

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.

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.
4. Integrate ZKP.
5. Run perf test for ZKP with a remote party with static identity.
6. How to access TrustZone.
7. Look at secure coding of android applications (downloaded).
8. Biometrics work follow.

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.

TODO: 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/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.

Commands 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. Intent:
create Intent : `newIntent()`,
add name-value pairs to it: `intent.putExtra(name, value)`,
start activity passing the intent: `startActivity(intent)`,
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.

14thMarch:

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

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

Auth app printed the passed infor 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 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.

15thMarch:

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..**

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>

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.

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 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.

New Ideas:

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

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?

REFERENCES:

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

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.