攻击安全去重方法, 由云服务商为每个目标文件随机设定一个去重阈值。去重阈值的定义为触发数据去重所需的数据副本数。在云端副本数量未达到阈值时, 云服务商不返回数据存在性响应。因此攻击者在收到要求上传数据的响应时, 无法确定对应的目标文件不存在还是未达到去重阈值。当云端副本数量达到阈值时, 由于目标文件在云端已经成为流行, 故不存在隐私泄露的风险。该方法的要点在于阈值的确定, 在上述方法中, 阈值在指定范围内被预先分配。Lee 等人[5]提出每次通过随机指定阈值以提高抗统计攻击的能力。除此之外, 也有研究者提出使用更为复杂的数学方法来设置阈值。Wang 等人[6]通过模拟攻击者与云端的博弈, 以数学建模的方式生成更符合真实场景的阈值。然而, 该类方法虽然在一定程度上可以实现数据存在性隐私保护, 但是攻击者很容易通过学习, 破解阈值设置原理, 甚至变换博弈策略导致建模结果偏离真实性, 从而继续发起边信道攻击。此外, 云端副本数量未达到阈值时, 用户仍需上传大量冗余数据, 这给用户也带来了额外的开销。

另一类抗边信道攻击的重复数据删除方法是采用混淆策略影响攻击者判断, 该类方法的基本思路为: 将文件分为若干块, 对块级去重请求的响应中加入混淆, 使得攻击者很难从返回值中窃取目标块的存在性隐私。Zuo 等人[7]提出了一种块级数据去重方法 RRCS, 该方案假设目标文件的所有敏感信息均存储在一个数据块中, 其余块为公开块。当云端判断出所检测文件的真实存在性后, 在命中文件和未命中文件的响应中分别添加不同数量的冗余块, 使两

种情况下要求检测者上传的数据块数量一样, 从而确保检测者无法判断所检测文件中敏感信息块的真实存在性。然而, RRCS 仍然无法抵抗附加块攻击, 并且当云端要求用户上传的文件块里不包含敏感块时, 存在性隐私仍然将暴露。在此基础上, Tang 等人[8]无论敏感块是否存在, 均要求请求者上传所请求的所有块的异或值, 实现了响应的无差异性。当敏感块未命中时, 云服务商可结合已知的公开块和接收到的异或值, 恢复出敏感块内容。当敏感块存在时, 要求用户上传的异或值作为实现混淆的冗余信息, 而开销仅相当于一个块的大小。此外, 他们提出了一种请求合并策略以应对附加块攻击。然而, 该方法仍然假定敏感信息存在于一个块中, 这与实际情况往往不一致。随后, Yu 等人[9]提出了一种数据块对(即两个相邻数据块)去重检测方法 ZEUS。该方法同时检测一个数据块对的存在性, 并根据检测结果生成模糊化的去重响应。该方法可确保攻击者无法窃取目标块的存在性隐私, 但需要云服务方维护一个独立的数据结构以记录所有用户检测过却未上传的数据块, 加大了云端的存储和计算开销。此外, 该方案无法严格保护云端目标文件的不存在性隐私。攻击者一旦接收到云服务商发送的要求用户上传两个完整的数据块的去重响应, 则可以推断出这两个数据块均为未命中块。随后, Pooranian 等人[10]在此基础上改进, 进一步提高了云端数据存在性隐私的安全性。然而, 他们的工作面临更大的开销。Vestergaard 等人[11]中提出了一种新的基于编码策略的跨用户去重方案 CIDER, 在 CIDER 中, 用户能够一次检查两个及以上的数据块, 为实现混淆引入的通信开销最多仅为一个块的长度。然而, 这类方案仍然存在安全问题。在攻击者已知去重请求中未命中块数的情况下, 若云服务商返回响应中要求上传的数据块数量等于未命中块数, 则请求中其余目标块的存在性隐私仍将暴露。

近年来, 有研究者提出广义数据去重方法, 将原始数据分解为基和偏移量。Vestergaard 等人在文献[12]中成功验证, 可以从一组传感器收集的时间序列数据的相邻值中提取相同的基, 并将此成果应用于数据压缩领域。基于这个工作, 文献[13]首次提出一种基于广义去重的安全去重框架 SRGD, 在该框架中对目标数据的基开展跨用户去重, 对偏移量开展云端去重, 使得攻击者不再可以从去重响应中推断出完整数据的存在性隐私, 从而彻底解决边信道攻击问题。然而, 该方法在基的提取中, 使用了重复数据删除、基于内容的分块方法等策略, 虽然提高了基的泛化能力, 但在客户端消耗了较多的计算开销。除此之外, 该方法对于偏移量因采用云端去重, 上传了大量冗余的偏移量数据, 降低了客户端的通信效率。因此, 在广义去重方法中, 如何在保证数据安全的前提下, 设计一个泛化能力强且计算复杂度低的基提取算法, 并降低偏移量的通信开销仍是一个重要的挑战。

3 准备工作

本节将介绍本文所提出方法对应的系统模型, 并通过定义威胁模型来分析安全风险。此外, 本节也将介绍引入的 Reed-Solomon 纠删码编码技术。

3.1 系统模型

在本文所提模型应用的数据托管场景中, 涉及到用户和云服务商两个实体。在这个场景中, 云服务商需向用户提供数据上传及下载服务, 承担数据维护和安全保障服务。用户需向云服务商支付数据保管费用和承担数据泄漏风险。因此, 对于云服务商来说, 首先需要具备充足的存储空间, 满足一定规模的用户数据存储需求。除此之外, 云服务商还应具备较强的计算能力, 支持冗余数据管理、重复数据删除和数据完整下载功能。最重要的是, 具备完善的数据安全保管方案, 保障用户的数据完整性和敏感数据的隐私不被窃取和泄漏。

我们考虑数据上传的整个流程。在一次完整的流程中, 请求上传的用户首先对数据进行预处理, 处理完后再向云端发起重复数据删除请求。预处理的过程为: 首先对数据重新进行 ASCII 码编码, 提取每个字节的基和计算恢复字节所需的偏移量, 并对基进一步使用 Reed-Solomon 解码生成标准基, 对标准基分组并计算每个块的标签信息。然后将标签信息和所有偏移量数据上传云端发起重复数据删除请求。云服务商根据标签信息检索数据, 确定数据的存在性。如果某块数据不存在, 则要求用户端上传对应块的标准基数据, 在云端存储数据副本且执行云端数据去重工作, 并将用户加入数据所有权列表中; 如果某块数据已存在, 则不再要求用户上传对应块基数据, 随后在云端进行数据去重处理。在此场景中, 即使数据副本已在云端保存, 用户也需要上传部分数据。该模型通过分解数据, 将数据去重工作分成两部分, 分别在用户端和云端进行去重。

3.2 威胁模型

在本文考虑的场景中, 假设云服务商完全可靠, 不考虑其窃取数据、丢失数据的可能, 并且能够为用户提供可靠的数据存储和去重服务。对于任意一个数据块, 其威胁风险来源于试图窃取其状态和信息

的外部攻击者。对于攻击者来说, 其通过使用边信道攻击试图获得该数据块的存储状态。那么, 为了判断该数据块副本是否存在于云端, 攻击者会发起去重请求, 随后根据云端返回的去重响应判断该数据块在云端存在或不存在。在传统的跨用户重复数据去重模型中, 对于具有固定模版格式的可预测文件, 一旦攻击者掌握部分文件内容或文件模版规律, 极易通过字典攻击猜测出剩余部分数据, 并通过去重响应判断其在云端的真实存在性。在这种场景下, 边信道攻击对云端数据的存在性隐私构成了极大的安全威胁。

3.3 Reed-Solomon 编码

Reed-Solomon 编码(以下简称 RS 编码)[19]是定义在伽罗华域中的一种纠删码, 伽罗华域是 RS 编码的重要理论基础。在伽罗华域中, 代数运算具有封闭性, 即代数运算的结果均在域内, 不存在数据溢出问题, 所以伽罗华域又称为有限域[19]。在伽罗华域中加法等价于异或运算, 乘法等价于逻辑与运算, 负数与正数相同。二维码是一种典型的 RS 编码结果, 其使用 *GF*(256)。在 *GF*(256)域中包含 0～255 的数字, 如果计算结果超出这个范围, 则继续对计算结果进行取模运算, 直至结果在 0～255 范围内。为了减少代数计算量, *GF*(256)中执行运算所需要的生成值已生成保存, 使用中可直接在表中查询, 无需反复计算。实验表明当数据运算量非常大时, 查表的时间远远小于反复计算出值所用时间。

RS 编码会将需要编码的流数据重新排列计算, 其公式如下:

$$RS(x) = M(x) + P(x) \tag{1}$$

其中, $M(x)$和$P(x)$分别表示原始数据多项式和纠错数据多项式, 其计算公式分别如下:

$$M(x) = m_{k-1}x^{n-1} + m_{k-2}x^{n-2} + \cdots + m_0x^t \tag{2}$$

$$P(x) = p_{t-1}x^{t-1} + \cdots + p_1x^1 + p_0x^0 \tag{3}$$

其中, k 表示原始数据长度, t 表示纠错数据长度, n 表示 RS 编码长度且满足$n = k + t$, 对于任意$m_i, i \in [0, k-1]$表示 k 个原始数据的数据值, $p_i, i \in [0, t-1]$表示 t 个纠错数据的数据值。

RS 编码的过程包括消息多项式、生成多项式的构造, 随后利用两个多项式计算纠错数据。解码原理则相对简单, 根据编码过程和矩阵运算原理, 即可查找到出错位并恢复正确数值。接下来将详细介绍 RS 编码过程:

消息多项式: 消息多项式使用数据码字作为系数, 例如数据: 00100000、01011011、00001011 的十进制值: 32、91、11, 那么其消息多项式为:

$$M(x) = 32x^2 + 91x + 11$$

生成多项式: 生成多项式$g(x)$,公式为:

$$g(x) = \prod_{j=0}^{t-1}(x - \alpha^j) \tag{4}$$

使用 *GF*(256), 其中α取值为 2, t表示纠错位数。

接下来, 我们以t取 2 为例, 介绍生成多项式的计算过程:

$$g(x) = (x - \alpha^0) * (x - \alpha^1) \tag{5}$$

1) 将变量系数转为α表示, 查表可知$\alpha^0 = 1$, 故多项式可以写为

$$g(x) = (\alpha^0 x - \alpha^0) * (\alpha^0 x - \alpha^1) \tag{6}$$

2) 展开多项式, 并进行系数合并

$$g(x) = \alpha^0 x^2 - (\alpha^1 + \alpha^0)x^1 + \alpha^1 x^0 \tag{7}$$

3) 在伽罗华域中, 负数等于正数, 即减法等同于加法, 故

$$g(x) = \alpha^0 x^2 + (\alpha^1 + \alpha^0)x^1 + \alpha^1 x^0 \tag{8}$$

4) 查表可知$\alpha^0 = 1, \alpha^1 = 2$, 则

$$g(x) = x^2 + 3x^1 + 2x^0 \tag{9}$$

5) 再次查表可知$\alpha^{25} = 3$, 用符号表示n取 2 时的生成多项式为

$$g(x) = \alpha^0 x^2 + \alpha^{25} x^1 + \alpha^1 x^0 \tag{10}$$

根据上述过程可知, 生成多项式与t的取值有关, 与原始数据码字无关。

生成纠删码: 从生成多项式最高项开始重复操作, 首先对生成多项进项乘法, 使其与消息多项式最高项系数相同, 然后消息多项式和变换后的生成多项式进行异或运算, 消去消息多项式最高项。重复多次, 结果完成后所剩余数项系数即为纠删码, 最后将生成的纠删码转换为添加到原信息码码后面。

数据恢复过程: 根据编码过程可知, 如果传输过程中不发生错误, 则满足$RS(x)\%g(x) = 0$ 的要求, 若余数不为 0, 表明发生错误。

$$E(x) = \frac{RS(x)}{g(x)} = Y_1 x^{e^1} + Y_2 x^{e^2} + \cdots + Y_l x^{e^l} \tag{11}$$

其中, l 表示可纠错的数据值个数且满足$t = 2l$, $Y_i, i \in [0, l]$表示出错位置, $e^i, i \in [0, l]$表示原始正确数值。

因此, 在解码过程中, 只需根据$E(x)$计算结果替换收到的编码数据, 最后将纠错数据删除即可恢复原始数据内容。

4 基于 Reed-Solomon 码的抗边信道攻击的云数据安全去重方法

4.1 算法框架

本文提出一种基于 RS 编码的抗边信道攻击的云

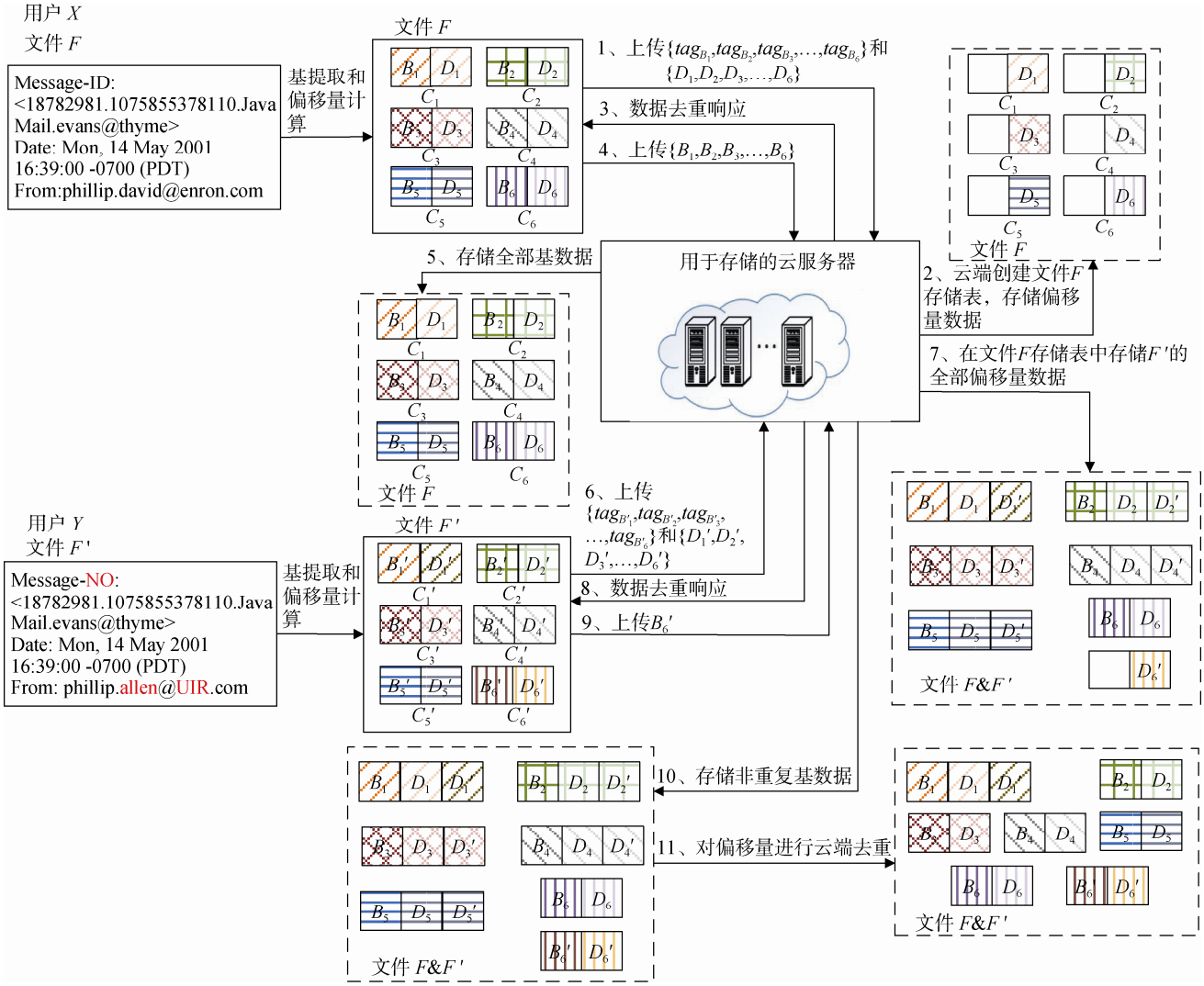


图 1　算法框架

Figure 1　The framework

数据安全去重算法，图 1 所示为算法整体框架图。假设文件 F 由其拥有者用户 X 首次向云端请求上传。在用户端，文件F的数据内容首先转为ASCII码二进制形式存储，经过基提取和偏移量计算算法处理后，文件F被分成c个块即，$F=\{C_1,C_2,C_3,\dots,C_c\}$，每个块由一组基和偏移量对组成，即，$C_1=\{B_1,D_1\},C_2=\{B_2,D_2\},\dots,C_c=\{B_c,D_c\}$，其中$B_i,i\in[1,c]$表示块$C_i$的基，$D_i,i\in[1,c]$表示块$C_i$的偏移量。基提取和偏移量计算的过程如图 2 所示，我们将在 4.2 节基提取和偏移量计算中进行介绍，这里不再展开介绍。随后，我们计算$\{B_1,B_2,B_3,\dots,B_c\}$的哈希结果$\{tag_{B_1},tag_{B_2},tag_{B_3},\dots,tag_{B_c}\}$，该哈希结果作为去重请求发送给云端。因文件 F 为首次上传，故云端收到哈希结果并检查后，发现云端不存在该数据备份。此时，云服务商要求用户上传全部数据，并在云端以数据块形式存储。假设用户 Y 拥有一份与文件F数据内容相似的文件F'，其经过算法处理后以$F'=\{C_1',C_2',C_3',\dots,C_c'\}$的形式存储。其中，$C_1'=\{B_1',D_1'\},C_2'=\{B_2',D_2'\},\dots,C_c'=\{B_c',D_c'\}$。如上图所示，由于文件$F$和$F'$内容相似，故两个文件间存在数据内容相同的文件块，图 1 中具有相同颜色和花纹的块代表相同的数据内容，因本方法所提基的泛化能力较强，可以从不同数据中提取出相同的基，故也存在数据块中基相同，而偏移量不同的情况。对文件F'而言，当用户 Y 生成基$\{B_1',B_2',B_3',\dots,B_c'\}$的哈希结果$\{tag_{B_1'},tag_{B_2'},tag_{B_3'},\dots,tag_{B_c'}\}$并将其作为去重请求发送给云服务器时，云服务器不再要求其上传与文件 F 相同的基块并对重复部分添加F'的引用，但对于偏移量数据$\{D_1',D_2',D_3',\dots,D_c'\}$则被全部要求上传。由上图所示，$B_1'$和$B_1$由于数据内容相同故没有上传云端，仅在存储列表中追加D_1'。随着文件F'的上传结束，用户端不再进行操作，云端将对

图 2　基提取和偏移量计算

Figure 2　Base extraction and deviation calculation

偏移量数据进行去重, 对相同的数据偏移量块仅保留一个副本。如图 1 所示, 数据偏移量块D_1'和D_1数据内容相同, 故仅保留其中一个。云端去重完成后, 文件F'的上传和去重流程结束。当用户请求下载文件F'时, 云端通过文件保存索引逐块还原后将数据返回用户。

序号	基 (数据恢复后)	基 (数据恢复前)					
1	*abcd*	1	*abcf*	2	*abcc*		
2	*abfc*	1	*abfd*	2	*abfg*	3	*abfh*
3	*abge*	1	*abge*	2	*abgf*	3	*abgd*
4	*bacd*	1	*bacd*	2	*bace*	3	*bach*
5	*cegh*	1	*cegh*	2	*cegd*	3	*cega*
6	…	1	…	2	…	3	…

图 3　数据存储结构图

Figure 3　Data storage structure

4.2　基提取

本方法将字节分解为基和偏移量, 对基进行用户端去重, 对偏移量进行云端去重。与基于块的数据去重方法相比, 在数据块重复的情况下, 增加了上传偏移量的通信开销, 降低了数据哈希计算开销和通信开销。在保证抗边信道攻击的安全前提下, 本方法融合 RS 纠删码原理和数据压缩方法, 设计了基提取和偏移量计算策略, 本小节首先介绍基提取策略。

假设文件 F 是由$\{x_1, x_2, x_3, \dots, x_n\}$, n 个字节组成, 对于每一个字节$x_i, (i \subseteq n)$都由基b_i和偏移量d_i组成。其中, 基b_i由 4 个比特组成, 偏移量d_i则由剩余的 4 个比特组成, 即文件 F 可以表示为:

$$F = \{(b_1, d_1), (b_2, d_2), (b_3, d_3), \dots, (b_n, d_n)\}$$

我们用 B 来表示文件 F 的基集合, D 代表文件 F 的偏移量集合, 即:

$$B = \{b_1, b_2, b_3, \dots, b_n\}$$

$$D = \{d_1, d_2, d_3, \dots, d_n\}$$

如图 2 所示, 文件 F 的基集合 $B = \{0100,0110,0111, \dots,0010, \dots\}$, 偏移量集合 $D = \{1101,0101,0011, \dots,0000, \dots\}$。接下来, 我们将按照

基提取、数据分块、偏移量压缩的步骤, 对集合 B 和 D 分别进行处理。

基于 RS 编码的数据恢复原理, 假设集合 B 是可能出错的数据结果, 对其进行数据恢复可以纠正出错位并将其还原为原始数值。那么, 对于 $B=\{b_1,b_2,b_3,\dots,b_n\}$,其数据恢复的结果可表示为 $B_{rs}=\{b_1,b_2{}^{correct},b_3,\dots,b_n\}$, 其中$b_2{}^{correct}$是纠正后的正确数值。

对于一个相似的集合$B'=\{b_1,b_2',b_3,\dots,b_n'\}$,其经过数据恢复, 可能获得结果 $B_{rs}'=\{b_1,b_2{}^{correct},b_3,\dots,b_n'\}$, b_2和b_2'经过数据恢复均可转为$b_2{}^{correct}$, 在这种情况下, 进一步提高了基集合的相似度。为了保证数据从$b_2{}^{correct}$可恢复至b_2或b_2', 需要额外记录b_2、b_2'与$b_2{}^{\text{correct}}$的对应关系。

根据第三节所介绍的RS数据编码和数据恢复过程中, 需要事先约定编码的长度, 本文所使用的长度为 4, 即每 4 个基为一组。如图 2 所示, {0100,0110,0111,0111}、{0110,0110,0110,0010}分别为一组基, 对其进行数据恢复后, 第 6 个数值从 0110 转为 0000, 从而第二组基转为{0110,0000,0110,0010}。此过程中, 可以看出相似文件可以提取出相同的基, 从而实现减少基的上传, 实现降低通信开销和存储开销的目的。然而, 重复利用 RS 编码原理进行数据恢复, 消耗了较多的计算开销, 且存在大量的重复计算。因此, 为了降低计算量, 本方法预先建立了一个基数据多对一关系的索引字典, 该索引分别保存于云端和用户端。在后续的数据上传过程和下载时, 仅需要查表代替重复计算, 索引构建方法将在 4.3 节着重介绍。在索引构建完成后, 为每组基进行编号, 如上图所示, 基{0100,0110, 0111,0111}对应序号为 8, {0110,0000,0110,0010}对应序号为 976。

在重复数据删除请求中, 我们需要对基集合进行哈希计算, 然后以哈希结果上传云端发起请求。为了避免过多的哈希计算次数, 我们设计了一个简单直接的分块方法, 选择若干组基组成一个块, 块的长度为固定值。由此, $B=\{B_1,B_2,B_3,\dots,B_c\}$, 其中变量 c 表示分块数量, 为了节约存储空间, 每个块中保存每组基的序号代替原始数值, 如图 2 所示数据中, $B_1=\{8,976,472,\dots\}$, 接下来计算每个块的哈希值用以进行跨用户去重。这里, 我们用tag_B表示计算后的哈希值集合:

$$tag_B=\{tag_{B_1},tag_{B_2},tag_{B_3},\dots,tag_{B_c}\}$$

4.3 偏移量计算

在上一节中, 本方法利用 RS 编码纠删原理, 提取基数据并通过多对一关系, 最大可能地从相似数据中提取出相同的基, 从而可以提高基数据的去重效率, 然而对偏移量数据进行云端去重, 仍然消耗了较多的通信开销。在本算法中, 为了降低通信开销, 我们引入数据压缩算法。针对数据文件的规模和特点, 选择合适的压缩算法可以极大的节约通信开销。在本方法中, 使用了 ZSTD 压缩算法[21], 它是一种轻量级的字典压缩算法, 尤其适用于小规模数据样本, 其压缩效率在使用有效字典后可提升至 80%左右, 字典无效时压缩率为 50%左右。未来, 可以利用机器学习算法, 引入效率更高、适应性更好的压缩手段。

4.2 节中, 算法会对基数据进行分组, 对应的偏移量数据也分为若干块, 在发起重复数据删除请求之前, 先对每个偏移量块内数据进行压缩, 压缩处理完成后其形式为:

$$D=\{D_1,D_2,D_3,\dots,D_c\}$$

如上图 2 所示的压缩过程中, 虚线部分表示不再存储的数据。

如图 1 所示, 我们处理后的哈希值集合tag_B和偏移量集合D上传云端, 云服务商在接收到它们后, 将tag_B与云端存储的基副本标签进行比较。根据第 4.1 节介绍的设计框架, 云端将对基执行跨用户数据去重, 对偏移量开展云端去重。因此, 云端根据tag_B检索结果, 判断是否需要用户上传基集合 B 中的未命中基块, 同时, 对偏移量集合D中的每个块与云存储中的偏移量数据开展云端去重。在这个过程中, 如果文件 F 是首次上传的全新文件, 则会在云端会建立一个新的数据存储字典$dicF[B]=D$,存储基和偏移量。如果所请求的文件F'中有一个数据块与云存储中的目标文件F的标签重复, 则认为F'与F为相似文件, 故满足以下要求时, 不再为 $F'=\{C_1',C_2',C_3',\dots,C_c'\}$重新建立字典, 而将数据直接索引在文件后。

1) 对于F'中的每一个块$C_j',j\in[1,c]$,假设存在一个块中的基 $B_j'\neq B_j$, 那么 $dicF$ 更新为 $\{B_1,B_2,B_3,\dots,B_c,B_j'\}$。

2) 对于F'中的每一个块$C_j',j\in[1,c]$,假设存在一个块中的基$B_j'=B_j$,偏移量$D_j'\neq D_j$,那么

$dicF$中$dicF[B_j]$的值更新为$\{D_j,D_j'\}$。

4.4 索引构建

本小节将介绍索引构建方法, 基数据对应关系的索引为了降低计算开销, 本方法利用 RS 纠删码思想预先构建索引, 为每一组基构建一对一和一对多的对应关系, 可以实现以下作用:

(1) 避免大量的重复计算。云数据的每次上传均

需要对其进行基的提取, 这个过程中需要大量的编码运算, 根据基数据的特点发现, 该运算过程存在较大重复计算, 通过索引结果, 可以将计算转为查询, 从而降低客户端的计算开销。

(2) 预先处理特殊结果。对于一组基数据, 本方法假设其是 RS 编码后的结果进行数据恢复, 故存在一定概率的无解情况, 在索引构建的过程中, 可提前定义无解情况, 避免客户端中的无效计算。

本方法所考虑的数据为标准 ASCII 码二进制形式, 将字节转为 ASCII 码后对其补零, 而后后对 8 位‘01’串进行划分为高四位和低四位。对于高四位即为上文所指基, 第四位为偏移量。索引构建主要构建基之间的关系。

通过观察 ASCII 码编码规则, 字节的高四位分别为 0000、0001、0010、0011、0101、0100、0111、0110, 8 种情况。0000 和 0001 分别对应使用频率较低的控制字符和通信专用字符, 其余分别对应常见的大写字母、小写字母、符号、数字等。

根据 RS 编码和数据恢复原理, 在有限范围内, 一组数值出错后可以恢复回正确结果, 简单来说则是, 正确数值序列和出错数值序列间存在对应关系。利用这一原理, 即将多组数值序列对应为一组数值序列, 即上文所述相似数据内容可提取出相同的基。

因此, 我们预先计算出每组基之间的对应关系, 可避免每次文件上传时的重复计算。这里我们选取 RS 编码的长度为 4, 即每四个基为一组, 多个基组可以转为一个相同的源, 从而进一步提高基的泛化能力。

在索引构建过程中, 首先计算出 0000、0001、0010、0011、0101、0100、0111、0110, 这 8 种情况的所有排列组合结果, 而后利用数据恢复原理, 计算其源码, 图 3 展示了相关数据的存储形式, 表示基源码和原始数据的对应结果。为了便于表述和理解, 在图 3 中, 我们以$\{a,b,c,d,e,f\}$表示{0010, 0011, 0100, 0101, 0110, 0111}, 例如值为{0010001101000111}的基, 源码值为{0010001101000101}编码而来。同样值为{0010001101000100}的基, 源码值也为{0010001101000101}。如此, 原本两个不同的基{0010001101000111}和{0010001101000100}转为同一个源码{0010001101000101}。将基通过所示表结构转为源码后, 经过测试, 存储解码前与解码后基对应关系所示表结构所需存储空间为 184512B , 为了便于基提取和数据恢复, 用户端和云端均需维护这个表结构数据。

5 实验验证及结果分析

为了验证本方法的安全性和性能, 我们选择了两个特点不同的真实数据集: Enron 电子邮件数据集[22]和 Sakila 样本数据集[23]。其中, Enron 电子邮件数据集包含 517401 个文件, 每个文件具有相同的模版, 主要包含发件人、收件人、时间、邮件内容等记录。图 4 展示了该数据集文件大小的分布状况。从图中可见, 大部分文件规模较小。Sakila 样本数据集则是由 16049 条数据记录组成, 是典型的数据库类型数据集。其中, 每条记录表示一笔支付数据, 包括订单时间、金额等信息。在这部分, 我们验证了所提方法的抗边信道攻击能力和去重效率。验证本方法的性能。具体地, 我们设计了如下三个实验:

1)计算性能验证: 通过与 SRGD[13]在相同文件提取基的计算开销和存储开销, 验证本方法在边信道攻击下的安全性和高效性;

2)去重效率验证: 通过与其他三种最新的跨用户数据去重方案 ZEUS[9]、RARE[10]和 CIDER[11]在上传相同文件时的通信开销和存储开销, 验证本方法可以提高去重效率, 节约带宽和存储空间;

3)在本实验中, 本方法所使用的 RS 编码长度为 4 个基, 文件块大小为 200 个基。在 ZEUS、RARE 和 CIDER 中, 文件被分成了固定长度为 128 字节的文件块, 并引入填充策略保证最后一个块的长度与其他文件块保持一致。

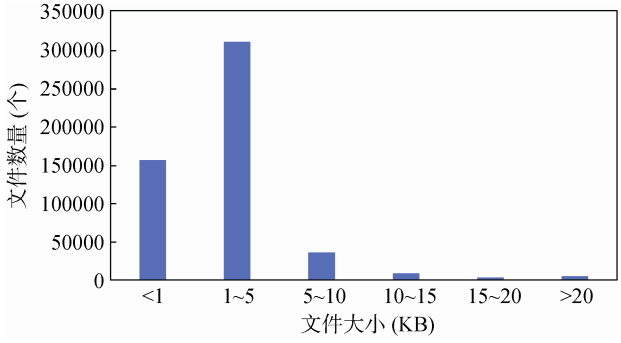


图 4 Enron 邮件数据集文件大小分布图
Figure 4 Enron file size distribution

5.1 计算性能验证

本方法提出一种字节级的数据去重框架, 对每一个字节提取基和偏移量, 对于基按照跨用户去重方法进行去重, 对于偏移量开展云端去重。当攻击者对于某一数据块发起边信道攻击时, 假设云端的相应基不存在, 那么不会发生数据存在性隐私的泄漏。假设云端对应的基存在, 云服务商返回确定性响应

阻止用户对基的上传。即使如此，攻击者仍然无法判断目标数据的云端存在性。这归结为本方法可以高概率从相似文件中提取出相同的基，即使基数据不需要上传，偏移量也仍然需要上传，故攻击者无法仅通过基的存在性判断其多对应字节数据的存在性。

在本实验中，我们首先将对算法的安全性进行实验验证。具体地，我们在 Enron 邮件数据集中，将发件人地址设定为敏感信息，并将其替换为其他地址，然后分别计算替换前后的文件数据基和偏移量。实验结果如图 5 所示。其中，图 5(a)和(b)分别表示邮件地址替换前后的两个相似文件。图 5(c)和(d)分别展示了图 5(a)和(b)两个文件前 128 个字节对应的基的情况。经过对比，可以看出两个结果完全一致，说明这两个相似文件按照所提方法可提取出相同的基。图 5(e)和(f)分别展示了两个文件前 128 个字节的偏移量计算结果。经过对比可以看出，图 5(e)和图 5(f)在第 119~124 字节的偏移量计算结果不同，对应两个文件原文的不同之处。经过该实验，我们发现本方法可以从相似文件中提取出相同的基和不同的偏移量，对于基数据可以有效的实现跨用用去重，并通过偏移量的上传，有效抵抗边信道攻击。

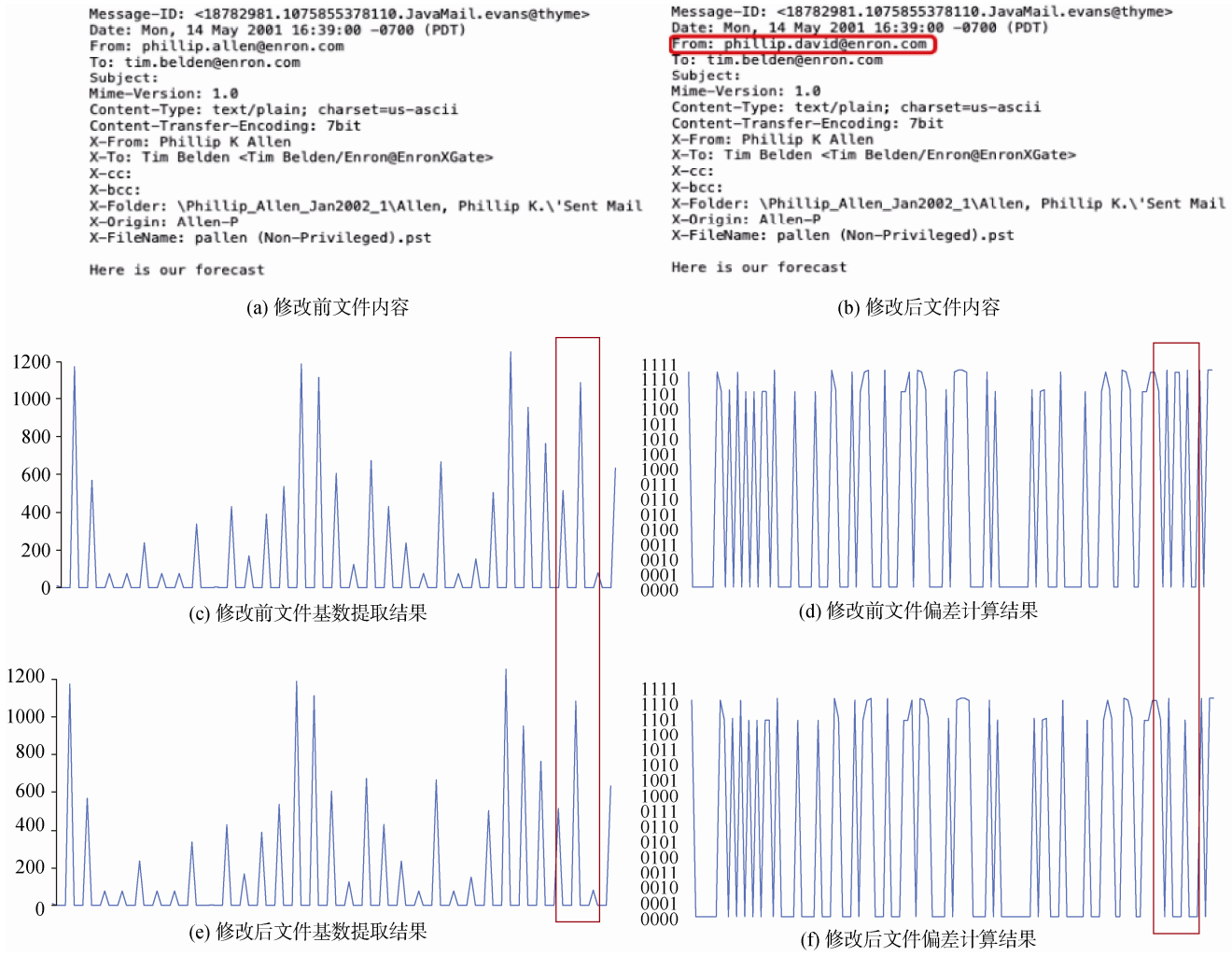


图 5　安全性分析实验结果

Figure 5　Security analysis experiment results

接下来，我们随机从 Enron 数据集中抽选 100 个文件，分别采用本方法和 SRGD 算法提取基和偏移量，对比基和偏移量大小，以及所消耗的时间开销。验证本方法在时间开销和存储开销上的优势。实验结果如图 6 所示。图 6(a)展示了对每个文件提取基和偏移量的计算时间对比情况，从结果看出本方法明显优于 SRGD 方法。这是由于本方法将计算量大的索引构建结果预先保存在本地和云端，避免了重复大量的重复计算，在提取基和偏移量时直接查表即可，而 SRGD 方法对于每个文件均需要重新进行计算，计算复杂度较大。图 6(b)和(c)分别展示了每个文件所提取基大小和偏移量大小的对比情况，从结果

看出本方法所需存储开销较小，说明所提取基具有更强的泛化能力，且偏移量数据间存在较多冗余，通过压缩可以提高存储效率。

5.2 去重效率验证

在本实验中，我们比较本方法与 ZEUS、RARE 和 CIDER 这 3 种最新的跨用户去重方案对相似文件去重的性能。具体地，我们从 Enron 邮件数据集中随机选取了大小为 14~26KB 的 1000 个文件，数据总量为 19135KB。从 Sakila 样本数据集中选取支付数据记录生成另一个长度为 2241B 的文件开展实验。在本实验中，假设这两个文件均在云端存在，分别替换部分数据内容进行上传，测试上传过程中的通信开销和存储开销。具体地，由于 Enron 邮件数据集文件较大，我们分别替换掉 10%、15%、20%、25%、30%随机选定的数据内容作为敏感信息，生成对应的相似文件，分别对处理后的文件生成对应的去重请求，并根据 4 种方法生成的响应上传数据后比较所需的通信开销和存储开销。

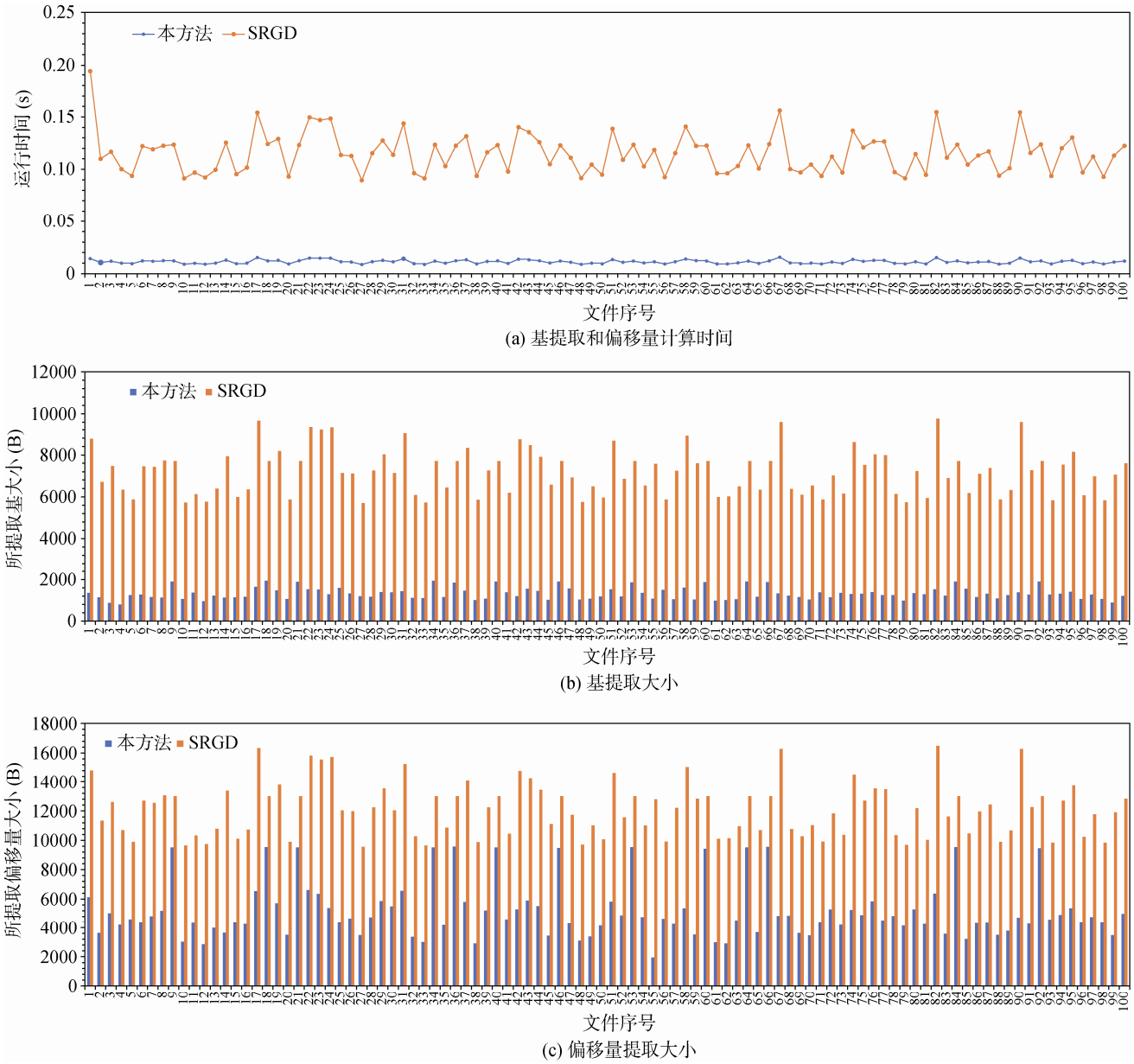


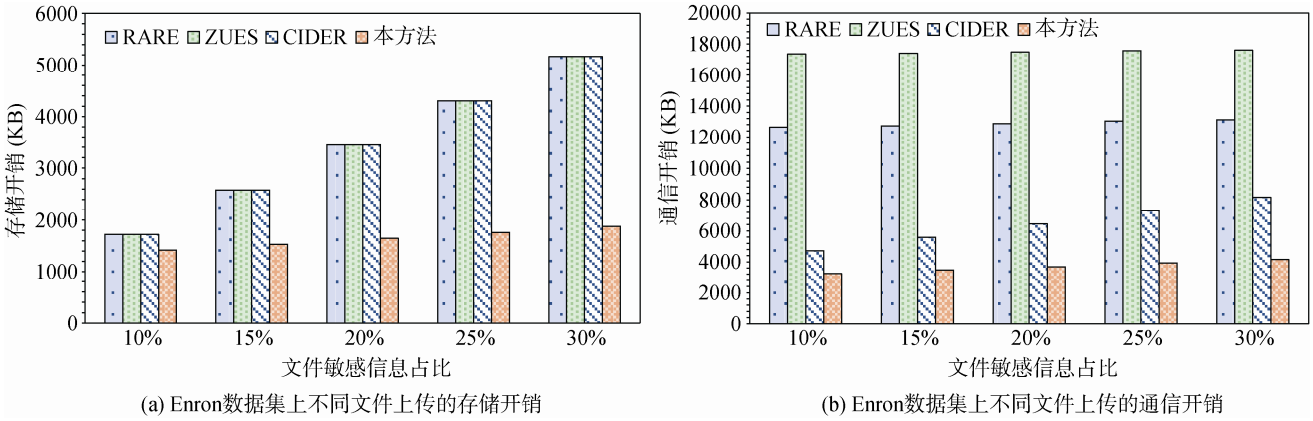
图 6 计算性能分析实验结果

Figure 6 Computational performance analysis experiment results

图 7(a)和图 7(b)分别展示了 Enron 数据集上的存储开销和通信开销实验结果。对于所选用的 Sakila 样本数据集，因其数据规模较小，数据模版特征较为清晰，我们分别替换掉 10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90%, 100%的数据作为敏感信息，对这些信息篡改替换后生成去重请求发送给云端。图

8(a)和图 8(b)分别展示了 Sakila 数据集上的存储开销和通信开销实验结果。如图所示, 在两个数据集中, 随着文件敏感信息占比的上升, 4 个比较方案的通信开销和存储开销均逐步上升, 但本方法所需开销增长趋势较为缓慢。主要原因在于 CIDER, ZEUS, RARE 都是块级数据去重方案, CIDER 的返回值随机在区间[未命中块数, 未命中块数+1]中随机选取, ZEUS 和 RARE 方案均对请求中的相邻块配对生成响应, 当其中至少一个块命中时, ZEUS 方案在响应中要求上传的块数为 1, 而 RARE 为 1 或 2, 因此 CIDER 的开销小于 ZEUS 和 RARE, RARE 的开销最大。而本方法仅对基开展跨用户去重, 对偏移量则开展云端去重, 所以在不同的敏感信息占比下, 需要的通信开销主要来源为偏移量数据和基数据哈希值上传, 存储开销主要来源为敏感信息的偏移量数据, 因此增加趋势较为平缓。与其他 3 种算法相比, 本方法字节级的去重方式, 可以降低敏感信息中基的存储开销, 且对偏移量数据压缩后可进一步降低开销, 因此, 在存储开销和通信开销结果中均表现出显著的优势。

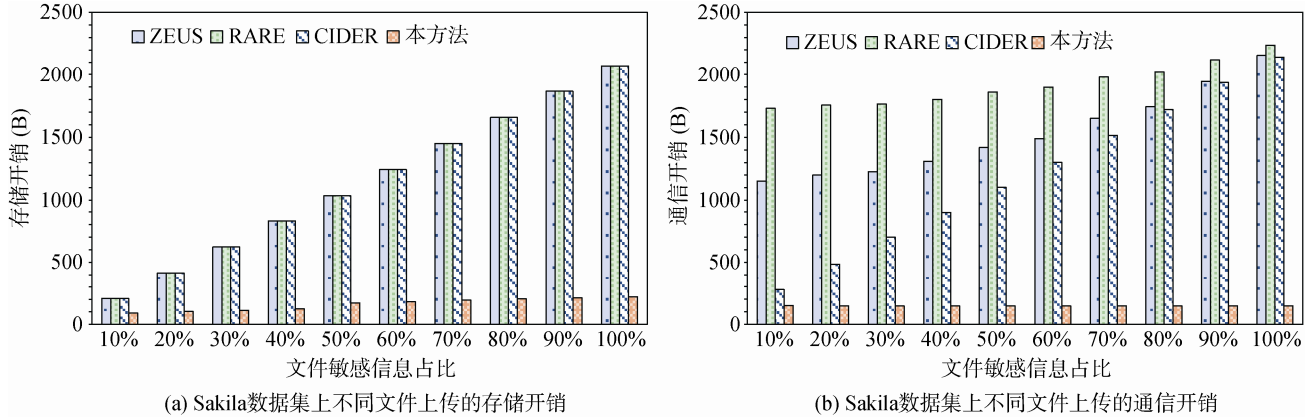
具体的, 在 Enron 邮件数据集中, 随着敏感信息占比从 10%增加到 30%, 本文所提方法的存储开销从 1417KB 增加至 1878KB, 低于 ZEUS、RARE、CIDER 3 个方案的存储开销, 同时通信开销从 3006KB 增加至 4155KB, 也低于 ZEUS、RARE、CIDER 三个方案的通信开销。本方法在 Sakila 样本数据集中表现出更优的去重效率, 相较于 ZEUS、RARE、CIDER 3 个方案, 存储开销可节约 50%至 90%。当敏感信息占比的较大时, 表现出更为显著的优势, 这是因为部分所提取的基已经存在于云端且具有较强的泛化能力, 即使敏感信息占比增大, 仍然可以实现数据的有效去重。而 ZEUS、RARE、CIDER 3 个算法均为块级去重策略, 无法从字节级别提取相似的文件内容。对于图 7 和图 8 实验结果, 发现本方法在 Sakila 样本数据集上具有更加显著的优势。这是因为 Sakila 样本数据集中数据内容更加规范, 所提取出的基之间重复度更高, 表明本方法更适合用于类似数据库等数据模版特征更为清晰的文件。



(a) Enron数据集上不同文件上传的存储开销　　(b) Enron数据集上不同文件上传的通信开销

图 7　Enron 邮件数据集去重效率验证实验结果

Figure 7　The experimental results of the deduplication efficiency verification on the Enron dataset



(a) Sakila数据集上不同文件上传的存储开销　　(b) Sakila数据集上不同文件上传的通信开销

图 8　Sakila 样本数据集去重效率验证实验结果

Figure 8　The experimental results of the deduplication efficiency verification on the Sakila Sample dataset

6 小结

本文提出了一种通用的跨用户重复数据安全去重方案, 可以有效的保护云端文件或块的存在性隐私在边信道攻击下的安全性。更进一步的, 本文所提出的基提取策略, 能够从相似的文件或块中提取以较高概率提取出相同模板, 具有良好的泛化能力。除此之外, 本文对偏移量在上传前进行压缩处理, 能够有效降低偏移量上传的通信开销。通过在两个真实数据集中的实验结果表明, 与现有技术相比, 本文所提方法具有更高效的去重效率和存储效率。由于本文所提方法中数据分块方法为固定大小, 可能存在边界平移问题, 后续我们将在数据分块方法上进行进一步的优化。

参考文献

[1] Meyer D T, Bolosky W J. A Study of Practical Deduplication[J]. *ACM Transactions on Storage*, 2012, 7(4): 14.

[2] Tang X, Zhou L N, Shan W J, 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.
(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.)

[3] Vestergaard R, Zhang Q, Lucani D E, et al. Enabling random access in universal compressors[C]. *IEEE INFOCOM 2021 - IEEE Conference on Computer Communications Workshops*, 2021: 1-6.

[4] 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.

[5] Lee S, Choi D, Processing C A, et al. Privacy-preserving cross-user source-based data deduplication in cloud storage[C]. *2012 International Conference on ICT Convergence*, 2012: 329-330.

[6] Wang B, Lou W J, Hou Y T, et al. Modeling the side-channel attacks in data deduplication with game theory[C]. *2015 IEEE Conference on Communications and Network Security*, 2015: 200-208.

[7] Zuo P F, Hua Y, Wang C, et al. Mitigating traffic-based side channel attacks in bandwidth-efficient cloud storage[C]. *2018 IEEE International Parallel and Distributed Processing Symposium*, 2018: 1153-1162.

[8] [8] Tang X, Zhang Y, Zhou L N, et al. Request Merging Based Cross-User Deduplication for Cloud Storage with Resistance Against Appending Chunks Attack[J]. *Chinese Journal of Electronics*, 2021, 30(2): 199-209.

[9] Yu C M, Gochhayat S P, Conti M, et al. Privacy Aware Data Deduplication for Side Channel in Cloud Storage[J]. *IEEE Transactions on Cloud Computing*, 2020, 8(2): 597-609.

[10] Pooranian Z, Chen K C, Yu C M, et al. RARE: Defeating side channels based on data-deduplication in cloud storage[C]. *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops*, 2018: 444-449.

[11] Vestergaard R, Zhang Q, Lucani D E, et al. CIDER: A low overhead approach to privacy aware client-side deduplication[C]. *GLOBECOM 2020 - 2020 IEEE Global Communications Conference*, 2021: 1-6.

[12] Vestergaard R, Zhang Q, Lucani D E, et al. Lossless compression of time series data with generalized deduplication[C]. *2019 IEEE Global Communications Conference*, 2020: 1-6.

[13] Tang X, Liu Z, Shao Y, et al. Side channel attack resistant cross-user generalized deduplication for cloud storage[C]. *ICC 2022 - IEEE International Conference on Communications*, 2022: 998-1003.

[14] Dang H, Chang E C, Communication N A B T, et al. Privacy-preserving data deduplication on trusted processors[C]. *2017 IEEE 10th International Conference on Cloud Computing*, 2017: 66-73.

[15] Chen R M, Mu Y, Yang G M, et al. BL-MLE: Block-Level Message-Locked Encryption for Secure Large File Deduplication[J]. *IEEE Transactions on Information Forensics and Security*, 2015, 10(12): 2643-2652.

[16] Tang X, Zhou L N, Huang Y F, et al. Efficient cross-user deduplication of encrypted data through re-encryption[C]. *2018 17th IEEE International Conference on Trust, Security and Privacy In Computing and Communications/ 12th IEEE International Conference on Big Data Science and Engineering*, 2018: 897-904.

[17] Vestergaard R, Zhang Q, Lucani D E, et al. Generalized deduplication: bounds, convergence, and asymptotic properties[C]. *2019 IEEE Global Communications Conference*, 2020: 1-6.

[18] Liu S, Tjuawinata I. On 2-Dimensional Insertion-Deletion Reed-Solomon Codes with Optimal Asymptotic Error-Correcting Capability[J]. *Finite Fields and Their Applications*, 2021, 73: 101841.

[19] Galindo C, Hernando F, Ruano D. Entanglement-Assisted Quantum Error-Correcting Codes from RS Codes and BCH Codes with Extension Degree 2[J].*Quantum Information Processing*, 2021, 20(5): 1-26.

[20] Luo C, Cui Y, Lin Y S. Container Migration Method Based on Bandwidth Prediction and Adaptive Compression[J]. *Computer Engineering*, 2022, 48(5): 200-207, 214.
(罗成, 崔勇, 林予松. 基于带宽预测与自适应压缩的容器迁移方法[J]. *计算机工程*, 2022, 48(5): 200-207, 214.)

[21] Enron email dataset, https://www.cs.cmu.edu/～enron/, 2015.

[22] Sakila sample database, https://dev.mysql.com/doc/sakila/en/, 2021.

刘小梅 于 2018 年在中国科学院大学计算机技术专业获得工学硕士学位。现任国际关系学院网络空间安全学院实验员。研究领域为信息隐藏、云计算安全。研究兴趣包括: 信息隐藏、云计算安全、人工智能。Email: liuxiaomei@uir.edu.cn

杨舒婷 于 2022 年在国际关系学院网络空间安全专业获得工学学士学位。研究领域为: 网络安全。研究兴趣包括: 云数据安全。Email: yangshuting1234@126.com

高语灿 于国际关系学院数据科学与大数据技术专业攻读本科学位。研究领域为云数据安全。研究兴趣包括云数据安全、信息隐藏。Email: gaoyucan@uir.edu.cn

唐鑫 于 2015 年在北京邮电大学计算机科学与技术专业获得工学博士学位, 2015—2017 年在清华大学电子工程系从事博士后研究。现任国际关系学院网络空间安全学院副教授。研究领域为数字内容安全、信息隐藏、云数据安全。研究兴趣包括: 云存储安全、信息隐藏。Email: xtang@uir.edu.cn

陈雄 于 2021 年在河南工业大学物联网工程专业获得学士学位。现在国际关系学院电子信息专业攻读硕士学位。研究领域为云数据安全。研究兴趣包括云数据安全、信息隐藏。Email: xchen@uir.edu.cn