2017 年全国大学生信息安全竞赛

作品报告

作品名称： 云环境下的密文图像安全去重与检索系统

电子邮箱： 1021922388@qq.com

提交日期： 2017.5.14

填写说明

1. 所有参赛项目必须为一个基本完整的设计。作品报告书旨在能够清晰准确地阐述（或图示）该参赛队的参赛项目（或方案）。

2. 作品报告采用A4纸撰写。除标题外，所有内容必需为宋体、小四号字、1.5倍行距。

3. 作品报告中各项目说明文字部分仅供参考，作品报告书撰写完毕后，请删除所有说明文字。(本页不删除)

4. 作品报告模板里已经列的内容仅供参考,作者可以在此基础上增加内容或对文档结构进行微调。

5. 为保证网评的公平、公正，作品报告中应避免出现作者所在学校、院系和指导教师等泄露身份的信息。

目　　录

摘要

随着计算机技术的飞速发展，信息网络的发展已成为社会进步的重要保障。信息网络涉及到国家的政治，军事，经济等诸多领域。其中有很多敏感信息，甚至是国家机密，所以难免会吸引不法分子的人为攻击（例如信息泄露，信息窃取，数据篡改）。通常利用计算机犯罪很难留下犯罪证据，所以这就需要加强对信息的安全处理，防止造成信息泄露和恶意的篡改。因此，在云计算环境下对数据隐私保护的研究有着十分重要的意义，对科技和经济发展也有很重要的促进作用。

本系统由客户端和服务器构成，其中包括图片的加密传输、查找相似的图片，主要实现的功能是图片的安全传输（上传与下载）和安全存储，防止信息泄露。

本系统包括图片加密，图片上传，可验证去重，近似检索这四大模块。主要实现的功能是通过云存储为个人或者企业提供图片的安全去重和检索服务。在系统的整个操作中的图片的是密文交互的。本系统实现了用户在客户端对图片进行加密，在服务器端进行密文存储，有效的保证了信息的机密性、可用性、完整性和实用性。

本系统主要针对保证网络的信息安全和安全存储进行研究。在目前这种网络环境中，本系统从客户与用户全方面考虑的产品的应用前景不言而喻。本系统可以广泛于医疗领域，军事领域等，能同时起到双向保密和节约存储空间的作用。目前市场上没有此类产品，也与普通的防毒软件有着本质的区别，基于云计算的安全图像检索系统能够依据以上优点更快的进入市场。

关键字：云存储；可验证去重；安全检索；

第一章 作品概述

1.1 应用前景分析

由于这个基于云计算的安全图像检索系统具有高度的保密性和信息安全性，这就为我们的应用前景提供了良好的优势。首先，对于医疗领域而言，在当今的医疗系统中，为了合理的利用各种医疗资源实现资源共享。各地的地方小医院只是增加了检查设备，具体的病情诊断与分析都需要让高一级的医院来实施。对于本系统来说就相当适合于这套体系。本系统双加密云存储可以让各地医院安全有效的实施医院图像的安全统一归档，高效跨区以及跨部门的图像共享，防止不法分子恶意篡改图片信息，避免对患者造成不必要的医疗事故。在医生使用图片时还能有效的检索存放在云端的图片信息，即使是服务器云端已经存储了海量的图片，本系统还是能快速的检索出图片，有效的解决了患者的就医问题。其市场而言就是针对于广大的人民群众，在这方面的市场前景自然是不言而喻。

其次就是应用于具有高度机密性的军事领域。在这么一个领域中，信息安全性就显得尤为重要了。这套系统有效的保证了信息的绝对安全，在本系统这个平台中就算是信息被截取，在复杂的算法加持下不法分子也不能得知用户的信息，有效的避免了军事信息的泄露，以此来避免不必要的损失。为信息安全事业添砖加瓦。

当然，本系统不仅仅是局限于此。在现在庞大的商业帝国中，文件的文件的保密性也是颇为重要的。尤其是在家喻户晓的电视剧《人民的名义》泄露之后就更加刺激了我们这类产品，让这套系统能够顺势进入市场，然后占领市场。

1.2 背景分析

随着时代的进步网络已经深入了我们的生活，已经和我们的生活密不可分，为我们的生活带来了极大的便利。在方便了我们的生活的前提下，也为我们的生活带来了极大的困扰。个人信息的泄露严重的影响了我们的生活。最常见的表现就是经常受到一些骚扰电话，像卖保险的，房屋中介之类的。台湾警方日前破获一起盗取并倒卖个人资料重大案件。该案件致使大约 1.7 亿多笔、超两千万人个人资料泄露，甚至不排

除台湾地区领导人蔡英文的个人资料。调查机构指出，嫌犯个资资料库容量高达200GB，粗估有1.7亿多笔资料，十分庞大惊人。不仅仅是我们的宝岛台湾面临着这样的窘况，在我们周围也是如此，腾讯安全发布《2017年第一季度反电信网络诈骗大数据报告》显示，全国因电信网络诈骗损失金额为33.4亿元，按13全国亿人算，人均被骗3元；从诈骗类型看，网络诈骗占比75%位居第一。

从这些赤裸裸的数据来看，网络信息的泄露已经严重的危害到了我们的经济和个人隐私安全。面对这样的周围环境，对于我们个人而言，应该尽量要避免网上各种单据的随意丢弃，像快递单、火车票和购物对账单等。因为这些单据里面包含了我们太多的个人信息；其次就是尽量避免网上聊天互动不经意“出卖”朋友或被朋友“出卖”，在这种公众的社交平台，尽量不要在昵称或者评论里出现自己或者他人的信息，因为有可能因为不经意的一句话就让一些“有心人”记在心里；最后就是身份证等个人信息要保管好;

对于信息安全，国家已经信息安全上升为国家战略。2014年2月27日，中央网络安全和信息化领导小组宣告成立，中共中央总书记、国家主席、中央军委主席习近平亲自担任组长，李克强、刘云山任副组长。这是中国共产党落实十八届三中全会精神的又一重大举措，既表明了网络信息安全目前面临的形势任务复杂和所处地位的重要，也标志着中国已把信息化和网络信息安全列入了国家发展的最高战略方向之一。所以面对这样的大环境，网络信息安全已经迫在眉睫。

在现在的这么一种网络环境下即使我们自己有了很大的防范意识，但是个人信息的泄露事件导致的一系列网络诈骗还在不时发生。这是为什么呢？因为网络已经深入了我们的生活细枝末节，我们已经离不开网络的世界了。我们在使用网络的过程中，又不能避免个人信息的传输。所以为了解决我们面临的这种矛盾的情况，我们必须将网络安全重视起来。

第二章 作品设计与实现

2.1 实现原理

2.1.1 提取特征值（pHash）

pHash。它将均值的方法发挥到极致。使用离散余弦变换(DCT)来获取图片的低频成分。离散余弦变换（DCT）是种图像压缩算法，它将图像从像素域变换到频率域。然后一般图像都存在很多冗余和相关性的，所以转换到频率域之后，只有很少的一部分频率分量的系数才不为 0，大部分系数都为 0（或者说接近于 0）。

2.1.2 局部敏感哈希（LSH）

在很多应用领域中，我们面对和需要处理的数据往往具有海量和高维的性质，如何快速地从海量地高维数据集合中找到与某个数据最相似的一个数据或多个数据成为了一个难点和问题。对此，我们通常采用一些类似索引的技术来加快查找过程，通常这类技术称为最近邻查找，LSH 就是这类技术中的一种。

我们知道，通过建立 Hash Table 可以使查找时间变为 O(1)，这其中的关键在于如何选取一个 hash function，将原始数据映射到相对应的桶内。实际上在对数据集进行 hash 的过程中，会发生不同的数据被映射到了同一个桶中的情况（即发生了冲突 collision），这一般通过再次哈希将数据映射到其他空桶内来解决。这是普通 Hash 方法或者叫传统 Hash 方法，其与 LSH 有些不同之处。

LSH 的基本思想是：将原始数据空间中的两个相邻数据点通过相同的映射或投影变换（projection）后，这两个数据点在新的数据空间中仍然相邻的概率很大，而不相邻的数据点被映射到同一个桶的概率很小。对原始数据集合中所有的数据都进行 hash 映射后，我们就得到了一个 hash table，这些原始数据集被分散到了 hash table 的桶内，每个桶会落入一些原始数据，属于同一个桶内的数据就有很大可能是相邻的，当然也存在不相邻的数据被 hash 到了同一个桶内。因此，如果我们能够找到这样一些 hash functions，使得经过它们的哈希映射变换后，原始空间中相邻的数据落入相同的桶内

的话，那么我们在该数据集合中进行近邻查找就变得容易了，我们只需要将查询数据进行哈希映射得到其桶号，然后取出该桶号对应桶内的所有数据，再进行线性匹配即可查找到与查询数据相邻的数据。因此将一个在超大集合内查找相邻元素的问题转化为了在一个很小的集合内查找相邻元素的问题，显然计算量下降了很多。

这些 hash function 需要满足以下两个条件：

1）如果 $d(x,y) \le d1$， 则 $h(x) = h(y)$的概率至少为 p1；

2）如果 $d(x,y) \ge d2$， 则 $h(x) = h(y)$的概率至多为 p2；

其中 d(x,y)表示 x 和 y 之间的距离，$d1 < d2$， h(x)和 h(y)分别表示对 x 和 y 进行 hash 变换。

满足以上两个条件的 hash functions 称为(d1,d2,p1,p2)-sensitive。而通过一个或多个(d1,d2,p1,p2)-sensitive 的 hash function 对原始数据集合进行 hashing 生成一个或多个 hash table 的过程称为局部敏感哈希。

2.1.3 AES 加密算法

AES 加密算法是密码学中的高级加密标准（Advanced Encryption Standard，AES），又称 Rijndael 加密法，是美国联邦政府采用的一种区块加密标准。这个标准用来替代原先的 DES，已经被多方分析且广为全世界所使用。Rijndael 密码的设计力求满足以下 3 条标准:

① 抵抗所有已知的攻击。

② 在多个平台上速度快，编码紧凑。

③ 设计简单。

当前的大多数分组密码，其轮函数是 Feistel 结构。

2.2 实现方案

2.2.1 图片特征提取模块

（1）缩小尺寸：pHash 以小图片开始，但图片大于 8*8，32*32 是最好的。这样做的目的是简化了 DCT 的计算，而不是减小频率。

（2）简化色彩：将图片转化成灰度图像，进一步简化计算量。

（3）计算 DCT：计算图片的 DCT 变换，得到 32*32 的 DCT 系数矩阵。

（4）缩小 DCT：虽然 DCT 的结果是 32*32 大小的矩阵，但我们只要保留左上角的 8*8 的矩阵，这部分呈现了图片中的最低频率。

（5）计算平均值：如同均值哈希一样，计算 DCT 的均值。

（6）计算 hash 值：这是最主要的一步，根据 8*8 的 DCT 矩阵，设置 0 或 1 的 64 位的 hash 值，大于等于 DCT 均值的设为”1”，小于 DCT 均值的设为“0”。组合在一起，就构成了一个 64 位的整数，这就是这张图片的指纹。

2.2.2 LSH 模块

离线建立索引

（1）利用 Jaccard distance 进行相似度计算。

（2）选取满足(d1,d2,p1,p2)-sensitive 的 LSH hash functions；

（3）根据对查找结果的准确率（即相邻的数据被查找到的概率）确定 hash table 的个数 L，每个 table 内的 hash functions 的个数 K，以及跟 LSH hash function 自身有关的参数；

（4）将所有数据经过 LSH hash function 哈希到相应的桶内，构成了一个或多个 hash table；

2. 在线查找

（1）将查询数据经过 LSH hash function 哈希得到相应的桶号；

（2）将桶号中对应的数据取出；（为了保证查找速度，通常只需要取出前 2L 个数据即可）；

（3）计算查询数据与这 2L 个数据之间的相似度或距离，返回最近邻的数据；

2.2.3 桶号的确定.

对于一个向量 P，得到一个 L 维哈希值，即 P|G，其中 L 维中的每一维都对应一个哈希函数 h。由于直接以上步中得到的 L 维哈希值做桶标号不方便，因而再进行第二步哈希，第二步哈希就是普通的哈希，将一个向量映射成一个实数。

$$h(v_1,...,v_k) = a_1.v_1 + ... + a_d.v_k \mod M$$

其中，a 是从[0,M-1]中随机抽选的数字。这样，就把一个向量映射到一个桶中了。

2.2.4 数据加密模块

数据加密采用了 AES 算法，将明文数据加密成密文数据。

2.2.4 去重模块

相同图片有相同的桶号，首先判断是否有相同桶号的图片，如果有就提示客户端，客户端发送发送随机序列验证（取出对应密文，然后利用余弦距离进行再次判定，如一致，则不上传图片。）

2.3 软件流程

上传图片去重见图 2.1；相似度查询见图 2.2

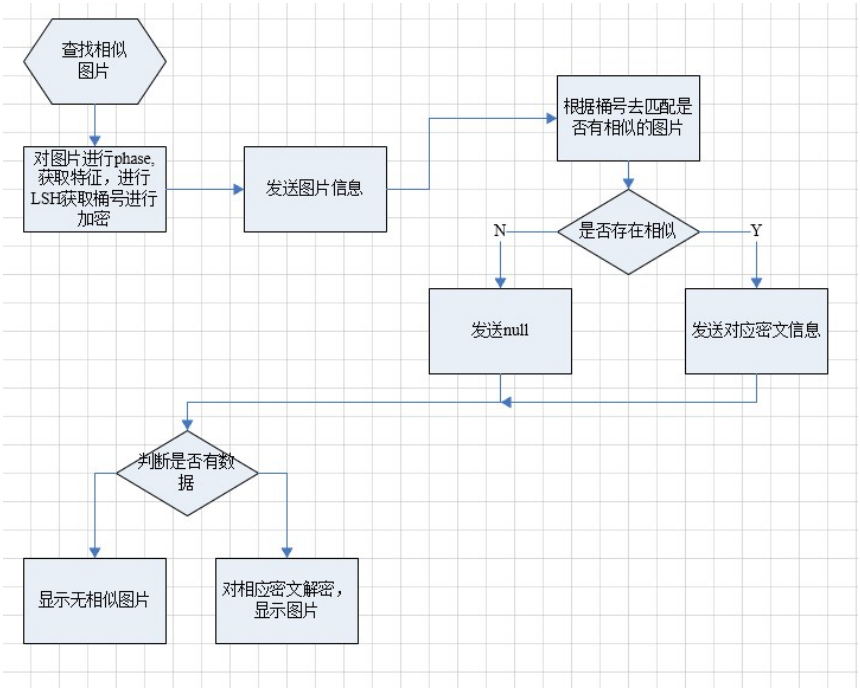

图 2.1 上传图片去重

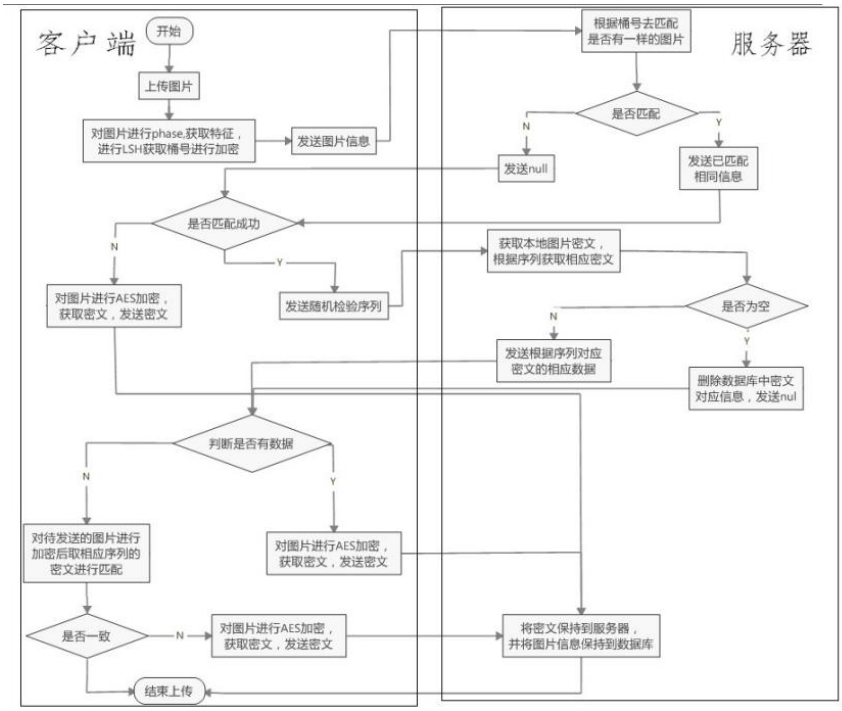

图 2.2 相似度查询

2.4 软件系统的实现

在软件系统的搭建中，服务器端使用的是阿里云服务器，以达到持续为业务发展提供完整的计算、存储、安全等解决方案；在服务器端我们需要进行以下的配置：

2.4.1 创建 Linux 实例

1 登录云服务器管理控制台。

2 定位到云服务器 ECS>实例。单击创建实例。

3 选择付费方式

4 选择地域

5 选择可用区

6 选择网络类型

7 选择实例

8 选择网络宽带类型

9 选择镜像

10 选择操作系统

11 选择存储、系统盘为必选，用于安装操作系统。您还可以根据业务需求，选择添加最多 4 块 数据盘，每块数据盘最大 32 TB。

12 设置实例的登录密码和实例名称。

13 单击页面右侧价格下面的 立即购买

实例创建好之后，您会收到短信和邮件通知，告知您的实例名称、公网 IP 地址、内网 IP 地址等信息。您可以使用这些信息登录和管理实例。

很多重要的信息都是通过绑定手机的短信接收，并且重要的操作（如重启、停止等）都需要手机接收验证码，因此请务必保持绑定手机通信畅通。

2.4.2 远程连接 Linux 实例

① 安装远程连接软件（putty）连接实例。下载地址：http://www.chiark.greenend.org.uk/~sgtatham/putty/

② 启动 Putty.exe 程序，进入 Putty 主界面。

③ 在 Host Name 中输入实例的公网 IP 地址。

使用默认端口 22。

在 Connection Type 中，选择 SSH。

在 Saved Session 中输入希望保存的名字，然后单击 Save 按钮，这样以后可以方便地调用而不需要每次输入 IP 地址。

④ 单击 Open 按钮进行连接。如下图 2.3 所示：

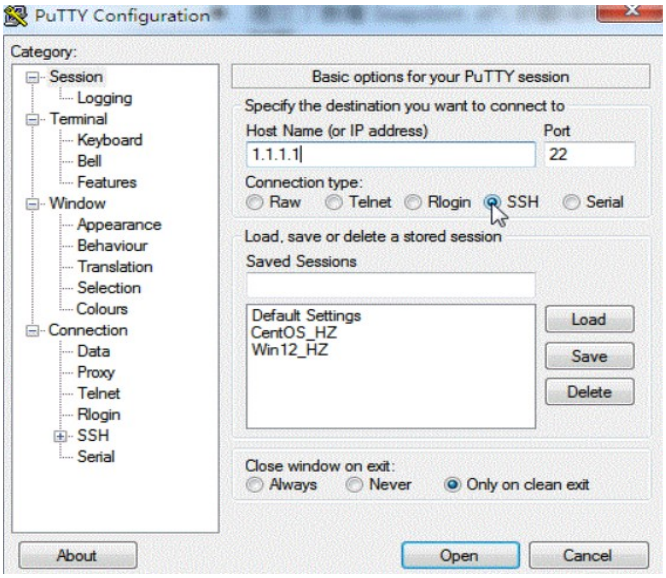

图 2.3 系统链接

⑥ 根据提示，分别输入您的 Linux 云服务器 ECS 实例的用户名和密码。输入完成后回车。如下图 2.4 所示：

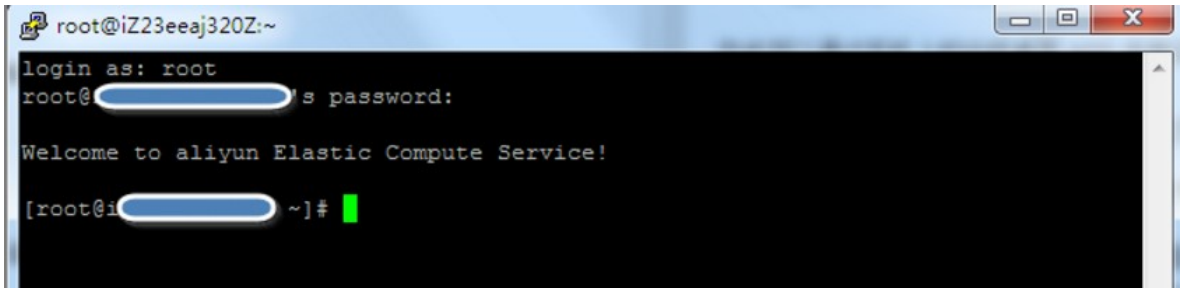

图 2.4 密码输入

2.4.3 安装 jre 及其配置

1.下载 jre1.8.0_25

2.安装 rpm -ivh jre-8u25-linux-x64.rpm

3.配置环境变量

文件默认安装于 /usr/java 下，接下来需要设置环境变量，让 linux 能找到 jdk 和 jre，环境变量设置方法为（这里给所有用户设置环境变量）：

vi /etc/profile

把以下内容加入文件最后：

########## JDK ##############

export JAVA_HOME=/usr/java/jdk1.8.0_25

export PATH=$PATH:$JAVA_HOME/

export CLASSPATH=.:$JAVA_HOME/lib/tools.jar:$JAVA_HOME/lib/dt.jar

########## Jre ##############

PATH=$PATH:/usr/java/jre1.8.0_25/bin

export JAVA_HOME=/usr/java/jre1.8.0_25

export CLASSPATH=$JAVA_HOME/lib

4. jdk 安装还需执行：

ln -sf /opt/java/jdk1.8.0_25/bin/java /usr/bin/java

ln -sf /opt/java/jdk1.8.0_25/bin/javac /usr/bin/javac

5.重启 linux 完成安装。

6.测试安装结果：

分别运行一下指令测试 jdk 安装是否成功

echo $JAVA_HOME

echo $PATH

echo $CLASSPATH

java

javac

运行 Java -version，测试 jre 是否安装成功。

2.4.4 安装 MySQL 及配置（yum）

1.配置 YUM 源

在 MySQL 官网中下载 YUM 源 rpm 安装包：http://dev.mysql.com/downloads/repo/yum/

```
# 下载 mysql 源安装包
shell> wget http://dev.mysql.com/get/mysql57-community-release-el7-8.noarch.rpm
# 安装 mysql 源
shell> yum localinstall mysql57-community-release-el7-8.noarch.rpm
检查 MySQL 源是否安装成功
shell> yum repolist enabled | grep "mysql.*-community.*"
```

2.安装 MySQL

```
shell> yum install mysql-community-server
```

3.启动 MySQL 服务

```
shell> systemctl start mysqld
```

4.开机启动

```
shell> systemctl enable mysqld
shell> systemctl daemon-reload
```

5. 修改 root 默认密码

mysql 安装完成之后，在/var/log/mysqld.log 文件中给 root 生成了一个默认密码。通过下面的方式找到 root 默认密码，然后登录 mysql 进行修改：

```
shell> grep 'temporary password' /var/log/mysqld.log
```

```
shell> mysql -uroot -p

mysql> ALTER USER 'root'@'localhost' IDENTIFIED BY 'MyNewPass4!';
```

注意：mysql5.7 默认安装了密码安全检查插件（validate_password），默认密码检查策略要求密码必须包含：大小写字母、数字和特殊符号，并且长度不能少于 8 位。

6.添加远程登录用户

默认只允许 root 帐户在本地登录，如果要在其它机器上连接 mysql，必须修改 root 允许远程连接，或者添加一个允许远程连接的帐户，为了安全起见，我添加一个新的帐户：

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'yangxin'@'%' IDENTIFIED BY 'Yangxin0917!' WITH GRANT OPTION;
```

7.配置默认编码为 utf8

修改/etc/my.cnf 配置文件，在[mysqld]下添加编码配置，如下所示：

```
[mysqld]
character_set_server=utf8
init_connect='SET NAMES utf8'
```

重新启动 mysql 服务，查看数据库默认编码。

2.5 技术指标

LSH 在线查找时间由两个部分组成：（1）通过 LSH hash functions 计算 hash 值（桶号）的时间；（2）将查询数据与桶内的数据进行比较计算的时间。因此，LSH 的查找时间至少是一个 sublinear 时间。为什么是“至少”？因为我们可以通过对桶内的属于建立索引来加快匹配速度，这时第（2）部分的耗时就从 O(N)变成了 O(logN)或 O(1)（取决于采用的索引方法）。

LSH 为我们提供了一种在海量的高维数据集中查找与查询数据点（query data point）近似最相邻的某个或某些数据点。需要注意的是，LSH 并不能保证一定能够查找到与 query data point 最相邻的数据，而是减少需要匹配的数据点个数的同时保证查找到最近邻的数据点的概率很大。

第三章 作品测试与分析

3.1 测试方案

为测试系统性能的技术指标，选取了 55.7KB、274KB、480KB、1.4MB、4.0MB 和 7.8MB 共 6 个文件分别对系统的信息预处理(特征提取，获取桶号)，加密上传，加密下载进行测评，分别记录各个功能所花销的世界与文件大小进行相除计算出速度指标，最后计算各个指标平均值

3.1.1 性能测试

3.1.2 测试环境

客户端	服务器
Windows 7 Intel core i5-4210M @ 2.6GHz 8G 内存 ADATA SP900 128GB Java 1.8	CentOS6.5 64 位 CPU1 核，内存 2G， 带宽 1M，系统盘 40G Java 1.8 (阿里云服务器)

3.1.3 测试环境的搭建

安装 Java JDK：

① http://www.oracle.com/technetwork/java/javase/downloads/index.html，登录网站下载最新的 jdk，建议使用 1.8

② 然后双击打开，进入默认打开，一直默认点击下一步，直至完成。至此 jdk 安装成功。

③ 配置环境变量,右键我的电脑，选择属性—>高级系统设置—>选择系统环境变量—>新建系统环境变量（变量值为 Java 安装路径下的 lib 文件夹路径）

④ 在系统环境变量里，点击“path”编辑,把%JAVA_HOME%/bin;加到最前面，点击确定完成

⑤ 至此，jdk 安装完成，我们来验证是否安装正确，win+R 或者 点击开始-->运行 输入“cmd”,打开系统命令提示框，输入"java -version",出现如图所示，说明安装成功。

3.1.4 测试数据与结果

测试三个性能指标的原始数据如下表 3.1：

表 3.1 性能指标

大小 M	预处理时间/s	上传加密时间/s	下载解密时间/s
0.054394531	0.17	0.22	0.23
0.267578125	0.19	0.35	0.37
0.46875	0.22	0.52	0.50
1.4	0.32	2.13	3.98
4	0.49	4.53	4.75
7.8	0.68	5.15	5.23

计算得速度指标数据见下表 3.2

表 3. 2 速度指标

大小 M	预处理时间 Mbps	上传加密时间 Mbps	下载解密时间 Mbps
0.05	2.56	1.94	1.90
0.27	11.03	6.12	5.74
0.47	17.36	7.17	7.46
1.40	35.44	5.26	2.82
4.00	64.78	7.06	6.74
7.80	92.44	12.12	11.92

三个指标的平均速度为见下表 3.3

表 3.3 平均速度

性能指标	预处理时间 Mbps	上传加密时间 Mbps	下载解密时间 Mbps
平均值	37.27	6.61	6.10

3.2 功能测试

3.2.1 上传图片功能

用户从客户端选择图片进行上传，这个功能将图片的加密与提取图片特征值包括在内，基于 AES 加密算法和 Phash 分别实现了加密与特征值得提取，以此确保图片的安全性和高效的进行相似图片检索。

点击浏览图片和上传图片按钮如图 3.1

图 3.1 图片上传

在网络传输过程中，对加密之前的图片和加密之后即将保存在服务器的图片进行了比对。如果在网络传输或者其他方式下我们的图片被截取了也不会造成信息的泄露，因为这些传输数据都是经过加密处理的。密文信息如图3.2：

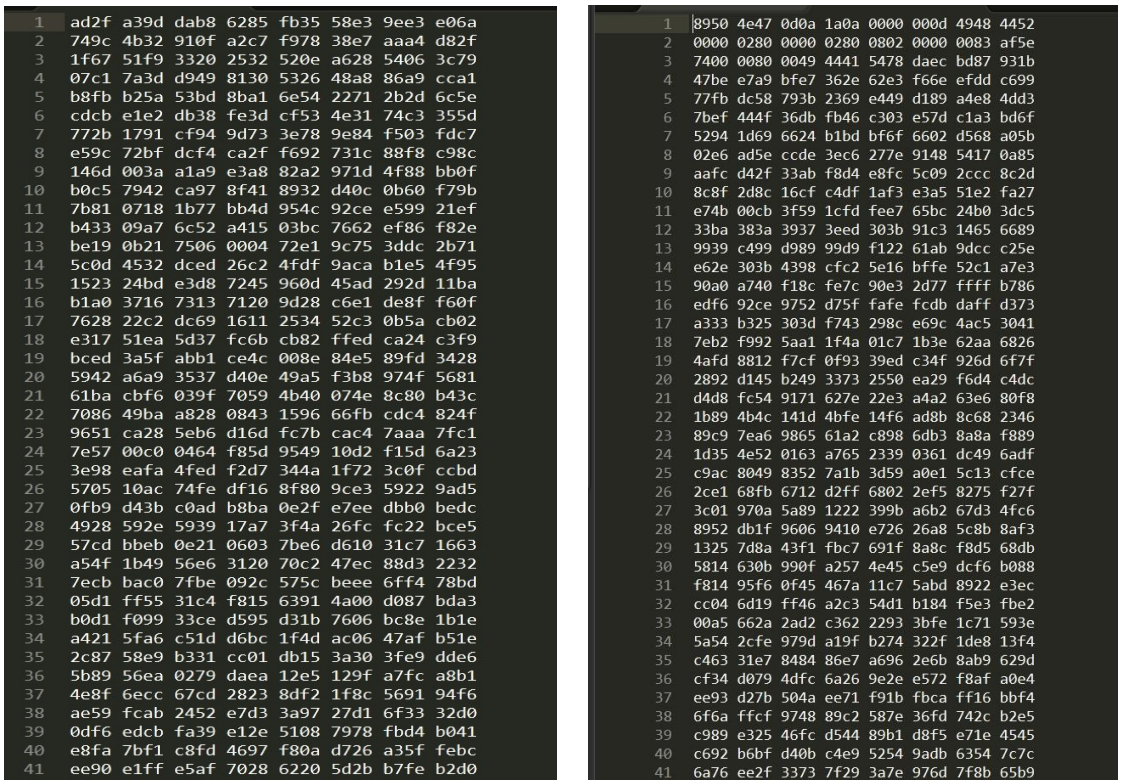

图 3.2 密文图片

上传成功的图片如下图 3.3

图 3.3 上传成功

3.2.2 查询图片功能

本系统的查询图片功能是利用图片的桶号对服务器的图片库进行检索，以达到快速检索的目的，并且在执行查询之前必须设置图片的相似度。

查询成功之后的界面如图 3.4 所示：

图 3.4 查询成功

3.2.3 验证图片功能

本系统当上传图片时会自动的对图片进行去重，当服务器有相同的图片时会提示用户服务器已经存在不能上传。此时，用户可以通过输入验证序列对是否有相同图片进行验证。验证界面如图 3.5 和图 3.6

图 3.5 验证界面

图 3.6 验证结果

第四章 创新性说明

要想保证网络的信息安全，安全的网络传输和网络安全存储是本系统探索的两大方向，本系统正是为了解决这种网络信息安全隐患而诞生的。本系统主要是针对图像的安全传输。产品在实现的时候主要是通过客户端对图像进行了加密，将本系统所需要传输的图像转换成了密文。因为数据是密文，即使在传输过程中被“攻击者”所截取到系统的信息，“攻击者”也不会知道传输的数据究竟是什么东西。由此也最大程度的保证了系统所传图像的隐私性。不仅如此，在用户将图像传到了服务器存储之后，服务器保存的也不是用户最原始的图像，而是在客户端加密之后的密文，如此一来，也不会出现服务器信息被窃取的现象。本系统正是利用这样的加密传输和加密云存储的模式来最大限度的维护用户的隐私。

此外，为了在保证用户信息安全的情形下本系统还增加了将图片与服务器是否有相似存储图片，利用了 LSH 算法和余弦相似度来快速、高效的实现相似图片匹配。不仅可以增加用户的使用需求和体验度，还能减小系统网络传输的负荷与降低服务器的数据冗余。在保证网络信息安全的前提下，不仅保证了用户的隐私和使用舒适度，还保证了客户在诸多用户下的成本。

在这种网络环境中，本系统从客户与用户全方面考虑的产品的应用前景不言而喻。在医疗领域中，为了实现医疗资源的合理利用，图像传输的使用屡见不鲜。在医院系统中很多医院的医疗条件还不能达到患者的需求，所以需要将检查的结构以图像的形式发送给等级更高的医院处理。对于患者和医院而言肯定是都不希望检查信息被泄露。“基于云计算的安全图像去重和检索系统”正是应运而生的来解决此类弊端的。

在大数据时代，数据量不断增大，使得各种存储设备十分受限，因此如何对数据进行精确的去重，特别是对密文数据的精确去重十分值的研究。

主要创新点：

1、 实现了密文图像的可验证精确去重，保护了密文图像的信息。

2、 实现了密文图像的可搜索性，在保护图像隐私信息的前提下能对图像进行模糊检索。

第五章 总结

基于云计算的安全图像检索系统由服务器和客户端所构成，包括图片加密，图片上传，图片去重，快速检索这四大模块。主要实现的功能是通过云存储为个人或者企业提供图片的上传和检索服务，其中更是增加了对图片的去重和云端密文存储的要求。在系统的整个操作中的图片的是密文交互的。有效的保证了图片在系统中的机密性，安全性和完整性。

在图片的加密模块中，是基于 AES 加密技术对图片进行加密，在保证图片安全性的同时支持图片的去重，其中在图片的去重模块中是经过取出密文，然后利用余弦距离进行再次判定，以此实现图片去重；最后在图片的检索模块是将检索的数据经过 LSH hash 实现图片的高效检索。

综上所述的四个模块，本系统实现了用户在客户端对图片进行加密，在服务器端进行密文存储，同时达到了保护用户图片机密性和数据去重的需求。言而总之，基于云计算的安全图像检索系统就是一套安全实现云存储的系统。

参考文献

[1] J. Furukawa, “Request-based comparable encryption,” in *Computer Security - ESORICS 2013 - 18th European Symposium on Research in Computer Security, Egham, UK, September 9-13, 2013. Proceedings*, vol. 8134, pp. 129–146, 2013.

[2] S. D. X., W. D., and P. A., “Practical techniques for searches on encrypted data,” in *2000 IEEE Symposium on Security and Privacy, Berkeley, California, USA, May 14-17, 2000*, pp. 44–55, SpringerVerlag, 2000.

[3] Y. Chang and M. Mitzenmacher, “Privacy preserving keyword searches on remote encrypted data,” in *Applied Cryptography and Network Security*, pp. 442–455, Springer, 2005.

[4] C. R., G. J. A., K. S., and O. R., “Searchable symmetric encryption: improved definitions and efficient constructions,” in *Proc. of the 13th ACM Conference on Computer and Communications Security, CCS 2006, Alexandria, VA, USA, Ioctober 30 - November 3, 2006*, pp. 79–88, 2006.

[5] J. Zhang, Y. Xiang, W. Zhou, L. Ye, and Y. Mu, “Secure image retrieval based on visual content and watermarking protocol,” *Computer Journal*, vol. 54, no. 10, pp. 1661–1674, 2011.

[6] J. Zhang, L. Ye, Y. Xiang, and W. Zhou, “Robust image retrieval with hidden classes,” *Computer Vision and Image Understanding*, vol. 117, no. 6, pp. 670–679, 2013.

[7] J. Guo, H. Prasetyo, and J. Chen, “Content-based image retrieval using error diffusion block truncation coding features,” *IEEE Trans. Circuits Syst. Video Techn.*, vol. 25, no. 3, pp. 466–481, 2015.

[8] P. Enser, “Progress in documentation pictorial information retrieval,” *Journal of Documentation*, vol. 51, no. 2, pp. 126–170, 1995.

[9] E. de Ves, X. Benavent, I. Coma, and G. Ayala, “A novel dynamic multimodel relevance feedback procedure for content-based image retrieval,” *Neurocomputing*, vol. 208, pp. 99–107, 2016.

[10] L. Al-Safadi, R. Alomran, and F. Almutairi, “An overview and evaluation of the radiologists lounge, a semantic content-based radiographic images retrieval,” *Multimedia Tools Appl.*, vol. 75, no. 1, pp. 607–625, 2016.

[11] W. Lu, A. Swaminathan, A. Varna, and M. Wu, “Enabling search over encrypted multimedia databases,” in *Media Forensics and Security I, part of the IS&T-SPIE Electronic Imaging Symposium, San Jose, CA, USA, January 19-21, 2009, Proceedings*, p. 725418, 2009.

[12] L. Weng, L. Amsaleg, A. Morton, and S. Marchand-Maillet, “A privacypreserving framework for large-scale content-based information retrieval,” *IEEE Trans. Information Forensics and Security*, vol. 10, no. 1, pp. 152–167, 2015.

[13] Y. Chun, N. Kim, and I. Jang, “Content-based image retrieval using multiresolution color and texture features,” *IEEE Trans. Multimedia*, vol. 10, no. 6, pp. 1073–1084, 2008

[14] Y. Chen, X. Li, A. Dick, and R. Hill, “Ranking consistency for image matching and object retrieval,” *Pattern Recognition*, vol. 47, no. 3, pp. 1349–1360, 2014.

[15] R. Bellafqira, G. Coatrieux, D. Bouslimi, and G. Quellec, “Contentbased image retrieval in homomorphic encryption domain,” in *37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBC 2015, Milan, Italy, August 25-29, 2015*, pp. 2944–2947, 2015.

[16] W. Lu, A. L. Varna, A. Swaminathan, and M. Wu, “Secure image retrieval through feature protection,” in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2009, 19-24 April 2009, Taipei, Taiwan*, pp. 1533–1536, 2009.

[17] B. Cheng, L. Zhuo, Y. Bai, Y. Peng, and J. Zhang, “Secure index construction for privacy-preserving large-scale image retrieval,” in *2014 IEEE Fourth International Conference on Big Data and Cloud Computing, BDCloud 2014, Sydney, Australia, December 3-5, 2014*, pp. 116–120, 2014.

[18] G. Liu and J. Yang, “Content-based image retrieval using color difference histogram,” *Pattern Recognition*, vol. 46, no. 1, pp. 188–198, 2013.