基于自适应MSB可逆信息隐藏的图像云数据密文安全去重机制

周艺腾，唐鑫，金路超

引用本文

周艺腾，唐鑫，金路超. 基于自适应MSB可逆信息隐藏的图像云数据密文安全去重机制[J]. 计算机科学, 2024, 51(12): 352-360.

ZHOU Yiteng, TANG Xin, JIN Luchao. Adaptive MSB Reversible Data Hiding Based Security Deduplication for Encrypted Images in Cloud Storage [J]. Computer Science, 2024, 51(12): 352-360.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.231100087

基于自适应 MSB 可逆信息隐藏的图像云数据密文安全去重机制

周艺腾　唐　鑫　金路超
国际关系学院网络空间安全学院　北京 100091
(ytzhou@uir.edu.cn)

摘　要　随着信息技术的飞速发展,越来越多以图像为代表的多媒体数据被重复上传到云平台进行存储,造成了用户通信开销和云端存储开销的极大浪费。此外,明文状态的图像数据存储在云端,导致数据机密性被破坏。尽管密文图像云数据去重技术在一定程度上解决了以上问题,但去重过程中产生的可区分响应为攻击者创建了一个侧信道,将泄露用户数据的存在性隐私。同时,为实现加密密钥在数据持有者间的传递,用户和云均需要付出巨大的额外代价。鉴于此,提出了一种基于自适应 MSB 可逆信息隐藏的高效密文图像安全去重机制,其能够在有效抵抗侧信道攻击的同时实现较低的通信开销和存储开销。具体来说,创新性地将密文域可逆信息隐藏技术引入密文去重框架,将用于传递随机密钥的辅助信息嵌入加密图像中并发送给云,从而消除辅助信息的传输和存储开销。此外,优化了现有的去重方案,即使请求图像并未存储于云端,用户也无需开展额外的密文上传工作,从而保证响应的不可区分性。安全性分析和实验结果表明,与现有方案相比,该方案能够以轻量级的方式抵抗侧信道攻击。

关键词:云存储;加密图像去重;侧信道攻击;自适应 MSB 预测;信息隐藏
中图分类号　TP309

Adaptive MSB Reversible Data Hiding Based Security Deduplication for Encrypted Images in Cloud Storage

ZHOU Yiteng, TANG Xin and JIN Luchao
School of Cyber Science and Engineering, University of International Relations, Beijing 100091, China

Abstract　With the rapid development of information technologies, more and more multimedia data represented by images are repeatedly uploaded to the cloud for storage, resulting in a great waste of communication and storage overhead. In addition, the plaintext images are directly stored in the cloud, which brings about the problem of confidentiality breach. Even though ciphertext deduplication is an effective means to deal with these problems, the differentiated response actually creates a side channel for attackers, which makes the existence privacy of data in cloud storage at risk. At the same time, in order to achieve key transferring between data owners, a huge amount of extra overhead is required. Thus, this paper proposes an efficient adaptive MSB reversible data hiding based secure deduplication (EMSD), which is able to effectively resist side channel attacks and save communication and storage overhead. Specifically, we innovatively introduce the reversible data hiding for encrypted images into ciphertext deduplication, and embed the auxiliary information for key transferring into the encrypted images before sending to the cloud. Thus the extra communication and storage overhead for auxiliary information are successfully eliminated. Furthermore, we optimize the existing deduplication scheme to ensure that even if the image in deduplication request is not duplicate, extra ciphertext uploading is not needed, thus indistinguishable response is achieved. Security analysis and experimental results show that, the proposed scheme is able to resist side channel attack in a lightweight way comparing with existing schemes.

Keywords　Cloud storage, Deduplication for encrypted images, Side channel attack, Adaptive MSB prediction, Data hiding

1　引言

随着云存储服务的迅猛发展,以及智能手机、摄影摄像等移动终端和社交媒体的普及,越来越多的用户开始将图像、音视频等多媒体数据发送至云平台进行存储。以图像数据为例,相较于文本数据而言,其本身具有更大的尺寸和更高的重复度,这导致云端存储数据呈现指数级的增长趋势[1]。IDC 最新报告显示,到 2025 年,全球产生的数据总量将达到

到稿日期:2023-11-15　返修日期:2024-04-02
基金项目:国家自然科学基金青年科学基金(62102113);国际关系学院中央高校基本科研业务费项目(3262023T33)
This work was supported by the Young Scientists Fund of the National Natural Science Foundation of China (62102113) and Fundamental Research Funds for the Central Universities, University of International Relations (3262023T33).
通信作者:唐鑫(xtang@uir.edu.cn)

175 ZB，其中近 75％的数据至少拥有一个冗余副本[2]。显然，数字图像等多媒体数据的重复存储对云服务提供商（Cloud Service Provider，CSP）的存储资源造成了极大的浪费。同时，由于需要反复上传相同的文件，用户也面临着巨大的通信开销浪费。

为解决这一问题，以数字图像为例，图像云数据去重技术[3-9]应运而生。在去重方案中，CSP 可以与用户合作，首先检查上传的图像是否已经存储在本地服务器，如果已经存在，则 CSP 只需为当前用户建立一个与所存图像对应的链接关系，而不需要在云上存储另一个副本；否则，要求用户上传完整数据。然而，图像去重技术在节省开销的同时伴随着潜在的安全威胁。一方面，CSP 返回给用户的可区分响应为外部攻击者创建了一个侧信道[10-13]，这将导致攻击者可以轻易地获取目标图像的存在性隐私。因为一旦响应没有要求上传完整的数据，攻击者便能推断出请求去重的图像已经存储于云端。进一步，攻击者可以针对目标图像中的低最小熵敏感信息发起暴力字典攻击，生成目标图像的所有可能版本，并依次发起去重请求，然后根据响应值最终确定目标图像的内容。另一方面，明文状态的图像数据存放在云端，面临被窃取的风险。加密是解决这一问题的有效手段。然而，由于加密密钥的差异，完全相同但来自不同用户的图像将会被加密为不同的密文，这会导致图像密文无法去重，从而带来额外的冗余存储和通信开销。

针对上述安全问题，现有的经典解决方案是利用消息锁定加密（Message-Locked Encryption，MLE）[14]解决图像机密性保护和密文去重之间的矛盾。这种方法使用图像的内容相关密钥来代替用户相关密钥以实现数据加密，保证即使数据来自不同用户，相同明文也能得到相同密文。其中最突出的实例是聚合加密（Convergent Encryption，CE）[15]，它选择图像的哈希值作为加密密钥，实现去重。遗憾的是，在 MLE 方法中，内外部攻击者仍然能够通过侧信道攻击，利用可区分响应窃取低熵图像的隐私数据。2020 年，Pooranian 等[16]提出了一种针对一般数据的，能够同时抵抗安全风险并实现密文去重的混合方案 LEVER。具体地，LEVER 使用短哈希代替哈希值进行重复检查，在一定程度上实现混淆的去重响应。这是因为所选的短哈希具有较强的碰撞性，同一短哈希值可对应多个不同的文件，即使响应值表明请求的短哈希存在，也并不意味着攻击者感兴趣的目标文件存在。此外，利用同态加密，LEVER 实现了随机选定的加密密钥（非 MLE 密钥）在数据所有者之间的秘密传递。实现原理是初始用户将用户密文、用于传递密钥的同态密文等辅助信息一起发送给云端进行存储，拥有相同文件的后续用户能够根据这个辅助信息获取初始用户使用的随机密钥，从而使用该密钥加密数据，实现去重。尽管 LEVER 面向一般数据去重，我们依然可以借鉴这一思路解决本文聚焦的图像云数据去重中的安全问题。然而，该方法实际上无法保护用户数据的存在性隐私。这是因为一旦用户请求去重的图像已经存储于云端，CSP 仅会在响应中返回辅助信息。否则，CSP 除了发送辅助信息，还会要求用户重新选择一个随机密钥，并将新的用户密文和辅助信息上传到云端。这种可区分的响应很明显地表明了攻击者请求去重的文件在云端的存在状态。此外，为了实现密钥传递，用户上传新文件时势必要额外传送辅助信息，增加了大量通信开销。与此同时，云端也要存储这些辅助信息，需要付出额外的存储开销。

鉴于此，本文提出了一种基于密文域可逆信息隐藏的改进去重方案 EMSD，其能够在有效抵抗侧信道攻击的同时实现较低的通信开销和存储开销。据我们所知，EMSD 是第一个将密文域可逆信息隐藏技术应用到图像云数据安全去重的方案。具体地，利用基于自适应 MSB（Most Significant Bit）预测的大容量密文域可逆信息隐藏技术[17]，将 LEVER 中的辅助信息隐写到图像密文中。在该方法的帮助下，用户只需要将携密密文发送给云端，CSP 就能在只存储携密密文的前提下通过信息提取完成密钥传递，极大地提高了方案效率，节省了用户端通信开销和云端存储开销。此外，简化交互过程，使得不论请求去重的加密图像在云端是否存在，云端都可生成无差异的响应，从而保证改进方案抗侧信道攻击的安全性。这是因为如果请求去重的图像并未存储在云端，用户在解密辅助信息时会得到一个随机比特串而非其他用户的随机密钥。将这个随机比特串作为当前图像的加密密钥，CSP 就无需要求用户重新选择一个随机密钥，并进行额外的用户密文和辅助信息的上传，以此确保响应和图像已存储在云端时相同。本文的主要贡献如下：

1）提出了一种支持密文域可逆信息隐藏的加密图像安全去重框架，在保证抗侧信道攻击安全性的同时极大降低了用户端的通信开销和云端的存储开销。在该框架下，用户能够把辅助信息嵌入到加密图像中，只需将携密图像发送给云端进行存储。当有后续用户请求去重时，CSP 可以利用信息隐藏的可逆性提取出携密图像中嵌入的辅助信息，以实现随机密钥在数据持有者之间的传递。

2）在加密图像去重框架的基础上，设计了一种抗侧信道攻击的图像安全去重机制。通过简化交互流程，即使请求文件并未存储于云端，用户也无需进行额外的密文上传，CSP 会在用户之前上传的携密图像中随机选择一个作为新文件存储，从而保证响应的不可区分性。这种新的混淆策略能够实现抗侧信道攻击的安全性。

3）对 EMSD 进行了安全性分析，并在真实数据集上进行了性能测试。理论分析和实验结果都表明，EMSD 能够以轻量级的方式在侧信道攻击下实现安全性。

2 相关工作

2.1 云数据去重

针对图像的去重技术，根据去重效果的不同可分成精确图像去重[3]和模糊图像去重两类。Agarwala 等[4]提出了一种基于双完整性收敛加密技术（Dual-Integrity Convergent Encryption，DICE）的近似图像去重方案。该方案将图像划分为独立块，对每个图像块应用 DICE 协议以提高去重效率。然而，DICE 协议依然使用哈希算法进行重复检测，只能处理相同的图像块，且容易受到暴力字典攻击。随后，Li 等[5]提出了第一个模糊图像去重方案，其使用感知哈希算法生成每幅图像的签名，并通过比较感知哈希之间的汉明距离来确定

重复图像。然而，该方案需要定期更换组密钥，计算较为复杂。此外，攻击者只要拥有较短的摘要信息，就能在该去重系统中获取目标文件的访问权限，而不需要证明具有该文件的所有权。随后，文献[6-8]提出了一系列基于认证的图像模糊去重方案。然而，攻击者可以通过在重复数据删除过程中监控网络流量来推断敏感数据内容。

针对一般形式的云数据去重，为抵抗侧信道攻击，有研究者提出了几种明文数据去重模型[18-19]，旨在通过混淆弱化不同情况下响应的可区分性。然而这些工作均无法保证数据在传输和云端存储过程中的机密性，因此密文数据去重得到了广泛研究。Pooranian 等[16]提出了一种基于随机密钥的密文去重方案 LEVER，其通过短哈希抵抗侧信道攻击，同时利用同态加密的密文特性实现随机密钥在数据持有者之间的传递。具体来说，用户用短哈希 $sh(f)$ 代替明文数据 f 的哈希值 $h(f)$ 发送给云端进行去重检查。其中，$sh(f)$ 可以由 $h(f)$ 的部分比特组成，因此，$sh(f)$ 具有较高的冲突率，即相同的 $sh(f)$ 可能对应不同的明文数据，攻击者不能通过 $sh(f)$ 是否重复来推断目标文件的云端存在性，从而降低了侧信道攻击的风险。此外，LEVER 允许初始用户在上传前采用随机密钥 k_f 对原始数据进行加密。为实现后续去重且保证数据机密性，k_f 必须加密后存储在云端。因此，LEVER 利用了同态加密对密文的处理可以同步传导到明文上这一特性，对 k_f 进行加密。具体地，LEVER 设计了一个依赖明文的随机密钥提取函数 $g(x,y)$，其中 x 为对 k_f 加密的辅助信息 $\Delta=\varepsilon(k_f \ominus h(f)) \| E_{\varepsilon(k_f)}(E_{h(f)}(k_f))$，$y$ 为原始明文数据 f，$\varepsilon(\cdot)$ 表示 Paillier 同态加密，$\ominus$ 表示加性同态减法。如果函数 $g(x,y)$ 的输入为正确的明文数据 f，则输出为随机密钥 k_f；否则为随机比特串 $bitstring$。利用这一特性，如果用户上传的 $sh(f)$ 在云中并未存储，用户会自主选择一个随机密钥 k_f，并计算密文和辅助信息以发送给云端进行存储。否则，CSP 会将存储中对应同一个短哈希的所有辅助信息发送给用户，用户利用持有的明文数据和辅助信息解密得到一系列密钥，并通过这些密钥重新加密得到一系列密文后发送给云端检查是否匹配。如果用户请求的文件已经存储于云端，解密出的密钥中会存在一个真正的 k_f 和其他随机比特串，同理发送的密文中也包含一个已经存储的密文和其他未存储密文，于是 CSP 在检查到匹配的密文后会丢弃用户上传的内容以实现去重。而如果 CSP 并未检查到匹配，这表明请求文件并未存储于云端，于是返回响应要求用户重新选择一个新的随机密钥，并将对应的辅助信息和密文上传到云中进行存储。然而，这种可区分的响应无疑会在侧信道攻击中泄露敏感数据的存在性隐私，同时，辅助信息的传输和存储也会增加用户的通信开销和云端存储开销。如何在实现去重方案安全性的同时提升效率，是本文方法 EMSD 改进的重点。

2.2 密文域可逆信息隐藏

信息隐藏[20-21]是一种在不占用额外信道的情况下将一些附加数据嵌入到载体介质中的技术。而密文域可逆信息隐藏 RDHEI(Reversible Data Hiding in Encrypted Image)通常包括图像加密和数据隐藏。根据数据提取是否独立于图像解密，现有的方法可分为联合 RDHEI 和分离 RDHEI。Zhang[22]提出了首个有效的 RDHEI 方法，首先将加密图像分成互不重叠的块，然后通过翻转块内一半像素的 3 个最低有效位(Least Significant Bit，LSB)来完成数据嵌入。在解码端，利用翻转函数识别翻转部分，进行数据提取和图像恢复。随后，Hong 等[23]引入边匹配技术和一种新的波动函数，简化了文献[22]中的提取方法，降低了提取错误率。为了获取更大的嵌入容量，Wang 等[17]提出了一种基于 MSB 预测的 RDHEI，这也是本文引入的方法。具体地，该方法首先将图像分成 2×2 大小的非重叠像素块，进行块级加密，以保留块内像素的相关性。加密像素块的结构如图 1 所示。

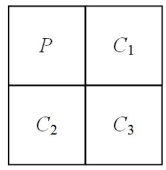


图 1 块内像素结构

Fig. 1 Pixel structure in a block

每个像素块由 P，C_1，C_2 和 C_3 4 个像素组成，通过压缩 C_1，C_2 和 C_3 像素的相同 MSB 位可以腾出嵌入空间。压缩过程中，首先将所有像素分解成 8 个比特，然后通过式(1)计算变量 d_1，d_2 和 d_3。

$$d_i=dif(P,C_i),i=1,2,3 \tag{1}$$

其中，$dif(P,C_i)$ 返回 P 和 C_i 最大的不相同 LSB 位数。例如，$dif(195,160)=7$，因为 $195=(11000011)_2$，$160=(10100000)_2$。接下来，计算 $md=8-\max(d_1,d_2,d_3)$，即 4 个像素共享的 MSB 的最小位数，可用 3 比特表示。随后，将块内所有像素总共 32 位进行重构，如图 2 所示。像素 P 占据前 8 位，变量 md 占据随后 3 位，预测误差 e_1，e_2 和 e_3(即 C_1，C_2 和 C_3 中除去公共 MSB 位后的剩余位)占据 $3\times(8-md)$ 位，n_c 表示可由待嵌入信息替换的剩余 $3\times(md-1)$ 位。由此，实现大容量密文域信息嵌入。

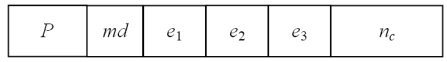


图 2 像素位重构

Fig. 2 Reconstruction of pixel

3 准备工作

3.1 系统模型

如图 3 所示，EMSD 的系统模型由用户和 CSP 两个实体组成。

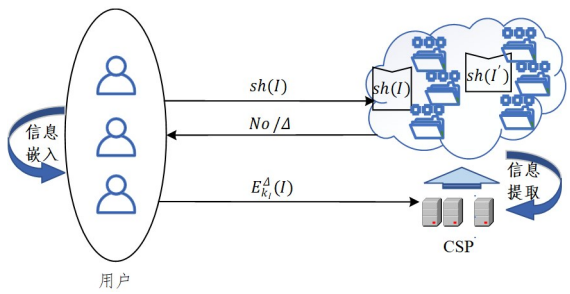


图 3 系统模型

Fig. 3 System model

用户:想要外包自己的图像数据以减轻本地存储负担的实体。具体地,用户在上传真正的图像数据 I 之前,先计算文件的短哈希 $sh(I)$作为去重请求来查询 I 在云端的存在状态。随后,根据去重响应,如果 $sh(I)$未命中,用户生成一个随机密钥k_I,并加密图像 I 生成密文图像$E_{k_I}(I)$。同时,为了保证将随机密钥传递给后续用户实现去重,用户还会计算辅助信息 $\Delta=\varepsilon(k_I\ominus h(I))\parallel E_{\varepsilon(k_I)}(E_{h(I)}(k_I))$,并利用 MSB 预测算法将其嵌入到$E_{k_I}(I)$中获取最终发送给云的携密图像 $H=E_{k_I}^{\Delta}(I)$。若 $sh(I)$命中,用户解密响应中的辅助信息得到潜在随机密钥,并利用这一解密密钥得到新的携密图像以发送给云。

CSP:为用户提供存储服务的实体。为了消除冗余存储并降低管理成本,同一数据在云中只存储一次。具体地,CSP 维护一个文件列表,列表中存储短哈希和对应的携密图像。在接收到图像 I 的短哈希 $sh(I)$后,CSP 将其与列表中的短哈希进行比较。如果 $sh(I)$未命中,表明图像 I 并未存储,CSP 返回响应“No”。否则,CSP 首先对该短哈希对应的所有携密图像进行信息提取,得到辅助信息,并以此作为响应返回给用户。此外,在接收到用户上传的携密图像后,如果 $sh(I)$未命中,CSP 直接将携密图像和短哈希存储在列表里。否则,将上传内容和列表中存储的携密图像进行比较,如果匹配,则丢弃所有上传内容;若不匹配,CSP 在上传内容中任选其一作为新一条记录,并将其存储在短哈希 $sh(I)$对应的列表里。

3.2 威胁模型

我们考虑合法但恶意的用户(外部攻击者)和诚实但好奇的云服务提供商(内部攻击者),他们企图获取正常用户的图像 I 或随机密钥k_I。

对于外部攻击者,其可以发动侧信道攻击,即伪装成合法用户发起去重请求,查询目标图像的云端存在性。他们通过观察 CSP 返回的不同响应,窃取目标图像的存在性隐私。此外,考虑到目标图像具有低最小熵的可预测性质,攻击者还可以发动暴力字典攻击,构造目标图像所有可能版本,查看哪个版本被 CSP 响应并阻止后续上传。

对于内部攻击者,即使目标图像可预测,也由于随机密钥具有高随机度,恶意云无法通过预测目标图像的方式获取随机密钥。因此,CSP 企图查看文件列表,通过短哈希和辅助信息进行解密运算,得到随机密钥k_I。一旦获取了k_I,CSP 就能通过解密密文得到用户原始图像,从而窃取隐私。

3.3 设计目标

考虑到上述系统模型和威胁模型,本文所提方案应该满足以下设计目标,以实现高效且安全的图像密文重复数据删除。

1)抗侧信道攻击的安全性:确保内外部攻击者无法通过预测目标图像的内容和分析重复数据删除响应窃取目标数据的存在性隐私。

2)基于密文域信息隐藏去重框架的可行性:确保密文域信息隐藏的可逆性和可分离性,即可以无损地提取出嵌入的辅助信息,但无法同时获取明文图像。

3)高效且轻量的去重流程:确保用户端的通信开销和云端的存储开销足够小。具体来说,云中只需存储相同图像的单一副本,无需存储额外的辅助信息。同时,消除用户与云的非必要上传内容与交互过程。

4 方案设计

4.1 算法框架

为了尽可能减少用户端通信开销和云端存储开销,EMSD 创造性地利用信息隐藏技术将一些附加信息嵌入到载体介质中而不占用额外存储资源的特性,将密文域可逆信息隐藏方案 MSB 预测算法[17]引入传统的图像密文去重框架,在保证密文去重安全性和可行性的基础上,减少用户发送给云以及云端存储的必要数据。具体来说,如图 4 所示,本文方案在 LEVER[16]的基础上,将去重框架划分成重复性检查、响应生成、图像处理和数据存储 4 个模块。其中,在图像处理阶段借助信息嵌入技术生成携密图像,在响应生成阶段引入信息提取技术得到返回给用户的响应。

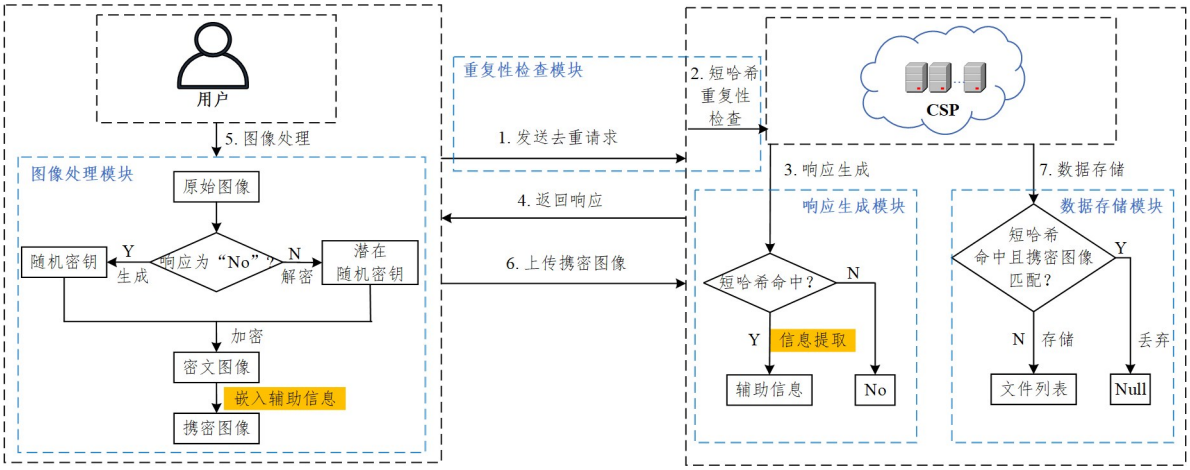


图 4 算法框架

Fig. 4 Algorithm framework

在重复性检查模块,用户生成明文图像的短哈希并发送给云端请求去重。由于短哈希具有较高的碰撞性,因此可实现对暴力字典攻击的抵抗。云服务器维护一个文件列表,列表里存储用户之前上传的携密图像、短哈希以及它们的

对应关系。于是，CSP 检查接收到的短哈希是否已经存储于列表，并将检查结果发送给响应生成模块。

在响应生成模块，CSP 根据重复性检查模块得到的检查结果生成不同响应。具体来说，如果短哈希命中，则对列表中该短哈希对应的所有携密图像进行信息提取，将提取到的辅助信息作为响应返回给用户；否则，响应值为“No”。借助信息隐藏的思想，CSP 并不需要在文件列表中额外存储辅助信息而浪费存储空间，也可以随时调用对实现密文去重至关重要的辅助信息以保证随机密钥可以在同一图像数据的不同持有者之间传递。

在图像处理模块，利用图像信息隐藏与图像云数据去重之间的共性，即减少冗余，设计双方相融合的全新图像处理方式。用户采取这种处理方式，根据响应生成模块返回的不同响应内容，最终得到携密图像并上传到云上。具体来说，如果用户收到响应“No”，就会自己生成一个高最小熵的随机密钥，否则便会分别解密响应中的所有辅助信息，获得一系列潜在的随机密钥。此外，由于 2.2 节提到的函数 $g(x,y)$ 的设计，用户不能区分解密出来的内容是真正的随机密钥还是随机比特串，因此无法通过响应判断请求图像的云端存在状态。随后，用户利用自己生成（解密得到）的潜在随机密钥来加密原始图像，并利用 MSB 预测方法将辅助信息嵌入密文图像中，最终将携密图像上传到云上。

在数据存储模块，在短哈希值未命中的情况下，CSP 将接收到的携密图像和短哈希一起存储在列表中。而在短哈希值命中的情况下，CSP 检查接收到的携密图像与列表中已有携密图像是否匹配。如果匹配，表明该图像已经存储在云端，将本次上传内容丢弃以实现去重；否则，意味着即使该请求图像的短哈希值是命中的，图像内容也并未存储在列表中。此时，LEVER 方案需要返回第二次响应要求，用户重新生成密钥并上传密文，导致了不同情况下的可区分响应，这为攻击者发动侧信道攻击以窃取目标图像的存在性隐私提供了便利。因此，所提方案 EMSD 在这一场景下设计了一种全新的处理方式，以实现抗侧信道攻击的安全性。具体来说，EMSD 简化交互流程，CSP 不会返回任何暗示用户携密图像的匹配结果的响应，而是直接在接收到的携密图像中随机选择一个存储在列表。这样处理既实现了新图像密文的云端存储，又在不经意间为该用户指定了对应新图像的随机密钥（即用户解密得到的随机比特串）。

4.2 短哈希未命中时的去重流程

当用户 1 第一次上传新图像 I 时，会首先计算 I 的短哈希，并将其作为去重请求发送给云。具体来说，用户在计算得到 I 的全长哈希值的基础上，采用特定过滤器提取全长哈希二进制表示中某些位置的比特来构成具有高碰撞率的短哈希 $sh(I)$。如图 5 所示，CSP 在文件列表中没有找到相同的 $sh(I)$。因此，用户 1 知道他是第一个上传 I 的人，随即生成一个文件 I 的随机密钥，记为 k_I。随后，用户计算用于传递随机密钥的辅助信息 $\Delta=\varepsilon(k_I\ominus h(I))\parallel E_{\varepsilon(k_I)}(E_{h(I)}(k_I))$ 以及携密图像 $H=E_{k_I}^{\Delta}(I)$ 并发到云上。CSP 接收到携密图像 H 后，将其与对应的短哈希 $sh(I)$ 一起存储在文件列表中。

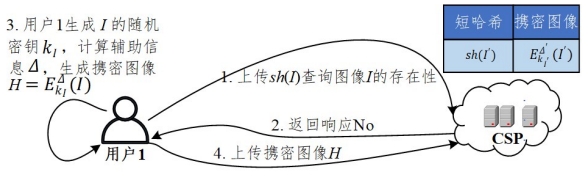


图 5 文件 I 的第一次上传

Fig. 5 First upload of file I

4.2.1 图像加密

如图 6 所示，在 EMSD 的加密算法中，用户输入明文图像 I、随机密钥 k_I，采用经典流密码加密方案 RC4，得到输出 $E_{k_I}(I)$。

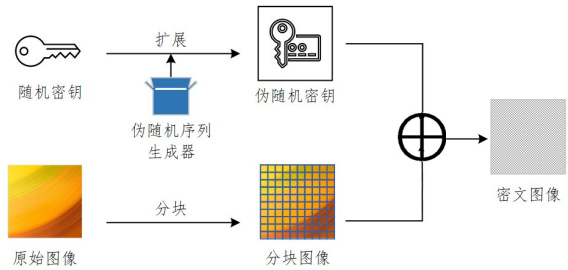


图 6 图像加密流程

Fig. 6 Process of image encryption

类似于文献[17]方案，EMSD 采用块级流密码加密算法实现图像加密。首先，用户将 $M\times N$ 的图像划分成 2×2 大小的非重叠块，并把块内所有像素分解成 8 位二进制比特。由于流密码采用和明文等长的流密钥对明文数据开展逐比特异或，因此在加密前用户会首先利用伪随机序列生成器将随机密钥 k_I 扩展为 $\left\lfloor\frac{M}{2}\right\rfloor\times\left\lfloor\frac{N}{2}\right\rfloor$ 的伪随机矩阵。利用式(2)对所有原始像素位进行加密，生成最终的密文图像 $E_{I_I}(I)$。

$$e_m^k(i,j)=b_m^k(i,j)\oplus r^k(i,j),k=0,1,2,\cdots,7 \tag{2}$$

其中，$b_m^k(i,j)$ 表示 (i,j) 位置（$i=1,2,\cdots,\left\lfloor\frac{M}{2}\right\rfloor$，$j=1,2,\cdots,\left\lfloor\frac{N}{2}\right\rfloor$）的图像块中第 m 个像素进行像素分解后的第 k 位比特值，$r^k(i,j)$ 表示伪随机矩阵中位于 (i,j) 处的元素的第 k 位比特值，“$\oplus$”表示异或运算。因此，$e_m^k(i,j)$ 为位于 (i,j) 处的块内第 m 个像素的第 k 位比特加密后的值。

4.2.2 辅助信息的嵌入

为了节省用户端通信开销以及云服务器存储开销，利用能够实现大容量密文域信息隐藏的可分离自适应 MSB 预测方法将辅助信息 Δ 嵌入到密文图像 $E_{k_I}(I)$ 中，得到最终的携密图像 $H=E_{k_I}^{\Delta}(I)$。这样，用户只需要将携密图像 H 上传到云端，在本地保存随机密钥 k_I，删除其他中间变量。

在辅助信息嵌入过程中，如 2.2 节所述，将辅助信息的长度记为 $|\Delta|$，利用 MSB 预测算法[17]进行像素重构以腾出嵌入空间 n_c。选择 $md>1$ 的块作为可嵌入块，并构造一张位置图记录所有可嵌入块的位置，将所有块的 n_c 之和记为 c。为保证信息隐藏的可逆性，需要将包含位置图、结束标志等在内的附加信息 Add 与辅助信息 Δ 一起嵌入。嵌入方式为依次替换每个可嵌入块的最后 n_c 比特，并重新将总的 32 比特反重构成 4 个独立像素，得到最终的携密图像。值得一提的是，为保证

方案的可行性，需保证密文图像可嵌入空间足够大，即 $|\Delta|+|Add|<c$。

4.3 短哈希命中时的去重流程

4.3.1 信息提取

随后，如图 7 和图 8 所示，用户 2 和用户 3 分别进行图像 I 的第二次上传和新图像 J 的第一次上传，向云端发送短哈希 $sh(I)$ 和 $sh(J)$，其中，$sh(I)=sh(J)$。CSP 在重复性检查中至少找到一个重复的 $sh(I)$，对所有与该短哈希匹配的携密密文 H 进行消息提取，并将提取到的每个辅助信息 Δ 发送给用户。得益于 MSB 预测方法信息提取、图像解密的可分离特性，提取信息后原始图像内容依然对 CSP 不可见，以此保证数据的机密性。

类似于文献[17]方案，CSP 提取信息时首先将 H 分成 2×2 大小的非重叠块，将前面的每个块分解 4 个像素以获得 32 位的比特序列，从中得到像素 P，md，e_1，e_2，e_3 和隐藏数据。然后将 C_1，C_2 和 C_3 恢复为：

$$C_i=Trunc(P,md)+e_i,i=1,2,3 \tag{3}$$

其中，$Trunc(P,md)$ 截取 P 的 md 个 MSB 位，“+”表示按位左连接。这样一个接一个地提取出位置图，然后根据位置图提取剩余的嵌入信息。

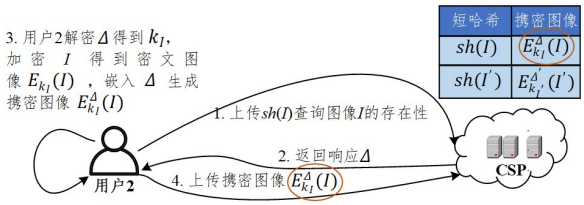


图 7 文件 I 的第二次上传

Fig. 7 Second upload of file I

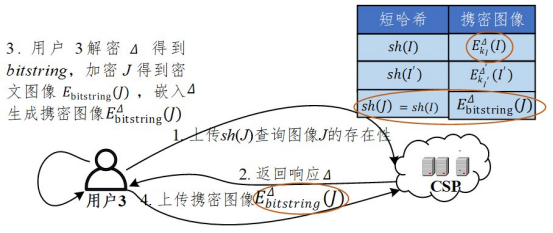


图 8 文件 J 的第一次上传

Fig. 8 First upload of file J

4.3.2 图像处理

假设用户 2 和用户 3 分别接收到 Q 条辅助信息，他们将依次从中解出 Q 个潜在随机密钥，并依据解出的密钥重新进行图像加密、信息嵌入（如 4.2 节所述），最后将 Q 个携密图像上传到云上。

在解密随机密钥时，类似于 LEVER[16]，用户对每条辅助信息执行相同操作。以用户 2 为例，首先计算 $\varepsilon(h(I))$，接下来计算 $\varepsilon(k_I)=\varepsilon(k_I\ominus h(I))\otimes\varepsilon(h(I))$，其中“$\otimes$”表示加性同态加法。最后，用户对 $E_{\varepsilon(k_I)}(E_{h(I)}(k_I))$ 执行两次解密，利用第一次的密钥 $\varepsilon(k_I)$ 和第二次的密钥 $h(I)$ 解出随机密钥 k_I。上述过程最终为 I 解密得到 Q 个潜在的密钥，其中 1 个为真正的随机密钥 k_I，剩余的 $Q-1$ 个为随机比特串 $bitstring$。而对于用户 3 来说，由于图像 J 实际上并未存储于云端，根据函数 $g(x,y)$ 的特性，输入不匹配的图像和辅助信息，最终将会得到 Q 个随机比特串，这类似于 4.2 节中没有其他用户上传过 J 的情况。值得注意的是，真正解出的随机密钥也是高熵的比特串，因此用户无法区分 k_I 和 $bitstring$，请求图像的存在性隐私得以保护。

4.3.3 数据存储

如图 7 和图 8 所示，CSP 在接收到用户 2 和用户 3 上传的 $E^{\Delta}_{k_I}(I)$ 和 $E^{\Delta}_{bitstring}(J)$ 后，在本地存储列表中查找匹配。考虑到 LEVER 方案会根据匹配结果生成可区分响应，从而导致安全风险，本文创新性地提出了一种抗侧信道攻击的数据存储方式。具体来说，$E^{\Delta}_{k_I}(I)$ 匹配成功，CSP 舍弃用户 2 上传的所有携密图像以实现去重。而 $E^{\Delta}_{bitstring}(J)$ 匹配失败，表明云中尚未存储 J 的密文。此时区别于 LEVER 的设计，CSP 不再返回第二个响应要求用户重新选择一个随机密钥并上传新的密文，而是在用户 3 上传的 Q 个携密图像中任选一个存储到 $sh(I)$（$sh(J)$）对应的文件列表中。这意味着，实际上由 CSP 而不是用户 3 自己为新图像 J 确定了加密密钥 $k_J=bitstring$。CSP 通过这种简化交互的方式实现了对用户 2 和用户 3 两种情况的无差异响应，保证本文改进方案 EMSD 抗侧信道攻击的安全性。

4.3.4 安全性分析

本节考虑外部攻击者企图通过发动侧信道攻击甚至暴力字典攻击来窃取目标文件的内容隐私的情况。根据 EMSD，外部攻击者在整个去重流程中只接收到了 CSP 返回的一次响应（对短哈希是否命中的回复），内容为“No”或者多条辅助信息。因为短哈希具有强碰撞性，同一短哈希可能对应多幅不同的图像。因此，短哈希的存在并不会泄露特定图像的云端存在状态。得益于 EMSD 对文献[16]方案去重流程的优化，对于可能被攻击者利用来发动侧信道攻击甚至是暴力字典攻击的关键点（携密图像的匹配结果），CSP 不再返回二次响应，从而消除了安全风险。

而内部攻击者即使能够通过执行信息提取来获得密文图像和辅助信息，也由于它对明文图像没有任何先验知识，且由哈希摘要生成短哈希的过程为单向计算，CSP 无法获取对解密随机密钥至关重要的哈希值 $h(I)$，也就无法解密得到真正的随机密钥 k_I。用户图像的内容隐私得以保护。

此外，随机密钥在不同用户之间传递的过程被认为是安全的。具体来说，请求上传的短哈希命中而图像未存储的恶意攻击者或正常用户，通过解密 CSP 返回的辅助信息得到与随机密钥等长且不可区分的随机比特串 $bitstring$。因此，只有实际拥有原始图像的正常用户才能得到真正的随机密钥。

5 性能分析

选取两个具有不同分辨率的真实世界数据集 SIPI[24] 和 Unsplash[25] 进行实验，比较 EMSD，LEVER[16]，CE 方案[15] 和原始去重方案 O-DD 在通信开销、存储开销和计算开销方面的差异。在 SIPI 和 Unsplash 数据集中分别选取 210 和 1 000 幅图像，这些测试图像的分辨率分别处于 256×256 到 2 250×2 250 和 1 280×774 到 10 000×6 667。在 Amazon 弹性计算云（EC2）实例上实现 CSP 功能，在配备 Intel Core i7-10875H CPU@3.3 GHz，16 GB RAM 和 7200 RPM 512 GB

硬盘的工作站上实现用户端程序。所有结果都使用 Python3.9 进行 20 次实验并取平均值，以保证可靠性。

5.1 嵌入容量

为保证去重方案的可行性，传递密钥所必须的辅助信息 Δ 的长度需要小于测试图像总嵌入容量减去信息嵌入过程中产生的位置图、结束标志等在内的附加信息 Add 的长度。将这一差值定义为可用嵌入容量，统计两个数据集每幅测试图像的可用嵌入容量，结果如图 9 所示。

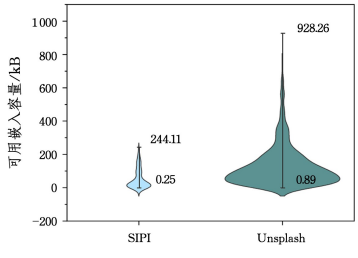


图 9 可用嵌入容量

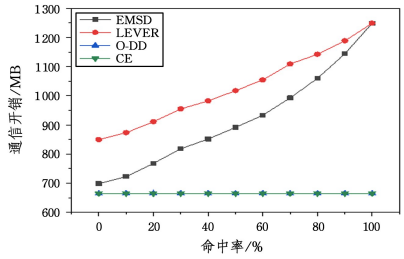
Fig. 9 Available embedded capacity

SIPI 和 Unsplash 数据集的测试图像中，可用嵌入容量的最小值分别为 0.25 kB 和 0.89 kB。而辅助信息 Δ 的长度取决于加密随机密钥过程中所使用的公钥算法的输出密文长度。以 RSA 算法为例，密文长度可选择 1 024 比特，远小于图 9 中的最小值。因此 EMSD 去重框架可行。

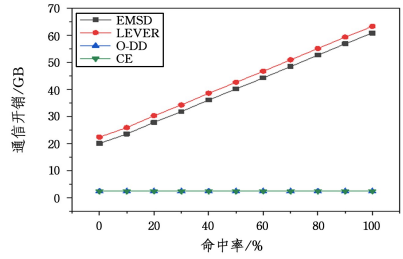
5.2 通信开销

为了评估所提方案在通信开销上的优势，假设每个数据集的实验图像中分别有 0%，10%，20%，…，100% 幅随机选择的图像被存储于云端。接下来对所有实验图像发起去重请求，计算不同图像重复率下的用户端通信开销。图 10(a)和图 10(b)分别给出了两个数据集中 4 种方案的比较结果。值得一提的是，为了在相同的情况下评估 4 种方案，将短哈希固定为 8 比特，且每次实验后都会将云存储恢复到其原始状态，使后续的比较不受影响。

如图 10 所示，EMSD 在通信开销上相较于 LEVER 实现了较好的性能。以 SIPI 数据集为例，当图像重复率从 0% 增加到 100% 时，EMSD 的通信开销分别为 698 MB，722 MB，768 MB，818 MB，851 MB，891 MB，933 MB，992 MB，1 060 MB，1 144 MB，1 248 MB，明显低于 LEVER 方案的 849 MB，873 MB，910 MB，954 MB，982 MB，1 017 MB，1 054 MB，1 108 MB，1 142 MB，1 188 MB，1 249 MB。这是因为，一方面，由于引入了信息隐藏技术，在短哈希未命中时，EMSD 将用于传递随机密钥的辅助信息 Δ 嵌入密文图像中，用户只需要上传与明文图像等长的携密图像。而在 LEVER 中，用户在上传密文图像的基础上，还要额外上传 $|\Delta|$ 位的辅助信息。另一方面，EMSD 优化了现有方案的去重流程，在短哈希命中但图像未存储于云中时，只需要 CSP 在用户上传的携密图像中任选一个存储，不需要用户上传额外信息。而 LEVER 方案却需要用户重新选择一个新的随机密钥，将密文图像和辅助信息一起发送给云。至于 CE 方案和原始去重方案 O-DD，其通信开销均近似维持在 665 MB。由于用户只需要上传密文图像和索引值，因此 CE 和 O-DD 在 4 个方案中实现了最低的通信开销。然而，EMSD 和 LEVER 均使用随机密钥对原始明文图像进行加密，保证 CSP 无法通过猜测图像内容获取存在性隐私。因此相较于 CE 和 O-DD 方案，EMSD 以较少的通信开销换取了强安全性保障。



(a) SIPI 数据集



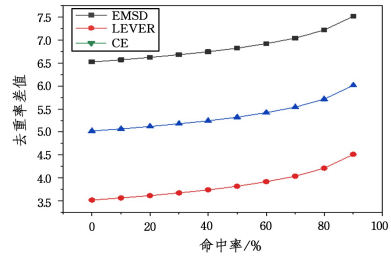
(b) Unsplash 数据集

图 10 通信开销的比较

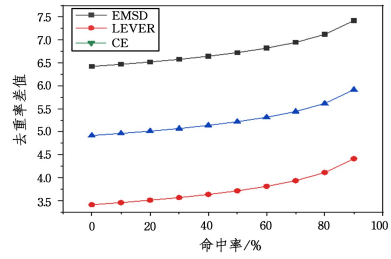
Fig. 10 Comparison of communication overhead

5.3 存储开销

与 5.2 节类似，通过改变数据集中图像的命中率，来计算不同方案的云端去重率。为了准确衡量云端存储开销，将去重率定义为 $R_d = 1 - d_s/d_r$，其中 d_s 表示去重结束后云中存储的数据量，d_r 表示用户请求去重的数据量。由于数据集中图像数据较大，为了更清晰地展示 EMSD 在存储开销上的优势，以 O-DD 方案去重率为基准，分别计算 EMSD，LEVER 和 CE 方案相比 O-DD 的去重率差值 $|\lg(R_d^{\text{O-DD}} - R_d^{\text{EMSD/LEVER/CE}})|$。O-DD 在不考虑安全性的前提下实现了理想的去重率，因此，去重率差值越大，去重率越高，方案节省的云端存储开销越多。两个数据集的实验结果如图 11 所示。



(a) SIPI 数据集



(b) Unsplash 数据集

图 11 存储开销的比较

Fig. 11 Comparison of storage overhead

如图 11 所示，相比 LEVER 和 CE，EMSD 节省了更多的存储开销。以 Unsplash 数据集为例，当图像重复率从 0%增加到 90%时，EMSD 的去重率差值分别为 6.42，6.46，6.51，6.57，6.64，6.72，6.82，6.94，7.12，7.42，明显高于 LEVER 方案的 3.41，3.45，3.50，3.56，3.63，3.71，3.80，3.93，4.11，4.41 和 CE 方案的 4.91，4.96，5.01，5.07，5.13，5.21，5.31，5.44，5.61，5.91。这是因为，尽管 EMSD 和 LEVER 均只保留同一图像的单一副本及其索引值（短哈希），看似实现了相同的去重率，但是得益于信息隐藏技术的引入，EMSD 可以将辅助信息 Δ 嵌入到密文图像中，保证 CSP 的存储列表只保存携密图像即可。而在 LEVER 中，CSP 还要额外存储 $|\Delta|$ 位辅助信息。此外，由于短哈希和全长哈希所占用的存储空间存在差异，EMSD 的去重率高于 CE 方案。

5.4 计算开销

在通信开销和存储开销之外，我们额外比较 4 个方案的计算开销。从用户角度出发，计算开销定义为用户接收到响应到根据响应上传数据这一过程所需的时间。分别计算两个数据集中所有测试图像的平均计算开销，结果如表 1 所列。

表 1 计算开销的比较

Table 1 Comparison of computing overhead

数据集	方案	平均计算开销/s
SIPI	EMSD	9.068
	LEVER	1.552
	CE	0.584
	O-DD	0.002
Unsplash	EMSD	27.204
	LEVER	4.969
	CE	1.988
	O-DD	0.008

从表 1 可以看出，4 个方案的计算开销按从高到低排序分别是 EMSD，LEVER，CE，O-DD。这是因为，O-DD 方案中，用户只需执行图像加密，而 CE，LEVER，EMSD 分别在上一个方案的基础上增加了索引值（哈希）加密、辅助信息生成和辅助信息嵌入操作。由于信息嵌入操作，EMSD 计算开销增长幅度较大。然而，这一操作恰好是降低用户通信开销和云存储开销，以及提高方案抗侧信道攻击安全性的关键步骤。

结束语 针对图像密文云数据去重中由可区分响应导致的侧信道攻击，以及为实现加密密钥传递引发的额外代价等问题，本文提出了一种基于自适应 MSB 的高效密文图像安全去重机制。在 LEVER 方案的基础上，创新性地将密文域信息隐藏技术引入去重框架，在密文图像中嵌入用于传递密钥的辅助信息，以节省其带来的通信开销和存储开销。在此基础上，优化交互流程，即使请求图像并未存储于云端，用户也无需开展额外的密文上传工作，从而实现响应混淆，保证抗侧信道攻击的安全性。安全性分析和实验结果表明了本文方案在安全性和性能上的优势。

所提算法针对数据量大、重复度高的图像云数据去重开展研究，虽然实现了预期设计目标，但只适用于精确图像去重领域，且增加了一定的计算开销。后续工作将考虑在降低计算开销的同时实现模糊图像去重。

参 考 文 献

[1] WANG C，ZHANG B，REN K，et al. Privacy-assured outsourcing of image reconstruction service in cloud [J]. IEEE Transactions on Emerging Topics in Computing，2013，1(1)：166-177.

[2] TANG X，CHEN X，ZHOU R，et al. Marking based obfuscation strategy to resist side channel attack in cross-User deduplication for cloud storage [C]// Proceedings of the 21th IEEE International Conference on Trust，Security and Privacy in Computing and Communications (TrustCom). Wuhan，China，2022：547-555.

[3] SHIN Y，KOO D，HUR J. A survey of secure data deduplication schemes for cloud storage systems [J]. ACM Computing Surveys，2017，49(4)：74.

[4] AGARWALA A，SINGH P，ATREY P. Client side secure image deduplication using DICE protocol [C]// Proceedings of 2018 IEEE Conference on Multimedia Information Processing and Retrieval(MIPR). Miami，USA，2018：412-417.

[5] LI J，CHEN X，LI M，et al. Secure deduplication with efficient and reliable convergent key management [J]. IEEE Transactions on Parallel and Distributed Systems，2014，25(6)：1615-1625.

[6] LI D，YANG C，JIANG Q，et al. A client-based image fuzzy deduplication method supporting proof of ownership [J]. Chinese Journal of Computers，2018，41(6)：1267-1283.

[7] TAKESHITA J，KARL R，JUNG T. Secure single-server nearly-identical image deduplication [C]// Proceedings of 2020 International Conference on Computer Communications and Networks(ICCCN). Honolulu，USA，2020：1-6.

[8] JIANG T，YUAN X，CHEN Y，et al. FuzzyDedup：secure fuzzy deduplication for cloud storage [J]. IEEE Transactions on Dependable and Secure Computing，2023，20(3)：2466-2483.

[9] LIU X M，TANG X，JIN L C，et al. Secure cross-user fuzzy deduplication for images in cloud storage [C]//Proceedings of the 7th International Conference on Data Mining and Big Data (DMBD). Beijing，China，2022：291-302.

[10] 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.

[11] 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.

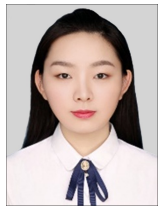
[12] TANG X，ZHOU L N. Response obfuscation based secure deduplication method for cloud data with resistance against appending chunk attack [J]. Journal of Computer Applications，2020，40(4)：1085-1090.

[13] 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.

[14] BELLARE M，KEELVEEDHIS，RISTENPART T. Message-locked encryption and secure deduplication [C]//Proceedings of the 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques(EUROCRYPT). Athens，2013：296-312.

[15] STORER M W，GREENAN K，LONG D D，et al. Secure data deduplication [C]//Proceedings of the 2008 ACM Workshop on

Storage Security and Survivability (StorageSS). Alexandria, 2008:1-10.
[16] POORANIAN Z, SHOJAFAR M, GARG S, et al. LEVER: secure deduplicated cloud storage with encrypted two-party interactions in cyber-physical systems [J]. IEEE Transactions on Industrial Informatics, 2021, 17(8):5759-5768.
[17] WANG Y M, HE W G. High capacity reversible data hiding in encrypted image based on adaptive MSB prediction [J]. IEEE Transactions on Multimedia, 2022, 24(1):1288-1298.
[18] LIU X M, TANG X, YANG S T, et al. Reed-Solomon coding based secure deduplication for cloud storage with resistance against side channel attack [J]. Journal of Cyber Security, 2022, 7(6):80-93.
[19] TANG X, LIU Z, SHAO Y, et al. Side channel attack resistant cross-user generalized deduplication for cloud storage [C]// Proceedings of the 56th IEEE International Conference on Communications(ICC). Seoul, South Korea, 2022:998-1003.
[20] TANG X, ZHOU Y T, CHENG Y X, et al. Weighted average-based complexity calculation in block selection oriented reversible data hiding [J]. Security and Communication Networks, 2022, 2022:1-15.
[21] TANG X, ZHOU L N, TANG G, et al. Improved fluctuation derived block selection strategy in pixel value ordering based reversible data hiding [C]// Proceedings of the 20th International Workshop on Digital-forensics and Watermarking(IWDW). Beijing, China, 2021:163-177.
[22] ZHANG X P. Reversible data hiding in encrypted image [J]. IEEE Signal Processing Letters, 2011, 18(4):255-258.
[23] HONG W, CHEN T S, WU H Y. An improved reversible data hiding in encrypted images using side match [J]. IEEE Signal Processing Letters, 2012, 19(4):199-202.
[24] SIPI. The USC-SIPI image database [DB/OL]. 1977, http://sipi.usc.edu/database/.
[25] Unsplash. The UNSPLASH image database [DB/OL]. 2013, https://www.unsplash.com/.



ZHOU Yiteng, born in 1998, postgraduate. Her main research interests include reversible data hiding and cloud data deduplication.

TANG Xin, born in 1987, Ph.D, associate professor, is a member of CCF (No. H9744M). His main research interests include reversible watermarking, cloud data deduplication, integrity auditing and scalable distributed data storage.

(责任编辑:柯颖)