Hardening Fingerprint Authentication Systems Using Intel's SGX Enclave Technology

Interim Progress Report

DELL-EMC Envision the Future Competition 2018

Table of Contents

List of Figures

List of tables

1. Refined Project Description

1.1. Problem statement

Authentication attacks have been a significant problem facing society for the past few years. Several evaluations have shown around 26% of people have been a victim of online account attacks [1]. A common reason behind such attacks is having a fragile password. As a result, various methods have been proposed in order to tackle this issue. Perhaps the most commonly used method is to have a multi-factor authentication system, which can increase the security of the system and that results on having the need to secure the biometric database to dodge surface attacks. Another approach is to use biometrics as a factor for multifactor authentications such as fingerprints to identify individuals.

However, fingerprints are still susceptible to several attacks. The following are possible attacks that would lead to compromising a fingerprint identification system:

- If a malware (e.g. rootkit) is able to hack the operating system, the attacker could read the system's memory and retrieve raw fingerprint data or the set of fingerprint features. The attacker could also modify the data or code in memory to make the system produce wrong match/no-match answers.
- Through an offline attack, the attacker can get access the patterns in the database stored on the local storage.
- The attacker could sniff the data stream as it moves from one system module to another.

1.2. Project scope and expected outcome

The aim of this project is to build a secure fingerprint authentication system that provides end-to end protection of the fingerprint patterns. This would include:

1. Encrypting the data coming from the fingerprint reader using the microcontroller.
2. Encrypting the sensitive fingerprint data while residing in memory or in local storage.

3. Preforming all fingerprint processing (e.g. feature extraction and pattern matching code) in a trusted running environment (SGX's enclave) to maintain its integrity and authenticity.
4. Preventing inappropriate access to main memory segments holding sensitive fingerprints data by other applications or even by the operating system.

Achieving the above goals would lead to hardening the security of fingerprint applications and significantly reducing their attack surface. The proposed approach can easily be extended to other biometric authentication systems such as face and iris recognition. The proposed approach will enable building biometric sensors with secure on-sensor matching (instead of on-host matching used in today's systems), which is a significant contribution as it enables separating the biometric sensing function from the host. The proposed approach can also be used for application scenarios when the biometric database is stored on the cloud.

2. Refined Project Plan

2.1. Milestones

- **Milestone 1: Getting started with the project**

At first we started by identifying the needed hardware components of the project such as SGX hardware, fingerprint scanner, LCD display, and keypad. Since SGX is a new technology that is available in new Intel processors, we had to purchase a computer that supports the SGX technology. Also, to understand this new technology we read the online documentation provided by Intel. Furthermore, we ran some sample programs provided by Intel to understand how SGX works. Then we wrote simple applications for SGX to further understand this technology.

For this project we need a fingerprint toolkit written in C/C++ so that we can integrate inside the SGX (since SGX applications can only be written in C/C++). Thus, we searched for an open source fingerprint SDK which provides reliable feature extraction and matching functions.

- **Milestone 2: High level architecture of the system**

This milestone involved designing the system as a whole by identifying the detailed software/hardware architecture of the fingerprint identification system based on the general framework shown in figure 1. As the project has two scenarios the figure illustrates the second scenario as it is the working prototype.

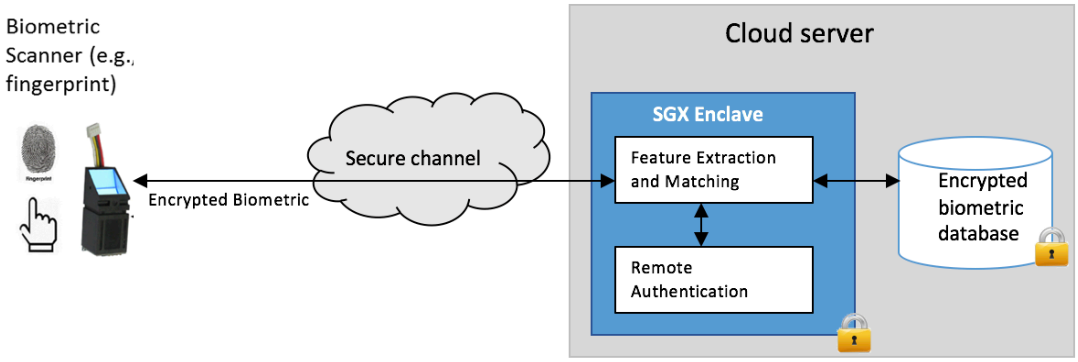

Biometric Scanner (e.g., fingerprint)
Secure channel
Encrypted Biometric
Cloud server
SGX Enclave
Feature Extraction and Matching
Remote Authentication
Encrypted biometric database

Figure 1. Proposed system architecture second scenario

- **Milestone 3: Writing a progress report**

We wrote an initial progress report to document the work that has been done in milestone 1 and milestone 2. First we identified the problem statement and read previous research efforts and related work to our project. We documented our literature survey in in the report. After that we performed requirements analysis of our project and documented these requirements that include design constraints and design standards.

- **Milestone 4: Hardware subsystem implementation**

for hardware subsystem implementation, we built a circuit that contains a fingerprint scanner, an input keypad, and an LCD display. The input keypad is used to allow the user to choose what operation to perform (e.g. adding or deleting fingerprints). The LCD display is used to display information to the user regarding acceptance or rejection of the fingerprint.

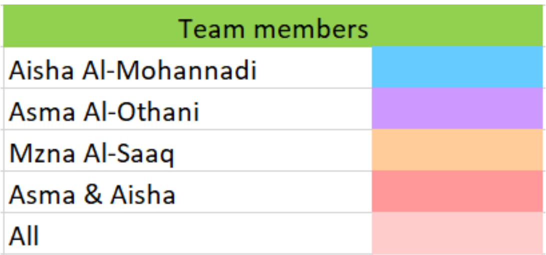
- **Milestone 5: Software subsystem implementation**

For the software implementation, we are working on an open source fingerprint SDK that provides feature extraction and matching functions. We will use this SDK to implement the enrollment and identification functions. The SDK is written in JAVA therefore we are working on converting it to C++ so that we can integrate it and run it inside the SGX enclave.

- **Milestone 6: Testing and evaluating the performance**

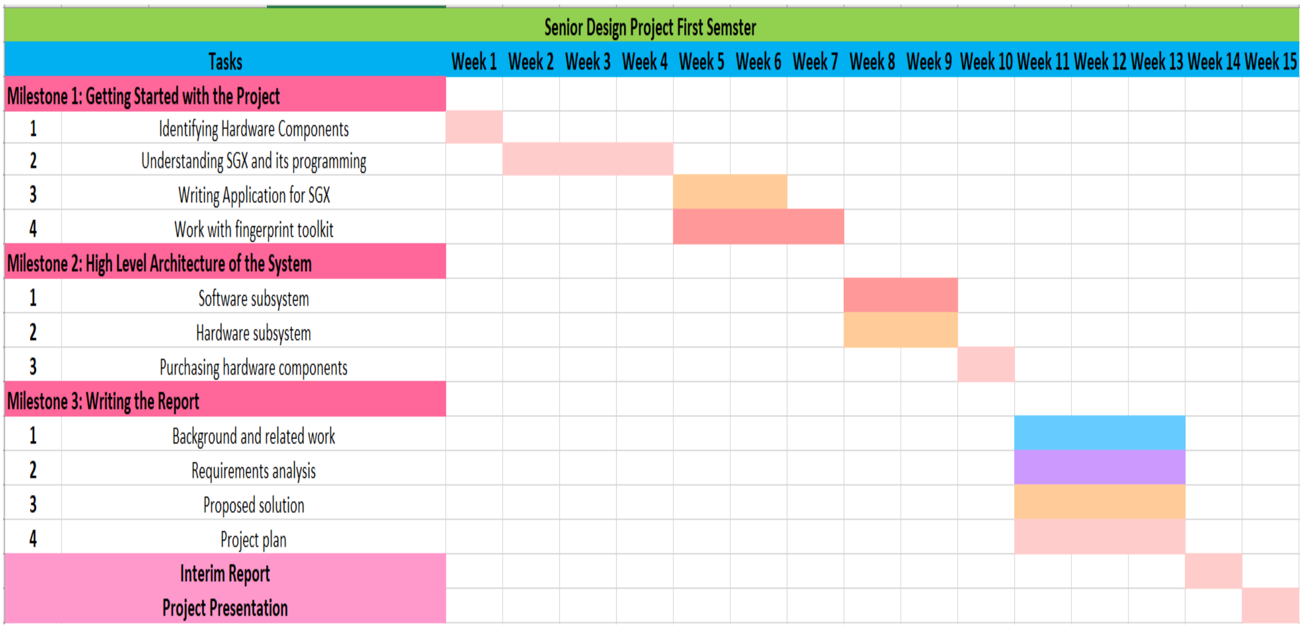
In this stage we will test the hardware and software subsystems of our project. First, to make sure that the hardware circuit is working, we will perform a unit testing by scanning and sending fingerprints to a cloud server. After that we will individually test the various required functions that has to run inside the SGX's enclave. After achieving a working prototype, we will analyze the system to evaluate its performance and its security.

2.2. Detailed schedule and team structure



Team members	
Aisha Al-Mohannadi	🟦
Asma Al-Othani	🟪
Mzna Al-Saaq	🟧
Asma & Aisha	🟥
All	▨

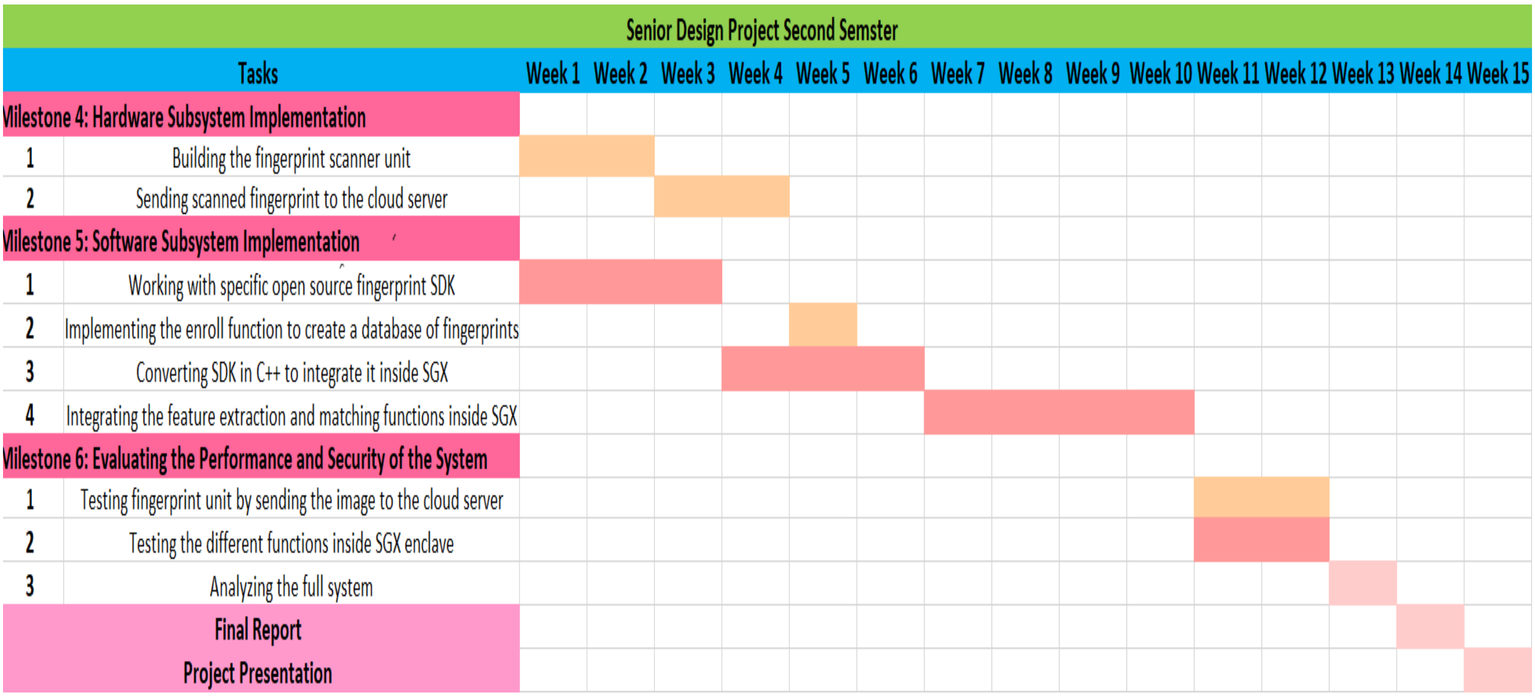
Figure 2. Team structure



Senior Design Project First Semster

	Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
Milestone 1: Getting Started with the Project																
1	Identifying Hardware Components	▨														
2	Understanding SGX and its programming		▨	▨	▨											
3	Writing Application for SGX					🟧	🟧									
4	Work with fingerprint toolkit					🟥	🟥	🟥								
Milestone 2: High Level Architecture of the System																
1	Software subsystem								🟥	🟥						
2	Hardware subsystem								🟧	🟧						
3	Purchasing hardware components										▨					
Milestone 3: Writing the Report																
1	Background and related work											🟦	🟦	🟦		
2	Requirements analysis											🟪	🟪	🟪		
3	Proposed solution											🟧	🟧	🟧		
4	Project plan											▨	▨	▨		
Interim Report															▨	
Project Presentation																▨

Figure 3. Detailed schedule of the first semester



Senior Design Project Second Semster

	Tasks	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6	Week 7	Week 8	Week 9	Week 10	Week 11	Week 12	Week 13	Week 14	Week 15
Milestone 4: Hardware Subsystem Implementation																
1	Building the fingerprint scanner unit	■	■													
2	Sending scanned fingerprint to the cloud server			■	■											
Milestone 5: Software Subsystem Implementation																
1	Working with specific open source fingerprint SDK	■	■	■												
2	Implementing the enroll function to create a database of fingerprints					■										
3	Converting SDK in C++ to integrate it inside SGX				■	■	■									
4	Integrating the feature extraction and matching functions inside SGX							■	■	■	■					
Milestone 6: Evaluating the Performance and Security of the System																
1	Testing fingerprint unit by sending the image to the cloud server											■	■			
2	Testing the different functions inside SGX enclave											■	■			
3	Analyzing the full system													■		
Final Report															■	
Project Presentation																■

Figure 4. Detailed schedule of the second semester

2.3. Contingency and risk mitigation

Table 1. Contingency and risk mitigation table

Risk	**Explanation**	**Approach to minimize the risk**
Incorporating the biometric processing libraries inside the SGX	One of the methodological risks that may be encountered is incorporating the biometric processing libraries inside the SGX. Not being able to do this task will require a much bigger effort to implement biometric libraries from scratch.	This risk is reduced by careful selection of a fingerprint reader that has an SDK with simple functions (e.g. feature extraction, and matching functions) which can be adapted to work inside the SGX.
Reaching maximum size of SGX memory	Reaching the maximum memory size of the protected area is a technological risk that may be faced while working on this project. That could happen because SGX technology has a limit of 64MB or 128MB of	The authentication functions load all of the database when preforming authentication. We will be changing the code so that the function will successively load small chunks of the database.

	protected memory. Therefore, the number active enclaves is limited.	
Side channel attacks	An example of side channel attacks is that SGX doesn’t support oblivious memory access, which will lead to attacks that would exploit knowledge of memory access pattern to reveal sensitive data.	This risk cannot be minimized in our project. However, the garbled computer project at Qatar University is working on a solution for this problem [2].

3. System Requirements

3.1. Requirements elicitation process

Identifying requirements involved performing security analysis on the existing fingerprint systems to identify weaknesses and possible attacks. The requirements should be defined such that any security gaps are appropriately covered. The requirements were also defined by eliciting feedback from fingerprint system users to identify system usability issues.

- **The following are the main requirements of the project:**

  a) Fingerprint image and user choice of what activity the system should perform:

  For the system to start doing its job it needs to have the user’s fingerprint and to know the operation to be performed such as: enroll, Identify, or delete fingerprint.

  b) Performing sensitive functions on the fingerprint image inside SGX’s enclave:

  Sensitive functions such as feature extraction and matching functions need to be executed inside the enclave so that user’s sensitive data are not exposed and the integrity of the functions themselves is preserved

  c) Storing the database in memory in an encrypted form:

  When the enclave gives data to the host application it should give it in an encrypted form, then the application stores them in the database without seeing them in the clear.

  d) Displaying the Result to the user:

After processing user’s data and performing the operation that was desired by the user, the system should display the result to the user about the success or failure of the requested operation.

- **Potential stakeholders and users of the proposed system:**

The proposed system has many critical application scenarios that may lead to commercialization activities beyond this project. For example, banks in Qatar (e.g. QNB) have started to use fingerprints to authenticate and authorize financial transactions. Another example is hardening the security of fingerprints to control access to sensitive facilities in Qatar during the hosting of World Cup 2022.

3.2. System requirements list

3.2.1. Functional requirements

The use case diagram shown in figure 5 illustrates the functional requirement, operations, and activities that the system must perform. The input to the system is the fingerprint image scanned using the fingerprint scanner. The proposed solution enables the user to scan his/her finger; after that the system will show a list of options that the user can choose to perform. The options are: Verify fingerprint, Login admin. If the user wants to verify his/her fingerprint, then scanned image will be encrypted and sent to the host machine. The host machine will perform feature extraction and then it will search for a matching fingerprint in the database. If it finds a match, it will inform the user that his/her fingerprint has a match. If the scanned image is the first image to be stored in the database, then it will be considered as the admin’s fingerprint. Therefore, when the admin scans his/her fingerprint, he/she can choose to login as the admin. The admin can perform the following activities: enroll new fingerprint, delete a fingerprint, or delete all fingerprints. When the admin asks to enroll a new fingerprint, the scanned image will be encrypted and sent to the host machine. The host machine will extract the features from the image and store it in the database. If the admin chooses to delete a fingerprint, then he/she will give the ID of the fingerprint to be deleted. After that, the host will delete the fingerprint with the specified ID from the database.

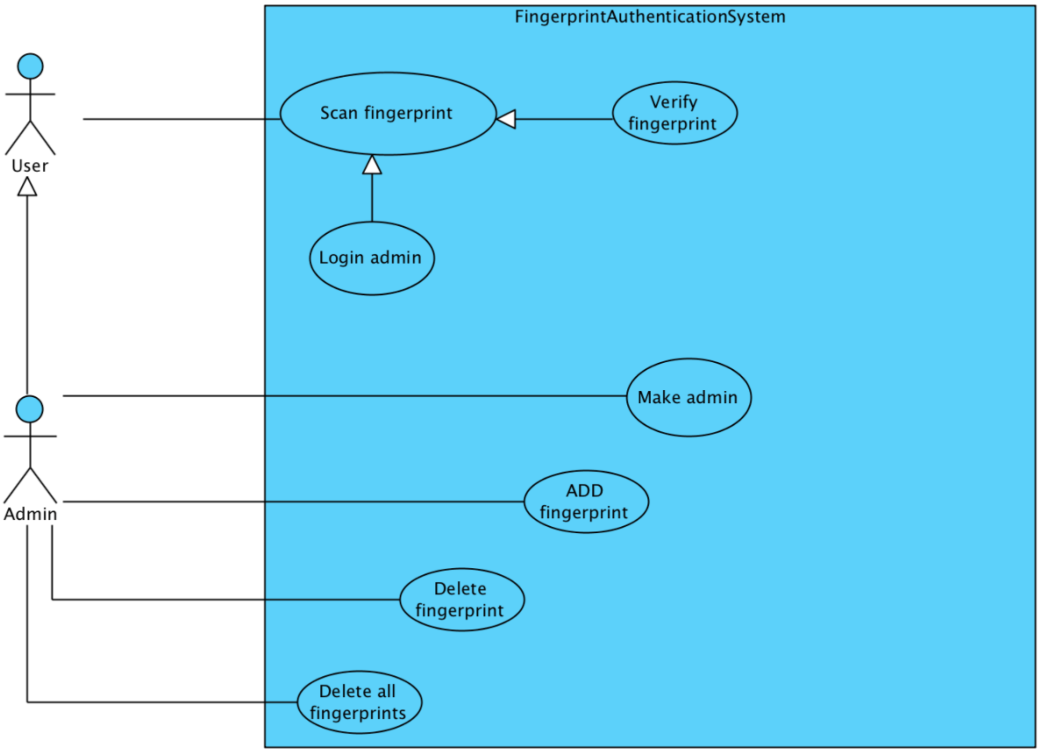


Figure 5. Use case diagram of the proposed system

3.2.2. Non-functional Requirements

Table 2. Technical design constraints

Name	Description
Scanner Output	Size of the output image from the scanner must be 180x 256 pixels.
Scanner Interface with the microcontroller	Connection between fingerprint scanner and microcontroller must be USB connection.
Fingerprint imaging time	The fingerprint scanner must take less than 1 second to capture the fingerprint image.
Size of SGX	The size of memory for all enclaves used in the system must not exceed

memory	64MB or 128MB.

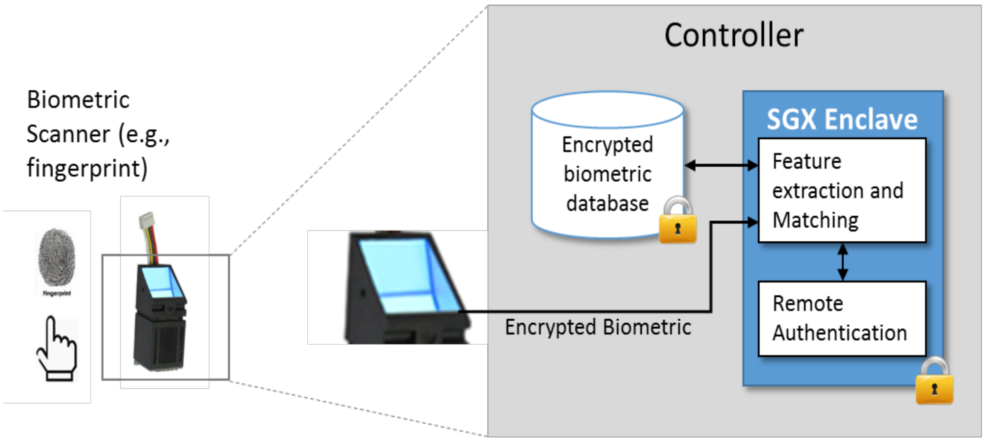
Table 3. Practical design constraints

Type	Name	Description
Economic	Cost	The overall cost of the system should not exceed QR 500.
Environmental	Temperature range	The fingerprint scanner must be operating in temperature between -10°C to +60°C.
	Humidity range	Fingerprint scanner must be operating in humidity that is between 0 - 95 % HR.
Social	Ease of use	The device must be easy for the public to use.
Ethical	Ethical	The system must protect the fingerprint database. Also it must secure the sensitive functions that performs feature extraction and matching.
Availability	Availability	Under normal operating conditions, the system should be available all the time.
Legal	Qatar privacy law	The personal data of an individual must be handled in agreement with standards including those of clarity, integrity and acceptable practices [3][3].
	Biometric privacy law	An individual must not use the system for inappropriate use such as selling the biometric data.

4. System Design

4.1. High level architecture

The proposed solution has two scenarios. In the first scenario, the fingerprint scanner will have the SGX technology inside it and it will perform the feature extraction and matching functions as well as the remote authentication. This way will require the database of fingerprint to be saved inside the scanner. Thus, it will create a safer system because it doesn't communicate with the outside world. However, this scenario is only suitable for single facilities. The second scenario, is suitable for multiple facilities because the fingerprint scanner will only scan the image, encrypt it, and send it to the cloud server. The cloud server will have SGX technology and it will have the database of fingerprints. Thus, the cloud server will be responsible of performing the feature extraction and matching functions and remote authentication.


Biometric Scanner (e.g., fingerprint)
Controller
Encrypted biometric database
SGX Enclave
Feature extraction and Matching
Remote Authentication
Encrypted Biometric

This project involved building and demonstrating the security of a prototype for a fingerprint identification system similar to the ones shown in figure 1/figure 6. The prototype is accomplished through two phases:

1. System design
2. Prototype implementation

System design

In the first phase, we designed a detailed software/hardware architecture of a fingerprint identification system based on the general frameworks shown in figure 4/figure 5. The system will consist of a fingerprint scanning circuit, and host machine or cloud server. The host machine has the encrypted fingerprint database and supports the SGX enclave technology. The code running inside the enclave will perform the feature extraction and remote authentication functions. In the other scenario, the cloud server has the encrypted fingerprint database, feature extraction and remote authentication functions which will be performed inside the server-side enclave. At first the user will scan his/her finger. Using a set of hardware components, the scanned image will be sent encrypted to the host machine. The host machine will perform feature extraction and matching of the image. These functions are trusted because they run inside the SGX enclave. An attacker cannot tamper with these operations to produce false positive answers. The sensitive data that these functions are processing are also stored in the enclave memory. Moreover, all data going out of the SGX enclave are encrypted. In a cloud matching scenario, the fingerprint image will be sent to the cloud server through a secure channel. The cloud server will perform all of the matching and authentication functions.

Prototype implementation

In the second phase, we will be implementing the system that was previously discussed. This project requires both software and hardware components. For the software implementation, we will use libraries, documentation and tools provided by Intel's SGX SDK to develop our project in C or C++. We will split the code of the project into two parts: trusted and untrusted parts. The trusted part will have the sensitive functions such as: feature extraction and matching of the fingerprint image as well as the remote authentication functions. Thus, the trusted part will run inside SGX enclave. This task will involve incorporating relevant parts of the fingerprint scanner SDK libraries to make them compatible with SGX.

The hardware subsystem will use a microcontroller connected to the fingerprint scanner. The microcontroller encrypts the scanned image before sending it to the host/cloud server. An input keypad will be added to the circuit in order to allow the user to initiate functions for adding, deleting or scanning his/her fingerprint. There will also be an LCD display in the circuit which will be used to display information to the user regarding acceptance or rejection of the fingerprint scanning and to inform the user about the success or failure of the requested delete or add operations.

5. System Implementation

5.1. Hardware and software platforms

Table 4. Hardware and software platforms

Hardware:	
Fingerprint Scanner	It will be used to detect the fingerprint and send the data to the microcontroller.
Microcontroller: Raspberry pi	The raspberry pi will be used to encrypt the raw data that is coming from the fingerprint scanner and sends it to host and the cloud server as well as it will print a statement on the LCD whether there is a match or not.
Host and Cloud server that has SGX	The SGX inside the cloud server will be working as a shield for the encrypted data coming from the microcontroller and only the SGX will be able to decrypt the data, process it, and then sends the output back to the microcontroller.
Keypad	An input to the system to allow the user to initiate functions for adding, deleting or scanning his/her fingerprint.
LCD	To print out the result for the user of either having a match or not from the microcontroller.
Software:	
Linux	The software part will be running on Linux operating system platform, as we will be working with it to create our SGX application and enclave using the terminal in the operating system.
SGX SDK	The SDK will help us in building the enclaves.

5.2. Modules/components acquired from external sources

Table 5. Components/ Modules acquired from external sources

Fingerprint SDK	The function of the toolkit that we will take advantage of is an open source SDK that provides feature extraction and matching functions.
Fingerprint scanning SDK	The SDK used for scanning the fingerprint image from the scanner is a licensed commercial SDK.

6. Other Relevant Issues and Challenges

6.1. Technical

The main technical challenges are related to identifying a usable and reliable fingerprint processing SDK and converting it to a format that enables its integration and execution inside SGX. The following summarizes the main technical challenges:

- Finding a simple fingerprint SDK with feature extraction and matching functions in C/C++.
- Converting the found SDK from java to C++.
- Removing the system input/output code from the fingerprint SDK so that it can be integrated inside the enclave. That is because the enclave only communicates with the application and does not allow any direct system input or output.
- Integrating the SDK inside the SGX.

7. References

[1] Impermium Study Unearths Consumer Attitudes Toward Internet Security. (2013,June 28) Retrieved from https://www.darkreading.com/attacks-breaches/impermium-study-unearths-consumer-attitudes-toward-internet-security/d/d-id/1140045

[2] Malluhi, Q. et al. “The Garbled Computer: Towards Computing without Seeing,” Computer Science and Engineering, Qatar University. Project supported by QNRF, NPRP Exceptional Program, 2016-2018.

[3] New National Privacy Law in Qatar. (2016, November 8). Retrieved from https://www.dentons.com/en/insights/alerts/2016/november/9/new-national-privacy-law-in-qatar