**PROGRAMMER MANUAL**

To set up the working environment for this project follow the given steps:

1. For android application project environment setup

a. Download and install Android Studio IDE.

b. Install the latest JDK version.

c. Start Android Software Development Kit(SDK). It comes integrated with the IDE.

d.Import the android project source file.

1. For XAMPP server setup

a. Download and install XAMPP server.

b. Import “cdac\_project” folder to the default document root path of XAMPP server. You can also change the default root path using configuration file.

c. Create database and tabels as described in the database section.

Note: Please configure or change the URL’s in the java files of android studio project according to the IP address of your machine.

**Flow of Source code**

1. **conn.php:** check the connection access to mysql database server.

2. **MainActivity.java(App):** When the app is started this is the first code which runs

3. **login\_id.java(App):** Here user creates his id, if valid proceeds to next page if not then enters another valid id. It also sets ‘Qi’ and generates public and private key unique to the user.

4. **registerandroid.php:** works along side login\_id.java to register the user on the server.

5. **registerimage.java(App):** displays 16 images for the user to set one of them ass the password.

6. **imagetag.java(App):** user sets his tag for the corresponding selected image.

6. **updatepwi.php:** ‘pwi’ is update in the database server.

7. **imageportfolio.php:** imageno are stored in the database server.

8. **loginuser.java(App):** user enters his login id here, and validity of server and client is verified.

9. **login\_password.java:** user selects his graphical password.

10. **setag.java:** user enters his tag, if validity approved user successfully logged in.

Various functions used in android studio project:

1. **gethash():** generates the SHA-256 of a string or an image depending on the argument this function takes and returns byte type output.

2. **bin2hex():** takes the return value of gethash() function and return the hex converted hash value.

3. **encriiptdata():** encrypts the data using private key.

4. **encriiptdata1():** encrypts the data using public key.

5. **xorHex():** calculates xor of two string of equal length.