The University of Hong Kong
Department of Computer Science

CSIS0801 Final Year Project (2011-2012)
Final Report

FYP11015 A Secure Mobile System to Support Citizen Journalism

Supervisor: Dr. S.M. Yiu

Lam Wan Yin, May 2009148398

Other Group Members:
Li Ka Yau, Ray 2009566813
Lee San Tai, Eric, 2009565273
Lam Harry To Chun, 2009177272
# Tables of Contents

1. **Introduction** ........................................................................................................... 3  
   A. Our Scheme ........................................................................................................... 4  
      i) Design Flow ......................................................................................................... 4  
      ii) Comparison to existing scheme ........................................................................ 8  
      iii) Other functions ................................................................................................. 9  
   B. Assumptions .......................................................................................................... 9  
   C. My Responsibilities ............................................................................................... 10  

2. **Progress of my part** .............................................................................................. 11  
   A. Windows Azure Platform ..................................................................................... 12  
      i) Image Service ...................................................................................................... 12  
   B. SQL Azure ............................................................................................................ 13  
   C. Web Portal ............................................................................................................ 16  
      i) Access Control .................................................................................................... 17  
      ii) Home Page ......................................................................................................... 19  
      iii) Private Page ....................................................................................................... 20  
      iv) Map Page ........................................................................................................... 23  
      v) Transaction Page ................................................................................................ 24  
      vi) Profile Page ....................................................................................................... 24  
   D. App ......................................................................................................................... 25  
      i) Capture ................................................................................................................ 25  
      ii) Access Control .................................................................................................... 25  
      iii) Session Key Generation .................................................................................... 27  

3. **Project Management** ............................................................................................ 28  
   A. Division of Labor .................................................................................................... 28  
   B. Project Time Line .................................................................................................. 29  

4. **Testing and Performance Evaluation** .................................................................... 30  
   A. Photo Encryption (AES) ....................................................................................... 31  
   B. Photo Decryption (AES) ....................................................................................... 32  
   C. Ring Signing Time .................................................................................................. 33  
   D. Ring Verification Time ........................................................................................... 34  

5. **Challenges** ........................................................................................................... 35  
   A. Problems on WCF services .................................................................................... 35  
   B. Encryption on phone .............................................................................................. 35  
   C. Immature SDKr ...................................................................................................... 36  

6. **Further Development** .......................................................................................... 37  
   A. Anonymous Photo Selling .................................................................................... 37  
   B. News Agencies Subscription ................................................................................ 38  
   C. Map Navigation ..................................................................................................... 38  
   D. Blacklisting and Whitelisting ............................................................................... 39  
   E. Anonymous Polling using Ring Signature ............................................................ 39  

7. **Conclusion** ........................................................................................................... 40
1. Introduction

Nowadays, many people have smartphones in their pocket. They use the phone camera to capture bits and pieces of everyday life. However, few people know that their photos are of much news value. In a vibrant and dynamic city like Hong Kong, news is always happening around us. News agencies want to get the first hand news and pictures of the news, but there is always a time lag between the happening of the event and the reporters arrive. In this situation, the smartphones of people near the site of event can help to capture the news in a much faster way. After that, they can sell the photos to news agencies. This is a win-win transaction as news agencies can get the latest news from the citizen photographers, and the photographers can have reward selling the photos.

Although photo sharing is easy on smartphones, it is often hard to have a secure channel to sell photos. Current implementation of photo sharing software doesn’t support encryption and digital signature. Photos may be leaked and tampered during the upload process. In sending photos to news agencies, one may even need to send photos through email. The one way process means people uploading the photo may not be guaranteed to have a reward if the photos are used in publication.

Furthermore, sometimes people may encounter dangerous situations that uploading photos may put the photographer at risk. To solve this problem, we need a tool to upload photos anonymously so that the photos are published but the identity of the photographer is not leaked.

Our project is to implement a smartphone photo-sharing platform that supports transaction of photos. 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

In the project, we aim to solve the problem of complicated photo selling process, as well as promoting citizen journalism to report breaking news in a faster way.
A. Our Scheme

In our project, we aim to speed up the process of photo transaction by using smartphone application and cloud technology. We also try to protect the interest of the seller and buyer by encryption and digital signature. Our project consists of three main pieces: the Windows Azure cloud server, the Windows Phone 7 application and the web portal hosted on Windows Azure.

i) Design Flow

First of all, when a user encountered an event that he thinks is of news value, he can immediately take a photo using the smartphone application. Our application design ensures that the user can finish the photo capture as quickly as possible.
After taking the photo, the user can categorize the photo by adding a tag so that people who are interested in such an event can find the photo quickly. The user can also add comments into the photo as a description to help the buyers to choose.

When the tags and comments are set, the user can then upload the photo. The user can choose to encrypt the photo in here. If the user chooses not to encrypt it, everyone can download the photo. If he does, other users have to buy the photo in order to download it.

![Car accident](image)

Comment: Car accident happened at 4:00am

After clicking the “upload” button, the photo will be uploaded to Windows Azure Blob storage. If the user intended to sell the photo, the photo will be encrypted with 1024bit AES encryption, while the key is stored in the phone waiting for other users to purchase. A thumbnail with watermark will also be uploaded to serve as a preview for customers to choose. The location of the photo taken is uploaded along with the photo.
Next, other users can view the thumbnail of photo along with its comments and tags in the application or web portal. If a user is interested in buying the photo, the user can click the “buy” button, along with the price he wants to give. A push notification will be sent to the seller of the photo for him to choose whether to sell the photo with the price or not.

If the seller agrees to sell the photo, the system will send a push notification back to the buyer, to notify him that the photo is ready to download.
When the buyer click the download button, the aforementioned amount of e-cash\(^1\) will be sent from the buyer to the seller, encrypted by the public key of the seller using 1024bit RSA encryption. At the same time, the AES key of the photo will be sent to the buyer, encrypted by the public key of the buyer, using 1024bit RSA encryption.

\(^1\) The e-cash platform is considered as an external system and is not being implemented in the project

Finally, after the buyer received the AES key, he can decrypt the original photo and save it into the photo library. There will be a check in hash value of the photo so as to ensure the photo is the correct photo the buyer is supposed to buy.
ii) Comparison to existing scheme

The table below shows a sketch of flow in selling photo using our platform, compared to the traditional way of selling photos to news agencies:

<table>
<thead>
<tr>
<th>Step</th>
<th>Traditional approach</th>
<th>Our scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Witness the event</td>
<td>Witness the event</td>
</tr>
<tr>
<td>2</td>
<td>Take a photo</td>
<td>Take a photo</td>
</tr>
<tr>
<td>3</td>
<td>Attach the photo to email</td>
<td>Choose tags and write comments</td>
</tr>
<tr>
<td>4</td>
<td>Send the photo to news agencies and wait for response</td>
<td>Upload the photo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Encrypt with AES</td>
</tr>
<tr>
<td>5</td>
<td>News agencies checked the photo and decided to use in</td>
<td>News agency checked the thumbnail and click buy with a</td>
</tr>
<tr>
<td></td>
<td>publication</td>
<td>price</td>
</tr>
<tr>
<td>6</td>
<td>News agencies contact the photo seller by email/phone</td>
<td>Photo seller decides the appropriate deal and sell the</td>
</tr>
<tr>
<td></td>
<td></td>
<td>photo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AES key uploaded</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Seller receives e-cash</td>
</tr>
<tr>
<td>7</td>
<td>Photo seller receives an award</td>
<td>News agency download the photo and use in publication</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MD5 photo verification</td>
</tr>
</tbody>
</table>

As compared to the original photo selling mechanism, the photo selling process using our scheme is speeded up by a wide margin and is much more secure. The seller does not need to wait for the news agencies for response (step 4 of traditional approach) and can guarantee to receive an award after buyer downloaded the photo (Step 6 of our scheme). Also, through the control in the application, we can limit the photos sold so that it cannot be sold twice. The notification and transaction is fully automatic so that the news agencies do not need to care about contacting the photo seller.
iii) Other functions

In uploading the photo, the user can choose to do a ring signature. The system will not upload the identification of the user who uploaded the photo, so as to protect him from being found by malicious people. The ring signature allows us to ensure the person who uploaded the photo is a valid user of our platform.

In addition to the photo selling mechanism, our platform allows users to find out the news happening around them. In uploading photos, no matter they are encrypted or not, the location of the photo will also be uploaded onto the server. Users can search on the map in the phone application or web portal for news nearby. For news agencies, this would be a handy function as they can find out the accurate location of the news event from the map.

The server application and database are built on Windows Azure cloud platform. This requires us to build services in order to allow communication between the smartphone application and the server. We are also building the application on Windows Phone 7 platform in order to take advantage of the sophisticated access control on Microsoft platforms.

B. Assumptions

During our project, we utilize 3G network and GPS technology. We have made the following assumptions in order to make our project possible:

i) The city has 3G network coverage with fast internet speed
ii) The firewall in the internet connection can allow communication between phone and cloud server
iii) Users has obtained digital signature from trusted authority
iv) The Windows Phone installed with the application is with the following minimum requirement:

- CPU: 1 GHz
- RAM: 512 MB
- Camera: 5MP
- OS: Windows Phone 7.0 or above
C. My Responsibilities

In this project, I am responsible to several parts including:

1. Getting the necessary resources (Windows Phone device and Azure account) for the projects from Microsoft Hong Kong limited
2. Set up and manage the Azure Platform (includes Windows Azure Platform, SQL Azure, Azure Blob and Access Control Service)
   
managing the Windows Azure Platform and the SQL Azure (includes the set up and the management) , development of the whole web portal (includes the function like browsing photos, finding photos via map, access control etc.), development of the image service, build up the access control on the app and develop the session key generation process on the app.

**Timeline of my part:**

<table>
<thead>
<tr>
<th>Activity</th>
<th>From</th>
<th>To</th>
</tr>
</thead>
<tbody>
<tr>
<td>Studying and setting up Azure</td>
<td>Early September</td>
<td>Late September</td>
</tr>
<tr>
<td>Set up of the Azure</td>
<td>Early September</td>
<td>Mid October</td>
</tr>
<tr>
<td>Building Image service</td>
<td>Late September</td>
<td>Late October</td>
</tr>
<tr>
<td>Set up of SQL Azure</td>
<td>Mid Sep</td>
<td>Late October</td>
</tr>
<tr>
<td>Building the web portal</td>
<td>Late October</td>
<td>On going</td>
</tr>
<tr>
<td>Linking up web portal and the SQL Azure and Azure Blob</td>
<td>Late October</td>
<td>Early November</td>
</tr>
<tr>
<td>Implementing the filtering function (Web Portal)</td>
<td>Early November</td>
<td>Mid November</td>
</tr>
<tr>
<td>Implementing the Map function (Web Portal)</td>
<td>Mid November</td>
<td>Late November</td>
</tr>
<tr>
<td>Implementing the “Like”, “Dislike” and “Buy” functions (Web Portal)</td>
<td>Early December</td>
<td>Mid December</td>
</tr>
<tr>
<td>Implementing the Access Control Features (Web Portal)</td>
<td>Mid December</td>
<td>Late December</td>
</tr>
<tr>
<td>Testing on the web portal</td>
<td>Mid December</td>
<td>Early January</td>
</tr>
<tr>
<td>Access Control (app)</td>
<td>Mid January</td>
<td>Early February</td>
</tr>
<tr>
<td>Session Key Generation</td>
<td>Early February</td>
<td>Late February</td>
</tr>
<tr>
<td>User Control (Web Portal and app)</td>
<td>Early March</td>
<td>Mid March</td>
</tr>
<tr>
<td>Push Function (Web Portal)</td>
<td>Mid March</td>
<td>Late March</td>
</tr>
<tr>
<td>Trading Function (Web Portal)</td>
<td>Late March</td>
<td>Late March</td>
</tr>
<tr>
<td>Testing and other enhancement</td>
<td>Early April</td>
<td>Mid April</td>
</tr>
</tbody>
</table>
2. Progress of my part

A. Windows Azure Platform

In our project, we are using Windows Azure Platform, a cloud computing platform, to host the web services, access control service, database and blob storage. The details of the will be mentioned later.

Azure Blob Storage is a storage account for saving all the photos. The blob storage account will generate a URL for each saved objects which allows users to get access in it via different services.

The three web roles (WebPortal, SQLservice and ImageServices) is hosting on Windows Azure which can be accessed in anywhere via internet. Web portal is a ASP.Net web portal for user to access. ImageService and SQLservice are Windows Communication Foundation which is created to connect the app to SQL Azure, Azure Blob and server other purposes.
i) **Image Service**

Image Service is the Windows Communication Foundation for connecting the Windows Phone app with the Azure Blob. Azure Blob is a storage space on Azure that allows us to store uploaded files. The image service is deployed on Azure and added to the app as a service for handling the upload process of the photos. Here is the flow of photo uploading process:

1. The user takes a photo and the photo will be saved at Isolated Storage which is allocated on the phone.
2. The phone app creates a new class called ‘CloudBlobUploader’ and save the photo as a stream file in it.
3. Then the phone app calls the Image Service (which should be deployed on Azure already) and passes the stream file to it.
4. The Image Service connects with the Azure Blob and uploads the stream file.

Here is the code for handling the upload process in the UploadService.svc:

```csharp
public Uri UploadUriWithPathSharedAccessSignature(string userID)
{
    try
    {
        InitializeStorage();
        var container = this.cloudBlobClient.GetContainerReference("Imageservice");
        container.CreateIfNotExist();
        var sas = container.GetSharedAccessSignature(new SharedAccessPolicy()
        {
            Permissions = ContainerSharedAccessPermissions, /* Permissions here */
            SharedAccessExpiryTime = DateTime.UtcNow + TimeSpan.FromMinutes(5)
        });
        var uriBuilder = new UriBuilder(container.Uri + "/" + userID) { Query = sas.TrimStart('?') };
        return uriBuilder.Uri;
    }
    catch (Exception exception)
    {
        throw new WebFaultException<string>(exception.Message, HttpStatusCode.InternalServerError);
    }
}
```

The container of the cloud blob is set to public. All people in the public can download the file on the Azure Blob as long as they have the URL of the file. Implementing session keys in the communication between the app and the Azure Blob will solve the security problems come along with this setting. The security problems aroused will be are discussed in the “Security Analysis” part in the report.
B. SQL Azure

SQLAzure is the Database which host on Windows Azure Platform. We save all the user and photo records on the database. The advantage of using Cloud database is that it has a strong firewall that blocks external access. We can configure the database so that only services we hosted on Azure can access the database. This can prevent malicious attack on our database.

We have created 6 tables in total in the database.

i. photo table
   This table is used to store the record of the photo, which includes url of the photo, uploader record, signatures and location records.

Table 2 - Design of table "photo"

<table>
<thead>
<tr>
<th>Name</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>photo</td>
<td>id (int)</td>
</tr>
<tr>
<td></td>
<td>security (int)</td>
</tr>
<tr>
<td></td>
<td>hash (text)</td>
</tr>
<tr>
<td></td>
<td>z2 (varchar (max))</td>
</tr>
<tr>
<td></td>
<td>h3 (varchar (max))</td>
</tr>
</tbody>
</table>
ii. **tags table**
  The tags table stores all the grouping information of photos. It also stores the number of views and rating of the photos.

**Table 3 - Design of "tags" table**

<table>
<thead>
<tr>
<th>Name</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>tags</td>
<td>photoid (int)</td>
</tr>
<tr>
<td></td>
<td>tag1 (int)</td>
</tr>
<tr>
<td></td>
<td>comment (text)</td>
</tr>
<tr>
<td></td>
<td>likenum (int)</td>
</tr>
<tr>
<td></td>
<td>dislikenum (int)</td>
</tr>
<tr>
<td></td>
<td>viewnum (int)</td>
</tr>
</tbody>
</table>

iii. **user_table**
  The table stores the Live ID, unique device ID, public key and push channel of the user. We check this table to ensure that one phone is only paired with one Live ID, in order to enable user access control. This can also ensure consistency when doing digital signature.

**Table 4 - Design of "user_table"**

<table>
<thead>
<tr>
<th>Name</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>user_table</td>
<td>id (int)</td>
</tr>
<tr>
<td></td>
<td>name (varchar 50)</td>
</tr>
<tr>
<td></td>
<td>liveID (varchar 50)</td>
</tr>
<tr>
<td></td>
<td>publickey (varchar(max))</td>
</tr>
<tr>
<td></td>
<td>currentacc (int)</td>
</tr>
<tr>
<td></td>
<td>channel (varchar(max))</td>
</tr>
</tbody>
</table>

iv. **buy table**
  The table stores all the information of a transaction. The user ID of sellers are buyers are stored here, and the ciphers of AES keys are stored here.

**Table 5 - Design of "buy" table**

<table>
<thead>
<tr>
<th>Name</th>
<th>Columns</th>
</tr>
</thead>
<tbody>
<tr>
<td>buy</td>
<td>buyerid (int)</td>
</tr>
<tr>
<td></td>
<td>photoid (int)</td>
</tr>
<tr>
<td></td>
<td>tranid (uniqueidentifier)</td>
</tr>
<tr>
<td></td>
<td>time (datetime)</td>
</tr>
<tr>
<td></td>
<td>id (int)</td>
</tr>
<tr>
<td></td>
<td>keyCipher (varchar(max))</td>
</tr>
<tr>
<td></td>
<td>price (int)</td>
</tr>
<tr>
<td></td>
<td>status (int)</td>
</tr>
<tr>
<td></td>
<td>money (varchar(max))</td>
</tr>
<tr>
<td></td>
<td>publickey (varchar(max))</td>
</tr>
</tbody>
</table>
v. passtable
The table stores used session keys of users. Session keys allow the users to access other services.

Table 6 - Design of "passtable"

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>passtable</td>
<td>id (int)</td>
<td>used (int)</td>
</tr>
</tbody>
</table>

vi. Push table
The table stores the push messages sent to users.

Table 7 - Design of "Push" table

<table>
<thead>
<tr>
<th>Name</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>buy</td>
<td>id (int)</td>
<td>message (100)</td>
<td>reciverid (int)</td>
</tr>
</tbody>
</table>

We have also created table views so that data can be easily obtain the data. The three (buysummary, summary and topview) table views link the tables of user_table, tags and photo for some features in the phone application and for the data collection in web portal.
C. Web Portal

The web portal is created to increase the accessibility of our system. It provides an alternative way for users (even for non-Windows Phone users) to access our platform and browse the photos easily without having the actual Windows Phone device or without the needs for browsing numerous photos on the limited size of screen of the Windows Phone device.

The web portal is implemented in ASP.Net and leveraging other services such as Access Control on Azure, SQL Azure, Azure Blob and Bing Map. It provides a various number of functions such as allowing user to browse the photos with different requirements, give rating to the photos and do transaction etc.
Access Control is one of the features provided by Windows Azure platform. It allows user to log in the system by different accounts (e.g. Windows Live ID, Yahoo, Gmail etc.) which most of the users already have created at least one account. We can make the login process more user friendly as users do not need to create another account and remember the other set of password.

After the users typed in the log in details (which is done on the page of the service provider), the Access Control will validate the user and issue a secure token to the application. In the process for the users to enter the log in details, the application will not and cannot access any information about the log in details. As a result, the confidentiality of the log in details can be kept.

Also, by using the Access Control, the web portal doesn’t need to implement a log in system and keep the confidential data in our system. Thus, this can enhance our security of the system. The figure below shows an overview of the flow in access control:
The actual flow of access control in the web portal is shown below:

1. User goes to the URL of the web portal and request to log in
2. The web portal will check whether the user have a secure token or not. If not, it will request one from the Access Control
3. The Access Control will direct the user to the login page of the service provider (Windows Live Account is used in the system in our project) and ask the user to log in.
4. The Access Control will validate the user
5. If the user is valid, the access control system will return a secure token to the web portal.
6. The web portal will check whether the token is valid or not. If it is valid, the web portal will direct the user to the home page of the web portal.

After adding the access control to the web portal, we use the following method to get the carrier information and the user ID to identify the user:

```csharp
protected void Page_Load(object sender, EventArgs e)
{
    var identity = User.Identity;

    //get provider
    provider = string.Empty;

    var claimsIdentity = identity as ClaimsIdentity;
    if (claimsIdentity == null)
        provider = string.Empty;

    var providerQuery = from c in claimsIdentity.Claims
                        where c.ClaimType == "http://schemas.microsoft.com/accesscontrolservice/2010/07/claims/identityprovider"
                        select c.Value;
    provider = providerQuery.FirstOrDefault();

    //get nameId
    nameId = String.Empty;

    claimsIdentity = identity as ClaimsIdentity;
    if (claimsIdentity == null)
        nameId = string.Empty;

    providerQuery = from c in claimsIdentity.Claims
                    where c.ClaimType == "http://schemas.xmlsoap.org/ws/2005/05/identity/claims/nameidentifier"
                    select c.Value;
    nameId = providerQuery.FirstOrDefault();

    adduser();
}
```

“nameId” (i.e. the token) is being used for identification. After getting the nameId, the portal will further check if it exists in database or not. If it exists, that means the user has registered his account by logging in our app using a Windows Phone. In this case, the user can enjoy all the function of the portal. However, if the nameId is not existed in the database, the user can only have limited functions (e.g. can’t buy photos) on the portal.
ii) Home Page

In the Home page, users can browse the public photos and give rating to each of it. By clicking on the photo, users can enlarge the photo and the portal will automatically add the number of view of that photo. Users can also like or dislike on particular photos.

After defining the connection string, we can use a List View to show the photos and use it in Configuration Manager to access the database.

To implement the “Like” and “Dislike” functions, we implemented a ListView OnItemCommand and execute the following SQL command for each function.

For adding number of like:
"UPDATE tags SET likenum = likenum + 1 WHERE photoid = (SELECT id FROM photo WHERE path = " + photopath + ");

For adding number of dislike:
"UPDATE tags SET dislikenum = dislikenum + 1 WHERE photoid = (SELECT id FROM photo WHERE path = " + photopath + ");

For filtering the photos by date or by categories, I added selectionchanged event handler on the two calendars SelectIndexChanged function to the ListBox of categories. Whenever the users change the preference by clicking on the calendars or on the ListBox, it will trigger the event handlers and change the condition of the SQL for selecting photos. Then the portal will be reloaded and display the photos under the new condition.

For filtering the photos by date or by categories, I added selectionchanged event handler on the two calendars SelectIndexChanged function to the ListBox of categories. Whenever the users change the preference by clicking on the calendars or on the ListBox, it will trigger the event handlers and change the condition of the SQL for selecting photos. Then the portal will be reloaded and display the photos under the new condition.
Similar to the Home Page, users can browse all the private photos in Private Page and enjoy all the functions in Home Page (e.g. filtering by date or categories, giving rating to the photos and view the photos).

Private photos are the photos which are taken from the users and for the purpose of selling. The main difference between the Home Page and the Private is that users can buy the photos on the Private Page.

By entering a desired amount of money in the textbox below each photos and click buy button, users can bid the photo. If the photo is already sold, the user will not be able to buy it.
For every successful bid, the following SQL command will be executed and the transaction record will be added to the buy table in database:

```csharp
string sqltran = "SELECT * FROM buy WHERE photoid = " + photoid;
SqlCommand cmdtran = new SqlCommand(sqltran, conn);
SqlDataReader readertran = cmdtran.ExecuteReader();
bool seen = readertran.HasRows;
if (seen)
{
    LabelBuy.Text = "Photo id: " + photoid + " is sold already";
} else
{
    TextBox txtprice = (TextBox)e.Item.FindControl("TextBoxprice");
    int price = Int32.Parse(txtprice.Text.ToString());

    string sqlbuy = "INSERT INTO buy (buyerid, photoid, tranid, time, price, status) VALUES (" + buyerid + ", (SELECT id FROM photo WHERE thumbpath = " + photopath + "), newid(), CURRENT_TIMESTAMP, " + price + ", 0)";
```
After the bidding is made, the record in the database will be updated accordingly. At the same time, a push notification will be sent to the seller to notice the seller that the photo has an offer.

The price and the photoid will be shown in the message by running the following code:

```csharp
TextBox txtprice = (TextBox)e.Item.FindControl("TextBoxprice");
int price = Int32.Parse(txtprice.Text.ToString());

string sqlbuy = "INSERT INTO buy (buyerid, photoid, tranid, time, price, status) VALUES (" + buyerid + ", (SELECT id FROM photo WHERE thumbpath = " + photopath + "'), newid(), CURRENT_TIMESTAMP, " + price + ", 0);"

SqlCommand cmdbuy = new SqlCommand(sqlbuy, conn);
int row_count = cmdbuy.ExecuteNonQuery();

string message = "Photoid: " + photoid + " is sold with the price " + price;
myClient.PushToast("Sold", message, uploaderid);
LabelBuy.Text = "You have bought Photo " + photoid + " with the price " + price;
```
The web portal also allows users to view the photos by locations. It will get the locations of all photos from the records on SQL Azure and get the photo from Azure Blob. Next, it will display those photos on the map by using pushpins on recorded location.

To implement the map page, first, we used the Configuration Manager in the Map.aspx.cs page to retrieve the photos records from SQL Azure and pass them to JavaScript function:

```csharp
StringBuilder sb = new StringBuilder("var lats = new Array(); var longs = new Array(); var paths = new Array(); var time = new Array(); var uploaderid = new Array();
sb.AppendLine();
int i = 0;
while (reader.Read()) {
    sb.AppendFormat("lat\ts[{0}]={1};longs[{0}]={2};", i, reader["latitude"].ToString(), reader["longtitude"].ToString());
    sb.AppendLine();
    sb.AppendLine("paths[{0}]=" + reader["path"].ToString() + ";");
    sb.AppendFormat("time[{0}]='" + reader["time"] + ";uploaderid[{0}]={2};", i, reader["time"], reader["uploaderid"].ToString());
i++;
}
ScriptManager.RegisterClientScriptBlock(this, this.GetType(), "PointArrays", sb.ToString(), true);
```

Next, in the JavaScript code, the details of the photos will be saved in arrays in the following format:

```javascript
lats[] = latitude;
longs[] = longitude;
paths[] = URL of the photo;
time[] = time;
uploaderid[] = uploaderid;
```
Then in the map.aspx, we used JavaScript to obtain the information of the photos from the arrays and add the pushpin for each photo:

```javascript
function LoadMap() {
    // Define the map
    map = new VEMap('BingMap');
    map.SetDashboardSize(VEDashboardSize.Large);
    map.LoadMap(new VELatLong("22.283068", "114.136587"), 9, VEMapStyle.Road);

    // Add pin for each photo
    for (var i = 0; i < lats.length; i++) {
        var pin = new VEShape(VEShapeType.Pushpin, new VELatLong(lats[i], longs[i]));
        pin.SetDescription(time[i] + " Uploader is " + uploaderid[i]);
        pin.ShowIcon();
        pin.SetPhotoURL(paths[i]);
        map.AddShape(pin);
    }
    map.ShowDashboard();
}
```

**v) Transaction Page**

<table>
<thead>
<tr>
<th>buyerid</th>
<th>photoid</th>
<th>tranid</th>
<th>time</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>124</td>
<td>8074116-7a5e-43a0-93cf-8670608a17d6</td>
<td>4/7/2012 2:28:09 AM</td>
</tr>
<tr>
<td>202</td>
<td>130</td>
<td>7804452c7b71-48f2-987a b58e030f8474</td>
<td>4/7/2012 2:20:25 AM</td>
</tr>
<tr>
<td>204</td>
<td>131</td>
<td>d46f36e-12eb-43f1-ac1c-66e5c2a4a27e</td>
<td>4/7/2012 11:31:46 AM</td>
</tr>
<tr>
<td>203</td>
<td>133</td>
<td>7865548-69c3-4264-bff2-692731b4d24f</td>
<td>4/7/2012 11:08:35 AM</td>
</tr>
<tr>
<td>202</td>
<td>137</td>
<td>21f8b32a-1b12-43f9-9866-6f5686e56bb</td>
<td>4/7/2012 4:33:22 PM</td>
</tr>
<tr>
<td>206</td>
<td>140</td>
<td>10c609e-1e44-4248-a168-55730487b59d</td>
<td>4/7/2012 1:53:05 PM</td>
</tr>
<tr>
<td>203</td>
<td>139</td>
<td>63894bd-ecb1-f599-a142-958e69e698fe</td>
<td>4/7/2012 1:58:01 PM</td>
</tr>
<tr>
<td>206</td>
<td>141</td>
<td>fbb9f5f-2233-4b2a-bb36-e19b9bb50a3</td>
<td>4/7/2012 1:58:40 PM</td>
</tr>
</tbody>
</table>

This page allows user to check the transaction records. Buyerid is the userid of the buyer, photoid is the id of the photo, time is the transaction time and tranid is the unique id for each transaction.

**vi) Profile Page**

```
Hi! MHRE

User name:  [MHRE]  [Change]

Your Transaction:
```

Profile Page allows user to customize their user profile such as renaming the name of user. Also, user can check all their transactions on this page.
D. App

i) Capture

The main feature of our Windows Phone App is to allow users to capture photos and share it with the others. Thus we aim to make the steps as simple as possible. When the user wants to share the news which may be passed in minute, he just needs to open our app, click the capture button to capture the scene. Then our app will save the photo and also save the location of the user. So the user can share the photos quickly together with their location without tedious process.

ii) Access Control

In order to enhance the user control on the app, users have to use their Windows Live ID to login before using the app. The user is required to login to the Live ID account in Microsoft server. After the user logged in, the app will give return a token to the application as the identification of the user. We cannot recover the Live ID of the user from the token so that the confidentiality of the log in process is kept. After receiving the token, the application will check the user records on the database to see whether the user has logged in before. If it is a first time user, new user record will be added to the database.

If the user is not a new user, the device id of the phone will be recorded in the database. Then the app will further check whether the user has logged in with the same Live ID before. If the user is not using the same Live ID as before on the device, the application will prompt the user to use the same Live ID. This is to ensure that other user cannot obtain the private key stored inside the application. If the user used the same Live ID to login before, the user can get into the app successfully and a new user records will be added to the database.
The `getid` method that return the token to the application on SQLservice.svc is implemented as below:

```csharp
public string getid(String deviceid)
{
    string conn_str = ConfigurationManager.ConnectionStrings["MHREConnectionString"].(ConnectionString);
    SqlConnection conn = new SqlConnection(conn_str);
    conn.Open();
    
    string sqlview = "Select liveID from user_table WHERE (name = '" + deviceid + ")"
    SqlCommand cmdview = new SqlCommand(sqlview, conn);
    SqlDataReader reader = cmdview.ExecuteReader();
    bool getliveid = false;
    string liveid = "null";
    while ((reader.Read()) && (!getliveid))
    {
        liveid = reader[0].ToString();
        getliveid = true;
    }
    conn.Close();
    return liveid;
}
```

On the phone, an event handler `SQLhelper_getidCompleted` will check whether the Live ID get from function `getid` is the same as current Live ID or not.

```csharp
private void SQLhelper_getidCompleted(object sender, SQLservice.getidCompletedEventArgs e)
{
    var simpleWebTokenStore = Application.Current.Resources["swtStore"] as SimpleWebTokenStore;
    string userNameIdentifier = simpleWebTokenStore.SimpleWebToken.NameIdentifier;
    
    if ((e.Result.ToString() == userNameIdentifier) || (e.Result.ToString() == "null"))
    {
        return main();
    }
    else
    {
        MessageBox.Show("Please use the same LiveID as before");
    }
}
```

A new user record should be created while every time the user reinstalls the app because the private key of the user will be deleted when the user uninstall the app. A new set of public key and private key should be generated and a new user record should be created.

However, the old user record is not deleted as that might be used as part of the ring signature in other photo. Deleting the old user record will cause the ring signature can’t function correctly.
iii) Session Key Generation

To enhance the security level on the app, we implemented the session key system in the app. For every time the user opens the app, a session key will be generated on the database and saved on the “passtable” on the database. After verifying the identity of the user, the app will get a valid session key on the passtable by calling the function “getpass” on SQLAzure service. Then for every request from the app to the services, a valid session key will be required for the purpose of ensuring the request is from a trusted authority.

```csharp
public string getpass(String liveid)
{
    string conn_str = ConfigurationManager.ConnectionStrings["MHREConnectionString"].ConnectionString;
    SqlConnection conn = new SqlConnection(conn_str);
    conn.Open();

    //Select the specific user
    string sqlview = "Select * from user_table WHERE liveid = "+ liveid + ";
    SqlCommand cmdview = new SqlCommand(sqlview, conn);
    SqlDataReader reader = cmdview.ExecuteReader();
    bool hasid = reader.HasRows;
    if (hasid)
    {
        //Get a valid sessionkey
        string sqlpass = "Select sessionkey from passtable WHERE (used = 0)";
        SqlCommand cmdpass = new SqlCommand(sqlpass, conn);
        SqlDataReader reader2 = cmdpass.ExecuteReader();
        bool gotpass = false;
        string s = "";
        while ((reader2.Read()) && (!(gotpass)))
        {
            s = reader2[0].ToString();
            gotpass = true;
        }

        //Update the status of the sessionkey
        string sqluse = "UPDATE passtable SET used = 1 WHERE (sessionkey = "+ s +")"
        SqlCommand cmduse = new SqlCommand(sqluse, conn);
        int row_count = cmduse.ExecuteNonQuery();
        //Add new sessionkey in the passtable for further usage
        string sqladd = "INSERT into passtable (used, sessionkey) values (0, newid())"
        SqlCommand cmdadd = new SqlCommand(sqladd, conn);
        int run = cmdadd.ExecuteNonQuery();
        conn.Close();
    }
}
else //if the record of LiveID does not exist
{
    return "no such user";
}
```

In our implementation, the session key is a uniqueidentifier. In further stages of development, the session key can be generated as other more secure crypto algorithm.
3. Project Management

A. Division of Labor

<table>
<thead>
<tr>
<th>System Infrastructure and Database Connection</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Setting up Windows Azure</td>
<td>May</td>
</tr>
<tr>
<td>Setting up SQL Azure</td>
<td>May</td>
</tr>
<tr>
<td>Setting up Access Control Service</td>
<td>May</td>
</tr>
<tr>
<td>Connecting between Windows Phone to SQL Azure</td>
<td>Harry</td>
</tr>
<tr>
<td>Setting up Image Service</td>
<td>May</td>
</tr>
<tr>
<td>Push Notification</td>
<td>Ray</td>
</tr>
<tr>
<td>Connecting between Web Portal to SQL Azure Database</td>
<td>May</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Security</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Building the crypto class</td>
<td>Harry</td>
</tr>
<tr>
<td>Ring Signature Scheme</td>
<td>Harry</td>
</tr>
<tr>
<td>Session Key</td>
<td>May, Ray</td>
</tr>
<tr>
<td>Water mark</td>
<td>Eric</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Apps</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>Ray</td>
</tr>
<tr>
<td>Application login and Access Control</td>
<td>May</td>
</tr>
<tr>
<td>Photo Upload Service</td>
<td>Harry</td>
</tr>
<tr>
<td>Transaction</td>
<td>Harry</td>
</tr>
<tr>
<td>Map Page</td>
<td>Eric</td>
</tr>
<tr>
<td>Album Page</td>
<td>Eric</td>
</tr>
<tr>
<td>Photos Storing</td>
<td>Harry</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Web Function</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Interface</td>
<td>May</td>
</tr>
<tr>
<td>Access Control</td>
<td>May</td>
</tr>
<tr>
<td>Album Page</td>
<td>May</td>
</tr>
<tr>
<td>Transaction</td>
<td>May</td>
</tr>
<tr>
<td>Map Page</td>
<td>May</td>
</tr>
</tbody>
</table>
## B. Project Time Line

<table>
<thead>
<tr>
<th>Tasks</th>
<th>Sep</th>
<th>Oct</th>
<th>Nov</th>
<th>Dec</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Plan</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Web Page</td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Studies on each topic</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Study on Security Linkage</td>
<td></td>
<td></td>
<td>Harry</td>
<td></td>
</tr>
<tr>
<td>Study on Windows Phone 7 apps</td>
<td></td>
<td></td>
<td>Eric</td>
<td></td>
</tr>
<tr>
<td>Study on Web Portal</td>
<td></td>
<td></td>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>Study on Windows Azure</td>
<td></td>
<td></td>
<td>May</td>
<td></td>
</tr>
<tr>
<td><strong>Implementation on App</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Encryption of the Photo</td>
<td></td>
<td></td>
<td>Harry</td>
<td></td>
</tr>
<tr>
<td>Map Function</td>
<td></td>
<td></td>
<td>Eric</td>
<td></td>
</tr>
<tr>
<td>Other functions</td>
<td></td>
<td></td>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>Frame and the linkage to Azure</td>
<td></td>
<td></td>
<td>May</td>
<td></td>
</tr>
<tr>
<td><strong>Implementation on Windows Azure</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Features</td>
<td></td>
<td></td>
<td>Harry</td>
<td></td>
</tr>
<tr>
<td>Azure with Bing Map</td>
<td></td>
<td></td>
<td>Eric</td>
<td></td>
</tr>
<tr>
<td>Other Function</td>
<td></td>
<td></td>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>Access Control</td>
<td></td>
<td></td>
<td>May</td>
<td></td>
</tr>
<tr>
<td><strong>Implementation on App and Web Portal</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Security Features on App</td>
<td></td>
<td></td>
<td>Harry</td>
<td></td>
</tr>
<tr>
<td>Photo Page on App</td>
<td></td>
<td></td>
<td>Eric</td>
<td></td>
</tr>
<tr>
<td>User Interface on App</td>
<td></td>
<td></td>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>Web Portal</td>
<td></td>
<td></td>
<td>May</td>
<td></td>
</tr>
<tr>
<td><strong>Mid-term Review and interim Report</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>All</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Implementation on App and Web Portal</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Transaction</td>
<td></td>
<td></td>
<td>Harry</td>
<td></td>
</tr>
<tr>
<td>Water Mark</td>
<td></td>
<td></td>
<td>Eric</td>
<td></td>
</tr>
<tr>
<td>Push Notification</td>
<td></td>
<td></td>
<td>Ray</td>
<td></td>
</tr>
<tr>
<td>Web Portal And Access Control</td>
<td></td>
<td></td>
<td>May</td>
<td></td>
</tr>
<tr>
<td><strong>Testing</strong></td>
<td></td>
<td></td>
<td></td>
<td>Eric and Harry</td>
</tr>
<tr>
<td><strong>Final Report</strong></td>
<td></td>
<td></td>
<td></td>
<td>All</td>
</tr>
<tr>
<td><strong>Presentation and Exhibition</strong></td>
<td></td>
<td></td>
<td></td>
<td>All</td>
</tr>
</tbody>
</table>
4. Testing

In the performance evaluation plan, we mainly focus on mainly 4 areas:

- Photo Encryption (AES)
- Photo Decryption (AES)
- Photo Signing with Ring Signature
- Ring Signature Verification

In the usage of the application, we found that the above areas are the most time consuming and resource hungry areas. Therefore, through analyzing the performance figures in the above areas, we can find out the bottleneck of the application.

The performance evaluation was carried out on the following 2 phones. Their specifications as shown:

<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 QSD8255 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 7.5 Build 8107</td>
</tr>
</tbody>
</table>

We have tested the phone in the above 4 areas under 4 different resolutions, which are 640x480 (0.3 MP), 1280x960 (1.23 MP), 1600x1200 (1.92 MP) and 2048x1536 (3.14 MP). Each resolution was tested 10 times on each phone, and the average time taken was used to plot the graph to see the performance.
A. Photo Encryption (AES)

Findings:

From the graph, we can see that the photo encryption time increases linearly with the number of pixels. This is a good sign as we can see that there is no bottleneck in encryption process under the tested resolutions.

From the figures, we can also see that the newer CPU in Radar has better performance. We can foresee that the time needed for encryption decrease as technology advance.
B. Photo Decryption (AES)

Findings:

The AES decryption time increases linearly with the increase in resolution of photo, which is the same as in the findings of AES encryption time.
C. Ring Signing Time

Findings:

From the graph, we can see that the ring signature time mostly remain constant in different resolution. This is expected because the ring signature signs on the 128 bit MD5 hash value of the photo. The figures obtained from performance evaluation confirm the prediction. Moreover, we can see that ring signature is a very slow process on the phone. We believe that the slow performance is due to unoptimized code in the ring signature method. The ring signature scheme is coded directly from mathematical formulas, so hardware accelerations are not supported. We believe that in optimizing the code, the performance of ring signature can be greatly increased.
D. Ring Verification Time

Findings:

Similar to signing, the time taken to verify a ring signature is independent to photo resolution. This is because the verification is done on the 128 bit MD5 hash value of photo. In addition, the verification is done in the cloud server, so that it would be less resource-hungry for the phone.
5. Challenges

We encountered several problems during the course of our project. Problems mainly lies on the WCF services, as the system architecture is relatively new to other platforms, there are not much references that we can make use of. Another problem is about the encryption and decryption process. As we have to take care of the performance of the process on slow phones, the encryption and decryption process on phone are a bit troublesome.

A. Problems on WCF services

At the very beginning, we tried to use the Windows Phone Azure toolkit to handle the upload of photos. However, the complicated toolkit drives us to use many custom built libraries and we have to amend the flow of our application in order to fit the service. Moreover, the usage of toolkit is overly restricted, so we decided to go for a more time consuming approach. After studying multiple references, we created photo-uploading services that have high reusability. This is one of the most challenging parts in our project.

B. Encryption on phone

Modern CPUs on phones provide hardware acceleration on encryption and decryption. However, slow phones still give very poor performance on the process. The initial implementation of the crypto class generates RSA keys in 30 seconds. The application will hang there for a very long time and being extremely unresponsive. The AES encryption will also make random crashes on our older phone. After tweaking the encryption process for several times, we are able to achieve acceptable performance on the slowest phone without compromising strength of security.

We are also encountering bugs in encryption and decryption process on the crypto library we used. The library randomly crashes while decrypting photos sent from another phone. It is hard to find out the problem until we solved it in a random occasion. This is one of the unfavorable results of not using more reliable and proved library such as Bouncy Castle. However, if the crypto library we see more and more bugs in the crypto library we are using, we may consider changing it.
C. Immature SDK

We encountered problems when using the Windows Phone SDK to develop applications. Our applications are developed using Silverlight but there are many references cannot be used in developing .Net application due to immature SDK. For example there no library supports the conversion between the images and bytes, which is used to be found in .NET applications, so we need to implement this function by our own. The implementation really wastes our time. Microsoft should add back some commonly used methods to the library to improve the experience in developing Windows Phone application.

Furthermore, there are issues in debugging and using emulators. The windows phone emulator does not reflect the true performance of the phone. It is running according to the configuration of your computer, which is not reasonable at all. The emulator also does not support multitouch, which is a popular feature on new applications. Debugging the application on Windows Phone is also troublesome, as we cannot open the camera while it is connected to the computer. That means we either take pictures without debugging, or debugging without taking pictures. This is not desirable for applications that have to make use of the camera. We have devoted more time on debugging because of the flaw of the SDK.
6. Further Development

The system developed in our project has lots of spaces for improvement. First of all, anonymous photo selling scheme can allow users to upload photos anonymously but also preserve the opportunity to make money out of it. Next, map navigation function can help users to navigate to the site of event in our application. Furthermore, an introduction of blacklisting system can block malicious users uploading fake photos or using photos inappropriately. Last but not least, as a spin off of the project, the ring signature scheme can be applied on the application polling systems so that we can verify a valid user but also protect the identity of the user.

A. 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.
B. 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.

C. 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.
D. 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, that 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.

E. 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 spin off 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.
7. Conclusion

Thanks to this final year project, I gain much more exposure to the environment of mobile apps and cloud computing. Since my major role in this project is focus on the Azure platform and the web portal, I have more chance to explore the possibilities of the cloud computing.

During the development of the project, I understand that this is a challenge since this technology of cloud computing is new and there’s not much reference can be found. But I like the process of discovering and learning during the implementation. And I found that there are many advantages of using cloud computing such as its Access Control features which really can help individual service to check the identity of the user easily without handling the tedious steps of account registration. Also, while using cloud computing, we can deploy a various number of services with ease. It does save much effort while without worrying the problems of building up and hosting a server by ourselves.

Other than Windows Azure platform, Windows Phone is another new technology in our project. While using some of the libraries in C# and Silverlight, it does help us to solve part of the problems. But the libraries are not fully equipped so we do need to use much effort to leverage the resources from different languages.

Overall we have implemented this secure mobile system which achieves the goal for supporting citizen journalism. The photo sharing app and web portal which equipped with photo selling function and a number of security protocols that does satisfy users by sharing the photos easily securely. Some of the features of the system such as photo sharing, transaction and anonymous upload are highly valued for news agent but should not be limited to them only. We hope that this secure mobile system can benefit every single user in various situations.