

# My Journal on My Research Work

Hasini Gunasinghe (huralali@purdue.edu)

## **RahasNym**

Until 12th Feb, I mostly worked on improving the paper to be submitted to SACMAT.  
Compare computation complexity with Camenish Lysyankaya signature scheme.  
If we can come up with the key exchange protocol, we do not have to worry about having the To: field.

# Biometrics Research Work

## Android Development:

### Some useful links for study android dev:

1. Starting another activity: <http://developer.android.com/training/basics/firstapp/starting-activity.html>
2. Intents and intent filters: <http://developer.android.com/guide/components/intents-filters.html>
3. Interactions between apps: <http://developer.android.com/training/basics/intents/index.html>
4. Layouts: <http://developer.android.com/guide/topics/ui/declaring-layout.html>
5. Build System: <http://developer.android.com/sdk/installing/studio-build.html>, <http://developer.android.com/sdk/installing/gradle.html>
6. Intent Class spec: <http://developer.android.com/reference/android/content/Intent.html>
7. Android Security Tips: <http://developer.android.com/training/articles/security-tips.html>

### RoadMap:

1. Get two activities in an app communicate. DONE. 1st March.
2. Get two apps communicate and return result. DONE. 14th March.
3. Get an app to communicate with a remote REST service. DONE. 17th March.
4. Integrate ZKP.
5. Run perf test for ZKP with a remote party with static identity.  
Perf test config: Connect laptop to network through wired connection. Run tomcat+services in Windows (so there wont be port forwarding overhead.)
6. Look at the new trend in replacing passwords with FRT and re-write the introduction of the journal. - DONE on 20th March for writing the proposal.
7. Send the update to Prof: Perf. results + introduction re-written. (DEADLINE: 7th April Mid Night.)
8. How to access TrustZone.
9. Look at secure coding of android applications (downloaded).
10. Biometrics work follow.

11. Do an experiment to see if classification gives better results than distance matching, so that we can claim it.
12. I think it is fundamentally wrong to say that you generate BID for a user by giving his/her biometrics features into a classification model that did not use his/her biometrics images to train the model. I think I should also test this during the experiments. But training a model is all about using it during prediction. But this will not preserve uniqueness. So I guess that we need to train a model for each enrolling user. I should do both:
  1. Train without individual's biometrics.
  2. Train with individual's biometrics.
13. I should claim results with statistical significance.
14. I should check if one app can invoke another app's activity through explicit intents - which is a security concern.

### **Wed Jan 20**

I installed Android studio. I referenced this good tutorial: <http://www.androidauthority.com/first-android-app-what-you-need-to-know-619260/>

However, in my Ubuntu, I got an error from android studio saying that SDK or libraries could not be installed.

Then this was the solution: <http://stackoverflow.com/questions/28804863/android-studio-how-to-install-android-platform-tools-on-ubuntu-14-04-64-bit>

It was because Android needs 32 bit libs and I have a 64 bit Ubuntu.

Here are some tips I found to make the emulator fast: <http://developer.android.com/tools/devices/emulator.html>  
linux

### **Feb 19**

Emulator runs Android in a kind of virtual machine, as an Android phone with an Intel processor. This is faster than emulating an ARM processor on your PC.

### **Feb 22**

After lunch, I worked on the mobile app dev. I am still at the very very beginning. Followed first app tutorial till end, and got a problem when running in the emulator. Emulator needs KVM *emulator: ERROR: x86 emulation currently requires hardware acceleration! Please ensure KVM is properly installed and usable. CPU acceleration status: KVM is not installed on this machine (/dev/kvm is missing).*

Then I tried to install KVM based on this tutorial:

<https://software.intel.com/blogs/2012/03/12/how-to-start-intel-hardware-assisted-virtualization-hypervisor-on-linux-to-speed-up-intel-android-x86-emulator>

However, there seems to be problems.

1. When I ran the command at the beginning of that tutorial to check if the CPU supports KVM extensions, I get the output as NO. However, since the error from Android studio shows some hope, I tried to install it.

2. Then the install command given in the tutorial doesn't work. Then I tried this: whose command works. <https://www.howtoforge.com/tutorial/kvm-on-ubuntu-14.04/>

It seems that now I need to relogin to enable KVM for my user accounts.

## **Feb 24:**

Since I had issues in running the hello world app in the emulator due to KVM enabling issue, I thought of checking my bios to see if KVM is enabled.

Before accessing the bios, I needed to backup my important files in bitbucket. Therefore, I spent sometime backing up my files.

Then I spent some fair amount time accessing the bios. First I tried F2 key as mentioned in many online articles which didn't work. Then F12 worked. Intel VT is enabled in the bios.

Then I checked the VMWare settings. So the problem was VT was not enabled in VMWare. When I enabled it, the emulator ran. But it was damn slow.

I searched about how to make the android studio fast. I got some answers:

1. <http://www.viralandroid.com/2015/08/how-to-make-android-studio-fast.html>

This suggested either to test in real device or in genymotion emulator: <https://www.genymotion.com/download>

**Option 1:** I tried to install genymotion emulator, but it requires VirtualBox to run. Hmmm. VirtualBox on VMWare!. So I thought of first running in the real device and see because I anyway have to install it in a real device.

**Option 2:** Option 2 is to install genymotion+virtual box on windows, compile the project in Linux and run it in the genymotion running in Windows. <https://dzone.com/articles/genymotion-simply-best-android>

I tried to install it in my phone (I upgraded my phone too for this reason. My phone is running Android 4.4.2). But somehow it didn't get installed. I was so tired by then and went to sleep.

Read how to make android studio faster:

1. <https://dzone.com/articles/how-speed-android-studio>

2. <http://www.codeproject.com/Articles/803935/How-To-Make-Android-Studio-Really-Fast->

On-A-Window

## Feb 25:

I continued my attempt in installing in the real device. But I still couldn't. I found that there is no apk built when I run the project. So the studio complains that no local path exists. So I need to understand what the heck is going through the Android build process. See how slow my progress is! :(

Oh... Android studio integrated gradle build is a crap!. It didn't even complain that the build was not successful. That is why there was no apk local path existed.

I ran gradle build from command line according to this tutorial:

<http://developer.android.com/training/basics/firstapp/running-app.html>

Steps:

1. invoke the assembleDebug build task using the Gradle wrapper script (gradlew assembleRelease).

```
chmod +x gradlew
```

```
./gradlew assembleDebug
```

I got an error when running ./gradlew: "libz.so.1: cannot open shared object file", followed by: "Exception in thread "png-cruncher\_4" java.lang.RuntimeException: Timed out while waiting for slave aapt process, try setting environment variable SLAVE\_AAPT\_TIMEOUT to a value bigger than 5 seconds".

It seemed that both errors are due to not installing the library: zlib1g as mentioned in: <http://stackoverflow.com/questions/21256866/libz-so-1-cannot-open-shared-object-file>

After doing: `sudo apt-get install zlib1g:i386`, the gradle gave me: BUILD SUCCESSFUL  
Total time: 1 mins 16.537 secs. :)

2. Then I wanted to deploy the app in my phone also from the command line (mentioned in the same above tutorial).

Added android-sdk/platform-tools to PATH and ran: `adb install app/build/outputs/apk/app-debug.apk`

**Yeyyy... I got my first android app installed on my phone! :) Now the floor is mine to do cool stuff.**

Command line worked so smooth compared to crappy android studio.

**Steps to build and run in the phone:**

```
./gradlew assembleDebug
adb install -r app/build/outputs/apk/app-debug.apk
```

I went step further, I edited my app and tried to run again in my app. First I failed because it was already installed. Solution was to install it with the -r option. :)

### **29th Feb:**

I didn't do much. I re-organized my research journal and went to apt with the hope of working more. But I was so tired and went to sleep.

### **1st March:**

Learned many new things. At the end of the day, I have an app running in my phone with two activities linked (<http://developer.android.com/training/basics/firstapp/starting-activity.html>).

1. How to add a new activity and what are the other parts of an android project related with an activity: AndroidManifest.xml, the activity.xml under the res/layout folder, strings.xml file.
2. How to change the layout of an activity and use layout weight.
3. How layouts are related with parent-layout (as indicated in the activity\_layout.xml and AndroidManifest.xml).
4. View and ViewGroups.
5. How to add a method for a view to be responsive. Method should satisfy following:  
Be public. Have a void return value. Have a View as the only parameter.
6. **Intents:**  
Invoke phase:  
create Intent : `newIntent()`,  
add name-value pairs to it: `intent.putExtra(name, value)`,  
start activity passing the intent: `startActivity(intent)`,

Receive Phase:

receive data sent thru an intent: `getIntent()`, `getStringExtra(name)`.

7. Accessing View elements of an activity through:  
`findViewById(R.id.jid_name;)` method.

**Next step:** getting two apps to communicate. I might want to understand intents and intent filters properly before that.

### 3rd March:

I learnt how to invoke an app from one app that accepts an intent and return a result. However, I could not implement. I am writing down notes while I am reading a tutorial which slows me down. IDK

Anyway, below is something I can modify to abstract the high level idea and mention in the implementation details of the journal.

#### Intents:

- Following is from: <http://developer.android.com/guide/topics/manifest/manifest-intro.html#ifs>

The core components of an application (its activities, services, and broadcast receivers) are activated by intents. An intent is a bundle of information (an Intent object) describing a desired action including the data to be acted upon, the category of component that should perform the action, and other pertinent instructions.

- Android locates an appropriate component to respond to the intent, launches a new instance of the component if one is needed, and passes it the Intent object.

- Components advertise their capabilities the kinds of intents they can respond to through intent filters. Since the Android system must learn which intents a component can handle before it launches the component, intent filters are specified in the manifest as `<intent-filter>` elements. A component may have any number of filters, each one describing a different capability.

- An intent that explicitly names a target component will activate that component; the filter doesn't play a role. But an intent that doesn't specify a target by name (implicit intents) can activate a component only if it can pass through one of the component's filters.

#### TODO:

I should check if one app can invoke another app's activity through explicit intents - which is a security concern.

### 14th March:

- I created a new project with two app-modules : authapp and clientapp and started adding the basic logic to it.

Actually, even this simple thing took me several hours to get right. Thing is, when I create a new android project, the app in that project is named just 'app', and doesn't hold the name that I give at the creation time. Although I tried renaming etc, didn't work. So, what I did was: create the android project, leave the default app that gets created, and create two new modules for my use with the names that I want.

- I made the client app to invoke the auth app, passing the SP URL to it.

Auth app printed the passed info and returned the session id to the client app which the client app printed in a toast.

#### **Troubles:**

First the app crashed because the name of the method written for the button didn't match.

Then I learned to debug the app while it is running in the phone.

Then I couldn't invoke the second app because the way I have declared the data field in the intent filter didn't match.

Everything worked after I removed the data field from the intent filter.

#### **15th March:**

I couldn't do much. But I searched for some good tutorial to learn how to invoke a REST service from mobile app.

**Then I found this good and complete tutorial (<http://programmerguru.com/android-tutorial/android-restful-webservice-tutorial-how-to-call-restful-webservice-in-android-part-3>) which uses async Http client (<http://loopj.com/android-async-http/>) which is based on the apache Http client.**

I also drew the sequence diagrams for enrollment and authentication using ZKP, before starting implementing. Then I realized that we need an additional step to allow users to obtain:

- different IDTs with different To fields, with same commitment. For this, user just have to send the IDT issued by the IDP at the enrollment time with To field having IDP name and carry out ZKP to prove that the token belongs to him/her.
- different IDTs with different To fields and different commitments to achieve unlinkability. For this, user needs to send a biometrics capture, IDP runs it through BID generator and create a new commitment. **We follow this approach because we do not store BID anywhere..**

**Question:** will this make the IDP know about the transaction time? No, actually it can be done at periodical windows of a day or week. The first will not require user involvement. And the second will require user involvement.

**TODO:** Measure the performance of the first case. How much time and cpu cost it takes to refill the commitments. But in this case, helper secrets needs to be stored - no, actually only an incrementer will be sufficient, which helps to derive the same secrets later.



**16thMarch:**

I started writing a REST service to be invoked by the android app.

I thought of going for Jersey rather than CXF because, jersey is said to be more light weight and meant for REST, compared with CXF which is more focused on SOAP.

I followed these tutorials to learn. It was very easy. You only need web.xml.

- (2015) <http://tutorial-academy.com/restful-webservice-jersey-maven/>
- (2011) <http://www.mkyong.com/webservices/jax-rs/jersey-hello-world-example/>
- (Not so good, but it has client code.) <http://www.vogella.com/tutorials/REST/article.html#installation>
- (Jersey modules and dependencies:) <https://jersey.java.net/documentation/latest/modules-and-dependencies.html>

**Troubles:**

I got an issue of not being able to access the REST service. It was due to the fact that in servlet filter pattern, we need to specify the path ignoring the path until the webapp's name.

Oh god, now I have basic REST service written in jersey to play around with.

I read a bit about android http clients before start writing the client side for the REST service.

This article gives a very good history: <https://packetzoom.com/blog/which-android-http-library-to-use.html>

But it doesn't mention about async http client. Also, the quora thread doesn't mention about it: <https://www.quora.com/What-is-the-best-library-to-make-HTTP-calls-from-Java-Android>

From the first article above, it seems that OkHttp is promising, but it seems that it requires some additional things to get REST, json, images etc working such as Retrofit..

The reason I didn't go with Volley: <https://developer.android.com/training/volley/index.html> (although it is from google) is that it has very few number of threads as per the first article above. But the plus point is that it is said to work with all types of devices including low power devices.

I thought of going with async http client as it is based on Apache HTTP client, but the only problem is that it says it is compatible with Android API 23 and higher. Let's see. If

problematic, I will switch to OkHttp or Volley.

### **HTTPS clients with Async Client:**

<http://stackoverflow.com/questions/21833804/how-to-make-https-calls-using-asynchttpclient>

### **Troubles and Solutions:**

1. The first problem I got was: the dependency for the async http client was not resolved by Android studio, although I could build the project using command line gradle. I tried several things: File->Invalidate Cache and Restart etc. Only thing which worked was: moving the entry for async http client one entry up in the gradle script. It was this stackoverflow thread which helped me: <http://stackoverflow.com/questions/19508649/android-studio-says-cannot-resolve-symbol-but-project-compiles>
2. There was another wired issue: I couldn't assign response's values to local variable defined in onAuthButtonClicked method of AuthActivity of AuthApp, because the response received in the onSuccess method of an inner class of async http client. I had to declare the variable as final and had to make it an array too. Related stackoverflow thread: <http://stackoverflow.com/questions/27558425/local-variable-access-to-inner-class-needs-to-be-declared-final>
3. So, finally, I was able to build the mobile app with async http client. But I wanted to return the response as JSON. Actually, Jersey has very good support for object to json conversion. This tutorial is a good starter: <http://www.mkymong.com/webservices/jax-rs/json-example-with-jersey-jackson/>.
4. Troubles continues: I couldn't invoke the rest service from the mobile app. I left Hicks undergrad, after my first try.  
Came home, did exercise, cooked, ate, talked with amma, thatththa and Thil and tried again.

I could access the service through host (windows) browser, but not from android. I debugged, and got the exception as: connection time out. Actually, in order to debug, I had to avoid the normal route of client app invoking the auth app, and implement AuthActivity invocation from the mainactivity of the Auth app itself, Silly me didn't realize that the IP of VM is not visible to outside. I went to sleep thinking it might be the reason that async HTTP client is said to be compatible only higher than API 23 or something

**Opportunity:** Prof. asked me to write a proposal for PRF. I would like to write it. I need to steal some good time from my busy schedule to write down one of my ideas for the

proposal. I am thinking of writing the GC idea for biometrics authentication which could avoid some of the drawbacks in our current method. I need to read the previous (NDSS paper) and get an idea where things stand today. May be I should spend the research three days of next week for writing the proposal (Sunday, Tuesday and Thursday).

17th March

Today morning I got up at 7.15 and thought that I will do only PL, cause I spent whole day yesterday on research proj. But I missed the bus at 8am so I thought I will try to connect mobile app to phone till next bus.

The journey:

1. First I tried to access the tomcat home page from phone and ipad browsers, which failed.
2. From this thread (<http://stackoverflow.com/questions/9887621/accessing-localhost-of-pc-from-usb-connected-android-mobile-device>), I tried to make a hotspot in phone and tried to connect to it from laptop. But, as soon as the hotspot is made, the normal internet connection in phone goes off. Seems like it should be made using the data connection in the phone. So that option is out.
3. From some other reading, I realized that firewall may be blocking. Then I tried to disable the firewall and tried to access the tomcat home of server in VM Ubuntu, from phone-no hope. Silly me didn't try to first access a server in host OS and see at least if it works.
4. Then I added an inbound rule in firewall to allow connections on port 8080, since I am scared of disabling firewall. It didn't work either.
5. I installed tomcat on windows and tried to access its home page from phone browser. It worked only after disabling the firewall. But still no luck accessing the tomcata in Ubuntu VM.
6. Then my understanding improved and I checked VM settings. The networking is enabled through NAT, which doesn't make VM visible to outside, that is why the phone couldn't access the tomcat running in VM. HMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM.
7. The solution was to make VM settings such that network mode is: bridge. But so so badly, for some mysterious reason, all the real network adaptors are not made available to VM Ware settings. I do not know if it is done by Windows 8 or if it is

due to some other reason.

I should check : just like enabling virtualization for CPU, whether I need to enable something from bios. But bridging for VMs seems to be something that trivially work for others, but not for me.

Following are some links that I found useful when searching to get over this issue:

- Good article: <https://pibytes.wordpress.com/2012/11/16/vmware-workstation-networking-basics/>

- Threads: <http://askubuntu.com/questions/629457/i-cant-connect-to-virtual-ubuntu-server-from-smartphone>, <http://askubuntu.com/questions/237461/how-do-i-access-ubuntu-server-running-in-virtualbox-from-outside>

As this thread (<http://superuser.com/questions/810097/vmware-player-bridged-networking-no-longer-works-host-win8-1-guest-mint-17-l>) says: lot of Virtual Box related adaptors are made available, which I unchecked in Bridge settings, but still no luck.

As this thread (<http://stackoverflow.com/questions/4601762/how-to-connect-wireless-network-adapter-to-vmware-workstation>) suggests, I also tried making a loop back, as described here: (<https://4sysops.com/archives/how-to-install-loopback-adapter-in-windows-8/>). Still no luck.

8. My final fall back was to use same old NAT for VM and use port forwarding, as suggested in this thread: <http://stackoverflow.com/questions/10355702/connecting-to-apache-web-server-that-is-running-on-a-vmware-from-any-device-in-t>  
The tool made available in this site (<http://www.quantumg.net/portforward.php>): saved me finally. I hope I can trust that is not a malware. Even if it is, I have no other option.

Well, I have one, which is: run tomcat on Windows-*i*copy .wars to it and access from phone. But I feel it will be very time consuming during the dev process. It will be good for perf testing.

## 5th April

I spent the time after 17th March to work on the proposal. I worked very hard on it and submitted on 28th March. Professor liked it.  
But that slowed down my work.

The whole week of 28th March, I could not work on any of my work. On 28th, Prof. asked if I can supervise an undergrad student. I liked it at first and said yes and tried to find ideas for projects. It ate my whole day of 28th Monday. It was about 3 am on Tuesday when I sent the ideas to Prof. I got late when I got up on Tuesday and Tuesday also, I replied to Prof's reply. Then I worked on my PL stuff the whole week - needless to say that I visited PUSH twice during that week, my leg is angry with me again.

I was back to my work on 4th April. I tried to get my old setup working. Unfortunately, my obstacles are not over. I have kept hope on trial port forward, but eats up all CPU when running. So I have to give it away.

My new way forward:

Develop in Ubuntu, build the war, copy it into tomcat running in windows and access from there.

**Baadha walatada Baadha genemin peratama yamu. If I work hard the remaining time with lots and lots of focus and effort, I will be able to succeed this semester. I should ruthlessly say no to anything else.**

Problems I faced:

- I couldn't deploy the war file built and copied from Ubuntu, in the tomcat running in the windows - it gave an exception - the reason was: I have compiled it in a higher version than the java version in which it executed.
- I installed java 1.8.77 on both Ubuntu and Windows, but still the error was there. Restarting didn't help. Then I checked the environment variable in Windows and it was still pointing to 1.7, although java -version on command line printed it as 1.8.
- It went away. But then I couldn't access my Windows IP, neither from the browser in windows nor in phone. I don't know why, but when I connected to Wifie, instead of the wired connection, I could. Phew!!!

Then I planned the next steps in the implementations and that is a lot. I will do it somehow. Then I went to physical therapy for 1 hour. Now I have to continue with my PL work.

## 6th April

Hmmm.... Our 3rd Anniversary. But Thil did not made me feel special. Anyway, I have lot more things to worry about than worrying about it.

I started writting the IDP service, integrating CryptoLib. CryptoLib only implments crypto primitives. It seems that I need to create a ZKP4IDLib for generalizing ZKP for Identity stuff on top of CryptoLib, which can be used by both RahasNym an PrivBioM-TAuth.

**Goal of ZKP4IDLib is to provide a reference implementation of FFS's ZKP identity concept, that can be used with any type of commitment scheme, with concrete implementations for pedersen commitment and FFS's commitment**

If you ever find that the project compiles in maven and errors shown in IDEA, right click pom-¿reimport.

Different ways to take input to RESTful methods: (very good explanation at: <http://stackoverflow.com/questions/10335483/how-do-i-post-a-pojo-with-jersey-client-without-manually-convert-to-json>)

1. payload is taken as an unannotated parameter
2. @FormParam
3. @HeaderParam
4. @QueryParam
5. @PathParam

I finished implementing the IDP service by 8pm I guess.

## 9th April

I started to work after gettig up at 5am and after doing my leg exercises.

I wanted to test the IDP service, first by writing a desktop client.

My goal was to use jersey's POJO-JSON converting ability to easily handle objects.

### Troubles:

1. First of all, it was hard to get post method invoked from jersey client. It is different from get method, which most of the samples provide. I did it in the way how this has done it: <http://stackoverflow.com/questions/10335483/how-do-i-post-a-pojo-with-jersey-client-without-manually-convert-to-json>  
But I could not get it to working. I tried several trouble shooting.  
First of all, the URL I used was wrong, which I got from this tutorial. <http://www.mkkyong.com/webservices/restful-java-client-with-jersey-client/>  
LESSON: You do not have to add get, post etc. at the end of the url.
2. Then I got some exceptions in the enroll method in the service due to some silly mistakes of mine in using AND OR.

3. Then I found there is a serious issue in POJO-JSON conversion support in jersey, because first it complains there is no reader/writer in the client side. When I configured the converters as in: <http://stackoverflow.com/questions/18808018/parse-a-json-response-as-an-object>  
But it gives a parsing error, when parsing the identityCommitment parameter, which is a big integer.
4. Then I tried only decoding manually at the client side, but it also failed, when creating the json object from the response string, the big integer format is changed due to some reason.
5. Then I wrote encoding also manually, to be used at the server side, then only it actually worked.

### **June 19th:**

This is the difference between privacy and security: <https://www.ghostery.com/intelligence/business-blog/privacy/understanding-the-differences-between-privacy-and-/>

### **July 6th, July 14th**

I went through my diary, tried to recap my work and tried get the setup working. I could access the tomcat service through the web browser but couldn't run the phone app. When I was debugging, Android studio suddenly gave the device not found error. This is where I am, trying to get out of it.

### **July 18th**

Too bad I didn't do anything during the weekend. Oh the problem is that USB debuggins is disabled. When I re-enabled it, it listed the USB device.

Commands for the adb and USB device:

1. adb devices
2. lsusb
3. ls -l /dev/bus/usb/002 (the device id)

### **July 23rd**

I worked in the coffee bar. It is a very nice place. I tried to get the mobile app connected to my server. But it didn't work. Could be a problem with the network. So I switched to planning what I need to do. I found certain ways to do performance testing of the mobile app:

<http://www.developer.com/ws/android/development-tools/android-app-performance-testing-an-end-to-end-approach.html>  
<https://developer.android.com/studio/profile/index.html>  
<http://stackoverflow.com/questions/1957135/how-to-test-the-performance-of-an-android-application>

I also found about the async tasks which I should use when invoking remote services from the mobile app:

<https://developer.android.com/training/basics/network-ops/connecting.html#http-client>  
Android suggests to use HttpURLConnection client. But I already use some other client:  
<https://developer.android.com/reference/java/net/URLConnection.html>

### **July 30th**

I was looking for a face dataset.

This seems to be for the algorithms to detect frontal face images - <http://vasc.ri.cmu.edu/idb/html/face/frontal>  
Which is not so useful to me.

This contains a list of datasets:

<http://www.face-rec.org/databases/>

This is a very good tutorial on fisher faces and SVM multi-class classification:

<https://github.com/JaimeIvanCervantes/FaceRecognition>

It gives mixed feelings to find this paper published based on Google ATAP project:

[http://www.cse.msu.edu/rgroups/biometrics/Publications/Face/Crouseetal\\_ContinuousAuthMobileFace\\_ICB](http://www.cse.msu.edu/rgroups/biometrics/Publications/Face/Crouseetal_ContinuousAuthMobileFace_ICB)

Another paper from blackhat:

<https://www.blackhat.com/presentations/bh-dc-09/Nguyen/BlackHat-DC-09-Nguyen-Face-not-your-password.pdf>

### **Aug 19th:**

Designed most of the end-to-end flow of the client and auth apps. I sketched the design.

Implemented little things in the mobile app.

An android activity can be with or without a view. <http://stackoverflow.com/questions/17346102/must-every-activity-have-a-layout>

### **Aug 20th:**

Saturday, I implemented the phone app based on the design I made.



### Aug 22nd:

I implemented the test client with Async tasks. But it didn't work!! Either I have done it wrong, or Async task is not for what I want to do. :(

I was right, according to this SO thread,

<http://stackoverflow.com/questions/19044670/should-i-use-android-async-http-client-in-dobackground-of-async-task>

I do not need to use Async Task if I use asynhttpclient.

So I need to revert the change and continue from there.

Also, I might need to increase the default timeout of async http client if needed.

### Aug 23rd:

Reverted the async task.

Porting the enrollment client to android. How to to a post from asynhttpclient: <http://stackoverflow.com/questions/30594572/send-json-as-a-post-request-to-server-by-asynhttpclient>

I also needed to add my libraries as dependencies into the android project: <http://stackoverflow.com/questions/19932793/syncing-android-studio-project-with-gradle-files>

If it is an external maven module use 'mavenCentral' in the repositories instead.

How to sync android project with updated gradle files: <http://stackoverflow.com/questions/19932793/syncing-android-studio-project-with-gradle-files>

Enrollment client was implemented. Two important links from which I learned how to do a post from asynhttpclient:

<http://stackoverflow.com/questions/30594572/send-json-as-a-post-request-to-server-by-asynhttpclient>  
<https://github.com/loopj/android-async-http/issues/642>

When building from command line in debug mode, I got an error:

Exception is: org.gradle.api.tasks.TaskExecutionException: Execution failed for task ':authapp:preDexDebug'.

There were several SO threads with the same issue, but no body has answered.

When I built from the android studio, it built fine - but later I found out there is no .apk built - I should not trust android studio build, with warnings-saying that my maven dependencies are in a higher java version. Then I noticed that android studio is pointing to java 1.7 - I upgraded it -warnings went away.**This is not Good, as I later read, android studio supports Java 1.7**

Then I was reading how to run gradle from command-line: <https://developer.android.com/studio/build/building-commandline.html>

There are two build types : debugMode and releaseMode. In debug mode, the .apk file is signed using a temp. debug key and in releaseMode, we need to sign it using our private

key by specifying its details in gradle script.

**Like python wrapper, there is a gradle wrapper as well. Need to include the gradle wrapper files in the git repo so that others can use them as well.**

I was further debugging the issue. I noticed that when failing, gradle complains about javax.xml stuff, when I searched about the error, it seemed that I should not conflict with android core libraries, when adding dependencies to the android app.

<http://stackoverflow.com/questions/10098088/dalvik-vm-error-exception-found-javax-xml-namespace-qname-class>

### **Aug 24th:**

Then I removed jersey libraries from ZKP4ID and downgraded the java version of the libraries to 1.7 and of android studio - then it was built fine. But let's see how it goes when running the app.

OMG, OMG, Finally, Finally, I was able to port the enrollment client to the mobile app and run it successfully and get the result. **It was after almost three and half months from I first invoked the IDP from desktop client, that I was able to invoke IDP from mobile client. Somehow I did it. Yeah... :)**

Then I was looking at how to store the enrollment data in the app. This gives a very good overview of android storage options: <https://developer.android.com/guide/topics/data/data-storage.html>

**article from developer.com** <http://www.developer.com/ws/android/storing-app-related-data-in-your-android-apps.html>

Options for me:

Internal Storage for files: classifier, classifier parameters, private keys etc.

SQLite DB for commitment data.

### **SQLite DB:**

This is a very comprehensive example on storing data in the database: <https://developer.android.com/training/storage/databases.html>

First create a contract class.

Then create the DBManager class which is a subclass of SQLiteOpenHelper, which is used to instantiate and obtain getWritableDatabase and getReadable database.

But they say it is recommended to use AsyncTask to getWritableDatabase and getReadable databases.

Then what happens is, when we run performance test, there will be multiple threads trying to read and write from/to the database and there can be problems. So we have to use locks

- like semaphores.

**This is a SO thread that talks about semaphores usage with sqlite db:** <http://stackoverflow.com/questions/11444444/threading-and-database-locking>

### **Aug 25th:**

Following are the recommended things to do to avoid issues when multiple threads accessing database:

1. Keep db instances local.
2. call close() on db in the same method it opened - in finally block.
3. call close on the cursors got from the db - in finally block.
4. Db wrapper that acquires/releases a global semaphore your DB access will be thread safe. Indeed this means that you could get a bottleneck because you are queueing the access to the DB. So in addition you could only wrap the access using semaphores if it's an operation that alters the database, so while you are altering the db no one will be able to access it and wait until the write process has been completed.

I finished implementing the persistence of IDTs based on what I read and thought was correct. But when I ran the app, all sorts of problems.

1. I get an error: Unfortunately, authApp has stopped working, every now and then.
2. When I tried to debug the app, it doesn't go beyond FilterActivity.

Then I got to know about logCat. Let's see if it can help me. Yes, I found it, it was because FilterActivity had a theme as NoDisplay, when I changed it to a normal theme, it worked. Well, now FilterActivity is visible.

**I might want to make filter activity a nice landing page of the authApp when it is invoked from other apps.**

Then I have done a very stupid mistake which ate my time a lot. Rather than using the intent object passed into the callback method: onActivityResult, I have done getIntent. It was so stupid. If I was in the present moment well, I would have detected it better.

Now, with that solved, execution proceeds to DBCode, but get an error due to the SQL query of : create table.

OK, that is solved. Lesson learned: you shouldn't have SQL keywords in column names

And now, the token gets stored in the DB. Yeah.... :)

Then I spend a little bit time desinging the next steps—to forward to AuthActivity from FilterActivity.

Then I had to find out how to pass IDT. It is by serialization. Then I serialized IDT class in ZKP4IDLib.

Now let's see what happens.

### **Aug 26th:**

Well, serializing IdentityToken didn't work, because of the attribute Public Pedersen Params. It is ok, I will just pass the identity token string and decode it from there.

Yeah.. I got the flow of the two app working end-to-end, given that enrollment is performed everytime.

Next steps:

In filter activity, if an appropriate token is there in the database, skip enrollment and proceed to authentication.

### **Note:**

To start an activity from an AsyncTask:

- instantiate the AsyncTask class with the Context passed as a constructor argument, from the first (current) activity.

- call the second activity as this:

```
((Activity) appContext).startActivityForResult(authActivityIntent, AuthConstants.REQUEST_CODE_ZKP_
```

- the result from the second activity will be returned to the onActivityResult method of the first activity.

Then I wrote the code for retrieving information from the SQLite database.

How to write query: <http://www.mysamplecode.com/2011/10/android-sqlite-query-example-selection.html>

### **Aug 27th:**

I didn't do any work because I went to temple.

### **Aug 28th:**

I complete the IdentityTokenReadTask in the FilterActivity and then completing missing parts in ZKP4Lib and writing AuthClient to fulfill ZKPK with the IDT obtained during enrollment.

I need to complete ZKP with SP today somehow. :-) But I couldn't do it.

**Aug 29th:**

I was working on completing the authentication phase, but it was slow - because I got very very sleepy in the morning, had TA meetings and was nervous about the meeting with the Professor. Meeting with Prof. was ok and in the evening I read the paper by ATAP project and learned somethings. I worked until like 9.30pm

**Aug 30th:**

I think I was late. I did the implementation. I had to implement the parts of ZKP4IDLib for authentication and service side and mobile client.

I went to stewart library and worked until late night, I finally drew both the flows at class level.

I was sad that I had a bad conv. with father. I cried and slept.

**Aug 31st:**

Mostly in a bad mode, I setup appointments for doc etc, read the project charters and went to the TA meeting to pick teams. Came home early, got the Water Filter from PVCC. Reconciled and told about my leg problem to parents.

**Sept 1st: Thursday**

Graded project charters and sent emails to team, and worked on scheduling meetings with them.

In the second half, I tested the phone app and services. Fortunately, the first phase of ZKP worked, but not the second phase.

I think the way I invoked was wrong. Second invocation within the response of the first invocation.

**Sept 2nd: Friday**

Couldn't do much as I was replying emails, updating the CV and preparing and meeting with three teams.

However, I tried to change the flow s.t after AuthActivity invokes the initial ZKP, I return the result to Filter activity along with the password etc., so that Filter activity can invoke the SP Auth service for the final auth call. I couldn't complete implementing though.

**Sept 3rd: Saturday**

Literally NO WORK AT ALL.

### **Sept 4th: Sunday**

Worked only in the evening at the PVCC. I realized that rather than getting the AuthActivity to to the initial ZKP call, let FilterActivity do it and if it is successful only, redirect the user to the AuthActivity, so that I do not have to pass the credentials back and forth. However, the initial way I did it was wrong-violating the asynchronacy.

### **Sept 5th: Monday**

This is the most important and new lesson I learned today: <http://blog.trifork.com/2014/07/14/how-to-remotely-debug-application-running-on-tomcat-from-within-intellij-idea/>

How to JAVA debugging works:

External libraries implementing debugger functionality (agents) is passed to remote JVM through java debugger wire protocol (jdwp).

This is related to the remote debugging options that we enable in the jvm: `-agentlib:libname[=options]` format

E.g: `-agentlib:jdwp=transport=dt_socket,address=1043,server=y,suspend=n`

Remote debugging worked after I changed the port to 80.

Finally finally finally - the long awaited static identity zero knowledge proof working end-to-end.

### **Sept 6th: Tuesday**

Started working at 11.20am.

Tips for polishing the apps:

1. Material Design: <https://developer.android.com/training/material/get-started.html#ApplyTheme>
2. Performance best practices: <https://developer.android.com/training/best-performance.html>
3. Testing for the app: <https://developer.android.com/studio/test/index.html>
4. Getting started with testing: <https://developer.android.com/training/testing/start/index.html>
5. Best practices for security and privacy: <https://developer.android.com/training/best-security.html>
6. Android studio overview: Profile your app: <https://developer.android.com/studio/profile/index.html>

For performance testing:

General end-to-end approach: <http://www.devx.com/wireless/Article/48170>

Important thing I learned from this is: there exists hybrid mobile applications in addition

to web based and native mobile applications.

Android specific example: <http://www.developer.com/ws/android/development-tools/android-app-performance-testing-an-end-to-end-approach.html>

Most search results I got was pointing towards using JMeter to capture screens like in webapp testing and running them against the server. But it was not what I wanted to do.

Important things I found: Android device monitor and its parts will be very useful for me.

### **Sept 7th: Wednesday**

I am starting almost at 12noon.

I found this really good tutorial in code labs: <https://codelabs.developers.google.com/codelabs/android-perf-testing/index.html>

It seems to be exactly what I need. But let's see.

There are two types of tests for android: unit tests and instrument tests.

Systrace helps monitor performance: <https://developer.android.com/studio/profile/systrace-commandline.html>

The Espresso library is used to assist with Android instrumental tests: <https://developer.android.com/topic/libraries/support-library/index.html#Espresso>

This is in testing support library: <https://developer.android.com/topic/libraries/testing-support-library/index.html>

The article also explains how to install it.

According to the above tutorial, we can write a perf test, run its gradle task so that what we manually execute will be executed by that test. Very cool.

I read part of the Prof. Dave Evans paper I should work towards that goal for at least 2 hours every day.

I read another remote biometrics authentication work in IEEE Info. Sec and forensics journal. It has references for other works. I should read them for my related work.

From the pointers I found from the above tutorial, I further read developer docs on android test support library and its components such as Espresso.

I guess these two tutorials will be helpful:

<https://developer.android.com/training/testing/ui-testing/espresso-testing.html>

<https://developer.android.com/training/testing/ui-testing/uiautomator-testing.html>

UI thread synchronization is handled by the Espresso framework, so that we do not have to put sleeps.

### **Sept 8th: Thursday**

I started work at 11pm.

Read Prof Evan's paper till 12.20 - understood basic privacy preserving euclidean distance protocol and their improved protocol.

TODO on that direction:

Learn homomorphic encryption better, read pailler and try to implement a small program using that. I could do a benchmark on that in my journal paper.

Learn packing, statistical hiding and perfect hiding

### **List of resources for testing:**

1. Testing support library: <https://developer.android.com/topic/libraries/testing-support-library/index.html> Mentions which dependencies should be added in order to use it.
2. Getting started with testing: <https://developer.android.com/training/testing/start/index.html#config-instrumented-tests>

Instrumented tests are built into an APK that runs on the device alongside your app under test. The system runs your test APK and your app under tests in the same process, so your tests can invoke methods and modify fields in the app, and automate user interaction with your app.

UI testing frameworks like Espresso allow you to programmatically simulate user actions and test complex intra-app user interactions.

3. Automating user interface tests: <https://developer.android.com/training/testing/ui-testing/index.html>
4. Testing single apps using Espresso: <https://developer.android.com/training/testing/ui-testing/espresso-testing.html>
5. Advanced - complete Espresso: <https://google.github.io/android-testing-support-library/docs/espresso/i>



6. Testing multiple apps using UI automater: <https://developer.android.com/training/testing/ui-testing/uiautomator-testing.html>
7. How to repeat tests: <http://stackoverflow.com/questions/39193268/run-espresso-test-multiple-times>, <http://stackoverflow.com/questions/1492856/easy-way-of-running-the-same-junit-test-over-and-over>
8. Test Assertions: <https://developer.android.com/training/testing/start/index.html#AssertionClasses>
9. HamCrest for matching: <https://github.com/hamcrest>

Yay.. I wrote and ran my first instrumentation test.

Steps for the blog:

1. Download the sdk and test support libraries.
2. add dependencies
3. set JUNIT Runner the default runner
4. Write the test
5. Run the connectedCheck gradle task (from android studio or ./gradlew :app:connectedCheck)
6. Run multiple times with @RunWith parameterized class.
6. Add logs.
7. Collect logs via Android Device Monitor.
8. Measure Memory, CPU and Network stats using android device monitor or android monitor.

I was really hurting my leg. Even after giving accupuncture. I actually came home to go to PVCC and work, with the great momentum I had. But there is always some force holding me back. I thought I should give my leg a rest. So I slept. Slept so much that I felt better the next day.

### **Sept. 9th: Friday**

I did zero work on my research. I was preparing for a stupid ineterview, which didn't work out at all.

Washed the clothes, searched for flights to go to GHC.

Walked to Indian shop. I felt everything was so dry. Everything around me.. Everything in my life..

I think I should not loose focus on work. If I keep myself busy with work, everything will be fine.

Then I tried to watch chef table. It started to thunder. Then I ate bread+egg coz I was hungry.

I was preparing to cook - then there came the tornado warning - siren went off three times - it was so scary. I went to the basement and came back.

I stayed in the bed - reading the book - I read read and read like crazy. It was so addicting.

### **Sept 10th: Saturday**

Spellbound again - despite the fact that amma and surprisingly, Thilina called me, I slept again and again and missed my accupuncture session which is bad. I don't know. My instincts say that I should go back to physical therapy - which is scientifically proven and have worked for people I know of - and they are covered by insurance.

I started reading the paper sent by Prof. Bertino in PVCC at about 8pm past. Then amma called and I came home to call.

### **Sept 11th: Sunday**

I went to Hicks, emailed teams, and started to read the rest of the paper. It is in vain that I put lot of time reading that paper. I came home by 10pm bus. Didn't even talk with Thil since he went to see a patient.

### **Sept 12th: Monday**

I was in a very bad mode as I got up at 5.30am and slept again and woke up at 8.20am. I was wondering what's wrong with me to sleep that much, even when I have a meeting with the advisor at 10am. Anyway, I made it early enough for the meeting, was in a bad mode after the meeting due to Pu akka's reply and chatted with Nangi about our plans. Started reading and grading product backlogs at about 12 past, went to the TA meeting and then went to career fair. Useless.

Went to the apt, did exercises, ate, talked with Thil, amma, thaththa. Slept.

### **Sept 13th: Tuesday**

Today was very very good. I wish everyday is like today. I got up with my alarm at 5.30am and slept again. Amma called me at 6.30am and stayed w/o going back to bed again. And it worked. I cooked, did PT and came walking at 7.30 since I had a meeting at TA meeting at 8am. I was so happy that I was able to teach them something.

Read Prof. Evan's paper for 1hour - understood their circuit.

Did grading.

Now back to my research work.

Tip : name the input field in the activities with the activity name as prefix so that it is easy in instrument tests.

I faced an issue that the button click didn't work in instrumentation. The reason is that I haven't cleared the soft keyboard after entering the last text.

Then again I faced an issue of not being able to get the auth activity fields. The issue was solved after deleting the toast.

Yay... I was able to instrument the auth app until it reaches the auth activity.

Not loading the auth activity soon enough gives issues intermittently. Therefore, I added a `Thread.sleep(x)`. But the more elegant approach seems to be in: <http://stackoverflow.com/questions/21417954/thread-sleep>

Yay... I got the complete flow automated... and ran multiple times. Then I read about how to print logs and collect them via Device monitor.

Logs: <https://developer.android.com/studio/debug/am-logcat.html>

Device Monitor: <https://developer.android.com/studio/profile/monitor.html>

### **Sept 14th: Wednesday**

I got up very hard very very hard at 8 o'clock and quickly got ready and went to Cline Clinic. I missed the bus while seeing it. Went to payless, bought some stuff, came to the next bus just in time. I was so hungry, called amma and started cooking. Before that, I graded the one last team. It took a while to cook and made a huge mess. Anyway, I ate and came to lab at 1.30 past. I was idling till and in between I meet teams. Then I came to the stewart library at about 6.00, ate and started to read Dr. Dave Evan's paper. Read for half hour and then slept sitting for half hour. Grabbed a coffee. Got lost in searching stuff. Back to project work at 8.40pm.

### **HOW CAN I MAKE PROGRESS BY WORKING LIKE THIS**

I put logs in several locations. But sleep time was not enough and all the logs of the complete flow is not printed.

### **Sept 15th: Thursday**

WOW.. today is good. Got up at 6.30am. PT, called mamma, came to lab at 8.30am.

Even 100ms of sleep is not enough to run the complete flow. So I increased to 200ms, then only it ran the complete flow all the iterations.

I need to implement the correct way of waiting as in: <http://stackoverflow.com/questions/21417954/espresso-thread-sleep>

Dalvik Debug Monitor Server (DDMS).

This is a very nice explanation on how DDMS interacts with VMs running different apps, via adb: <https://developer.android.com/studio/profile/ddms.html>

“When DDMS starts, it connects to adb. When a device is connected, a VM monitoring service is created between adb and DDMS, which notifies DDMS when a VM on the device is started or terminated. Once a VM is running, DDMS retrieves the VM’s process ID (pid), via adb, and opens a connection to the VM’s debugger, through the adb daemon (adbd) on the device. DDMS can now talk to the VM using a custom wire protocol”.

How to measure:

**Memory:** In Device Monitor, select the device → select the process → select ‘update heap’ → cause GC.

There is a separate tutorial on RAM profiling which I didn’t look into: <https://developer.android.com/studio/profile/ram.html>

**Network:** Can measure Tx and Rx bytes and packets. There is a way to tag the sockets used for each operation and get network stats for each of them as per the aforementioned tutorial. However, it is only possible with android’s default http client - URL connection.

**CPU:** For this, I used Android Monitor.

Setting up: <https://developer.android.com/studio/profile/am-basics.html#byb>

Running CPU monitor: [https://developer.android.com/studio/profile/am-cpu.html?utm\\_medium=android-studio](https://developer.android.com/studio/profile/am-cpu.html?utm_medium=android-studio)

I think we can better measure from Android Device Monitor’s System Information tab.

Related:

Capture screen : <https://developer.android.com/studio/debug/am-screenshot.html>

System Info: <https://developer.android.com/studio/profile/am-sysinfo.html>

**Important thing I learned today:** Usage and difference between Android Device Monitor and Android Monitor.

I worked effectively only half a day. I was hungry, but I went to fountain to read Prof. Evan’s paper, then went to Stewart. Was very hungry. Got a stupid sandwich from

Flatbread. Ate and came to lab. and started reading Naked Security news letter. Slept on the desk for about 45 mins. Got up, read it again. Found some interesting things related to security, most importantly, the TV series. Then I was idling, Manisha called, went home, cooked, watched the 1hr tv series. Talked with Amma only over phone, Thil and Thil's amma. Then wrote an email to Nirosh and went to sleep.

### **September 16th: Friday**

Today is very bad again. Woke up at 9.30. Came to lab at 11.30. Started work at 12noon.

Read Prof. Evan's paper for about 2hours.

**Note:** I should write the difference between biometrics authentication and biometrics identification at least in the paper that I am going to do next and say that there are PP approaches for identification but not for authentication.

Now back to project.

I am learning how to do cross-app UI automation: <https://developer.android.com/training/testing/ui-testing/uiautomator-testing.html>

For that, I need to use UI automator API compared to the espresso API.

When I added uiautomator dependency to client app's build.gradle, I got an error. I got the same error for auth app as well. But I do not remember what was the fix. Then I compared the two build files and found that minSdkVersion should be 18.

In order to do cross app testing, I need to make sure that the UI component is accessible to the testing framework.

Next steps in static identity:

1. Run it multiple times - Yay.. @3.33 on Sept 13th, I automated it 10 times using @RunWith Parameterized.class as mentioned in: <http://stackoverflow.com/questions/1492856/easy-way-of-running-the-same-junit-test-over-and-over>
2. Collect the logs. - DONE: 15th Sept.
3. CPU, MEMMORY, Network Usage - DONE: 15th Sept.
3. Re-factor the flows.
4. Inter-app instrumentation
4. benchmark, benchmark, benchmark

### **MyNexts in end project:**

1. Port the IDP client into mobile - DONE - Aug 24th.
2. Save the commitment in the mobile - DONE - Aug 25th

3. Implement ZKP with SP - almost done and tested on 1st Sept - the initial request goes and response received but there is a problem with challenge-response. - completely done end-to-end - on 5th Sept.

3.2. Polish the app (When I click back from activities, it shouldn't try to connect etc.). 4. Run performance tests - create a performance test harness which can be re-used. Wrote my first instrument test on Sept 8th.

Standard deviation and 99th percentile rank and statistical tests when reporting.

5. Integrate BIOMETRICS.

6. Implement correct way of sleeping in the end product: <http://stackoverflow.com/questions/1492856/easy-way-of-running-the-same-junit-test-over-and-over>

### **Questions to solve:**

1. What is the best way to authenticate to IDP, in order to get the tokens? What harm that could happen if that authentication mechanism is compromised.

Tentative solution: password or certificate based mutual SSL, because no sensitive info. will be stored at the IDP about biometrics, bank account, health care or credit card. But, how the IDP verifies that the correct owner is sending the biometrics is out of the scope of this project. May be that each time the user obtains/renew/revoke, they need to be physically present at the IDP with a photo ID.

Non-of the information can be accessed just by logging into the IDP account.

2. Revocation:

Push based revocation.

3. How to secure the data passed between apps:

i.e: session id s - use encryption.

4. How to write a formal security proof:

**To Dos:**

1. Study the mobile phone's security architecture and find about the latest work in the secure architecture/trust zone.
2. Discuss the attack surface/vulnerability window and argue that it is very small.
3. Look at Daniel's journal paper to see how to prepare a journal paper.
4. Do an experiment to see if classification gives better results than distance matching.

**Literature Survey:****Feb 19**

Today, I was just searching zero knowledge biometrics authentication for remote services. I got a bunch of results - papers and a commercial product.

**Sedicii**

This commercial product : sedicii (<https://www.sedicii.com>), seem to be doing exactly what I have done: ZKP based identity verification/authorization. They say that they do credit card authorization as well as biometrics authorization in ZKP - exactly my two works. They have not described how they do biometrics authentication in ZKP, however, it should be similar to their website logging scenario: I have written how their credit card authorization is comparable to ours in my RahasNym journal.

## Brain Storming:

### Feb 2

I went through their attack descriptions. Their attack target is Android fingerprint authentication/authorization framework, in which the features extracted from user's fingerprint is stored in encrypted form, for matching during authentication time.

They show that these can be stolen if TrustZone (a form of TEE) is not used and that they can be permanently lost since they are not revocable.

Since we do not store the biometrics features as it is (and not even the biometric identifier, which is revocable as well), our approach is secure against the rooting attack they mention. However, since we had proposed to use the TEE to securely store other artifacts, I need to look into the attacks against the TEE further, in the third presentation linked above. I will update you on that too.

### Feb 3

I read the white paper on Exploiting Trust Zone in Android [1].

Summary is: by exploiting a vulnerability in the kernel of TEE, any user-mode application could read from/write to any physical memory location. This has made possible for a malicious local application to read the fingerprint image from sensor which is supposed to be read only by the trusted application related to fingerprint scanning.

However, under the responsible disclosure note at the end of the paper, it says "These vulnerabilities were disclosed to Huawei PSIRT in March 2015, a fix was provided by Huawei in May 2015. CVE IDs were assigned as CVE-2015-4421 and CVE-2015-4422."

### Feb 19

Lot of biometrics based authentication mechanisms are defined for authenticating to devices. Once authenticated into the device, different services that the user accesses are already logged in with username/password security. In such cases, critical remote services are relying on the device biometric authentication, which is not usually strong.

#### **TODO:**

See how device biometrics authentication works in Android and Apple.

Also, if a malware is installed by some mistake by the user, client of the remote service is at risk (password can be stolen, session stolen etc.).

#### **TODO:**

See how bank apps work in mobile devices.

This shows the requirement for remote services to have their own authentication of user to make sure that the genuine user invokes some request, with strong verification, beyond username/password, and without relying on device authentication.



ZKP is a good candidate. There are some previous works, suffers from some drawbacks. Main issue is identity is not static.

The works differ by the approach they address this non-static nature of the biometrics.

This should be a standard mechanism, that any app resides in user's device-communicating with the remote service can integrate easily.

**TODO:**

See this could be developed as a service in Android which could be invoked by other apps.

**Contributions of our work:**

- Secure protocol for remote authentication using biometrics. Preserves good properties of biometrics (i.e: uniqueness). Avoids non-desirable properties of biometrics (i.e: non-repeatability, non-revocability).
- Prototype implementation that is a proof of concept. That can be integrated to any app.
- Security Analysis and Performance Analysis.

**Feb 20**

I started documenting what I wrote down on paper during the weekend. I felt I need a brainstorming/mind mapping tool. And I got FreeMind and noted down different aspects.

**16th March:**

I found these articles which talks about Amazon, Alibaba and Google adopting selfies or FRT to make payments instead of password. But I think it is bad. Coz now, instead of passwords, you store templates to pattern match, which has higher risk than passwords, coz passwords can change and you can use different passwords at different SPs unlike your frt template. You need a secure (in the sense that biometric template is not stored) and privacy preserving (in the sense that user's biometrics are not exposed to third party SPs that the user does not trust.)

**17th March:**

1. <https://thestack.com/security/2016/03/15/amazon-wants-to-replace-passwords-with-selfies-and-videos/>

**Motivation:** Hard to type passwords in small screens.

2. <https://theSTACK.com/cloud/2016/03/03/google-hands-free-facial-recognition-trial-san-francisco/>

**Motivation:** To make it super easy: users do not even have to take out wallet or phone. To protect CCN.

Picture taken from in-store camera is verified against your Hands-free profile picture and the temp pics are deleted.

Hands Free never shares your full credit card number with the store and all your payment details are stored securely and shared only with the payment processor.

The cashier can only charge you when Hands Free detects that your phone is near the store. The cashier then verifies your identity to make sure that they are charging the right person. You'll get instant notifications after every purchase, so you can check purchase details right away.

3. <https://theSTACK.com/security/2015/03/16/alibaba-demonstrates-facial-recognition-payment-system-at-cebit/>

**Motivation:** People forget passwords.

4. <https://theSTACK.com/iot/2016/02/22/mastercard-rolls-out-selfie-verification-for-mobile-payments/>

**Motivation:** To prevent the cost of false decline of transactions.

Consumers will be asked to upload their pictures online to be stored on MasterCard servers. These registered images will then be used as a reference every time a user opts for facial verification during a transaction.

5. <https://theSTACK.com/world/2016/02/19/hsbc-voice-biometric-online-banking/>

**Motivation:** quicker and easier for customers (15 million of its customers by summer 2016).

Biometric banking has been used by other banks in the past, notably Barclays, which rolled out voice recognition to its top 300,000 wealthiest corporate customers in 2014. In the U.S., Citibank has approximately 250,000 customers who have signed up for voice authentication since the company launched the technology in April 2015.

6. <http://www.americanbanker.com/news/bank-technology/banks-embrace-biometrics-but-will-customers-1078867-1.html>

Citi: When customers call in, their voices are matched to the prerecorded data.

7. <http://www.americanbanker.com/news/bank-technology/biometric-tipping-point-usaa-deploys-face-voice-recognition-1072509-1.html>

The key thing, and what may turn out to be USAA's secret sauce, is the company uses device identification in the background, so each time a member logs in, an encrypted token is sent from their phone to USAA that is matched against the ID of the device registered at enrollment. So for a fraudster to successfully impersonate a member with a photo or video (or trying to mimic their voice), they would also have to steal the member's mobile device.

The other safety mechanism is that USAA requires the member to blink, which rules out the use of a static photo or video replay, because video can't blink at the right moment.

**GREAT!** Swenson said USAA's technology analyzes facial bone structure and dimensions, allowing it to see through such alterations (beard, glasses, aging etc.).

8. Apple Touch ID: Activates the scanner on contact which then takes a high-resolution picture of your fingerprint. That fingerprint is then converted into a mathematical formula, encrypted, and carried over a hardware channel to a secure enclave on the Apple A7 chipset. If the fingerprint is recognized, a "yes" token is released. If it's not, a "no" token is released.

Secure Enclave: The A7 also includes an area called the "Secure Enclave" that stores and protects the data from the Touch ID fingerprint sensor on the iPhone 5S and iPad mini 3.[10] The security of the data in the Secure Enclave is probably enforced by ARM's TrustZone/SecurCore technology.

9. ARM's TrustZone: Wikipedia: [https://en.wikipedia.org/wiki/ARM\\_architecture#Security\\_extensions...](https://en.wikipedia.org/wiki/ARM_architecture#Security_extensions...) It provides a low-cost alternative to adding another dedicated security core to an SoC, by providing two virtual processors backed by hardware based access control. This lets the application core switch between two states, referred to as worlds. Open Virtualization[81] and T6[82] are open source implementations of the trusted world architecture for TrustZone. In practice, since the specific implementation details of TrustZone are proprietary and have not been publicly disclosed for review, it is unclear what level of assurance is provided for a given threat model.

**March 21st and March 22nd til 4pm and March 24th from 4pm to mignight  
and March 25th from 11.30am-2pm**

I wrote PFR proposal. It takes lot of time to write.

Professor liked the proposal very much.

She said that she is very confident that I could do the prelim early this Fall.

## **New Ideas:**

I attended Prof. Dongyan Xu's research award talk. I liked how he presented his research work as branches of a tree.

### **0.1 Privacy Preserving Personalized Medicine**

The research that I wanted to do so eagerly, once upon a time.

After the discussion with Fang-Yu about that, he mentioned about indistinguishability obfuscation, in addition to GC based solution that I had in mind.

Today (18th March), I found this paper on that topic: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6244444>  
Candidate Indistinguishability Obfuscation and Functional Encryption for all circuits (Extended Abstract) by Sanjam Garg IBM Research Craig Gentry IBM Research Shai Halevi IBM Research Mariana Raykova IBM Research Amit Sahai UCLA Brent Waters UT Austin

### **PrivBioAuth and PrivBioGeneAuth**

I can have two projects under the umbrella: PrivBioAuth. Two will be: PrivBioMTAuth and PrivBioGeneAuth.

### **How to address the security issues in health care data collected from IoT devices using biometrics.**

Professor suggests to see how to generate keys from biometrics to protect the medical data.

### **Fuzzy Zero Knowledge Scheme:**

**Feb 23:**

I worked on the research the whole afternoon after lunch - but on an apparently useless one. I tried to come up with a fuzzy zero-knowledge protocol based on the ideas from fuzzy identity based encryption. But I had no luck.

I need to look at fuzzy commitment and fuzzy vault before talking about this with anyone else.

Professor mentioned we can talk with Professor Atallah or Italian professors.

### **Storing the machine learning model by encoding it in a garbled circuit**

**Feb 19**

- I got an idea yesterday that I could use the same method of obtaining a circuit from IDP for authentication verification later.

But the circuits can be used only once, which is a problem.

**QUESTION:**

- Good question I can ask in the summer school: are garbled circuits re-usable?

- Also, I argue that due to memory forensics techniques etc, secret information unfolded in the memory can be compromised in our previous approach. Wouldn't this solution have any secret information unfolded in memory? Not even keys? I doubt.

**REFERENCES:**

This is a very nice project website - a reservoir of resources. <http://www.mightbeevil.org/> and <http://www.mightbeevil.org/mobile/>

**27th March**

Completed writing a proposal on this idea to be submitted for PRF funding. Professor Bertino liked it.

## Meetings with the advisor:

### Feb 25:

- I spent lot of time getting ready for the meeting - how I explain what I did, and my ideas to the Professor. At the beginning, she was not in a very good mood. But the end was very good.
- It is ok to have 35% overlap with the conference paper. If I could get the implementation on the mobile phone and do the Performance evaluation, that would be an excellent extension for the journal paper. Need to get it by the end of this semester. Usually we do not submit new work for journals. Only extensions.
- I need to look at Daniel's journal to see how I can prepare my journal paper.
- I need to study the mobile phone android security architecture and write about it.
- Professor mentioned about the new idea of protecting the medical data.
- Professor was interested in two of my new ideas. Professor mentioned we can ask Prof. Atallah or her Italian collaborators. She said it is a very specialized area.
- Professor gave me the two recommendations for the communication requirement and the travel grant application.

### Next Meeting:

I should mention about the issue of recovering from memory images. May be we can ignore it for the moment.

Can we mention about the counter measures that we took in order to avoid the threats arise due to specifics of our architectural decisions and the specifics of the framework?

May be we can mention them as the security best practices for client implementations.

### March 22nd:

I met with Professor to talk about the proposal. I worked hard and sent an initial draft of the proposal to her just about 1 hour before the meeting. She said it looks reasonable. Then I mentioned briefly

### August 19th:

It was a very hard meeting. I knew it was going to be so.

Professor said: you keep saying this to me for two years, I hope you will get it done finally. This time you do not have any excuses and you do not get sick. She took as example Daniele. He has completed the PhD in 2 1/2 years with a 4.0 GPA. She said if he can

do it, she doesn't see any reason why I can't do it. She said I really do not know how to manage time. I should not wait for the perfect time in the future to do the research. In order to publish in top conferences, the student should be really hard working.

Notes to me: I should re-build HASINI.Inc. from the worn out tools and establish my identity.

### **August 29th:**

It was not as bad as the previous meeting.

I told that I completed the enrollment phase end-to-end. I studied about storage options and decided to go with the private database, because Android doesn't allow us to use Trusted Execution Environment.

**Then the professor said that I need to deeply analyze what it means to be private. How they guarantee that other apps can not access it.**

I should find about this and write it in the paper.

Professor also said that I need to measure the performance - that is actually my next big step.

Professor said I am good in research although it take a while for me to get things done - I think that is why she still keeps me as a student, although I didn't get any new publication for three semesters.

She said if I make the paper very competitive, with lot of insights, we can submit to a top journal.

She also asked me to analyze what other people have done and compare ours with them.

My next big step will be to work on the PRF idea with Professor Dave Evans - I really really want to get this going this semester.

I asked why the PRF was rejected - she said they do not give feedback - but asked me to trust my instincts.

I should meet her on next Monday - with performance results and also with some insights about biometrics tests.

### **Sept 5th:**

I sent an email to Prof. Bertino that I completed ZKP end-to-end and asked for a meeting time on Thursday. She only replied that we can meet on next Monday.

### **Sept 12th:**

Met Prof. Bertino. She was not in a good mood. I said about the implementation - she doesn't care about the implementation details at all. Then I said that I changed the flow a



little bit to get tokens with To: identity as SP identity. She said that then we are back to the case where IDP is involved in the loop and asked to get the SP to sign the challenge. But it actually doesn't work when I later thought about it.

Anyway, she said that I can do the prelim once the journal paper is accepted.

I also said that I read the paper she sent. She also said that it is very badly written.

## **Real world stories:**

### **Biometrics:**

1. From: <https://thestack.com/security/2015/03/16/alibaba-demonstrates-facial-recognition-payment-system-at-cebit/>

New York City University's Center for Catastrophe Preparedness and Response (CCPR) published an interesting report [PDF] in 2009 detailing some of the more pervasive problems of FRT systems:

As the size of the identification database increases, the probability that two distinct images will translate into a very similar biometric template increases. This is referred to as the biometric double or twin. Obviously, biometric doubles lead to a deterioration of the identification system performance as they could result in false positives or false negatives.

Ninety per cent of the factors individuating one face from another occur in just 10% of the facial area, and as the enrolled database of volunteered face images grows, the differences can become trivial enough to obtain either a false ID (the wrong person succeeds in identifying as someone else) or a double-match (the probe picture matches more than one face in the database).

Australia's SmartGate FRT technology (pictured right), designed to speed passengers more quickly through airport security, was duped by two similar-looking journalists (Ibid) when first launched at Melbourne Airport in 2002. The two similarly-featured journalists had previously succeeded in duping other facial recognition systems, and successfully got through the SmartGate system after swapping passports.

According to the report, FRT accuracy is very sensitive to the aging effect, stating: For 18 to 22 year-olds, the average identification rate for the top systems was 62%, and for 38 to 42 year-olds, 74%. For every ten-year increase in age, performance increases on average 5% through age 63,

## Getting my theory right:

I really need to learn the primitives I have used in my projects right, so that I can teach them to someone else correctly.

## Zero Knowledge Proof of Knowledge:

1. Pedersen Commitment
2. Zero Knowledge Proof of Identity paper
3. Formal Language for ZKP by Camenish and Stadler (get the reference from DAA paper.)
4. DAA - the first real world implementation of ZKP? Sure? What about Idemix?
5. Implementations:
  - (a) Bringing Zero-Knowledge Proofs of Knowledge to Practice. Endre Bangerter Stefania Barzan Stephan Krenn Ahmad-Reza Sadeghi Thomas Schneider and Joe-Kai Tsay (<https://eprint.iacr.org/2009/211.pdf>)
  - (b) On the Design and Implementation of Efficient Zero-Knowledge Proofs of Knowledge (same authors) (<http://www.lsv.ens-cachan.fr/Publis/PAPERS/PDF/BKSST-speedcc09.pdf>)
  - (c) An implementation of zero knowledge authentication (NARWHAL): <https://courses.csail.mit.edu/6.cheu-jaffe-lin-yang-zkp-authentication.pdf>
  - (d) Implementation and Evaluation of Zero-Knowledge Proofs of Knowledge (<https://securewww.esat.ku206.pdf>)

## Garbled Circuits:

1. First one by Yao
2. Security By Lindel
3. Formalization by Rogway

## Biometrics Features:

1. Refer that paper I reviewed.
2. FisherFaces nice tutorial: <http://www.bytefish.de/blog/fisherfaces/>
3. MultiModal/Fusion:

- (a) Robust Multi-Modal Biometric Fusion via Multiple SVMs : Sabra Dinerstein, Jonathan Dinerstein, and Dan Ventura (<http://axon.cs.byu.edu/papers/dinerstein.smc07.pdf>)
- (b) An Ensemble Approach to Robust Biometrics Fusion : <http://ieeexplore.ieee.org/stamp/stamp.jsp?>