

# 北京理工大学

## 本科生毕业设计(论文)

基于国产密码算法的云计算网络信息传输认证系统设计与实现

Design and implementation of information transmission and authentication system in cloud computing network based on domestic cryptography algorithm

学    院:	计算机学院
专    业:	软件工程
学生姓名:	侯添久
学    号:	1120161912
指导教师:	闫怀志

2020 年 5 月 10 日

## 原创性声明

本人郑重声明：所呈交的毕业设计（论文），是本人在指导老师的指导下独立进行研究所取得的成果。除文中已经注明引用的内容外，本文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。

特此申明。

本人签名: \_\_\_\_\_ 日期: \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日

## 关于使用授权的声明

本人完全了解北京理工大学有关保管、使用毕业设计（论文）的规定，其中包括：①学校有权保管、并向有关部门送交本毕业设计（论文）的原件与复印件；②学校可以采用影印、缩印或其它复制手段复制并保存本毕业设计（论文）；③学校可允许本毕业设计（论文）被查阅或借阅；④学校可以学术交流为目的，复制赠送和交换本毕业设计（论文）；⑤学校可以公布本毕业设计（论文）的全部或部分内容。

本人签名: \_\_\_\_\_ 日期: \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日

指导老师签名: \_\_\_\_\_ 日期: \_\_\_\_\_ 年 \_\_\_\_\_ 月 \_\_\_\_\_ 日

# 基于国产密码算法的云计算网络信息传输认证系统设计与实现

## 摘 要

随着云计算的发展，越来越多的应用都在使用云计算，然而，数据的隐私安全也变得格外重要，作为云计算服务的使用者，他们不希望自己的数据服务提供者所得到，只是借助云计算强大的计算和存储能力来帮助完成一些任务。如果服务使用者发送密文数据到相应的云上，云服务对密文进行运算，得到的密文结果再发送给服务使用者，服务使用者解密拿到的结果和直接使用明文进行相同的操作得到的结果是相同的，若密码算法具备这种性质，则称为同态加密算法，这也是目前解决云计算数据隐私安全最强有力的手段。

本课题主要研究国产密码算法的同态性，研究的密码算法主要是非对称加密算法 SM2，对称密码算法 SM4，数字摘要算法 SM3 三种密码算法，通过实验确定其是否具有同态性。

研究主要通过 socket 通信，通信双方分为发送者和接收者模拟用户和云服务器，发送者为客户端，接收者为服务端，对数据的加密操作在客户端进行，客户端再将加密后的数据通过网络发送给服务端，服务端对客户端传递来的数据进行某种计算，并且将得到的密文结果返回给客户端，客户端解密后拿到结果，再判断与使用明文进行相应计算得到的结果是否相同。通过此种方式模拟用户数据传输到云服务器，云服务进行只进行相应的计算操作，客户端拿到密文后解密得到明文的结果。

研究最后得出在国产加密算法 SM4 的 ECB 模式下，使用分割字符串的方法拿到的结果解密之后与明文运算的结果相同。

**关键词：国产密码算法；同态加密技术；云计算；网络通信**

## **Design and implementation of information transmission and authentication system in cloud computing network based on domestic cryptography algorithm**

### **Abstract**

With the more and more scene use cloud computing, but the security of data has also become an very important problem. As users of cloud computing services, they do not want their data to be gotten by cloud computing providers, and they only want to use the powerful computing of cloud computing to help complete some problems. If the service users send the ciphertext data to the remote cloud, and the cloud service calculates result using the ciphertext and sends the ciphertext result to user. Users decrypt ciphertext result and get the plaintext result. It is the same as execute those operation on the plaintext directly. The cipher algorithm with this property is called homomorphic encryption algorithm. This is one powerful way to solve the problem of data security on the cloud.

In this paper, we mainly study the homomorphism of domestic cryptography algorithms. We study some existing domestic cryptographic algorithms. They are SM2, SM3, and SM4.

In this paper, I use socket to communication. Coding the socket of client and server independently, then encrypting the data in client, then client sends the encrypted data to server. Server calculates result, and returns the encrypted result to the client. Client decrypts the result, then judges if it is the same as using the corresponding operation in plaintext. In this way, It is simulated to transmit data to the cloud server, and the cloud service only performs the corresponding computing operation. The client gets the ciphertext result and decrypts it to get the plaintext result.

At last, it is concluded that under the ECB mode of the domestic encryption algorithm SM4, the result by using the method of splitting string is the same as result using plaintext to operate.

**Key Words: Domestic Cryptography Algorithm; Homomorphic Encryption; Cloud Computing; Network Communications**

## 目 录

基于国产密码算法的云计算网络信息传输认证系统设计与实现.....	I
摘 要.....	I
Abstract.....	II
第 1 章 绪论.....	1
1.1 研究的背景与意义.....	1
1.2 研究的主要内容与方法.....	2
1.3 组织结构.....	2
1.4 本章小结.....	3
第 2 章 密码算法及同态性.....	4
2.1 密码算法简介.....	4
2.1.1 对称密码算法.....	4
2.1.2 非对称密码算法.....	5
2.1.3 数字摘要算法.....	5
2.1.4 应用简介.....	6
2.2 加密算法的同态性.....	7
2.2.1 完全同态加密技术.....	8
2.2.2 部分同态加密技术.....	8
2.2.3 应用简介.....	9
2.3 分组密码中的四种模式.....	9
2.3.1 ECB 模式.....	9
2.3.2 CBC 模式.....	10
2.3.3 CFB 模式.....	10
2.3.4 OFB 模式.....	11
2.4 本章小结.....	11
第 3 章 国产密码算法.....	12
3.1 国产非对称加密算法.....	12
3.1.1 SM2 加密过程.....	12

## 北京理工大学本科生毕业设计（论文）

---

3.1.2 SM2 解密过程.....	12
3.1.3 SM2 算法实现.....	13
3.2 国产对称加密算法.....	13
3.2.1 SM4 加解密过程.....	13
3.3.2 SM4 算法实现.....	14
3.3 国产数字摘要算法.....	14
3.3.1 SM3 加解密流程.....	14
3.3.2 算法实现.....	15
3.4 本章小结.....	15
第四章 网络通信.....	16
4.1 网络通信简介.....	16
4.2 传输层通信.....	17
4.2.1 TCP.....	17
4.2.2 UDP.....	18
4.3 数据传输过程.....	19
4.3.1 数据流转.....	19
4.3.2 零拷贝.....	20
4.4 网络通信实现.....	20
4.5 本章小结.....	21
第五章 实验流程.....	22
5.1 实验方案.....	22
5.2 实验流程.....	22
5.3 实验结果.....	25
5.4 本章小结.....	25
结 论.....	26
参考文献.....	27
致 谢.....	28

## 第1章 绪论

### 1.1 研究的背景与意义

近二十年来, 云计算技术的高速发展, 使得我们的生活发生了翻天覆地的变化, 并且也对许多传统行业影响是深远的。云计算的发展给我们带来方便的同时, 也随之带来了严峻的隐私数据安全问题。近年来, 云计算时常发生严重的安全问题, 其中用户隐私数据安全态势显得格外严峻。

2018年, 国外知名互联网公司Facebook被曝出发生数据泄漏事件, 3月Facebook上5000万名用户个人信息遭一家名为剑桥分析公司的泄露; 在9月Facebook上有大约3000万用户信息泄露, 导致这次是事件的原因是安全系统存在一定的漏洞, 而该漏洞被黑客攻击; 在12月Facebook上有大约有6亿人的私隐照片被泄露。

2019年8月, 美国的大型商业银行Capital One发生数据泄露, 黑客利用他们基础设施中一个特定的配置漏洞, 获得了约1亿美国人以及600万加拿大人的基本信息, 这些信息涵盖了2005年至2019年初的个人信息, 其中包括大约14万个社会安全号码和8万个关联银行账号。

由上述众多数据泄露事件可以看出, 在云计算, 大数据等新的科学技术越来越多的应用在我们平时的生活中, 其所服务的用户数据量回事越来越多, 相应地用户的隐私数据被泄露的可能性也随之增高。随着云计算成为众多企业的首选数据存储处理方案, 其应用场景也是一直在变化, 云计算服务提供者需要保证在任一场景下用户隐私数据都是安全的, 因此云计算不得不面对一个巨大的难题, 即如何在保证用户数据安全的基础上进行相应的数据处理操作。一般地, 我们为了使得数据更加安全, 往往会对用户数据进行加密, 然后将加密后的密文发送给云服务商, 云服务商得到的只会是数据密文, 但是用户数据传输到云服务商处是需要对其进行相应的计算操作, 因为其是密文无法进行相应的操作, 即使进行计算, 也会使得得到的结果无法正确解密亦或得不到预期结果。这种情况下, 云服务只是单单存储数据, 并没有用到其巨大的计算能力。如果使的加密算法具有同态性, 就可以在不解密情况下对密文进行任意的计算操作, 此时对密文的处理等价于对明文的处理, 用户拿到密文结果解密后依然会得到预期明文结果, 通过同态密码算法可以解决目前云计算中的用户隐私数据安全问题。如此云的算力可以很好地发挥出来, 同时又解决了用户隐私数

据安全问题。

## 1.2 研究的主要内容与方法

本课题研究国产加密算法在云计算网络信息传输认证的设计与实现，主要是基于现有的国产密码算法，通过实验判断其是否具有同态特性。密码算法的同态性是指对于密文进行某种操作，得到相应的密文结果，对密文结果进行解密后得到的结果与直接使用明文进行运算得到的结果是一样的，即对密文的操作等价于在明文上进行相同的操作，满足此类性质的密码算法称为加密算法的同态性。

当在云计算中使用的加密算法具有同态性时，我们可以在客户端对数据进行加密，将密文发送给云服务，云服务根据密文进行相应的计算，返回给我们需要的密文结果，客户端对收到的密文解密。整个过程对于云服务来说，都是在对密文进行操作，不需要进行任何的解密，也无法拿到密文对应的明文，即使数据泄露也可以保证数据的隐私安全，所以密码算法的同态性可以保证我们隐私数据的安全。

研究国产密码算法的同态性，主要采用socket实现客户端和服务端，模拟实际的环境，在客户端加密，将加密后的结果通过socket发送给服务端，服务端只是对密文进行计算，最后返回给客户端，客户端解密拿到结果。

## 1.3 组织结构

本课题主要研究国产密码算法在云计算网络中信息传输认证的原型设计与实现，论文一共分为五个章节。

第一章主要介绍了研究的背景和意义，阐述了目前云计算所存在的问题，研究的主要内容和方法的简要概述以及论文的组织结构等。

第二章研究了密码算法的分类，主要分为对称密码算法，非对称密码算法以及数字摘要算法。同时对密码算法的同态性，分组加密的模式介绍。

第三章研究相应的国产密码算法的加解密过程，SM2国产非对称密码算法、SM3国产数字摘要算法、SM4国产对称密码算法。

第四章研究了网络通信，主要对网络数据传输的socket做了详细的介绍，同时介绍了现在五层和七层的网络模型。

第五章主要介绍、详细的实验方案，实验的过程以及最后得到的实验结论。



#### 1.4 本章小结

本章主要介绍了云计算目前存在的一些问题、研究的背景以及研究的意义。简短阐述了研究的方案。

下一章主要对论文中使用的密码算法做整体的阐述、具体国产密码算法的研究、密码算法的同态性的介绍以及分组密码算法工作模式的介绍。

## 第2章 密码算法及同态性

### 2.1 密码算法简介

密码算法主要就是一种数学函数，将我们需要发送的数据经过相应的数学函数进行计算得到与之前数据不同的数值，将得到的数值发送给通信的另一方，通信的另一方收到该数值后，需要使用该数学函数的反函数进行计算得到原来的数据。上述发送者通过数学函数计算的过程称为加密，接收者使用反函数计算的过程称为解密。一般地，使用的加密算法需要一定的可逆性，否则无法解密得到结果。

明文消息 $M$ 、加密函数 $E$ 、解密函数 $D$ 、密文 $C$ 会满足以下性质：

$C = E(M)$ ，表示对明文消息 $M$ 使用加密函数 $E$ ，得到密文 $C$ 。

$M = D(C)$ ，表示对密文 $C$ 使用解密函数 $D$ ，得到明文 $E$ 。该形式等价于 $M = D(E(M))$ 。

对称密码算法，非对称密码算法，数字摘要算法是目前密码算法主要三大类。

#### 2.1.1 对称密码算法

对称加密算法是指加密的所使用密钥与解密所使用的密钥是相同的，在一般的网络数据传输中，数据的加密都是采用对称加密算法。通过密钥我们可以控制加密和解密的流程，算法相当于数学函数，加密和解密操作都需要通过该数学函数实现。计算量小，加密速度很快，加密效率高是对称加密的优点。但是使用对称加密算法加密数据，如果密钥被第三方得知，那么作为第三方可以获取通信双方传输的内容，所以只使用对称加密的安全性不高。因此对称加密的安全性不能只考虑所使用的加密算法还需要考虑密钥如何去高效管理不会造成密钥泄露。因为加密和解密都使用同一个密钥，所以当时使用非对称加密时，需要考虑如何安全的去分发密钥到解密者手中。密钥管理是对称加密技术的关键，密钥安全性的保证非常重要。

常见的对称加密算法有DES，AES，3DES，IDEA，PBE等。其中DES是数据加密的标准，加密的速度较快，对于需要加密的数据量很大的场景可以使用DES；AES是下一代的加密算法的标准，加密速度很快，并且安全级别高，加密的密钥的位数也是不同的；3DES是三重DES的简称，其基础还是DES，相比于DES，3DES通过三个不同的密钥对一个数据组加密，从而使得其安全性高于普通的DES；IEDA常用于电子邮件的加密中，其工作模式只有ECB；PBE是综合了消息摘要算法和对称加密算法，工作模

式只有CBC，不具有安全性。一般地，对称加密因为其加解密速度快，效率高，常用在对数据进行加密中，但是因为其加密密钥和解密密钥使用的同一个密钥，密码算法的安全性也存在一定的挑战，可以采用多级密钥管理或者配合其他加密算法使用。

### 2.1.2 非对称密码算法

非对称加密是指加密所使用的密钥和解密所使用的密钥是不同的。一般地，在非对称加密中会有公钥和私钥两种密钥，其中公钥是可以被别人得到的，用来对将要传输的数据进行加密，私钥用来对接收到的数据进行解密。非对称加密的安全性很高，但是因为存在公钥和私钥两种密钥，计算量较大，加密速度较慢，加密的效率较低。非对称加密还需要考虑公钥的分发策略，其更多的使用在通信双方的身份认证中。

常见的非对称加密算法有RSA，ECC，DSA等。RSA主要是基于大素数分解，其安全性主要与其密钥长度相关，密钥长度越长，安全性越高，但是需要考虑到程序计算的开销，密钥长度的选择是需要从安全性和程序性能做一个相应的平衡；ECC的原理是在椭圆曲线上的有理点构成Abel加法群上椭圆离散对数的计算非常困难，其主要的优势在于可以使用更短的密钥，提供相当或更高等级的安全性。在给定密钥长度的条件下，ECC是安全性最强的非对称加密算法，在某些对带宽具有要求的场景时可以使用该算法；DSA 是基于整数有限域离散对数难题，DSA不仅仅具有私钥和公钥，还具有数字签名。非对称加密算法也可以作为数字摘要算法使用，数字签名通过私钥生成，数据和签名的验证通过公钥实现。数据在网络中传输可能会被修改，当另一方收到数据后，需要通过数字签名来判断数据是否被修改。非对称加密因为其计算的开销非常大，存在公钥和私钥两个密钥，如果单纯使用非对称密码算法加密数据，需要考虑密钥分发策略，程序的性能也会下降。一般地，非对称加密多用在身份认证中，用来确认通信双方的身份。

### 2.1.3 数字摘要算法

数字摘要通过某个数学函数，以需要传输的明文数据为参数，得到一个固定长度的消息，通过生成的固定长度的消息可以判断数据是否被修改，数字摘要函数也被称为Hash函数。数字摘要所使用的函数是单向不可逆的，并且对数据的修改是敏感的，对于不同的明文数据，通过同一数字摘要函数得到的消息总是不同的。通过此种性质可以在网络通信中保证数据的完整性，因为其是单向不可逆的，当网络

中的密文数据包被第三方获得，纵使第三方拿不到相应的明文数据，但是可以对其进行增减操作，从而改变了数据的内容。通过数字摘要，在发送数据时便对明文进行计算出相应的数字摘要，发送给通信另一方，对方收到数据解密后，使用相同的数字摘要算法计算出相应的数值与传送来的进行比对，如果相同，则可以证明没有修改过，若是不同，那么数据在传输中被截取修改过。

常用的数字摘要算法有md5，SHA-1等。其中md5是一种较为常用的散列函数，其通过相应的摘要函数会产出一个128位的散列值，通过该散列值可以确保信息传输的一致性，但是无法防止碰撞，所以不适用于安全性认证；SHA-1是一种密码散列函数，它可以生成一个被称为消息摘要的160位散列值，通常是由40个十六进制整数表示。数字摘要算法常用在判断数据是否被修改过，主要应用在网络数据传输，下载文件等等。一般地，在网站上的文件可能在下载时会被攻击者篡改，当我们下载网站上的文件打开后，可能会破坏我们的计算机系统，在下载完文件后，应该对比hash值是否与网站上文件显示的hash值相等，相等则证明没有被修改，否则，文件可能被修改了。

### 2.1.4 应用简介

现如今的网络信息时代，我们常常会使用浏览器或者一些app来浏览网页，浏览网页的主要内容是HTML，CSS等，但是这些文件是放在远程服务器上，需要通过网络传输到我们的浏览器，浏览器解析后才会显示出我们所看到的内容。现如今，很大部分的浏览器都使用了HTTPS协议，因为其可以确保数据在网络中传输的安全性，主要就是通过上述的密码算法来实现网络信息传输的安全。

HTTPS较HTTP更为安全，其安全性主要体现在数据的加密传输，身份认证以及hash值保证数据完整性。数据在网络中传输，可能会被第三方获得传输的数据包，如果对传输的内容使用对称加密算法进行加密，第三方拿到的内容也只会是密文，而无法拿到对应的明文。当网络中的数据被第三方拿到，其拿到的是密文，虽然无法看到具体内容，但是可以对内容做一定程度的破坏，增加、删除或修改任意一段密文，就会造成信息的混乱，这时就需要使用相应的数字摘要算法，在数据发送之前的明文作一次计算获得哈希值，另一端再收到数据解密后使用相同的数字摘要算法计算得到一个哈希值，比较两个哈希值是否相同给，若是不相同，则证明数据在传输过程中被修改，否则，则说明数据未被修改。网络中通信双方在一开始需要建

立连接，确认身份，常用的做法是通过非对称加密实现，一端通过公钥加密一段相应的内容，然后发送给另一端，另一端通过私钥解密，并且相应。对于使用非对称密码算法进行通信的双方，需要考虑公钥的分发策略，如何获取到公钥，一般地公钥都会放在一些可信的机构，比如：公钥存放在CA证书中，请求方拿到CA证书，通过CA证书解析出需要的公钥即可。

HTTPS建立连接的过程如下：

用户端生成一个随机数1，并将自己的所支持的密码算法列表、数字摘要函数列表和该随机数1发送给服务端。

服务端接收之后，选择适当的密码算法和摘要函数，并生成一个随机数2，之后将CA证书，随机数2发送给用户端。

用户端接收之后，先进行CA证书验证，如果CA证书有效，可以从CA中获取其公钥，然后生成随机数3，并使用公钥加密该随机数3，发送给服务端，此时，只有服务端相应的私钥可以解密得到该随机数。

服务端接收到相应的数据后解密得到该随机数，根据双方三个随机数计算确定对称密码的密钥，之后通信双方就可以进行相应的数据传输。

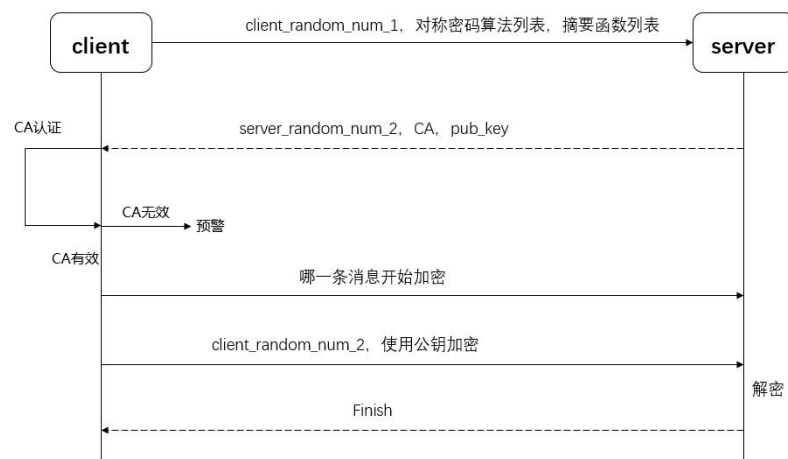


图 2-1 HTTPS 加密流程

HTTPS只可以保证数据在网络中传输的安全性，但是在云环境中，我们需要保证用户的一些隐私数据对于云服务提供商不可见，同时当发生数据泄露，需要保证用户数据隐私性，这就需要通过密码算法的同态性来实现。

## 2.2 加密算法的同态性

目前云计算平台对于用户数据只是简单的以密文的形式存储，无法对密文做任

何操作，云服务商对密文的任何操作，都可能导致解密后无法得到预期的结果。同时，由于近年来频繁发生用户隐私数据泄露，云计算的安全问题显得日益严峻。云计算拥有的是强大的计算能力，如果可以对密文进行任意操作的效果等价于对明文的操作，在解密后的结果与直接使用明文计算得到结果一致，就可以完美解决云计算目前存在的问题，这也是同态加密技术的核心。

### 2.2.1 完全同态加密技术

完全同态加密算法是指对于密文进行任何的计算操作，得到的结果解密后与直接使用明文计算拿到的结果一致，主要强调对于执行的计算操作没有限制，任何的计算操作都可以。

完全同态加密系统是一种允许客户端除了进行基本的加密与解密操作之外，还可以在加密数据上进行任何的计算操作而不需要解密数据。假设，我们有两个明文  $m_1$  和  $m_2$ ， $c_1 = \varepsilon(m_1)$  和  $c_2 = \varepsilon(m_2)$  分别是其加密后的密文，完全同态加密可以让我们得到  $c_1 + c_2 = \varepsilon(m_1 + m_2)$  和  $c_1 \cdot c_2 = \varepsilon(m_1 \cdot m_2)$ ，因此，用户可以将自己复杂的计算交给远程的云服务器去完成，云服务在它自己的运行环境中可以提供无限制的计算。

### 2.2.2 部分同态加密技术

部分同态加密算法是指对于密文进行某些特定计算操作，得到的结果解密后与直接使用明文进行该运算拿到的结果一致，主要强调对于某种特定的计算操作，而不是对于所有的计算都满足上述性质。

常见的满足部分同态加密算法有RSA，Paillier，其中RSA满足乘法的同态性，而Paillier满足加法的同态性。对于未进行填充的RSA加密系统，可以认为是可以相乘的同态，即，对于明文  $m_1$  和明文  $m_2$ ，我们可以得到  $\varepsilon(m_1) = m_1 e \% N$  和  $\varepsilon(m_2) = m_2 e \% N$ ， $N$  是 RSA 的模数， $e$  是 RSA 公钥的指数，因此，可以得到  $\varepsilon(m_1)\varepsilon(m_2) = (m_1 e \cdot m_2 e) \% N = (m_1 \cdot m_2) e^2 \% N = \varepsilon(m_1 \cdot m_2)$ 。RSA允许我们只是仅仅对密文进行乘法操作而不能进行加法操作，关于加法的同态加密方案是Paillier的加密系统，如果公钥是模  $N$  并且是基于  $g$  的，那么加密的信息  $m$  则是  $\varepsilon(m) = g^m r^N \% N_2$ ，对于  $r$  是区间  $[1, N-1]$  之间任意的数，Paillier 加密算法的同态特性则是  $\varepsilon(m_1)\varepsilon(m_2) = (g^{m_1} r_1^N) \cdot (g^{m_2} r_2^N) \% N_2 = g^{m_1 + m_2} (r_1 r_2)^N \% N_2 = \varepsilon(m_1 + m_2)$ 。

由上可以看出，并不是所有的密码都具有同态性，具有完全同态特性的密码算法是理想化的解决方案，实际很多的密码算法都只可以满足某种计算操作的同态性，

也就是部分同态性。

### 2.2.3 应用简介

在云计算领域中，密码算法的同态性是很重要的，当企业选择使用云来部署服务以及存储数据时，都会不可避免的涉及到数据安全性问题，用户的隐私数据属于个人信息中的重要信息，不希望被其他人得知，当需要使用到用户的一些隐私数据时，此时可以利用密码算法的同态性，对隐私数据的密文进行操作，其效果等同于明文的操作，此时不需要解密使用明文，保护了用户的隐私数据。对于数据泄露的情况，如果服务端计算和存储都使用密文，那么即使数据泄露，攻击者拿到的也只是数据的密文，而无法得到明文，更加保障了数据的安全。

密码算法同态性除了用在云计算领域来保证用户数据的安全之外，还可以用于搜索，搜索很多时候都是针对明文搜索的，这种方式当发生数据泄露时，用户的隐私数据就会以明文的形式展现出来。如果使用的密码算法具有同态性，那么当用户搜索输入某个关键字后，这个关键字被加密成密文发送给服务端，服务端对密文的搜索等同于使用明文搜索，那么就不用关心数据的具体内容即可拿到预期的结果，即使数据泄露也是以密文的形式展现出来，保障了数据的安全。

## 2.3 分组密码中的四种模式

一般的对称加密算法根据加密方法的不同又分为分组加密和序列密码两个类。其中分组密码也称为块密码，对明文按一定的位长进行分组，每组称为一个块，例如：按每64位为一组，每个分组也就是一个块，加密的时候对每一个明文块进行加密，明文块经过加密运算后变为密文块，密文块经过解密运算后也会得到相应的明文块，每次以一个块的方式加解密，不需要多次使用加解密函数，减小了加解密过程中的计算量，极大的提高加解密的效率。序列密码也称为流加密，对明文每次都只加密其中的一位，这种加密方式对明文中的每一位都会通过相应的数学函数运算，相比分组加密方式，计算量增大，加解密的效率降低。

ECB，CBC，CFB，OFB是分组密码算法中常见的四种加密模式。

### 2.3.1 ECB 模式

ECB模式也被称为电子密码本模式，是最简单的一种运行模式。它先将明文按64位的长度进行分组，然后对每组分别进行加密，每组加密所使用的都是同一个密

钥。所以，在此加密模式下当加密的密钥确定后，对于每一个明文分组，都会有相应的密文与之对应。使用ECB模式进行加密操作，每组独立加密，可以认为存在一个大的密码本，每个明文都会在该密码本中与某个密文惟一对应。因为是分组加密，当明文最后一组的长度小于64位时，需要进行填充至64位。解密过程也是对密文按相同的长度分组，每次解密一个分组，加解密使用的密钥是相同的。

通过上述可以知道，ECB模式下每组加密解密操作都是独立的，不存在组与组之间影响传递的情况，当某一组加密出现错误，不会传递到后面的分组中。分组独立加解密可以并行的加解密，提高加密的速度。但是对于两个明文相同的分组，会被加密成密文相同的两个分组，所以无法抵抗统计分析攻击。所以，ECB模式适于加密小消息，例如：密钥的保护。

### 2.3.2 CBC 模式

CBC 模式也称为加密块链模式，它先将明文按一定的位长进行分组，然后对每组分别加密，加密时，需要考虑前一组加密的结果对本组的影响，前一组加密的结果与本组将要加密的明文分组进行异或操作，再使用密钥对得到的结果进行加密生成相应的密文。当明文分组中第一个明文分组加密时，由于其前面没有相应的密文分组结果，所以需要初始化一个向量与第一个明文分组进行异或操作再加密。CBC 模式加密中，各个分组之间不相互独立，前一个分组会对后一个分组有影响，对于明文分组来说，同一个明文分组会对应多个密文分组，这种加密模式破解起来更加困难，并且通过简单的调换密文分组的顺序无法造成攻击。

通过上述可知，CBC模式不容易被主动攻击，并且密文分组直接相互依赖，安全性更高，当传输较长的数据时可以使用CBC模式。但是由于相互依赖，当某一分组计算错误，会使得误差传播到后面的分组，导致后面分组计算结果不符合预期。同时相互之间的依赖不适合并行加解密，降低了加解密的速度。对于第一个分组的加密，需要确定初始化向量。

### 2.3.3 CFB 模式

CFB模式也称为加密反馈模式，它先将明文数据进行分组，前一组加密的结果会有后一组的部分数据进行异或操作，每组加密都会选择行丢弃一定长度的数据。因为数据的表现形式多样化，当需要按照字符进行加密时，就可以使用CFB模式。此模式可以对数据按任意的格式进行加密操作，当加密字符流时，加密每个字符即



可；当加密比特流时，加密其中每个比特即可。

通过上述可知，CFB模式可以加解密多种形式的密文，并且某次加密出错，误差的传播也只会是有限的。但是因为其分组间相互依赖不适合使用并行加密，第一组加密时需要初始化向量，整体导致加密的效率变低。

#### **2.3.4 OFB 模式**

OFB模式也称为输出反馈模式，该模式在结构上与CFB模式相似，它们惟一差别是CFB将上一组的密文输出作用到下一组的加密中，而在OFB中，其是将初始化向量加密的输出作用到下一组加密中。

OFB模式可以将分组密码转化为序列密码，传输过程中某一个比特出现错误，不会影响后面的结果。但是因为是相互依赖不适合并行计算，加解密效率较低。

### **2.4 本章小结**

本章主要介绍了密码算法的分类，主要的三类密码算法以及实际中的应用；对密码算法同态性做了阐述，介绍了加密的模式。

下一章主要对具体的国产密码算法做相应的介绍，SM2国产非对称密码算法、SM3国产数字摘要算法、SM4国产对称密码算法的加密解密操作做相应的介绍。

## 第3章 国产密码算法

### 3.1 国产非对称加密算法

本课题主要研究了国产非对称密码算法中的SM2加密算法。目前较为流行的非对称加密算法是RSA，但基于椭圆曲线上点群离散对数计算难题的SM2算法的安全性是高于RSA的。密钥长度越长，安全性越高，但是当SM2的密钥长度为256位时，其安全性是高于2048位的RSA。

#### 3.1.1 SM2 加密过程

用户的原始数据，椭圆曲线的系统参数，长度为 $k$ 比特的消息 $m$ 以及公钥 $P_B$ 。  
产生一个随机数 $k$ ， $k \in [1, n-1]$ 。

- (1) 计算椭圆曲线上的点  $C_1 = [k]G = (x_1, y_1)$ 。
- (2) 计算椭圆曲线上的点  $S = [h]P_B$ 。
- (3) 判断 $S$ 是否为0，若为0，则报错，否则继续向下执行。
- (4) 计算  $[k]P_B = (x_2, y_2)$ 。
- (5) 计算  $t = KDF(x_2 \parallel y_2, k)$ 。
- (6) 判断 $t$ 是否全0，若是，返回第一步，否则，继续向下执行。
- (7) 计算  $C_2 = M \oplus t$ 。
- (8) 计算  $C_3 = Hash(x_2 \parallel M \parallel y_2)$ 。
- (9) 输出密文  $C = C_1 \parallel C_2 \parallel C_3$ 。

#### 3.1.2 SM2 解密过程

用户的原始数据，椭圆曲线系统参数，密文  $C = C_1 \parallel C_2 \parallel C_3$ ，私钥 $d_B$ 。

- (1) 从密文中取出 $C_1$ 。
- (2) 验证 $C_1$ 是否满足曲线方程，若不满足，则报错退出；否则，继续向下执行。
- (3) 计算椭圆曲线点  $S = [h]C_1$ 。
- (4) 判断 $S$ 是否等于0，若是，则报错退出；否则，继续向下执行。。
- (5) 计算  $[d_B]C_1 = (x_2, y_2)$ 。
- (6) 计算  $t = KDF(x_2 \parallel y_2, k)$ 。
- (7) 判断 $t$ 是否全0，若是，则报错退出；否则，继续向下执行。
- (8) 计算  $M' = C_2 \oplus t$ 。

(9) 计算  $u = Hash(x_2 \parallel M' \parallel y_2)$ 。

(10) 判断  $u$  是否等于  $C_3$ ，若不相等，则报错退出；否则，数据明文  $M'$ 。

### 3.1.3 SM2 算法实现

本课题实现了SM2算法，对于非对称加密既可以用于签名也可以用于对数据的加密使用，本课题主要研究了对于数据的加密，对于SM2的加密解密操作都封装为一个SM2EncDecUtils的类，该类是一个SM2算法的工具类。

输入字符串“hello world”，程序运行结果如下所示：

```
输入的明文数据: hello world
加密使用的公钥: 04B834D657EE7E8490E66EF577E6B3CEA28B739511E787F84F71B7F38F241D87F18A5A93DF74E90FF94F4EB907F271A36B295B851F971DA5418F4915E2C1A23D6E
解密使用的私钥: 0B1CE43098BC21B8E82B5C065EDB534CB86532B1900AA49D49F3C53762D2997FA
加密的结果: 045408d7f2f9c44ef26d68913bea5e1c4eca3d8deab8f9c32d242ccff3dcd5e3c9b93aa0ee36afcd5b91947fdacb0aea0c9d301b859d8d74170e7f27a550185d47060ae81ac
解密后的结果: hello world
```

图 3-1 SM2 加解密结果

如图3-1所示，程序会将输入的数据进行加解密，第一行为输入的明文；第二行为加密的公钥，此公钥在代码中已经确定；第三行是解密的私钥，此密钥在代码已经确定好的；第四行是输入数据通过公钥加密后的结果；第五行是通过私钥解密后的结果。

## 3.2 国产对称加密算法

本课题主要研究了国产对称密码算法中的SM4加密算法。SM4密码算法主要用于对于数据的加解密操作，通过SM4算法来保证数据的安全性。在对称密码算法中，密钥的长度越长，相应的其安全性也就越高。在SM4和AES具有相同的密钥长度分组长度的条件下，AES的安全性是高于3DES的，因此SM4在安全性上高于3DES算法。

### 3.2.1 SM4 加解密过程

定义反序变换R为：

$$R(A_0, A_1, A_2, A_3) = (A_0, A_1, A_2, A_3), A_i \in Z_2^{32}, i = 0, 1, 2, 3。$$

说明文输入为  $(X_0, X_1, X_2, X_3) \in (Z_2^{32})^4$ ，密文输出为  $(Y_0, Y_1, Y_2, Y_3) \in (Z_2^{32})^4$ ，轮密钥为  $rk_i \in Z_2^{32}$ ， $i = 0, 1, 2, \dots, 31$ 。则本算法的加密变换为：

$$X_{i+4} = F(X_i, X_{i+1}, X_{i+2}, X_{i+3}, rk_i) = X_i \oplus T(X_{i+1} \oplus X_{i+2} \oplus X_{i+3} \oplus rk_i), \quad i=0,1,2,\dots, 31.$$

$$(Y_0, Y_1, Y_2, Y_3) = R(X_{32}, X_{33}, X_{34}, X_{35}) = (X_{35}, X_{34}, X_{33}, X_{32})$$

SM4加密变换与解密变换的结构相同，惟一不同的是轮密钥的使用顺序不同。

加密时轮密钥的使用顺序为： $(rk_0, rk_1, \dots, rk_{31})$ 。

解密时轮密钥的使用顺序为： $(rk_{31}, rk_{30}, \dots, rk_0)$ 。

### 3.3.2 SM4 算法实现

本课题实现了SM4算法，对于对称加密只是单纯的对数据进行加密来保护数据的安全性。对于SM4的加密解密操作都封装为一个SM4Utils的类，该类是一个SM4算法的工具类。同时实现了SM4加密算法的ECB模式和CBC模式。

输入数据“hello world”，程序的运行结果如下所示：

```
plainText = hello world
ECB模式加密
加密后的密文: 449930d6f9d21f47c999d9727eee2bbc
解密后的明文: hello world
-----
CBC模式加密
加密后的密文: 64be68cb00b44284c3a072b4ab7cdd32
解密明文: hello world
```

图 3-2 SM4 加解密

如图3-2所示，第一行为我们输入的明文数据，后面会使用ECB模式与CBC模式分别加密，将得到的密文展示出来，之后会对相应的密文进行解密，拿到相应的明文数据。

## 3.3 国产数字摘要算法

本课题主要研究了国产的数字摘要算法中的SM3算法。数字摘要算法需要保证数据的完整性，而SM3算法可应用在数字签名和验证消息认证码的生成与验证等众多场景需要中，满足应用的安全需求。对于数字摘要算法，其安全性与最后生成的散列值长度有关，所以，一般的产生的散列值不能太短，否则无法确保安全性。相比于MD5和SHA-1，SM3计算得到的散列值长度为256比特，其安全性会更高。

### 3.3.1 SM3 加解密流程

(1)填充。假设消息 $m$ 的长度为 $l$ 比特，首先将比特“1”添加到消息的末尾，再添加 $k$ 个“0”， $k$ 是满足 $l+1+k \equiv 448 \pmod{512}$ 的最小非负整数。然后添加一个64位的比特串，该比特串是长度为1的二进制表示。填充后的消息 $m'$ 的比特长度是512的倍数。

(2)迭代。将填充后的消息 $m'$ 按512比特进行分组： $m' = B^{(0)}B^{(1)}\dots B^{(n-1)}$ ，其中

$n = (1 + k + 65) / 512$ 。对 $m'$ 按下列方式迭代：

FOR  $i = 0$  TO  $n - 1$

$V^{(i+1)} = CF(V^{(i)}, B^{(i)})$

ENDFOR

其中 $CF$ 是压缩函数， $V^{(0)}$ 为256比特初始值 $IV$ ， $B^{(i)}$ 为填充后的消息分组，迭代压缩的结果为 $V^{(n)}$ 。

(3) 消息扩展。将消息分组 $B^{(i)}$ 按以下方法扩展生成132个字 $W_0, W_1, \dots, W_{67}, W_0', W_1', \dots, W_{63}'$ ，用于压缩函数 $CF$ 。

(4) 压缩函数。令 $A, B, C, D, E, F, G, H$ 为字寄存器， $SS1, SS2, TT1, TT2$ 为中间变量，使用压缩函数 $V^{i+1} = CF(V^{(i)}, B^{(i)})$ ,  $0 \leq i \leq n - 1$ 。

(5) 杂凑值。 $ABCDEFGH \leftarrow V^{(n)}$ ，输出256比特的杂凑值 $y = ABCDEFGH$ 。

### 3.3.2 算法实现

本课题实现了SM3算法，对于数字摘要算法是需要生成一个长度固定的散列值，并且对于数据的改变是敏感的。

输入数据“hello world”，程序的运行结果如下图：

```

输入的明文数据: hello world
计算得到的散列值: 44F0061E69FA6FDFC290C494654A05DC0C053DA7E5C52B84EF93A9D67D3FFF88
-----
修改明文为: I love the world
计算得到的散列值: 9195748EBF6D1E243931966C1539A829BA0B46032426F02CA69756C65A093860
    
```

图 3-3 SM3 生成散列值

如图3-3所示，第一行显示了我们输入的数据，第二行是SM3算法根据我们输入数据计算得到的散列值，当输入的数据改变时，计算的散列值与前面计算得到的散列值不相同。

## 3.4 本章小结

本章主要介绍了相应的国产对称密码算法、国产非对称密码算法、国产数字摘要算法的加解密过程。

下一章主要介绍网络通信，网络的模型以及传输层的通信，因为本课题的实验模拟了客户端上传数据到服务端，服务端部署在云上，整体采用socket进行网络通信。

## 第四章 网络通信

### 4.1 网络通信简介

数据在网络中的传输需要遵守一定的规则，需要考虑传输的介质，如何去发送数据，使用什么协议等等。现在的网络可以使用五层或七层的模型进行描述，七层模型由下到上依次是：物理层，数据链路层，网络层，传输层，会话层，表示层，应用层。本课题实验所采用的socket属于网络中的传输层。

物理层。该层主要规定了有关传输介质的特性，一般的物理层规范中包括连接头、帧、编码及光调制等，不同的传输介质承载的数据形式也不相同。

数据链路层。该层定义如果让格式化的数据以帧的形式进行传输，以及如何对物理介质进行访问，还提供了错误检测和纠正来保证数据的可靠传输。

网络层。该层定义了数据包如何在主机之间传输的。通过对网络中路由器抽象，将路由器标识为逻辑节点，标志相应地址，包在网络中如何路由，路由器如何进行主动学习等。因为传输的带宽有限，该层还定义了分包策略，将一个大的数据包进行拆分为更小的数据包，分开发送。

传输层。该层定义了主机在网络中通信的协议，一般的协议有两种，一种是面向连接可靠的协议，会有差错校验、超时重传、拥塞控制以及包排序等；另一种是用户数据报协议，主要是进行包传输，但是不去关心包是否成功发送、网络是否拥塞等。

会话层。该层定义了对于通信双方一个会话是如何开始，控制和结束的。通过对双向消息的控制，可以在数据传输只完成部分时便通知上层处理，提高处理速度。在某些情况下，如果表示层收到了所有的数据，则用数据代表表示层。示例：RPC，SQL等。

表示层。这一层的主要功能是定义数据格式及加密。因为在网络中传输的数据格式多种多样，网络层做了封装，但是对于应用来讲，需要具体的数据格式，通过表示层改变数据呈现的形式。

应用层。该层主要是一些可以直接为应用进程提供服务的协议。常见的邮件协议SMTP，web访问协议HTTP和HTTPS等。

网络的五层协议由下至上分别为物理层，数据链路层，网络层，传输层，应用

层。网络中的每一层都会有相应的协议去做一些约定，在物理层主要考虑数据以什么样的形式在网络中传输。数据链路层需要考虑传输的介质，不同的传输介质可以传输的数据形式不一样，有的介质可以传输多种形式的数 据，但是会存在适配的问题，只有传输某种特定形式的数 据，传输的效率才会更高。网络层更多的对不同传输介质中的数 据形式做一种抽象，使得对于上层应用来讲，不需要关心底层数 据的具体格式，在什么介质中传输，其主要核心功能还是如何进行域内和域间路由。

数 据在网络中传输，如果使用HTTP，HTTPS，FTP等协议，都会经过网络的五层，自上向下的逐层封装数 据，最后发送出去。若是使用socket套接字进行简单数 据的传输，只会经过给物理层，数 据链路层，网络层和传输层，不会经过会话层，表示层和应用层，整体上速度更快，效率更高。

## 4.2 传输层通信

### 4.2.1 TCP

TCP是一种面向连接的，可靠的数据传输控制协议，其面向连接主要是在数据传输之前会进行三次握手，数 据传送完毕之后通过四次挥手来断开连接。其可靠性的保证主要通过ACK确认机制以及超时重传，同时，TCP也提供了流量控制和拥塞控制的功能。

TCP的三次握手主要是客户端与服务端的通信协商。先由客户端发起主动连接，会发送一个相应的数 据包给服务端，其中TCP的首部的SYN会被标为1，表示这个数 据包是连接的数 据包。服务端收到该数 据包后，会返回给客户端一个数 据包，其SYN位也被标为1，表示服务端同意与客户端建立连接。客户端收到数 据包后，会发送给服务端一个相应的数 据包，其中ACK位标为1，表示客户端收到了服务端传输的数 据，同时也会商定双方传输的包的序号。

TCP的四次挥手主要是客户端与服务端数据传输完毕后断开连接。先由客户端主动断开连接，发送一个数 据包，其FIN位被标为1，服务端收到客户端的数 据包后，会先返回给客户端一个数 据包，其ACK位标为1，表示服务端已经收到客户端传递的数 据包，但是此时服务端可能还会有数 据传送给客户端，所以无法立即断开连接。数 据传输完毕后，服务端会发送一个数 据包，其FIN位标为1，表示这是一个断开连接的数 据包，客户端收到后会发送给服务端一个数 据包，其ACK位标为1，表示客户端收到了服务端断开连接的包，客户端发送完毕后，会等待2倍的最大报文传送时间，

如果考虑到最后一个包丢失，便会重新传送最后一个包来保证断开连接。

TCP的可靠性通过ACK确认和超时重传策略。客户端向服务端发送一个数据包，发送完后，会开启一个计时器，如果在规定的时间内，未收到服务端返回的ACK，那么就会重新传输之前发送的数据包。若是在规定的时间内收到了服务端返回的ACK，就表示这个数据包服务端已经成功接收了，开始传输下一个数据包。

TCP的拥塞控制主要是通过拥塞窗口实现的，如果网络中发生了拥塞，那么在规定的时间内发送的数据包太多了，此时就可以动态调整拥塞窗口的大小，当网络中拥塞消失后，可以增加拥塞窗口大小。一般的，最初初始拥塞窗口的增长是以2倍的方式增长，达到一定阈值时，变为线性增长，当发生拥塞时，将阈值和拥塞窗口变为原来的一半，之后按照线性的方式增长。

本课题实验中所使用的socket通信属于传输层，其是基于TCP实现的，当网络中发生拥塞时，可以进行相应的拥塞控制，因为其是可靠的，传输的数据一定会到达服务端，不会发生数据包的丢失。

#### 4.2.2 UDP

UDP是全称为用户数据报协议，它是无连接的，所以通信的双方不需要在数据传输之前建立连接，当通信的一方需要传输数据时，只需要将数据尽快的发送到网络中去，不会做超时重传，拥塞避免等。在发送端，UDP传送数据的速度依赖于应用程序数据的生成速度、计算机的能力和传输带宽的限制；在接收端，UDP把每个消息段放在队列中，应用程序每次从队列中读一个消息段。数据传输之前不会建立连接，数据传输完后也不会断开连接，同时不会去做确认以及超时重传机制，只是在一端将数据发送出去即可。UDP在数据传输前不会建立连接，所以也就不会有连接状态的维护等操作，对于一台服务器来讲，此时可以同时与多个客户端进行数据传输。

UDP数据包的首部很小，基本只有8个字节，而相较于TCP，其首部需要有ACK，建立连接标识位，断开连接标志位等字段，所以TCP的首部比UDP的首部要大，基本为20个字节。所以，对于UDP来说，其传输的速率较高，不受拥塞控制算法的调节，只是网络的带宽，主机的性能会对其有影响。

UDP和TCP协议的主要区别是TCP在传输数据之前会建立连接，而UDP不会建立连接；TCP对于传输的数据包会进行计时，若是超时则重发，UDP只是将数据发送到



网络中去，不会对进行超时重传；TCP对于数据在网络中传输速度收到拥塞控制算法的调节，当网络发生拥塞时，会相应的减小拥塞窗口并且调整窗口的阈值。UDP不会收到拥塞控制算法的调节，只受网络传输带宽，通信主机的性能的影响。因此，通常将UDP称为不可靠的传输协议。

TCP 是面向连接的传输控制协议，而UDP 提供了无连接的数据报服务；TCP 具有高可靠性，确保传输数据的正确性，不出现丢失或乱序；UDP 在传输数据前不建立连接，不对数据报进行检查与修改，无须等待对方的应答，所以会出现分组丢失、重复、乱序，应用程序需要负责传输可靠性方面的所有工作；UDP 具有较好的实时性，工作效率较 TCP 协议高；UDP 段结构比 TCP 的段结构简单，因此网络开销也小。TCP 协议可以保证接收端毫无差错地接收到发送端发出的字节流，为应用程序提供可靠的通信服务。对可靠性要求高的通信系统往往使用 TCP 传输数据。

## 4.3 数据传输过程

### 4.3.1 数据流转

本实验中数据是通过控制台输入的，对于客户端程序来讲需要获取输入，输入的内容是在内核的缓冲区中，内核缓冲区中的数据需要拷贝到我们相应的用户进程中，此时用户进程拿到相应的数据，进行相关处理计算操作后，调用相应的系统调用函数，将数据先拷贝到内核的另一块缓冲区，再由内核线程将数据拷贝到socket缓冲区中，最后由内核线程将数据发送出去。在用户进程等待数据输入以及数据发送过程中，用户进程是被阻塞的。在客户端发送数据是经过了上述五层网络协议中的传输层，网络层，数据链路层和物理层，发送端的顺序由上至下，逐层对数据封装，加上一些相应的首部构成。数据在发送时可能会被分拆成多个数据包，在网络中通过路由器进行路由转发，达到服务端。服务端收到客户端发送的数据后，会先进行由下至上的解包，顺序是物理层，数据链路层，网络层和传输层，最后拿到实际我们传输的数据。

上述操作在网络中的部分不会更改，但是在客户端接收和服务端处理上会存在频繁的数据拷贝以及系统调用时频繁的在用户态和内核态切换，会造成一定的性能消耗。

#### 4.3.2 零拷贝

为了减少数据的拷贝次数以及用户态与内核态频繁切换，socket使用了相应的零拷贝机制。主要的零拷贝技术有直接IO与MMAP的方式。

直接IO允许用户进程不需要通过相应的系统调用，直接可以从磁盘上读取数据，读取后的数据直接存储在用户进程相应的缓冲区域，不需要进行数据的拷贝。这种方式减少了一次用户态到内核态的切换，同时也减少了一次数据从内核缓冲区拷贝到用户缓冲区的操作，降低了CPU的消耗，典型的直接IO的应用有RDMS。

MMAP的方式通过在磁盘开辟一块缓冲区进行映射，此缓冲区为用户进程与内核进程共享的空间，通过共享空间就可以避免频繁的数据拷贝和用户态与内核态切换。客户端程序获取用户输入的内容后，其内容一直存放在缓冲区，该缓冲区被用户进程和内核进程共享，只需要两次系统调用，整体就可以实现数据的输入和发送，不存在数据频繁从内核缓冲区拷贝到用户缓冲区的问题。当用户进程处理完数据后，内核线程直接从共享的区域读取数据，再将数据发送拷贝到socket缓冲区，最后把数据发送到相应主机，此种方式很大程度提高了数据传输的效率。

#### 4.4 网络通信实现

用户端使用Java语言进行开发，使用Java的socket，在创建socket对象时，传入服务端程序监听的端口和所在机器的IP，需要连接服务端的IP和端口均采用配置化的方式编写，将IP和端口写在一个配置文件中，在用户端启动时，需要从配置文件中获取相应的IP和端口。

客户端的socket使用单例的类主要防止每次在对数据进行加密和解密时需要频繁的创建该类的对象，造成CPU会频繁的调度垃圾回收线程导致程序的性能下降。在初始化单例对象时进行了加锁，保证操作的原子性，在多个线程同时访问时，也只会初始化一个该类的对象，得到预期的效果。

服务端使用Java语言进行开发，服务端启动时需要绑定主机的某个端口，监听的端口也是配置化的方式，通过在配置文件里写入监听的端口，在创建服务端socket对象时，传入相应的端口即可。服务端采用线程池来对客户端的请求响应，使用的阻塞队列是链表阻塞队列，拒绝策略使用的是当任务过多，线程数达到最大线程数时，会让调用线程池的线程执行该任务。使用线程池进行响应避免了每次响应都创建线程，造成频繁的线程创建与销毁，提高程序的运行效率。对于每个客户端的连

接都会有一个变量进行统计，表示这是第几次连接，该变量使用了Java中原子类，在多线程并发访问下是线程安全的，通过该变量就可以实现正确的数值统计，不会存在多个客户端同时连接，计数错误的情况。服务端的socket对象也是单例的，当有用户请求与其进行连接时，会返回一个相应的socket对象，通过该对象与用户端进行通信。

对于服务端使用的线程池，其工作流程是当某个任务提交给线程池后，先判断当前的线程数是否大于线程池的核心线程数，如果小于核心线程数，那么便创建一个线程去执行该任务，否则，将该任务加入到所设置的任务队列中，如果任务队列中的任务数量没有达到最大的限制，那么便将该任务加入任务队列，否则，判断线程池中的线程数是否大于最大线程数，如果小于最大线程数，则创建一个线程去执行该任务，当线程执行完该任务后，便会存活一定的时间，在规定的时间内，该线程没有执行其他任务，那么该线程对象会被回收。如果大于最大线程数，则进行执行相应的任务拒绝策略。创建线程池所使用的是默认的工厂创建。本实验中核心线程数设置为CPU数量加1，因为服务端的任务主要是对客户端的请求响应，属于CPU密集型的任务，不需要创建过多的线程，造成频繁的线程切换影响程序性能。最大线程数设置的100，线程存活时间设置为60秒，采用链表的阻塞队列，设置了队列的最大长为1000。为了让每一个任务都会被执行得到一个结果，采用的拒绝策略是让调用线程池的线程执行该任务。

使用原子类操作主要是为了保证多个客户端连接计数的正确性，其主要是通过CAS操作实现，即比较和交换。在进行修改的该数据的值时，先会比较内存值与旧值是否相同，若是相同，则说明在某个线程操作期间数据没有被修改过，就可以进行该数据的修改，如果被修改过，两个数值就会不相同，会进行失败重试直到成功。

## 4.5 本章小结

本章主要介绍了网络的七层和五层模型，同时对传输层的两个主要的协议TCP和UDP做了详细的介绍，对于网络数据发送做了阐述。

下一章主要介绍本课题研究的主要方案、在编程中用到的具体技术、实验的过程以及最后实验结果。

## 第五章 实验流程

### 5.1 实验方案

本课题实验主要采用国产密码算法的SM4的ECB模式，使用socket通信模拟数据网络传输。将需要传输的数据在用户端进行加密，然后通过网络将加密后的数据发送到服务端，在服务端对密文进行一个相加的操作，此处相加的操作只是指字符串的一个拼接操作，即将两个字符串合并为一个字符串的操作。服务端最后将处理完的密文发送给客户端，客户端接收到密文后进行解密操作，拿到解密的结果再与客使用明文进行相同操作得到的结果比对，如果两个结果是一致的，那么说明满足同态性，否则，则不满足同态性。

绝大部分的密码算法只是满足部分同态性，所以本实验采用通过对服务端传输到客户端的数据进行分组解密，服务端会将客户端发送过去的数据也一并返回，客户端根据每次的字符串对整体计算的结果进行分割字符串，然后分组解密拼接结果。

实验开发工具使用IDEA，JDK版本是1.8。

### 5.2 实验流程

第一步，配置好服务端监听的端口，此实验配置的端口是7878，然后启动服务端程序，若成功启动，会提示“服务器启动成功，等待客户端发送数据”。




图5-1 服务端启动

第二步，配置好客户端需要连接的服务端的IP和通信的端口，本实验配置的IP是127.0.0.1，端口7878，然后启动一个客户端应用程序，若是启动成功会在客户端提示输入字符串，同时，服务端会提示本次是第几次连接，并且等待客户端输入数据，如图5-2所示。



图 5-2 客户端连接成功

图5-3展示了客户端第一次连接到服务端，此时服务端会打印出第1次连接，若是此时有新的客户端连接会体是后续的连接次数。

```
服务器启动成功，等待客户端发送数据...  
-----  
第1次连接  
|
```

图 5-3 服务端连接成功

第三步，客户端通过控制台输入相应的数据，先输入一个hello，在客户端会显示出相应的明文以及加密后的结果。

```
请输入您的第1个字符串：  
hello  
加密后的密文为：a432c83e2fb356d420c9a674959b8832  
请输入您的第2个字符串：
```

图 5-4 客户端输入数据后变化

图5-4显示的即将客户端加密前的明文以及加密后的密文都打印出来。当客户端每输入一个数据，便会进行加密，然后通过socket发送给服务端，服务端接收到对应的密文，该密文也会在服务端展示出来。

```
服务器启动成功，等待客户端发送数据...  
-----  
第1次连接  
客户端发送的信息：a432c83e2fb356d420c9a674959b8832
```

图 5-5 服务端接收输入

据图5-5可知服务端接收到的知识客户端发送给他的加密后的数据，客户端并没有发送相应的明文信息，服务端完全是对密文进行操作。

第四步，输入一个world，同时结束输入，本实验中当输入的字符串符号为“#”时，客户端默认用户输入完毕，后面便不会提示用户继续输入，只需要等待服务端处理后返回结果，如图5-6所示。

```
请输入您的第1个字符串：  
hello  
加密后的密文为：a432c83e2fb356d420c9a674959b8832  
请输入您的第2个字符串：  
world!  
加密后的密文为：35cbe2a4014c9a2845d55ffc83e36b4c  
请输入您的第3个字符串：  
#
```

图 5-6 客户端输入完毕

第五步，服务端会根据客户端输入的密文进行一个拼接操作，然后将密文传递

给客户端，客户端拿到服务端传递的数据后，会进行整体以及分隔字符串的方式解密，对比结果。

```
第1次连接
客户端发送的信息: a432c83e2fb356d420c9a674959b8832
客户端发送的信息: 35cbe2a4014c9a2845d55ffc83e36b4c
客户端发送的信息: #
服务端发送给客户端的数据:
a432c83e2fb356d420c9a674959b8832,35cbe2a4014c9a2845d55ffc83e36b4c,a432c83e2fb356d420c9a674959b883235cbe2a4014c9a2845d55ffc83e36b4c
```

图 5-7 服务端收到的全部数据

图5-7是客户端传递到服务端的所有密文，每次都会将发送的密文打印出来，当服务端完成相应的操作后，会将结果返回给客户端，同时，也会把每次传递的密文以列表的形式返回给客户端。

```
请输入您的第3个字符串:
#
-----
开始接收服务端返回的数据...
总的解密的结果是: hello world!
分开解密的结果是: hello world!
```

图 5-8 客户端收到的数据

据图5-8可知，客户端会对接收的数据进行解密，有两个解密结果。其中第一个是整体解密结果，其含义是服务端发送给客户端一个字符串列表，客户端使用列表的最后一个字符串进行解密，解密得到的结果如上图展示，如果整体解密，密文中间会存在一些解析错误的字符，也就是乱码；第二个是分组解密的结果，从服务端返回给客户端的数据可以看出，服务端返回的列表中不仅仅只返回了处理的结果，还返回了客户端传递给服务端的数如数据，通过返回输入的数据，客户端可以根据这些输入的数据对服务端处理结果进行分隔字符串，因为返回的是密文，所以是根据列表中前面N-1个字符串对最后一个密文字符串进行逐个的拆分操作，拿到每个拆分的结果后再逐个解密，最后将解密结果拼接起来，得到一个分开解密的结果，这种解密方式拿到的结果与我们预期的结果是一致的，通过次种方式解密是可以得到正确的结果。

第六步，重新启动一个客户端程序，会在服务端显示是第二次连接，客户端逐次输入数据“I love the world!!!”，客户端每次对于输入的数据都是会进行加密，然后将数据发送给服务端，所以用户输入的数据没有长度的限制。服务端收到数据后，进行相应处理后返回给客户端列表，客户端根据返回的列表进行整体解密和分开解

密操作做对比。

```
第2次连接
客户端发送的信息: 001ba7de7b5312bf17aa35542bc8e77e
客户端发送的信息: fbaf355c78c8de3d76506b7791839f17
客户端发送的信息: 2cdf5b1476c2baf6d48a1362ed853226
客户端发送的信息: b67f710fe958ed66dcf4d26056894221
客户端发送的信息: #
服务端发送给客户端的数据:
001ba7de7b5312bf17aa35542bc8e77e,fbaf355c78c8de3d76506b7791839f17,2cdf5b1476c2baf6d48a1362ed853226,b67f710fe958ed66
```

图 5-9 客户端收到的数据

图5-9展示客户端第二连接到服务端后，传输用户的输入到服务端，服务端显示的客户端的输入以及数据处理后发送给客户端的列表。

### 5.3 实验结果

通过上述实验过程可以看出，客户端接收用户的输入然后对数据进行加密，将加密后的密文发送给服务端，服务端对密文进行了拼接操作，结果返回给客户端，客户端解密后得到的结果与直接使用明文操作得到的结果一致，说明了国产密码算法SM4的ECB模式下采用分组解密的方式，可以对字符串密文进行简单的拼接操作，是一种部分同态性的体现。

本课题研究的国产密码算法有SM2，SM3，SM4，经对比，使用SM4加密解密的速度都较快，加解密的效率高。SM4属于对称密码算法，相比于非对称的密码算法，数字摘要算法，其本身的加解密的效率都是要高于其他两种加密算法，同时因为使用ECB模式进行加密操作，每组加解密所使用的密钥都是相同的，所以使用SM4来进行加解密操作可以使程序的运行速度更快，效率更高。

### 5.4 本章小结

本章主要介绍了实验的方案，实验过程中所使用的技术原理，实验的流程，每次操作客户端的变化提示与服务端的变化提示，最后得到相应的实验结果。

## 结 论

使用国产密码算法SM4的ECB模式，分组解密的方法是可以得到预期的结果，即客户端加密数据传输给服务端，服务端只是对密文进行相应的计算操作，将计算结果返回给客户端，客户端解密拿到预期结果。

本课题研究的创新点在于对于国产密码算法的同态性进行研究，随着国产密码技术的发展，国产密码算法的应用场景会越来越多，相应密码算法需要更多的特性来支撑各种场景。在云计算领域，密码算法的同态性至关重要，用户希望自己的数据传递上去不被云服务提供商所获得，云服务提供商只是提供相应的计算和存储能力，这个场景下，就需要云服务提供商只是对密文进行下相应的计算操作，将计算结果返回给用户，解密操作在用户端进行。同时，如果云服务提供商没有相应的解密方案，对于数据泄露问题也就可以得到有效的解决，即使数据泄露，但是数据是以密文的形式呈现的，无法获取到相应的明文信息，保护了用户隐私数据的安全。

本课题研究的密码算法只有SM2，SM3，SM4三种国产密码算法，密码算法选择的范围较小，今后可以选择更多的国产密码算法研究其同态性，在一个更大的范围去寻找全局最优解，也会去考虑将机器学习等与密码算法同态行进行结合。



## 参考文献

- [1] 杨竞. 同态加密关键技术研究[J]. 电子科技大学, 2019.
- [2] 李婷. 浅析云计算安全技术[J]. 机电信息, 2019 (33): 113-114.
- [3] 李曾鹏,马春光,周红生. 全同态加密研究[J]. 密码学报, 2017(06): 561-578.
- [4] 徐海霞. 云计算环境中改进的整数上全同态加密算法研究[J]. 科技通报, 2019(06): 87-92.
- [5] 杨浩淼,金保隆,陈诚,吴新沿. 一种高效的同态加密方案及其应用[J]. 密码学报, 2019(06): 611-619.
- [6] 洪家军,陈俊杰. 一种基于全同态加密的密文检索算法[J]. 廊坊师范学院学报(自然科学版), 2018 (04): 15-18+30.
- [7] 巩林明,李顺东,郭奕旻. 同态加密的发展及应用[J]. 中兴通讯技术, 2016(01): 611-619.
- [8] 程晋格. 国密算法在数据存储及码流数据传输中的应用[J]. 中国集成电路, 2018(07):15-18.
- [9] 王栋,李国春,俞学豪,陈智雨,葛冰玉,谢磊,谭静. 基于量子保密通信的国产密码服务云平台建设思路[J]. 电信科学,2018(07):171-178.
- [10] 刘悦,贾忠田,张波.结合国产密码算法的应用密码学课程教学探讨[J]. 计算机教育,2018(03):10-13.
- [11] 鲍海燕. 基于同态加密算法的网络信息安全保护[J], 现代计算机. 2019(24):22-25.
- [12] 桑杰,许雪姣,刘硕,蔡子凡. 基于国密算法的分布式加密存储研究[J], 数据通信.2020(01):9-12.
- [13] 卢希. 国产密码算法的安全、可信之路[J], 智能建筑与智慧城市, 2019(03):11.
- [14] 吴红英. 云计算下数据安全存储技术研究[J], 计算机产品与沟通, 2020(07):10
- [15] 黄延伟. 云计算背景下计算机安全问题策略[J], 网络安全技术与应用,2020(04):90-91.
- [16] 李鹏,李华,石永红. 云计算下的信息安全体系研究[J], 机械工程与自动化,2020(02):225-226.
- [17] 王佳,张远,刘超,杨婷婷.探讨云计算安全问题及其技术对策[J],科学技术创新,2020(10):52-53.
- [18] 贾凌杉,关艳. 同态加密技术在物联网中的应用[J], 科技风, 2018(17):90.
- [19] 王全福,宋文爱,杨顺民. 云环境中数据安全的同态加密方法[J], 计算机工程与设计,2017(01):42-46.
- [20] 柳玉东,王绪安,高忠石. 基于同态加密算法的欧式距离外包计算协议[J], 计算机工程与应用,2019(05):110-116.

## 致 谢

值此论文完成之际，首先向我的导师闫怀志老师表示感谢。我的论文是在老师的指导下修改完成的，正是因为他细心帮助和耐心的指导，我才会完成本论文，在此，对于闫老师表示由衷的感谢。

感谢所有在大学期间传授我知识的老师，不积跬步无以至千里，本论文能够顺利完成，也归功于各位任课老师的认真负责，使我能够很好的掌握和运用专业知识，并在实验中得以体现。正是有了你们的悉心帮和支持，才使我的毕业论文工作顺利完成，在此向北京理工大学计算机学院的全体老师表示由衷的谢意，感谢你们四年来的辛勤栽培。

感谢我的爸爸妈妈，在日常交流中您们用的最多的两个字就是加油，感谢父母一直以来给我不断的鼓励，这些鼓励转化为我前进的动力，学无止境，我将努力做一个对社会有用的人，感谢亲人的关心，愿爸爸妈妈永远健康快乐。

最后，我要感谢专业的同学们，感谢他们的鼓励和支持，感谢他们和我一路走来，让我在此过程中倍感温暖！

感谢所有关心和帮助过我的老师、同学、朋友，谢谢你们！