

基于分割的数字图像云存储机制

吕骁博 郭 耀 陈向群

(高可信软件技术教育部重点实验室(北京大学) 北京 100871)

(北京大学信息科学技术学院软件研究所 北京 100871)

(lvxb12@sei.pku.edu.cn)

A Splitting-Based Cloud Storage Mechanism for Digital Images

Lü Xiaobo, Guo Yao, and Chen Xiangqun

(Key Laboratory of High Confidence Software Technologies (Peking University), Ministry of Education, Beijing 100871)

(Institute of Software, School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

Abstract Due to the development of cloud computing and datacenters, the task of data processing and storage has migrated to the cloud platforms gradually. As a special kind of data, digital images contain a lot of private information, and require a huge amount of storage resources with frequent addition and deletion. Although storing digital images on cloud servers could solve some of these challenges, the quality of service provided by some of these servers is not reliable enough, especially when privacy issues are concerned. This paper proposes a splitting-based cloud storage mechanism for digital images. After splitting images into several blocks or layers and storing them into different cloud storage platforms, the privacy and reliability of these images can be enhanced. Two image splitting algorithms are proposed: block-based splitting and layer-based splitting. An image storage tool for privacy protection is implemented to demonstrate the mechanism. Experimental results show that the proposed techniques are able to provide privacy protection, at the same time enhancing the reliability and performance when storing digital images to the cloud. Performance under different cloud storage services, different splitting algorithms and different splitting granularities are also provided and analyzed. Finally, privacy-related issues and challenges are discussed.

Key words cloud computing; data storage; image splitting; privacy protection; quality of service

摘要 随着云计算和数据中心的发展,数据处理和存储工作逐渐转移到云平台。通过云服务器存储数字图像可以解决图像存储和共享的问题,但是一些云服务器可能难以保证这些数字图像数据的可靠性和可用性,导致用户隐私泄露。提出一种基于分割的数字图像云存储机制,并采用了2种不同的图像分割算法:分块分割和分层分割。为了验证该机制,实现了一个支持隐私保护的数字图像云存储工具,针对不同图像分割方法进行了比较分析与实现,对所选的不同云服务器服务质量进行了测试。通过一系列实验,结论表明该机制可为数字图像云存储提供隐私保护、提高可靠性,并在一定程度上提高云存储的性能。

关键词 云计算;数据存储;图像分割;隐私保护;服务质量

中图法分类号 TP316.8

收稿日期:2012-09-28;修回日期:2013-02-04

基金项目:国家“九七三”重点基础研究发展计划基金项目(2011CB302604);国家自然科学基金项目(61103026);国家“八六三”高新技术研究发展计划基金项目(2011AA01A202);国家自然科学基金中港合作项目(60931160444)

通信作者:郭 耀(yaoguo@sei.pku.edu.cn)

随着云计算以及分布式存储技术的发展,数据处理和存储的工作逐渐转移到云平台上.数以亿计的用户每天访问网络,并且利用网络进行数据的存储与管理,以此节省本地的计算与存储资源.

数字图像作为一种特殊的数据,具有信息量大、数据量大、添加删除频繁等特点.数字图像在日常生活中是一种普遍存在的数据,随着智能手机以及移动无线网络(3G, WIFI)的普及,人们对数字图像上传、下载、分享的需求日益增加,2011年官方数据显示,Facebook用户日均上传照片数超过1亿张.与此同时,因为数字图像包括大量用户信息,如人物面部信息、场景信息等,这些敏感信息涉及到用户隐

私,需要加以特殊保护.

基于上述特点,用户有意愿将数字图像存放在云服务器中,但由第三方提供的公共云服务器可能难以保证这些数字图像数据的可用性和可靠性.以此为背景,本文提出了一种基于分割的数字图像云存储机制.

数字图像包含大量用户隐私信息,通过图像分割将图像中的信息分离.然后,将分割后的数据存储到不同云服务器中.通过这种方式可以提高用户图像的安全性.基于分割的图像存储机制的基本工作流程如图1所示:

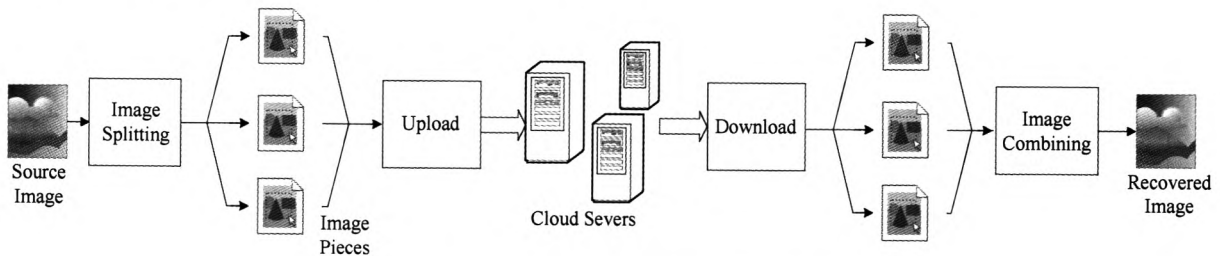


Fig. 1 Workflow of cloud storage mechanism.

图1 云存储机制工作流程

这一机制通过对图像进行分割,并将其上传到多个云服务器,具有如下的优点:

- 1) 为数字图像的云存储提供隐私保护;
- 2) 提高数字图像存储的可靠性,降低图像信息丢失的概率;
- 3) 在一定程度上提高云存储的性能.

1 图像分割算法

1.1 理论分析

1) 隐私保护. 如果因为某个云服务器的可靠性问题,导致存储在该服务器上的图像片段泄露,由于每个图像片段仅保存部分原始图像信息,与不分割存储整个图像相比,可以减少泄露的总信息量,由此提高对图像隐私信息的保护.

2) 可靠性. 假设图像片段一共存储在 K 个云服务器,第 i 个云服务器上存有 N_i 个图像片段. 设每个图像片段包含的信息量为 INF_{ij} , 每个图像片段丢失(或泄露)的概率为 P_{ij} , 其中,下标 i 表示该片段位于第 i 个云服务器上($1 \leq i \leq K$),下标 j 表示该图像片段是存储在该云服务器上的第 j 个图像片

段($1 \leq j \leq N_i$).

则通过分割后进行云存储能保存下来的图像信息量为

$$INF_{\text{PRESERVE}} = \sum_{i=1}^K \sum_{j=1}^{N_i} (1 - P_{ij}) INF_{ij}.$$

原始图像总信息量为

$$INF = \sum_{i=1}^K \sum_{j=1}^{N_i} INF_{ij}.$$

若不进行分割,则最坏情况下能保存下来的图像信息量为

$$INF_{\text{PRESERVE}}^* = (1 - \max(P_{ij})) INF.$$

易得 $INF_{\text{PRESERVE}}^* \leq INF_{\text{PRESERVE}}$, 由此可见对图像进行分割可提高在进行云存储后获取的图像总信息量,从而提高数字图像存储的可靠性.

3) 传输性能. 对图像进行分割后,可利用多线程技术同时与不同云服务器建立连接,进行传输. 假设传输总时间为 T , 与云服务器建立连接的时间为 T_{connect} , 传输时间为 T_{trans} , 断开连接的时间为 $T_{\text{disconnect}}$.

不分割的情况下:

$$T^* = T_{\text{connect}} + T_{\text{trans}} + T_{\text{disconnect}}.$$

进行分割的情况下:

$$T = ALL_T_{connect} + ALL_T_{trans} + ALL_T_{disconnect},$$

其中, $ALL_T_{connect}$, ALL_T_{trans} , $ALL_T_{disconnect}$ 分别是有分割的情况下与云服务器建立连接, 传输数据, 断开连接的总时间. 因为是多线程并行处理, 所以 $T_{connect} = ALL_T_{connect}$, $T_{disconnect} = ALL_T_{disconnect}$, ALL_T_{trans} 可近似为分割后图像片段的传输时间, 假设有 N 个图像片段, 则有:

$$ALL_T_{trans} = T_{trans}/N.$$

由此得到 $T^* > T$, 结论为对数字图像进行分割可以在一定程度上提高云存储的性能.

1.2 算法设计

本节将描述 2 种高效、简便的分割算法, 分别是分块分割算法与分层分割算法.

数字图像可用像素矩阵来表示, 假设一幅图像的长和宽分别是 H 和 W , 单位为一个像素. 可以用一个二维函数 $f(x, y)$, $x \in [0, H]$, $y \in [0, W]$ 表示一幅图像, 其中, x, y 是空间坐标, f 在任何坐标点 (x, y) 处的振幅为该点的亮度.

1.2.1 基于分块的图像分割算法

基于分块的图像分割算法以像素点坐标 (x, y) 为分割依据, 将图像分为相同大小的多个块.

该算法的一般性描述是: 用户规定的分割粒度用 $(\delta x, \delta y)$ 表示, 其中 δx 和 δy 均为整数, 根据分块算法, 将图像分为 $H/\delta x \times W/\delta y$ 块, 每块的大小为 $(H/\delta x, W/\delta y)$, 第 (i, j) 块的像素点坐标 (x, y) 满足: $x \in [iH/\delta x, (i+1)H/\delta x]$, $y \in [jW/\delta y, (j+1)W/\delta y]$, 其中 x, y 均为整数, $i = 0, 1, \dots, \delta x, j = 0, 1, \dots, \delta y$.

1.2.2 基于分层的图像分割算法

对于一个彩色图像而言, 其表示方式有多种^[1]. 常用的彩色图像格式采用 RGB 彩色空间表示法.

基于分层的图像分割算法以像素点的灰度值 $f(x, y)$ 为分割依据, 将图像分为和原图像大小相同的多个层, 每个块包含原来图像的部分信息.

该算法的一般性描述是: 用户规定的分割粒度用 δx 表示 (δx 为整数), 根据分层算法, 将图像分为 δx 层, 每层的大小与原始图像相同. 所有层的坐标

为 (x, y) 的像素点满足: $f(x, y) = \sum_{i=1}^{\delta x} f_i(x, y)$.

在实现时, 分层的分割算法首先选取 RGB 三个通道作为 3 层, 然后用随机数的方法继续分层.

1.3 图片到云服务器的映射

将分割后的图像片段映射到不同的云服务器

上, 需要预先指定服务器个数及每个云服务器的地址. 首先与云服务器建立连接, 包括获取许可、检查密钥等步骤 (与具体的云服务器有关). 然后按一定映射规则将图像片段分别上传到不同的云服务器上, 在本文工具实现中, 采用的是随机映射的方式, 将分割后的图像片断平均分配到多个云服务器上.

2 实验与结果分析

2.1 云服务器搭建

云存储是一种较为成熟的技术. 目前已经有许多基于开放式云存储的商业领域应用, 而许多网络服务运营商也提供了相应的云存储服务支持, 例如 Amazon Web Services^[2], Enomalism Elastic Computing Infrastructure, Azure Cloud Storage^[3] 等.

本文实验中, 搭建了数个云服务器 (如表 1 所示) 来模拟更为真实的网络云存储环境, 以达到所需的实验效果. 其中, 云服务器的服务质量 (QoS) 可以用传输时间来衡量.

Table 1 Cloud Servers Used in Experiments

表 1 实验中用到的云服务器

Server	Details
Server ₁	Implemented in the local PC. Its speed is the fastest.
Server ₂	Based on an FTP server in the local network, which has the second fastest speed.
Server ₃	A cloud server based on the APIs provided by Sina VDisk (a commercial network storage service). A remote server located within the same continent.
Server ₄	A server based on the Amazon EC2 Micro Instance. The server is located on another continent, whose speed is the slowest.

2.2 图像分割实验结果分析

我们选取图 2 作为测试图像, 格式为 jpeg, 大小为 203 KB. 该图像包含 RGB 三个通道, 每个通道中像素矩阵大小为 978×650 , 每个像素用 uint8 格式表示.



Fig. 2 Testing digital image.

图 2 测试用图像

将测试图像采用分块的方法进行分割,分割粒度 $(\delta x, \delta y) = (8, 6)$, 得到的结果是 8×6 个图像片

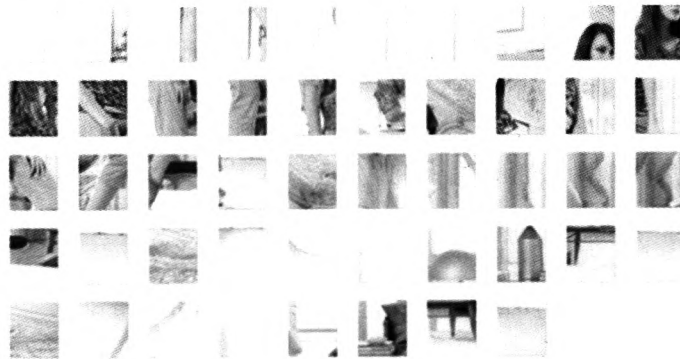


Fig. 3 Results of block-based image splitting.

图3 分块分割结果

将测试图像采用分层的方法进行分割,分割粒度 $\delta x = 6$, 得到的结果为图 4 所示的 6 个图像片段. 实验中进行了 5 次测试, 平均耗时 8 770 ms.

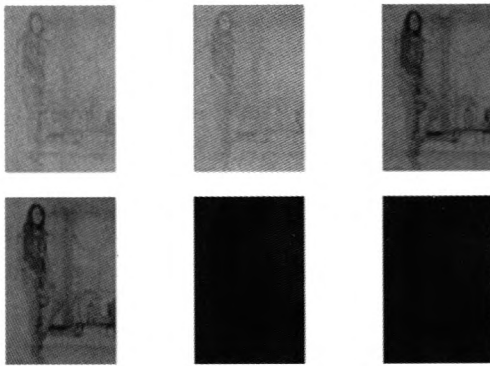


Fig. 4 Results of layer-based image splitting.

图4 分层分割结果

无论是哪种分块方式,分割粒度越大,每个图像片段所包含的原图像的信息越小,因此对隐私保护的强度越强,但是分割开销也会越大.

对基于分块的图像分割方式而言,所得的图像片段细节(例如测试图像右上部分墙面的纹理)明显,但不会显示原来图像的轮廓,对基于分层的图像分割方式而言,所得的图像片段细节模糊,但会显示原来图像的轮廓.

就效率而言,分块算法的效率较高,这是因为分块算法不需要按像素值进行计算.

这 2 种分割方式都可以很好地对图像信息进行隐私保护. 虽然对图像进行隐私保护没有特定的评价标准, 仍可采用更多的方法对其进行优化, 具体细节可见本文第 3 节的讨论.

2.3 实验结果

无论是分层还是分块算法,所得到的图像片段

段,如图 3 所示. 在实验中进行了 5 次测试,分块分割的平均耗时 238 ms.

并没有本质的区别,所以在下述实验中,我们采用分块算法得到的图像片段作为实验样例,进行云服务器的上传和下载测试.

首先对不分割的原始图像上传下载作测试,得到如表 2 所示的平均上传下载时间:

Table 2 Transmission Time without Splitting

表 2 未分割-云服务器传输时间 ms

Server	Uploading Time	Downloading Time
Server ₁	50	31
Server ₂	817	894
Server ₃	1 422	1 172
Server ₄	41 888	40 284

在对图片进行分块分割之后,我们对每个服务器分别进行了多次测试,其上传和下载时间如表 3 所示. 从表 3 显示的数据来看,不同云服务器 QoS(传输时间)差异较大. 其中 Server₁ 的 QoS 最好,而 Server₄ 的 QoS 最差. 在实验中,我们在对 Server₃ 进行上传实验,还遇到了云服务器故障,而导致文件无法上传的问题,可见进行云存储时,数据安全性和可用性问题确实存在. 上述结果与我们的预期假设一致.

Table 3 Transmission Time with Splitting

表 3 分割-云服务器平均传输时间 ms

Server	Uploading Time	Downloading Time
Server ₁	448	113
Server ₂	4 792	5 106
Server ₃	61 239	51 987
Server ₄	612 559	607 155
Multi-Servers	164 891	163 022

在进行单个云服务器上传实验后,我们进行了多云服务器上传下载实验.

我们将相同的实验数据集(图像片段)同时上传到不同云服务器上,然后进行下载实验.由于 Server₃ 不太稳定,我们选择了 Server₁, Server₂, Server₄ 作为多个云服务器测试用的云服务器.测试结果如表 3 所示,可见使用多个云服务器的 QoS 明显较高,约为单个云服务器的 1/3,这是因为我们一共使用了 3 个云服务器,且 Server₁ 和 Server₂ 的服务质量明显优于 Server₄.

从表 2 和表 3 的对比来看,未分割的图像与经过分割的图像的上传时间存在一些差异,这是由于具体的实现细节产生的.为此,我们针对性能最好的 Server₁ 和性能最差的 Server₄ 进行了对比实验,并采用多线程的方式进行了图像传输的优化,结果如图 5、图 6 所示:

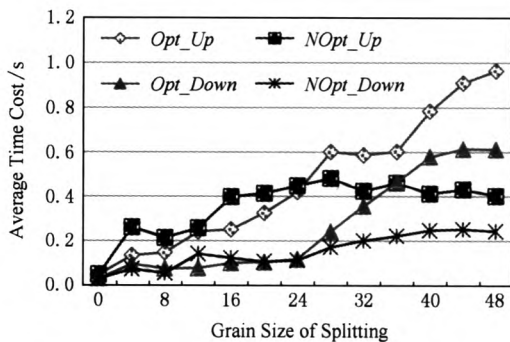


Fig. 5 Result of optimization on server₁.

图 5 Server₁ 不同粒度分割优化结果

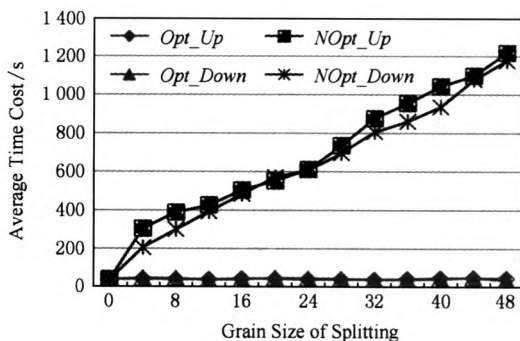


Fig. 6 Result of optimization on server₄.

图 6 Server₄ 不同粒度分割优化结果

图 5、图 6 中纵轴为传输时间,横轴为分割粒度(块数),Opt_Up, NOpt_Up, Opt_Down, NOpt_Down 分别是优化上传时间、未优化上传时间、优化下载时间、未优化下载时间.

从结果来看,Server₁ 的传输时间较短,服务质量较高,分割的粒度增加,对上传下载的时间影响不

大.而分割与未分割(分割粒度为 0)相比,时间有差别,这是因为在实现时,为了保证图像质量,将 jpeg 的压缩率调得较低,而使得分割后数据量变大,所以导致传输开销的增大.在 Server₁ 上采用多线程的优化方式,对整个传输过程的时间优化并不明显,这是因为原来的传输时间就很短,在分割粒度较高时,优化计算的额外开销将导致传输性能下降.

对 Server₄ 而言,传输时间与压缩粒度大致成正比关系,这是因为网络传输的关系,导致每次传输的时间较长,而数据量的多少成为次要因素.采用多线程的优化方式可以使传输过程的时间开销极大降低,甚至比未分割的情况更好,这也符合前面的理论分析结果.我们的优化策略适合于传输时间较长的云服务器.

从上述结果来看,分割所造成的额外开销(包括分割与融合的时间开销,以及传输时间的额外开销)较小,在一般用户的可容忍范围内.该结论说明本文的云存储机制可以在进行隐私保护,提高可用性和可靠性的同时保证较小的额外开销,甚至提高云存储的传输性能.

通过优化实现细节,还可以进一步减小额外开销,接近理论上分割与不分割传输时间相等的情况.

3 相关问题及讨论

3.1 分割算法

在实际应用中,图像的每个像素包含的信息量并不相同.例如一幅图像中含有纯色背景和人物,则背景像素的信息含量较小,而人物像素的信息含量较大.在人物像素中,脸部所在位置的像素信息量又比其他部分更大.我们把含有信息量较高的部分(例如人脸)称为一个图像的重要信息区域.如果采用一些其他数字图像处理的技术^[4],可以提取这些重要信息区域,再对这些重要信息区域进行更细粒度的划分,例如提高分块(或分层)粒度,以及采取一些数字图像处理方法进行细节处理.这样做会导致算法性能下降,但是会提高安全性,具有更好的隐私保护功能.

与分块分割方法相比,分层算法灵活性较强,可以采用许多不同的方法对图像进行分层.分层方法比分块方法能更好地保护重要信息区域(例如照片中的人脸),这是因为重要信息区域可能会集中于某个较小区域.在分块方法中这一块区域可能并没有被分割,而分层的方法基于图像像素的灰度值,可以

更好地拆分重要信息集中区域的信息,但是分层的算法相对来说更加复杂,效率也较低。

3.2 功能优化

随着智能手机的普及,以及手机照相功能、网络功能的提高,手机平台对于图像云存储的需求会越来越高,可以将本文实现的工具移植到移动平台上,例如 Android 平台、IOS 平台、Windows Phone 平台等,以满足这方面的需求。

如果发生某个云服务器上数据的遗失,本文工具仍可以将从其余云服务器上获得的信息进行整合,得到一个包含大部分原始信息的融合图像,以此提高云服务器容错率。如果遇到某个云服务器访问或传输过慢,本文实现的工具可以提前整合从其他云服务器上获得的图像片段,得到一个包含大部分原始信息的融合图像,以此提高用户体验。此外,如果发生部分图像片段丢失,还可使用一些图像修复的方法,将丢失了部分图像片段的融合图像进行修复,以恢复图像信息。

我们还可以对云存储策略进行优化。根据用户所在位置、网络环境等信息,为用户选取最优的云服务器进行存储,以降低开销,提供传输效率,Tran 等人提到了类似方法^[5]。

该云存储机制也可以推广到其他形式的数据上,例如数据库表单等,将这些数据采用一定方式进行分割,存储到不同服务器上,以达到隐私保护的目的。

3.3 与加密存储的比较

另外一种常见的保护数字图像隐私的方式是在上传到服务器之前对图像进行加密。与加密方法相比,基于分割的存储机制拥有以下优点。

首先,分割方法通过信息分散来提高可用性与可靠性。因为每个片段仅保存了原始图像的部分信息,因此当信息发生泄漏时,具有风险的仅仅是部分图像信息,如果采用加密的方式,一旦加密方式泄漏或被他人破解,则所有图像信息将会泄漏。

此外,在开销上采用分割的方式,不会增加上传的总数据量,而采用加密的方式会在原始数据之上增添额外数据,从而导致开销增大。

4 相关工作

4.1 图像分割算法

Gonzalez 等人^[1]、Pal 等人^[6]总结了一系列的基础图像分割算法。这些图像分割算法一般基于亮

度值的 2 个基本特性之一:不连续性和相似性。包括如边缘检测、阈值方法、区域生长、区域分离、聚合、分水岭、基于神经网络的图像分割方法等。在这些基础图像分割算法上,Cheng 等人^[7]总结了将这些图像分割算法应用到彩色图像中的方法。

这些分割算法希望达到的目的是将图像细分为构成它的子区域或对象,以提取需要的图像内部信息。我们可以采用这些高级算法,进一步提高图像分割的能力,对图片关键信息进行保护。

4.2 云存储的隐私保护

公共云服务提供商提供的云计算和云存储基础设施的安全性往往得不到保障,用户存储在这些云服务器上的信息可能泄露或丢失。

Hwang 等人^[8]总结了以下对存储在云上的数据进行隐私保护的方法:

1) 为云服务器和用户建立信用评级系统。例如 Trust overlay networks^[9]信用评级系统可以使云服务使用者和数据中心所有者进行安全可靠的数据传输。同建立信用评级系统的方式相比,本文采用的方法不需要对云服务器端的服务数据进行采集,从而降低了用户端与云服务器端收集与管理信息的开销。同时,也避免了针对这些信息收集而产生的权限问题。

2) 为数据着色或加水印以防止恶意篡改。该方法主要是为了保护隐私、安全和版权。通过加密和防火墙保护用户隐私,例如 Kamara^[10]提出了基于加密算法的虚拟私人云存储服务。加水印的方式与加密类似,在 3.3 节中已作出了相关讨论。

3) 通过用户认证(user authentication)的方式保护数据^[11-12],这种方法主要是通过对用户的访问控制来实现。与用户认证的方式相比,本文方法不需要收集和维持用户信息,减少了和用户相关的资源开销,并且使图像在云端的共享变为简单可行。

与上述方法相比,本文的方法不依赖于服务商提供的隐私保护机制,也可以与之一起协作,进一步提高数字图像存储的安全和隐私保护能力。

4.3 基于分割的数据隐私保护

Sweeney 对数据隐私保护进行了许多理论分析,并在文献[13-14]中提出了 k-anonymity 模型,用于对数据库信息进行隐私保护的。在该工作基础上,研究者提出了若干针对具体数据的隐私保护机制^[15-16]。这类方法与本文工作的出发点类似,但是没有专门针对数字图像分割的相关工作。对图像分割而言,每个信息片段并没有关联到一个主体,而是

对其本身的信息进行隐私保护,不需要关心数据之间的映射关系等具体信息。

5 结束语

本文针对数字图像这种特殊数据,提出了一种基于分割的数字图像云存储机制,为了验证该机制,设计并实现了一个支持隐私保护的数字图像云存储工具,并进行了相关实验。

实验结果表明,该机制可以对数字图像云存储进行有效的隐私保护,同时可以提高云存储的数据的可靠性及可用性,在大多数情况下,额外开销较低,甚至在一定程度上提高云存储的性能。

在今后的工作中,将会考虑进一步提高数字图像云存储的隐私保护能力,提高存储数据的可靠性和可用性,以及对性能和传输效率进行优化。

参 考 文 献

- [1] Gonzalez R, Woods R. Digital Image Processing [M]. 3rd ed. Englewood Cliffs, NJ: Prentice Hall, 2007
- [2] Murty J. Programming Amazon Web Services [M]. Sebastopol: O'Reilly Press, 2008
- [3] Calder B, Wang J, et al. Windows Azure storage: A highly available cloud storage service with strong consistency [C] // Proc of the 23rd ACM Symp on Operating Systems Principles (SOSP'11). New York: ACM, 2011
- [4] Li S, Jain A. Handbook of Face Recognition [M]. 2nd ed. Berlin: Springer, 2011
- [5] Tran N, Aguilera M, Balakrishnan M. Online migration for geo-distributed storage systems [C] // Proc of the 2011 USENIX Conf on USENIX Annual Technical Conference (USENIX ATC'11). Berkeley: USENIX Association, 2011
- [6] Pal N, Pal S. A review on image segmentation techniques [J]. Pattern Recognition, 1993, 26(9): 1277-1294
- [7] Cheng H, Jiang X, Sun Y, et al. Color image segmentation: Advances and prospects [J]. Pattern Recognition, 2011, 34(12): 2259-2281
- [8] Hwang K, Fox G, Dongarra J. Distributed and Cloud Computing: From Parallel Processing to the Internet of Things [M]. San Francisco: Morgan Kaufmann, 2012
- [9] Hwang K, Li D. Trusted cloud computing with secure resources and data coloring [J]. IEEE Internet Computing, 2010, 14(5): 14-22
- [10] Kamara S, Lauter K. Cryptographic cloud storage [G] // LNCS 7601: Proc of Financial Cryptography: Workshop on Real-Life Cryptographic Protocols and Standardization 2010. Berlin: Springer, 2010: 136-149
- [11] Subashini S, Kavitha V. A survey on security issues in service delivery models of cloud computing [J]. Journal of Network and Computer Applications, 2011, 34(1): 1-11
- [12] Yan L, Rong C, Zhao G. Strengthen cloud computing security with federal identity management using hierarchical identity-based cryptography [G] // LNCS 5931: Proc of 1st Int Conf on Cloud Computing (CloudCom'09). Berlin: Springer, 2009: 167-177
- [13] Sweeney L. K-anonymity: A model for protecting privacy [J]. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, 2002, 10(5): 557-570
- [14] Sweeney L. Achieving k-anonymity privacy protection using generalization and suppression [J]. International Journal on Uncertainty, Fuzziness, and Knowledge-based Systems, 2002, 10(5): 571-588
- [15] Li N, Li T, Venkatasubramanian S. T-Closeness: Privacy beyond k-anonymity and l-diversity [C] // Proc of ICDE'07. Piscataway, NJ: IEEE, 2007: 106-115
- [16] Bayardo R, Agrawal R. Data privacy through optimal k-anonymization [C] // Proc of ICDE'05. Piscataway, NJ: IEEE, 2005: 217-228



Lü Xiaobo, born in 1990. Received the BSc degree in computer science from Peking University, Beijing, China, in 2012. Since 2012, he has been a Master candidate in computer science from Peking University, Beijing, China. Student member of China Computer Federation. His current research interests include operating system, mobile cloud computing.

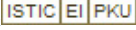


Guo Yao, born in 1976. Associate professor in the Institute of Software of the School of Electronics Engineering and Computer Science at Peking University. Senior member of China Computer Federation. His general research interests include operating systems, embedded systems, low-power computing and software engineering.



Chen Xiangqun, born in 1961. Professor in the Institute of Software of the School of Electronics Engineering and Computer Science at Peking University. Senior member of China Computer Federation. Her general research interests include operating systems, embedded system and software engineering.

基于分割的数字图像云存储机制

作者: [吕骁博](#), [郭耀](#), [陈向群](#), [Lü Xiaobo](#), [Guo Yao](#), [Chen Xiangqun](#)
作者单位: [吕骁博, 陈向群, Guo Yao, Chen Xiangqun \(高可信软件技术教育部重点实验室\(北京大学\) 北京 100871\)](#)
[, 郭耀, Lü Xiaobo \(北京大学信息科学技术学院软件研究所 北京 100871\)](#)
刊名: [计算机研究与发展](#) 
英文刊名: [Journal of Computer Research and Development](#)
年, 卷(期): 2014, 51(5)

本文链接: http://d.g.wanfangdata.com.cn/Periodical_jsjyjfz201405019.aspx