A Secure Mobile System to Support Citizen Journalism

4/11/2012

CSIS0801 Final Year Project Final Report

Supervisor: Dr. Yiu Siu Ming
2nd Examiner: Dr. Lucas Hui
Student Name: Lee San Tai (2009565273)
A Secure Mobile System to Support Citizen Journalism

Introduction

Nowadays, most people around us are using smart phone. They use their phones for calling, taking photos, listening to music, surfing the internet, and so on. People are eager to share their daily life with their friends, such as where do they go for shopping, what they have in lunch. Sometimes, they may encounter some events which contain news value. Then, we started to think, if they have an application in their smart phones to capture those critical moments for new agencies, we will be able to read fresh news as there are always some people next to scene of the news.

There are 3 main parts in our projects and the security concepts behind. They are Windows Phone Application, Web Portal, SQL Azure Database.

Our project aims to implement a smartphone photo-sharing platform which supports photo transactions. There are 4 main functions of the platform:

1. A two-way process selling photos with encryption and digital signature
2. Ring signature for uploading photos anonymously
3. A map to let people know the news around them
4. Categorization of photos to improve organization

Concept of Our Application

Fig. 1
Users can capture some photos and upload them to our server. They can choose the photo to be public or private before uploading. If the photo is public, everyone can see the photo. If the photo is private, other users can only read the thumbnail on the web portal. After uploading, users can see the photos on the map so that they know what is happening around them.

The following figure shows the security concepts of our application.

![Security Concepts Diagram](Fig. 2)

When the user chooses the photo to be private and uploads it. The photo will be encrypted with 1024-bit AES encryption. In this project, we simulate the certificate authority to identify users and generate public and private keys on our own. When a buyer purchases the photo, the key will be encrypted by the public key of the buyer using 1024-bit RSA encryption and sent to the buyer, so that the buyer can decrypt it with his own private key.

In uploading the photo, the user can choose to upload it anonymously by applying ring signature. The system can hide the identification of the user who uploaded photo, and protect him from being discovered by malicious people. Moreover, the ring signature allows us to ensure the user is a valid user of our platform.
Phone Interface

The following figure shows the interface of our main page:

The Fig. 3 shows the main page of our application, users can choose corresponding icons to capture photos, show photos in map, show photos in album page, and read notifications. The capture photo interface looks like Fig. 4, users can choose the photo to be public or private by sliding the bar. After that, the user can tag the photo and leave comments to it. Finally, press the upload button to upload the photo. If the photo is choose to be public, it will not be encrypted, and vice versa. The user can also click “Ring” to do an anonymous upload.
The map interface is in Fig. 5. It shows a map zoomed to a certain level with the current location of the windows phone and shows pushpin of photos on it, so that users can figure out what are happening next to them.

Fig. 6 shows the interface of the photo album. It shows the thumbnails of photo sorted in dates on the original page. The users can swipe the page to show photos sorted by like number, dislike number, and view number. Those thumbnails can also be filtered in different categories in the application bar, such as headline, local, sports, etc. (the interface is shown in Fig.24)

The notification interface looks like Fig. 7. In this page, users can view their own notifications, such as buyer request for photo sale, a deal is completed, etc.
Web Portal

After uploading the photo to our server, the web portal is also updated. The interface of our web portal looks like the following figure:

![Web Portal Interface](image)

Fig. 8

In the web portal, users can log in with their own live ID. They can filter the photos by the categories on the top right hand corner. Moreover, they can like or dislike photo, and also view the thumbnail of a photo by clicking the small photo of it. Users can also see what is happening around them by clicking “Map” on the menu bar. A map with some pins will be shown. A small photo will be shown when pointing to the pin, and the thumbnail of the photo can be downloaded by clicking the small photo.
Windows Azure

Windows Azure is a Cloud Platform. The interface of Windows Azure is in Fig. 9. It allows us to deploy SQL Azure which is for processing SQL queries, and Azure Blob which is for data storage on it. It also has access control which handles the user log in and passwords.

My Works and Challenges

In this project, my main contributions are the development of the application on Windows Phone, especially the watermark rendering, map page, album page, the rating scheme, and performance evaluation. Since Windows Phone platform is new to us, so it takes some time to get familiar with the development tools. Visual studio contains the drag and drop functions in creating graphical user interface, but some libraries are missing. Hence, I have to download the corresponding .dll file and add it as a reference.
Watermark Rendering

![Fig. 10](image1.png) ![Fig. 11](image2.png)

When the user captures a photo, a thumbnail of the photo will be generated and masked by a colourful watermark. The thumbnail is of low resolution and quality, and is used as a preview of the original photo. In the photo-selling scheme, the watermarked thumbnail is an important piece of the solution. This is because the watermark can prevent malicious users use the resized photo on unauthorized purposes. Users have to buy the photo in order to remove the watermark and obtain the authorization of the seller to use on publications.

In the design of the watermark, the text of the watermark is designed with changing colours. This is to make the watermark visible on any background colour.

The watermark function is called just after a photo is captured. The function will render a text block with specified font size, font family, and font colour to the photo before uploading.
private void RenderString(WritableBitmap bitmap, string stringToRender)
{
    TextBlock textBlock = new TextBlock();
    textBlock.Text = stringToRender;

    // Colour the watermark text
    LinearGradientBrush brush = new LinearGradientBrush();
    GradientStop a = new GradientStop();
    a.Color = Colors.Red;
    a.Offset = 0.0;
    GradientStop b = new GradientStop();
    b.Color = Colors.Yellow;
    b.Offset = 0.25;
    GradientStop c = new GradientStop();
    c.Color = Colors.Blue;
    c.Offset = 0.50;
    GradientStop d = new GradientStop();
    d.Color = Colors.Green;
    d.Offset = 0.75;
    brush.StartPoint = new Point(0, 0);
    brush.EndPoint = new Point(1, 1);
    brush.GradientStops.Add(a);
    brush.GradientStops.Add(b);
    brush.GradientStops.Add(c);
    brush.GradientStops.Add(d);

    textBlock.Foreground = brush;

    // Set font, size, etc. on TextBlock
    textBlock.Margin = new Thickness(50, 50, 50, 50);
    textBlock.Height = 150;
    textBlock.Width = 150;
    textBlock.FontSize = 150;
    textBlock otherTextBlock.

    bitmap.Render(textBlock, null);
    bitmap.Invalidate();
    Image img = new Image();
    img.Margin = new Thickness(5, 0, 0, 0);
    img.Height = 450;
    img.Width = 450;
}

Fig. 12
Map

![Map with pushpins](image)

Fig. 13

When the map page is loaded, the Bing map with pushpins on it will be shown. The user can zoom in or zoom out the map by clicking the “+” and “-” sign at the bottom of the map respectively. With this map function, users will be able to figure out what are happening next to them easily.

The original design for the map page will show the thumbnail of photos on the map. However, if two photos are taken at a near position on map, they will become overlapped. This is not convenient for the user to see what are happening around them. Therefore, we changed all thumbnails on map to pushpins. The pushpins are in red for eye catching.
When a pushpin is clicked, the photo page of the corresponding photo will be shown. Users can view the information about the photo. On the photo page, the user can see the thumbnail, comment, number of like, number of dislike, number of view of the photo. The user can also rate the photo and offer a price to purchase the photo from the seller. The seller can accept or deny the offer here. Moreover, once the user owned the photo, he or she can download the photo here.

When a pushpin is clicked the photo id and the session key will be transferred to the photo page.

```csharp
void MouseButtonUp(object sender, MouseButtonEventArgs e)
{
    Pushpin pin = (Pushpin)sender;
    string id = photo_info[Int32.Parse(pin.Name)];

    NavigationService.Navigate(new Uri($"/photoPage.xaml?id={id}&session={sessionKey}, UriKind.Relative"));
}
```

Then, the photo page will access the database and request for the information of the photo by providing the photo id.
After getting the corresponding information, the photo page is shown.

When the photo page is loaded, the view number of corresponding photo will be increased by 1 automatically. As mentioned above, the user can rate the photo on this page by clicking “like” or “dislike”. This rating system can allow user to express their feelings of the photo in a fundamental but convenient way, and also facilitate the photo categorizing function on album page.

**Album Page**

When the album page (left-bottom button on main page) is clicked, the album page with a list of photos and their id will be shown. Album page is implemented in panorama, which allows user to swipe the page easily and see photos listed in different order. In our application, it allows users to view photo listed in the order of date, number of likes, number of dislikes, and number of views.
Moreover, it has an application bar to allow users viewing photos listed in different categories: headline, local, sport, and entertainment. The application bar also enables multiple selections on category, which means users can choose to view photo listed in two or more categories. Both panorama page and application bar can work together for showing photos in different criteria. For example, when a user goes to the “Top Like” page and tick both headline and local in the application bar. Those photos in headline and local category will be shown in descending order of their like numbers.

Similar to the map page, when a user clicks on a photo listed in the album page, the photo id and the session key will be sent to the photo page for database query. After getting the information of the photo from database, the information is shown on the photo page.

```csharp
private void photo_clicked(object sender, RoutedEventArgs e)
{
    string id;

    Button source = (Button)sender;

    id = photo_info[Int32.Parse(source.Name)];
    NavigationService.Navigate(new Uri("/photoPage.xaml?id" + id + "&session=" + sessionKey, UriKind.Relative));
}
```
**Performance Evaluation**

In the performance evaluation plan, we focus on mainly 4 areas to test. They are:

- Photo Encryption Time
- Photo Decryption Time
- Photo Signing Time with Ring Signature
- Verified Ring Signature Signed Photo

We focus on these 4 areas because we would like to know how the time taken for encryption, decryption, signing, and verifying varies with the resolution of a photo. The performance evaluation was carried out on the following 2 windows phones. Their specifications are shown in the following table:

<table>
<thead>
<tr>
<th></th>
<th>HTC 7 Mozart</th>
<th>HTC Radar</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Chipset:</strong></td>
<td>Qualcomm QSD8250 Snapdragon</td>
<td>Qualcomm QSD825 Snapdragon</td>
</tr>
<tr>
<td><strong>CPU:</strong></td>
<td>1 GHz Scorpion</td>
<td>1 GHz Scorpion</td>
</tr>
<tr>
<td><strong>RAM:</strong></td>
<td>576 MB</td>
<td>512 MB</td>
</tr>
<tr>
<td><strong>OS:</strong></td>
<td>Windows Phone 7 Build 7740</td>
<td>Windows Phone 7</td>
</tr>
</tbody>
</table>

Table 1

For the above 4 different cases, we test each with photos with 4 different resolutions, which are 640x480, 1280x960, 1600x1200 and 2048x1536. Each resolution with each case was tested 10 times on each phone, and the average time taken was used to plot the graph to see the performance. After doing the performance evaluation, the results are as follows:

Both encryption and decryption time increase linearly with the resolution on both phones. It is because when the resolution is higher, the file size will be larger, so it takes more time to encrypt and decrypt.

Both signing and verification time are almost constant on both phones. It is because they are carried out on the 128-bit MD5 hash value of the photo on Windows Azure.

For details, please refer to the group report.
Difficulties Encountered

In the implementation of these two pages, I faced both technical and non-technical difficulties. For technical difficulties, both pages need SQL service to get photo thumbnails from the database on Windows Azure. In the photo page, users can filter the photos by checking buttons. Therefore, I need to check which buttons are checked and construct the query to get corresponding thumbnail paths from the database by referencing two tables with the query “SELECT thumbpath FROM photo WHERE photo.id IN (SELECT photoid FROM tags WHERE tag1 = x)”. However, sub-query is not supported in SQL service. There were two methods I thought could handle the problem. One method is to get the photoid from the table tags and store them in an array, then use a for-loop to get information from the table photo “SELECT thumbpath FROM photo WHERE photo.id = [photo_id_array]”. However, this is absolutely not a good method as the number of photos can be huge and it is crazy to execute millions queries. Another method is to attach all photo ID stored in the array to the where clause by the function “OR”. Hence, the query will look like “SELECT thumbpath FROM photo WHERE photo.id = 1 OR photo.id = 2 OR ……”. This method is workable but it makes the query very long and strange.

Therefore, I used another way to solve this problem. It is to create a view on the database to combine information from the two tables first, and then get corresponding thumbnail paths from it.

In the implementation of the map page, since showing thumbnails of photos on map will cause overlapping when two thumbnails are too close. We planned to use customized pictures to show on the map as pushpins. However, there are some unknown bugs caused the pushpins cannot be shown. After spending some time to try to fix the problem, I eventually give up and try another method to do the pushpin as long as it can solve the overlap problem on map.

In the late stage of development, we planned to develop a photo page (Fig. 18 & 19) to show photos as well as facilitating other functions, like rating, buying, and selling. However, the structure of the architecture in album page did not support this further development. It is because the listed of photos in album page did not have an id to identify themselves, so we did not know which photo to show on photo page. Therefore, I have to reconstruct the album page.

The structure is as follows:
To simplify the concept, the structure is to put the image, text block for showing photo id into a stack panel. Then, set the content of a button to the stack panel, and use the name of the button as its id. Finally, display the button list.

For non-technical problems, it is about the cooperation with other group mates. At the beginning of the project, our parts are relatively independent. As the project goes on, our parts are much more interrelated. Every time when there is a change to the table in database or SQL service, I have to modify all parts related to the SQL parts.
My parts need the SQL service to get information from the database. However, SQL service needs to be developed on our own and it is on the hind part of my teammate’s schedule. Therefore, I have to develop on the user interface and other parts first.

The application bar does not support check box. Therefore, I have to use buttons instead of check boxes in filtering. When the user clicks a button, an event is triggered to store the status of the button and change its appearance.

Future Development

In the future, we will improve our application in different aspects, such as to make the application more user-friendly, and add the following new features:

1. **Anonymous Photo Selling**
   The current ring signature scheme supports anonymous upload, but only in an unencrypted way. With little modification in the ring signature scheme, a user should be able to sell a ring signed photo on our system.

   Here we provide a sketch of flow in the modified scheme. First of all, when the photo is encrypted, an AES key is generated. At the same time, a new RSA key pair is generated and the public key is sent to the transaction record. We do ring signature on the AES key, as opposed to the hash value of the photo in the original scheme. When a buyer buys the photo, the AES key is sent to the buyer using RSA technology. The buyer can verify the ring signature on the AES key obtained. Consequently, the buyer upload the e-cash encrypted with the new public key on the transaction record. Only the photo seller with the corresponding private key can decrypt the e-cash. Since it is hard to trace the e-cash, anonymity is preserved.

   There are few issues remain unsolved in the proposed scheme. First of all, the newly generated RSA key pair is not from trusted authority. There might be security issues aroused by the unsecure RSA key pair. Furthermore, there might be an ethical issue aroused in the scheme, as users can upload forbidden photos in order to make money.

2. **News Agencies Subscription**
   We believe that it is the news agencies that are most interested in citizen journalism platforms. Therefore, one possible improvement in our system is to allow news agencies to subscribe to our system.
The subscribed users can have a dedicated button in the page when photo sellers are choosing the tags and comments. The sellers can choose to notify the subscribed users about the photo uploaded. The subscribed users are then immediately notified of the event and decide to buy the photo or not. This is a win-win situation for the photo sellers and subscribed users as the sellers can increase the probability of selling the photo, and the subscribers can get the information of the event as soon as possible.

3. **Map Navigation**
   The system we have implemented can show the users events happening around them. Some of the users interested in the event may like to go directly to the location to witness the event. To improve user experience, the system should guide the user from the current location to the site of event.

   The map navigation can be made possible by integrating navigation services from Google Map or Bing Map.

4. **Blacklisting and Whitelisting**
   We have implemented the “like” and “dislike” function for photos in the system and counting the number of views of the photo. However, there should be a further development in the system in order to rate a photo according to popularity of photos. If a photo is of high popularity, the user should be awarded, and if a user is uploading too many inappropriate photos, we should punish the user.

   We can implement a whitelisting scheme, which allows users of high rating to have priority in photo ranking, so that the photos taken by them are more easily seen by others.

   We can also implement a blacklisting scheme, which bans those users uploading many inappropriate photos. Since we registers users with the Device ID of the phone and the Live ID account, banning the user means the user have to buy a new phone and use a new Live ID in order to join the platform again.

5. **Anonymous Polling using Ring Signature**
   Recently, the Public Opinion Programme of the University of Hong Kong has hosted an online poll on Chief Executive Election. However, to poll using the system, the user has to provide their Hong Kong ID number in order to identify themselves as a valid voter. Some people are concerned that the polling system may link their poll with their Hong Kong ID number and discloses their vote and identity.
As a spinoff of our project, we suggest that the ring signature scheme can solve the aforementioned problem. The system can use ring signature to sign on the vote, so that we can make sure the voter is valid without disclosing their identity.

There are still problems like double voting if we implement ring signature on the voting system, but this is outside the scope of this project.

Conclusions

Recalling the 4 main goals of our project, which are: (1) A two-way process selling photos with encryption and digital signature; (2) ring signature for uploading photos anonymously; (3) a map to let people know the news around them; (4) categorization of photos to improve organization; we have created a photo sharing platform that have met the goals.