Report for Project4
Shaoyan Yu

Goal:

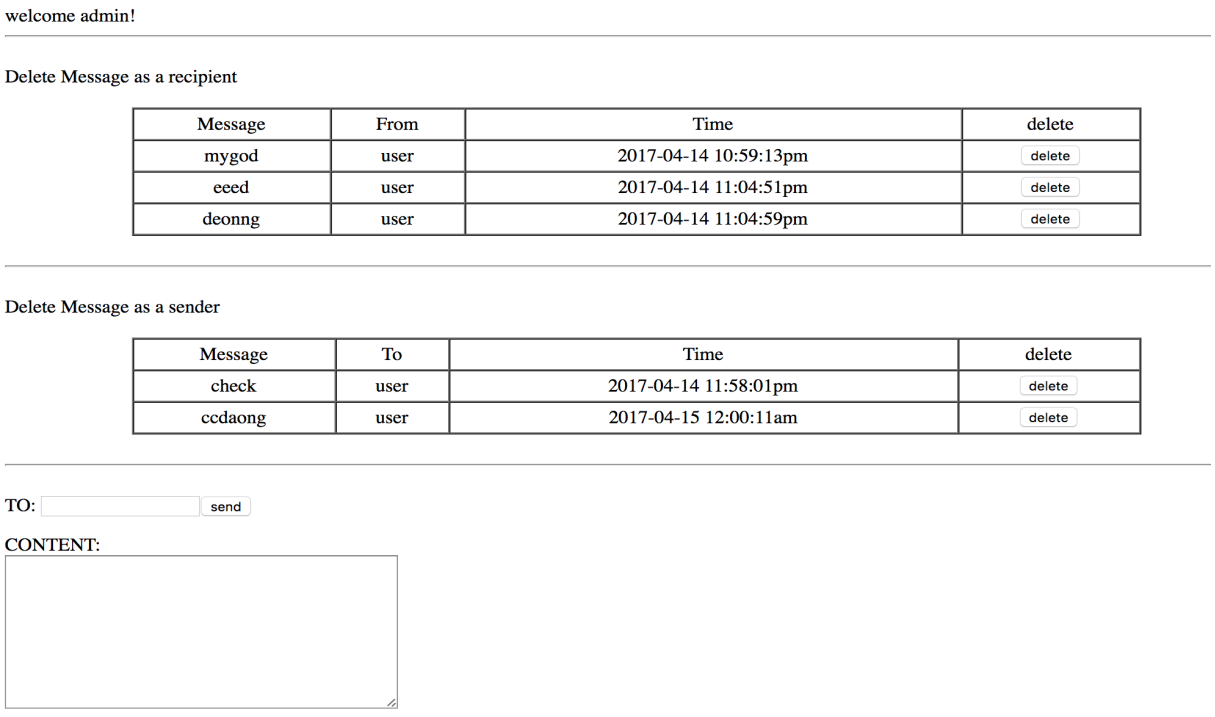
In this project, I will continue to work on project1 and add a function to send messages from a user to another user. When the user log in the system, he can see the messages he sent and received. Then he can delete the messages he sent and received or send a message to another user. However, the databases can be accessed by untrusted users, the messages should be encrypted in the databases. Also, the user can see the messages even if he changes his password.

The improvement I made on project1:

The “user.php” is the file which can implement the functions of project4. In this php file, it is divided into three parts:

1. Show the messages received.
2. Show the messages sent.
3. Show the text area to send messages to another user.

Here is the screenshot of my user file. This the time when I log in as admin.



To build this file, I created a new table on my database. The name of this table is “Messages”

Here is my commands to create the table:

```
CREATE TABLE messages
(
sender varchar(255),
receiver varchar(255),
content varchar(255),
encrypted_KE varchar(255),
encrypted_KI varchar(255),
encoded_IV1 varchar(255),
encoded_IV2 varchar(255),
encoded_IV3 varchar(255),
sendtime varchar(255)
);
```

Here is the screenshot of the information in “Messages” table. As you can see, the content of the messages is encrypted in the tables.

sender	receiver	content	encrypted_KE	encrypted_KI	encoded_IV1
admin	user	f9GyJass1778OG+fSpYKepDqHv42qTStVeA41...	uc0bCxfgssC2otiSkdXM8tDZUaBxAo8FaStW7...	vJmS1h1WJIdyfDbi0liHIDq1Em5qXJ9I3M5GJq...	SAhOkwTO6q11rM629rtmHA==
user	admin	DomyxreLX0x6Jsz4Xt/tKNoLBDy8GXadIZXUs+...	924HuRrbDzfUZgqannei9Mgmw1U4WjNpgRN...	fkQhZPGVE881d/r05mUWytv0rpRoTuSj3jNI3B...	T7LP02Dm8J39mNyGhe0opQ==

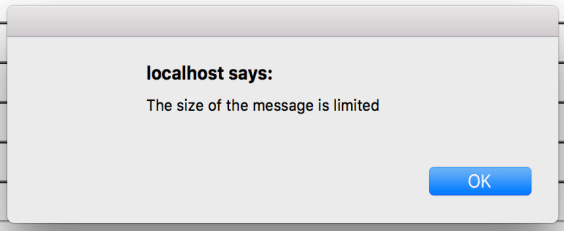
	encrypted_KI	encoded_IV1	encoded_IV2	encoded_IV3	sendtime
DZUaBxAo8FaStW7...	vJmS1h1WJIdyfDbi0liHIDq1Em5qXJ9I3M5GJq...	SAhOkwTO6q11rM629rtmHA==	iMapNxSCOegXYEwaZfRwCg==	oQmyeRODp2MmE9J6B/EQWQ==	2017-04-22 07:34:52pm
Mgmw1U4WjNpgRN...	fkQhZPGVE881d/r05mUWytv0rpRoTuSj3jNI3B...	T7LP02Dm8J39mNyGhe0opQ==	bv8Fd+iJVC3b4kk7czGwSA==	at/TDwMOdX0dUvKVB/fjhA==	2017-04-22 07:40:34pm

The last thing I should do is limit the length of the messages when sending messages to another user. Here is the screenshot.

welcome admin!

Delete Message as a recipient

Message	From
mygod	user
eeed	user
deonng	user



Delete Message as a sender

Message	To	Time	delete
check	user	2017-04-14 11:58:01pm	delete
ccdaong	user	2017-04-15 12:00:11am	delete

TO: send

CONTENT:

oppdnapenvnvopepnv

This is my javascript code to implement this function:

```
<script>
function  validateForm()
{
    var s = document.forms["theForm"]["message"].value;
    if (s.length > 10) {
        alert("The size of the message is limited ");
        return false;
    }
}
</script>
```

The scheme I used to encrypt the messages:

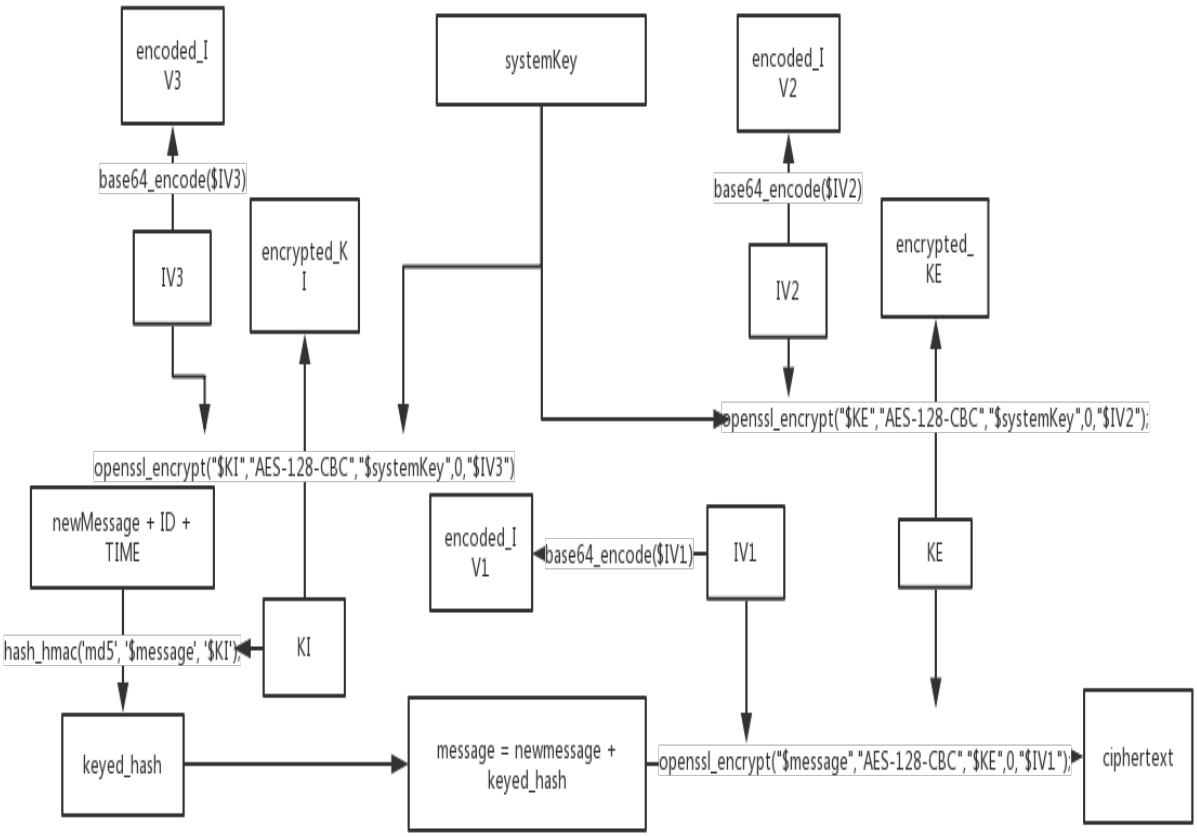
The code I used to implement the encrypt function is in “user.php”. I encrypt the messages before sending them. It used some functions in the PHP.

Here is my code to implement the encrypt function:

```php
if($_POST["add"]==="send"){
    $newMessage=$_POST["message"];
    $receiver = $_POST["receiver"];
    $idsql = "SELECT ID FROM students WHERE username = '$myName'";
    $idresult = $conn->query($idsql);
    $id = 0;
    if ($idresult->num_rows > 0) {
        // 输出每行数据
        while($row = $idresult->fetch_assoc()) {
            $id = $row[ID];
        }
    }
    date_default_timezone_set("America/New_York");
    $newTime = date("Y-m-d h:i:sa");
    $message = $newMessage . $id . $newTime;
    $KI = openssl_random_pseudo_bytes(16);
    $KE = openssl_random_pseudo_bytes(16);
    $method = 'AES-128-CBC';
    $ivlen = openssl_cipher_iv_length($method);
    $IV1 = openssl_random_pseudo_bytes($ivlen);
    $IV2 = openssl_random_pseudo_bytes($ivlen);
    $IV3 = openssl_random_pseudo_bytes($ivlen);
    $keyed_hash = hash_hmac('md5', '$message', '$KI');
    $message = $newMessage . $keyed_hash;
    //ciphertext
    $ciphertext = openssl_encrypt("$message","AES-128-CBC","$KE",0,"$IV1");
    //encrypt KE
    $encrypted_KE = openssl_encrypt("$KE","AES-128-CBC","$systemKey",0,"$IV2");
    //encrypt KI
    $encrypted_KI = openssl_encrypt("$KI","AES-128-CBC","$systemKey",0,"$IV3");
    //encode IV1,IV2,IV3
    $encoded_IV1 = base64_encode($IV1);
    $encoded_IV2 = base64_encode($IV2);
    $encoded_IV3 = base64_encode($IV3);
```

```php
    //insert
    $addsql = "INSERT INTO Messages(sender,receiver,content,
    encrypted_KE,encrypted_KI,encoded_IV1,encoded_IV2,encoded_IV3,sendtime)
            VALUES ('$myName', '$receiver', '$ciphertext',
            '$encrypted_KE','$encrypted_KI','$encoded_IV1','$encoded_IV2','$encoded_IV3','$newTime')";
    mysqli_query($conn,$addsql);
```

Here is the diagram to show the encrypt procedure:



After receiving the messages, I decrypt the messages before showing them. I implement the decrypt function in the “user.php”.

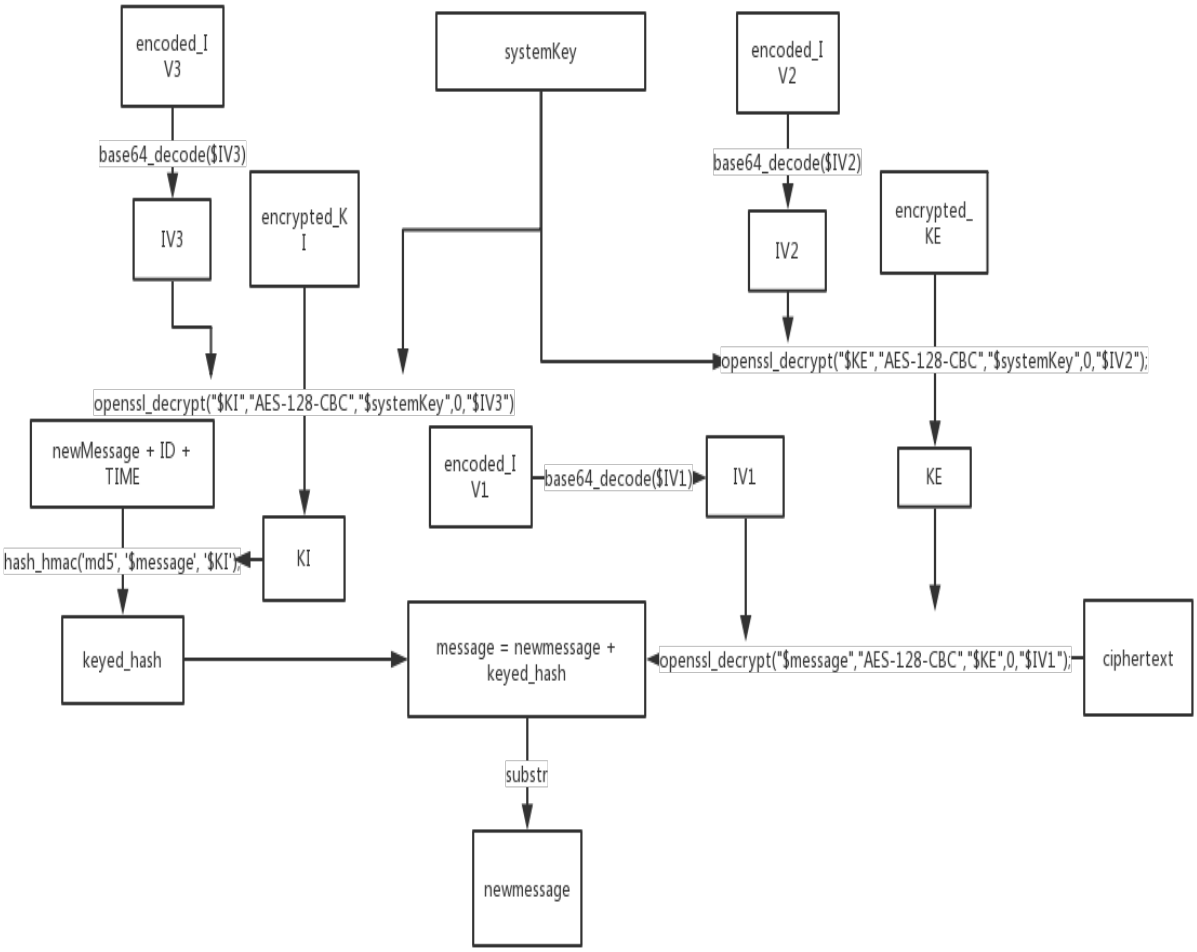
Here is my code to implement the decrypt function:

```php
$sql0 = "SELECT sender,content,encrypted_KE,encrypted_KI,encoded_IV1,
encoded_IV2,encoded_IV3,sendtime FROM messages WHERE receiver = '$myName'";
$result0 = $conn->query($sql0);
if ($result0->num_rows > 0) {
    while($row = $result0->fetch_assoc()) {
        if ($conn->connect_error) {
                    die("can not connect: " . $conn->connect_error);
        }
        //uptdate
        echo "<form action='user.php' method='post'>";

        $sendtime0 = $row['sendtime'];
        $sender0 = $row['sender'];
        echo " <input type='hidden' name='sendtime0' value = '$sendtime0'>";
        echo " <input type='hidden' name='sender0' value = '$sender0'>";
        //decrypt
        $ciphertext = $row['content'];
        $IV1 = base64_decode($row['encoded_IV1']);
        $IV2 = base64_decode($row['encoded_IV2']);
        $IV3 = base64_decode($row['encoded_IV3']);
        $encrypted_KE = $row['encrypted_KE'];
        $encrypted_KI = $row['encrypted_KI'];
        $KE = openssl_decrypt("$encrypted_KE","AES-128-CBC","$systemKey",0,"$IV2");
        $KI = openssl_decrypt("$encrypted_KI","AES-128-CBC","$systemKey",0,"$IV3");
        $temp0 = openssl_decrypt("$ciphertext","AES-128-CBC","$KE",0,"$IV1");
        $length0 = strlen($temp0);;
        $temp0 = substr($temp0,0,$length0 - 32);

        echo "<tr align='center'>";
        //$temp0 = encrypt($row["content"],'D',$row['salt']);
        echo "<td>" . $temp0 ."</td>";
```

Here is diagram for decryption:



Any security concerns of my project:

1 the user can do online dictionary attack

The user can attack the encrypted messages by online dictionary attack. So I should set up a function to let the server monitor the actions of the user. If the user is concerned suspicious by the server, the user will be held for some hours and can not to log into the system.

2 the authentication protocols when the users communicate with each other

We can try to build a public key and private keys for each user. The user should use all the keys to decrypt the messages they received and make sure the identity of senders.

3 the attacker may want to know the length of the plaintext

Add a key to protect the integrity of the plaintext.

4 the attacker may attack the database to get the keys or messages.

Encrypt the keys and messages stored in the database as well.