Executive Summary

This report investigates design and implementation details of Fingerprint Based Authentication System (FBAS). This design consists of hardware and software specifications.

The system implemented involves the Verifi P4000 sensor and OmniPass software, based on the Bioscrypt Core algorithm. Simple Systems Solutions Consulting (SSSC) has completed this project to the best of our abilities, establishing knowledge base for the future projects in this area.

Simple Systems Solutions Consulting has included in this final report the detailed approach, an in-depth evaluation and analysis of the results obtained. The total cost of implementing the resulting Fingerprint Based Authentication System is approximately $130.00.

Table of Contents

List of Figures

Nomenclature

Binarization – a filtering process done to a grayscale image that goes over all pixels of the image to mark them as either black or white.

Biometrics - methodology that enables an automated identification of a person's biological traits for the purpose of verification of his or her unique identity.

Bitmap image -a data file or structure representing a rectangular grid of pixels. The color of each pixel is individually defined.

Equal Error Rate (EER) - performance indicator of a biometric system, the point where FAR and FRR are the same.

False Acceptance Ratio (FAR) - a performance indicator of a biometric system based on a ratio of falsely to correctly accepted matches.

False Rejection Ratio (FRR) – a performance indicator of a biometric system based on a ratio of falsely to correctly rejected matches.

Failure to Enroll (FTE) - a performance indicator of a biometric system based on a ratio of rejected to accepted enrollments.

Fingerprint - an imprint made by the pattern of ridges of a human finger.

Graphic User Interface (GUI) - is a method of interacting with a computer through a direct manipulation of graphical images and widgets on the screen.

Hashing - an algorithm for summarizing or probabilistically identifying data. Such summary is known as a hash value and the process of computing such a value is known as hashing.

Minutiae - the points of interest in a fingerprint, such as bifurcations, ridge endings or ridge enclosures.

Radio-frequency (RF) – imaging method where a fingertip is energized with a low-intensity radio wave, skin acts as a transmitter and distance variation between ridges and valleys can be detected by an array of suitably tuned antenna pixels

Ridge bifurcation - the point where a ridge forks or diverges into branch ridges

1. Introduction

1.1 Problem Statement

The requirements of embedded systems are continuously becoming more demanding, with security being proposed as a new required design dimension. The safety of information on such devices is important in today's society where unauthorized access can be detrimental. The primary concern should be the security of these embedded devices with the architecture being developed accordingly. Using an embedded system that requires user authentication can be a solution to ensure the security of information transactions. The traditional methods of user authentication that involves assuming surrogate identity such as swipe cards and passwords, pin numbers, etc. are limited as security measure because they fail to provide a unique and permanent method of identification for every individual. Biometrics offers a method of identifying individuals by their measurable unique physical or behavioral characteristics.

Biometrics can be used to further increase the security of the information stored in all electronic devices. The biometric approach is generally superior to conventional method of assumed surrogate identity when a higher level of security is required without typical drawbacks of increased inconvenience to users. In this project we will investigate the potential solutions to support secure and efficient Fingerprint Based User Authentication on embedded systems.

1.2 Objectives

The purpose of this project was to research a method of implementing a fingerprint based authentication sensor on an embedded computing device. The recommended system should satisfy the security issues of unauthorized access while maintaining the basic principles of an embedded device.

1.3 Constraints

The recommended system must adhere to the following constraints:

- Sensor should be capable of operation from –30C to +45C
- System should not generate false positive matches
- System should not generate rejected enrolls

1.4 Criteria

The recommended design solution must have the following criterions:

- Fast and reliable access
- Minimize access and processing time
- Modularity is an important factor because this design must be re-usable for future improvements and upgrades.
- Intuitive end user interface

1.5 Assumptions

- All users must have at least one finger and are physically capable of placing it on the sensor.

2. Background

2.1 Biometrics

Biometrics is the technology or discipline that recognizes a person's biological or behavioral characteristics thereby verifying the identity of the corresponding individual, [10]. A more restricted definition of biometrics refers to the methodology that enables an automated identification of a person's biological traits for the purpose of verification of his or her unique identity.

Biometric systems can be used in two different modes: identity authentication and user identification. Identity authentication occurs when the user claims to be already enrolled in the system (presents an ID card or login name); in this case the biometric data obtained from the user is compared to the user's data already stored in the database. Identification occurs when the identity of the user is unknown. In this case the user's biometric data is compared against all the records in the database; the user's data can match a record in the database or he/she is not in the database. This project will focus on the authentication mode of the biometric system.

2.2 Fingerprinting

Fingerprints are the most widely used biometric features for personal authentication. The traditional method of obtaining (or enrolling) fingerprints used ink to get the fingerprint onto a piece of paper. Then this piece of paper was scanned using a traditional scanner. This method is becoming obsolete as it is replaced by faster automated methods. Currently, fingerprint sensors are used to obtain fingerprints.

Fingerprint sensors are commonly based on optical, thermal scanning, radio-frequency, pressure and a number of others principles. Optical fingerprint sensors are based on measuring changes in luminous intensity between light reflected from valleys and light from ridges. Thermal fingerprint sensor's measures the temperature differential between the skin ridges and the air caught in the fingerprint valleys. Using the radio-frequency method a fingertip is energized with a low-intensity radio wave; the skin acts as a transmitter and the distance variation between ridges and valleys can be detected by an array of suitably tuned antenna pixels. The pressure method uses a pressure-sensitive pixel array that can be constructed from piezo-electric elements that capture the pattern of ridges in a fingerprint pressed against it. All biometric fingerprint sensors capture a fingerprint as a bitmap image. In the scope of this project, we will use the radio-frequency method for the Fingerprint Based User Authentication system. We chose this method as one of the more tamper-proof, reliable and readily available sensor technologies.

2.3 Fingerprint Based User Authentication

There is a wide range of security products available from various companies that revolve around fingerprint-based user identification. A variety of products are available from companies such as: Atmel, AuthenTec, Bioscrypt and Verifi. Most of these companies share component design and software algorithms.

The key functionality of a typical fingerprint-based authentication system can be categorized into enrollment, authentication and storage. Schematic diagrams of enrollment and authentication systems are shown in *Figure 1* and *Figure 2*, respectively.

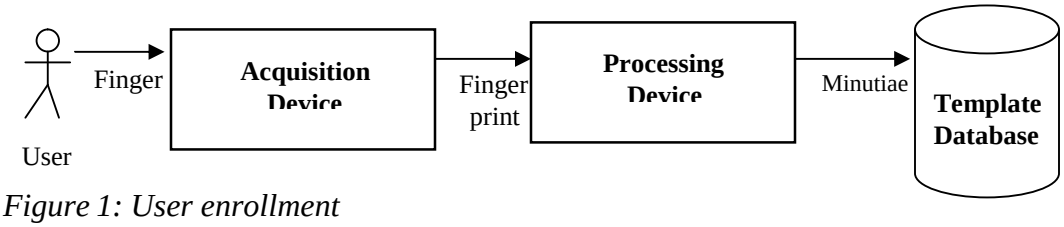

Figure 1: User enrollment

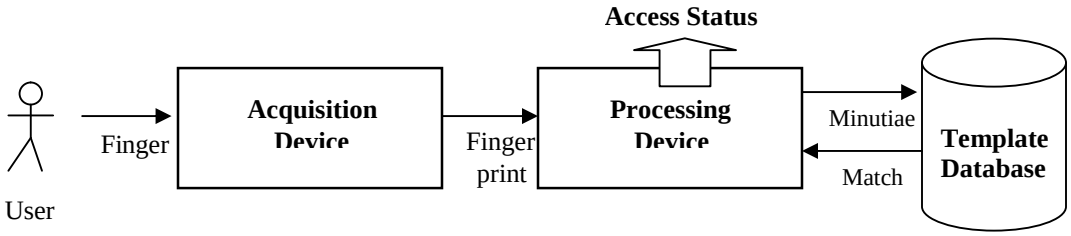

Figure 2: User authentication

During the user enrollment phase an acquisition device captures an image of the user's fingerprint. A series of image processing procedures are applied to the fingerprint image in order to identify it. Processed fingerprints are stored in a template database for future use. During user authentication phase the system compares presented fingerprint image against the reference template database. A reference score is then calculated and the user is considered authenticated if the score exceeds a specified threshold.

2.4 Biometric System Performance Evaluation

The main difference between any biometrics system and a traditional authentication is that the biometric system cannot generate an exact answer. In the case of biometrics, biological information or its reading could change so matching scores against templates could change accordingly. As a result, even a valid person may be rejected or a wrong person may be accepted. The ratios developed to evaluate the probabilities of the two cases are called False Rejection Ratio (FRR) and False Acceptance Ratio (FAR). Additionally, the number of rejected enrollments, Failure to Enroll (FTE), is also important. The point where FAR and FRR are the same is called EER (Equal Error Rate), and is used to evaluate performance of the system.

3. Methodology

3.1 Fingerprint Sensor System

The implemented Fingerprint Based Authentication System uses the Verifi P4000 sensor that is based on the AuthenTec TruePrint RF technology. TruePrint technology is based on the principles of the radio-frequency (RF) electronic imaging. This type of a sensor scans the fingerprint pattern from a conductive layer of skin that lies beneath the skin's dry outer surface layer. The electric fields created between the conductive surface of the sensor and the skin replicates the shape of the conductive skin layer in the amplitude of the RF field, refer to *Figure 3.*

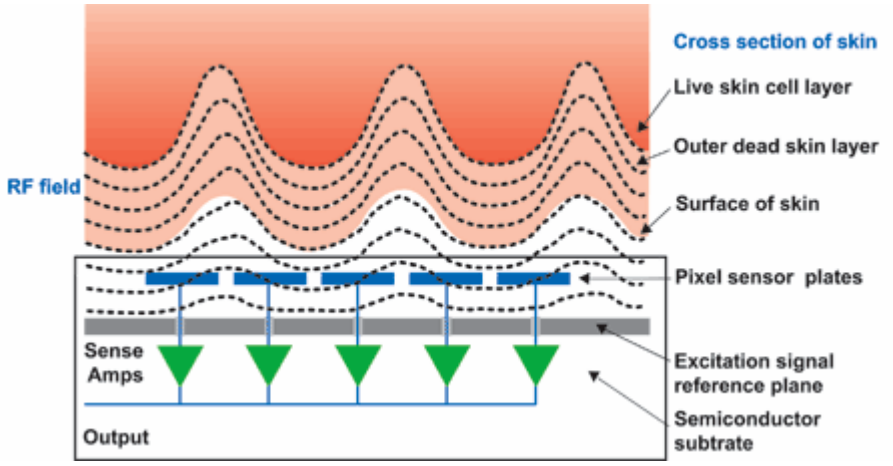

Figure 3: Cross-section of the fingertip skin in relation to the sensor, [3].

The sensor does not depend on the air gap between the sensor and the fingerprint valley in order to get a reading. Therefore, the different types of fingerprints, that are difficult to capture, can be obtained successfully using this technology. TruePrint based

affected by the common skin conditions, such as dry, worn, calloused, dirty or oily skin that would normally distort the fingerprint image.

Sensor selection can be independent from the software used for sample extraction and template comparison of the fingerprint. The Verifi P4000 sensor that was purchased is bundled with Omnipass software from Softex Inc. This software package uses the Bioscrypt Core Algorithm for minutiae matching and custom image processing software that was optimized to work with the sensor's outputs. This software package implements the basic functions of enrollment and matching, image acquisition, image reconstruction and sample matching.

3.2 Fingerprint Authentication Algorithm

3.2.1 Minutiae vs Graph approaches

Fingerprint identification and comparison in most automatic systems can be classified as correlation-based, ridge or feature-based techniques or minutiae-based, shown in *Figure 4*. Correlation-based and ridge or feature-based techniques are both graph-based approaches and are based on performing a correlation of the direct image or of ridge patterns extracted from the image. Minutiae-based methods attempt to detect and match local discontinuities in the fingerprints pattern, which represent terminations and bifurcations (branching) of the ridges. Graph-based methods are less sensitive to image quality and minutiae-based methods require less information storage and processing time.

Figure 4: Minutiae in the fingerprint image

3.2.2 Enrollment and Matching

The OmniPass is a Win2K/XP-based Graphic User Interface (GUI) with OS interconnections that invokes customizable fingerprint modules based on user actions. It manages access, login and encryption permissions. The key system functionality still remains Fingerprint Based User Authentication with user enrollment and matching as its core functionality. The initial step of the enrollment process consists of capturing and storing a reference version of the fingerprint of an identified user. OmniPass requires the administrator to create a single master password and it is used every time the application is accessed. The enrollment process for OmniPass is managed by the Enroll Wizard GUI. This wizard guides the user through the process of enrolling a fingerprint, a sample interaction screen is shown in *Figure 5*.

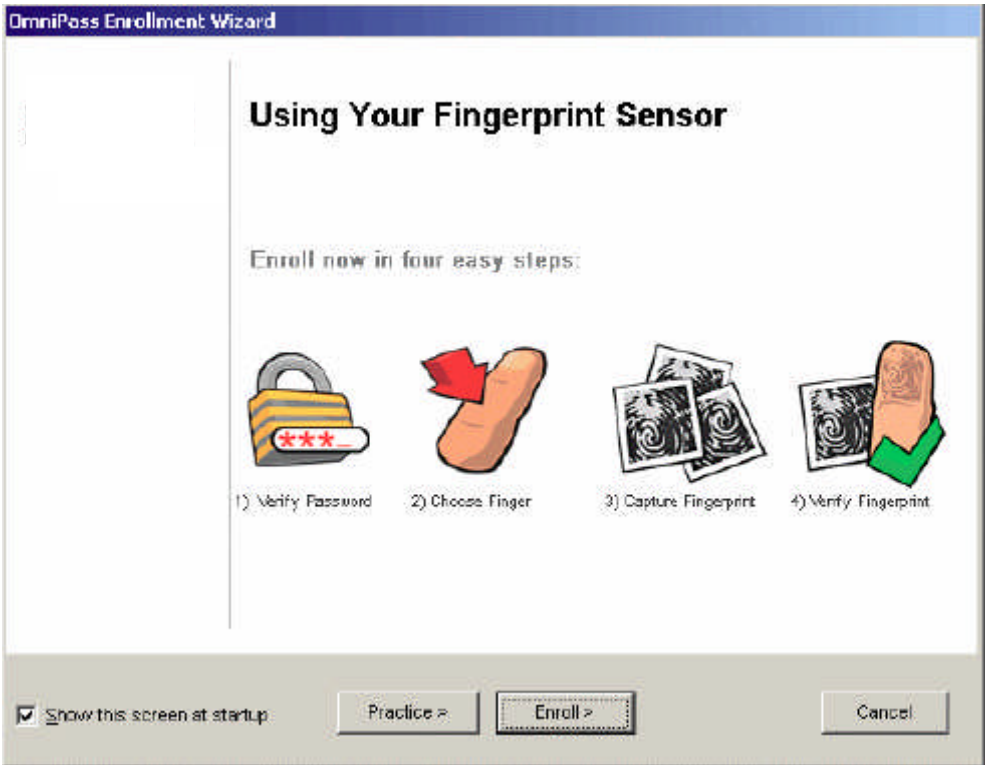

Figure 5: Enroll Wizard First Screen

The enroll button takes the administrator to the verify screen in which the initial password created must be entered correctly. The next step is to select the finger to enroll. This is done by clicking on the image of the finger on the screen that represents the finger you want to enroll but the system does not verify your choice. The next screen is the Capture Fingerprint and this is where the user is asked to place previously selected finger on the sensor and the image is captured into OmniPass. Once the image is enrolled it is processed, enhanced and then compressed creating a template. The template is returned with the image statistics, which reflect the quality of the enrolled image. The final step is the Verify Fingerprint dialog box. The finger is placed on the sensor again and the wizard will let you know if you were successful or not. In this final step a score is also returned indicating the similarity of the user's image and the template. This score is compare to a threshold to make a decision on whether or not the image and the template were derived from the same finger. The threshold is determined by the system administrator and specifies the numerical requirements for a successful match. If attempt was unsuccessful the user can repeat the entire process again until they are successful.

3.2.3 Operation and performance

In an ideal fingerprint biometric system, a person registers with the system, processed by an enrollment algorithm that never rejects any users, and then entered into a database. When authorized user attempts to log in, all fingerprint readings are exact matches and the system grants access every time. When unauthorized user tries to log in readings do not ever sufficiently match and the system denies access.

Real-world biometric fingerprint system cannot generate definitive answers. In the case of fingerprint authentication, information could change in terms of its shape or quality, so matching scores against stored templates could vary. As a result, even a valid person may be rejected or a wrong person may be accepted.

Performance of a biometric system is usually defined by standard terms. These include the false accept rate (FAR), the false reject rate (FRR), and the failure to enroll rate (FTE). FAR and FRR can be traded off against each other by changing operational parameters such as processing time spent on image filtering and the number of data points used for generating templates. FTE is usually inversely correlated with FRR, as a result more desirable lower scores usually seen on systems with higher FRR. One of the common measures of real-world biometric systems is the rate at the setting at which both accept and reject errors are equal (EER), shown in *Figure 6*. The lower the EER, the more accurate the system is considered to be.

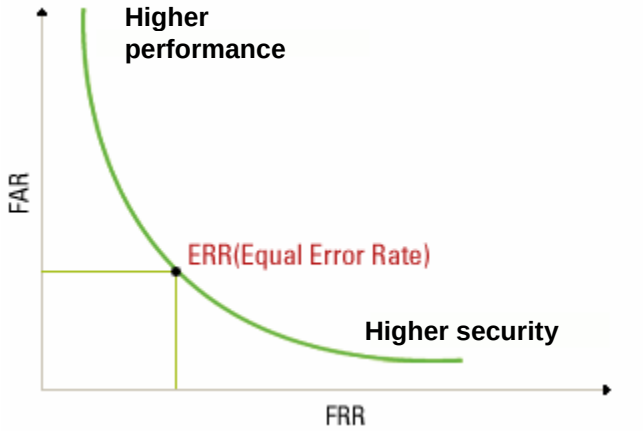

Figure 6: Calculating ERR

In the scope of this project security is prioritized and as a result the system is configured to minimize FAR resulting in higher FRR. The rationale of this decision is that potential consequences of granting access to an unauthorized user are by far greater than inconvenience of increased number of login attempts typical for a system with a higher FRR rating.

3.2.4 Image processing

A critical step in fingerprint matching is to automatically and reliably extract minutiae from the input fingerprint image. The performance of a minutiae extraction algorithm relies heavily on the quality of the input fingerprint images [11]. The ridge structures in fingerprint images are not always well defined and the use of an enhancement algorithm to improve the clarity of the ridge structures is required. Fingerprint image enhancement algorithm receives an input fingerprint image, applies a set of intermediate steps on the input image, and finally outputs the enhanced image.

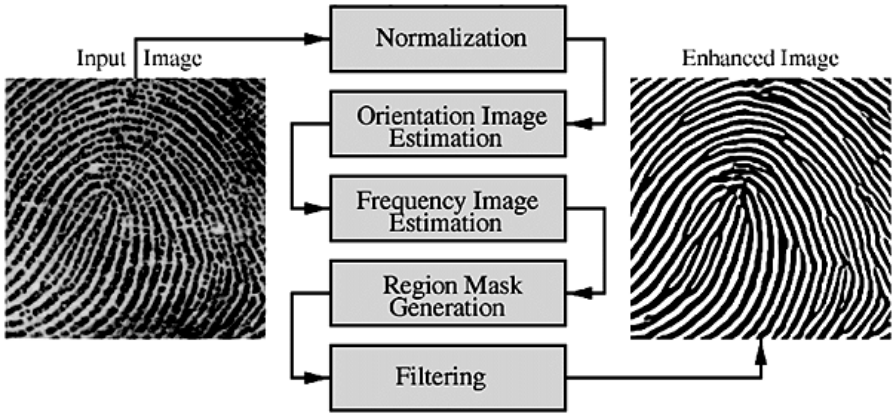

Figure 7: Fingerprint image enhancement algorithm

Typical steps for image enhancement are normalization, image orientation, frequency estimation, region mask generation and filtering, shown in *Figure 7*. Normalization refers to process of applying a color shift to an entire image to a pre-specified mean and variance. Normalization is an off-line process and it requires two

passes. The first pass determines the highest peak, mean and variance. The second pass applies changes to the entire image in order to bring mean and variance to predetermined range of values. Image orientation estimation is a process of creating an orientation image which contains the coordinates of the ridges and furrows in the fingerprint. Frequency image estimation examines each pixel and those in it local neighborhood. The rationale is that, in a local neighborhood in which no minutiae are present, the gray levels between the ridges and furrows can be modeled as a sinusoidal wave [2]. Based on the frequency image, a region mask is created where parts of the original input image are marked as recoverable or unrecoverable. Gabor filtering is performed on the recoverable parts to remove noise and highlight the ridges and furrows. Binarization is done on the pixels of the image to mark them as either black or white. Finally, the minutiae are extracted by examining each pixel and its immediate neighbors, [2].

3.3 Security Issues

One of the most significant disadvantages of the biometric recognition system is that they cannot be easily recalled [1]. If one of the fingerprints used to access the system is compromised, it cannot be reused since it is impossible to change, which means it is compromised forever. Additionally, since one person only has a limited number of fingers, different applications might use the same fingerprint. A person's fingerprint template stolen from one application could mean security compromise of multiple systems. As a result the secure storage of the biometric data is extremely important.

Traditionally fingerprint-based authentication system stores templates on a central server. The fingerprint captured by the sensor is then sent to the server and the processing and matching steps are performed on the server. In this case the safety of the data cannot be guaranteed during transmission or on the server.

Embedded biometric recognition systems solve this problem by performing signal processing and matching on the embedded device and output only the result. This approach can avoid the attacks on communication and server, but it is very easy to compromise the plaintext storage of the template in the embedded device. To secure templates storage all templates should be encrypted before being stored.

One possible way of encrypting biometric template data is that instead of storing plain-text sensor data on the embedded device, the system could store its noninvertible transformed version, for instance, a hash, in the enroll phase, [1]. During recognition, the input biometric information is first encrypted using the same non-invertible transform. Then matching is performed in the transformed space. Different applications can use different non-invertible transforms or different parameters of the same transform. Thus a template would be usable only by the application that created it.

Since security of authentication system is a high priority for this projects all template data is hashed using division reminder method, where you divide the key value by some fixed number and take the remainder as the key location. This encryption can be effectively performed without detrimentally affecting performance of entire system.

However, this method is inadequate against determined hackers and it is still possible to reconstruct original data.

4. Results

4.1 Fingerprint Based User Authentication System Overview

The project was broken into four phases, each focusing on separate and distinct element of FBAS. The recommended system was developed on the basis of these four sequential phases listed below. The development of this design is also outlined in our Project Development Timetable in Appendix A.

4.1.1 Phase 1: Sensor Identification

During this phase it was imperative to identify and recommend a fingerprint sensor in the early stages of the project, so that more time can be allotted to implementing the fingerprint processing hardware, software and all other necessary interfaces. This phase was completed with the recommendation and actual purchase of the sensor from Atmel, the FingerChip Biometric Module. The decision for the sensor was based on the results of an exhaustive investigation. According to our research SSSC concluded the advantages such as: image processing, embedded standalone system, algorithms and source code are provided, the system is immune to transmission and server identity theft, and many more outweigh the possible drawbacks. Due to budgetary constraints it was necessary to revisit this phase and select and purchase an alternative sensor with a lower overall implementation cost. Our initial reports indicate that the Verifi P4000 sensor was the next best replacement. The advantages of this sensor are interface is already implemented, USB is sufficient to power the sensor and easy to interchange components.

4.1.2 Phase 2: Hardware Implementation

Once the sensor selection phase was completed it was time to determine if it can be implemented on an embedded system. Due to the time remaining for project completion it was decided that it would be in the best interest of SSSC, to use this sensor with pre-package software from the manufacturer. Unfortunately the sensor interface was not meant for embedded applications, but is Windows based. The focus here was to ensure the sensor will function if the interface present. The sensor was tested for basic functionalities; for example, the requirements for a successful fingerprint scan and image capture. Initially, this phase was dedicated to determine the inefficiencies of the hardware and interface then recommended course of action. However, hardware optimization was replaced and now the focus of the project was to get the sensor operating and determine a method of interfacing with this sensor, if not provided. There were many changes made to accommodate the delay within the first phase. These changes can be seen in the Project Development Timetable in Appendix A.

4.1.3 Phase 3: Software Implementation

This phase was initially scheduled to occur once the hardware implementation was already underway, but it was rescheduled. The focus of this phase was to determine the functionality of the software algorithm. Fingerprint algorithms are complex and as a result require time to understand the basic principles. The algorithm was benchmarked using the standards set-out by the Fingerprint Verification Competition (FVC) 2004, [5]. Implementation of this algorithm was closely based on available code.

4.1.4 Phase 4: System Optimization

Due to setbacks in our project development, system optimization will simply be recommended.

These phases were in actuality a constant iteration, as is any design process. This can be seen in the Project Development Timetable, which is an explicit reflection of the progress of this project and the constant design optimizations that were required.

4.2 Fingerprint Sensor System

Figure 8 demonstrates functionality of the password automation feature. One of the convenience features of the software package allowing for direct replacement of passwords in existing systems with the biometric access.

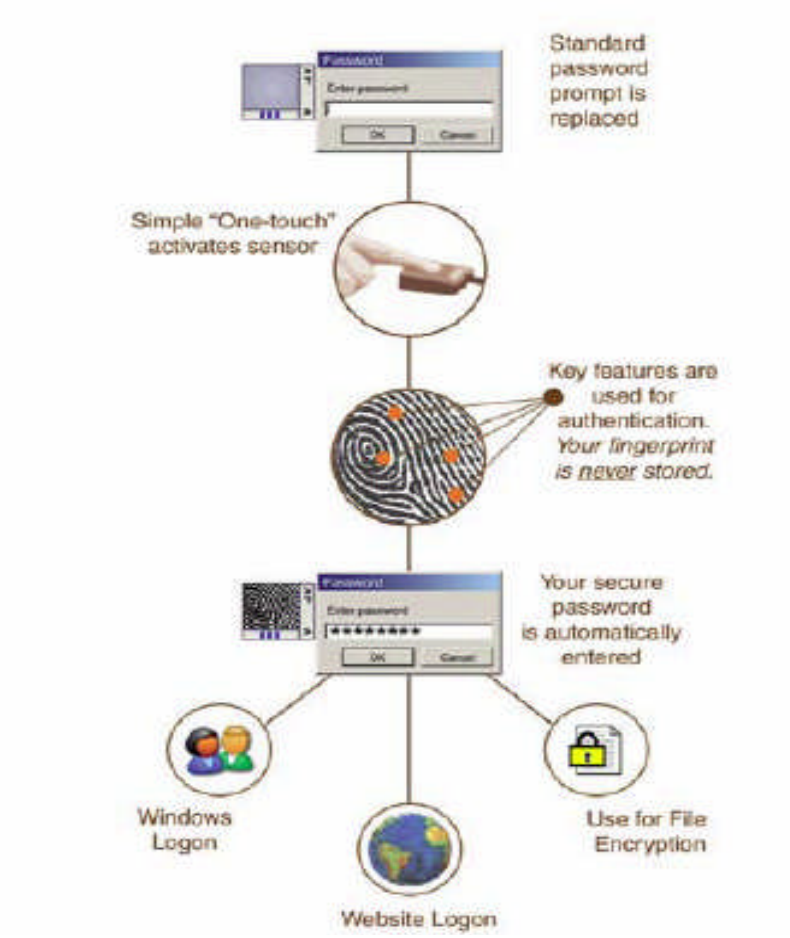

Figure 8: Overview of the fingerprint system [14]

Figure 9 is a system flow diagram and reflects two modes of the system – enroll and verify. Each mode shares scanning and minutiae extraction algorithms and differ only in template processing stages. Due to security issues enrollment is a separate mode and must be specified before fingerprint scanned and processed.

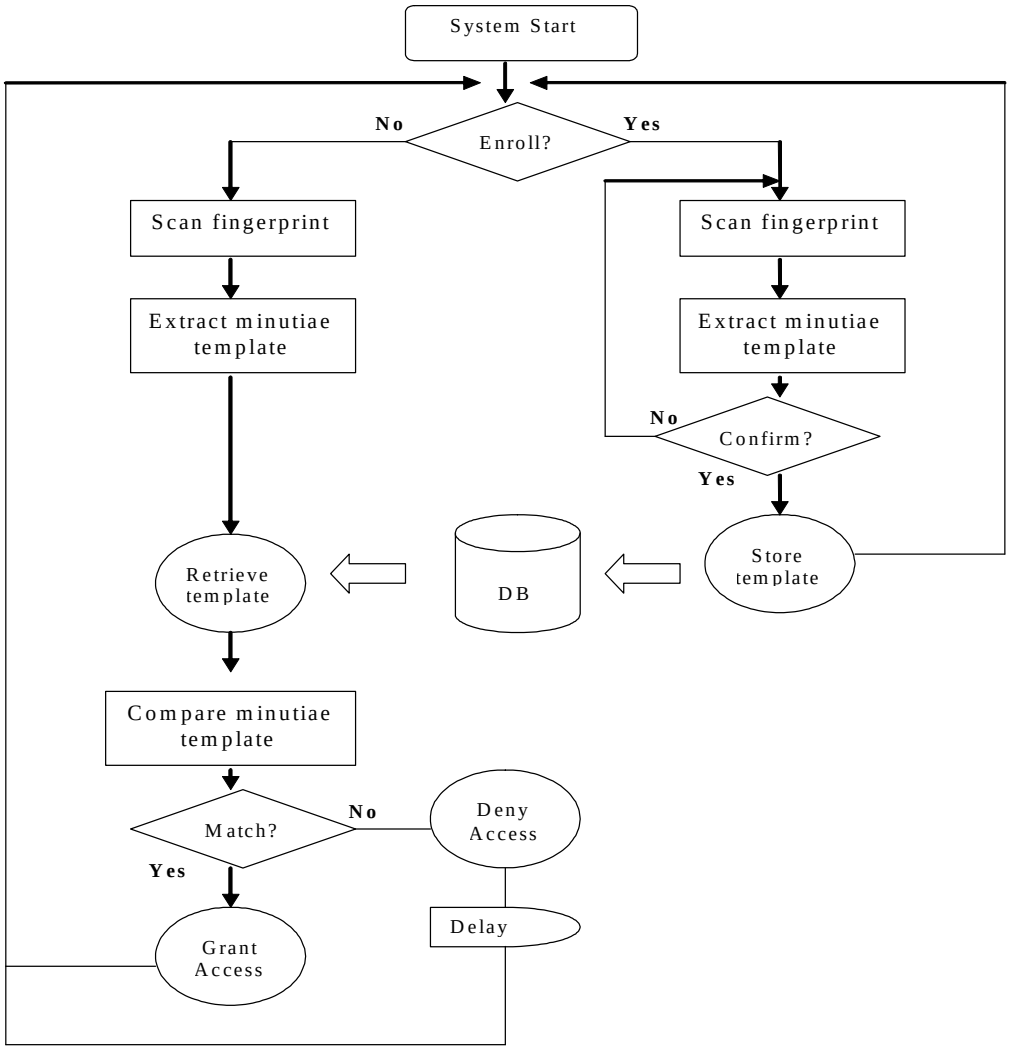

Figure 9: System Program Flowchart

4.3 Fingerprint Recognition Algorithm

4.3.1 Fingerprint Matching Algorithm

The recommended fingerprint comparison algorithm is Bioscrypt developed by Bioscrypt Inc. It is also available through Bioscrypt [6] corporate web site for evaluation and licensing.

Using FVC2004 [5] testing methodology and evaluation software we confirmed performance evaluation of this algorithm in controlled conditions. While decreased fingerprint quality will result in detrimentally affected system performance, we believe that the database used for testing is representative of fingerprint image readings acquired with Verifi P400 sensor.

> According to the standardized tests Bioscrypt algorithm had following scores:
>
> Number of rejected fingerprints during enrollment – 0%
> Number of rejected fingerprints during genuine matches – 0%
> Average enrollment time - 0.08 sec
> Average matching time - 0.16 sec
> Maximum template size - 1.2Kb
> Maximum amount of memory allocated - 3044Kb

With perfect values for rejected enrollments and genuine matches, impressive enrollment and matching times and reasonable memory usage makes this the leading algorithm choice. While the final system will not show perfect EER values due to potential problems involved in physical aspect of fingerprint scanning, the comparison algorithm will not be limiting factor in this regard.

4.4 Fingerprint System Hardware

4.4.1 Sensor Specifications

The final system implementation uses a standalone Verifi P4000 fingerprint reader, shown in *Figure 10*. This sensor is based on Authentec AES 4000 EntrePad sensor which uses TruePrint RF technology to acquire 250dpi image making it possible to achieve top EER scores. The sensor is designed to work with a server (i.e. PC) connected via USB port. It comes with all necessary authentication software. The Software Development Kit (SDK), VeriFinger, is also available from the manufacturer [13]. Verifi P4000 sensor weights 2.5 ounces and is 2.5” X 2.0” X 0.75”

Figure 10: Verifi P400 finger print reader

If there are any instabilities with the connection between the USB and the PC, it is automatically detected and corrected by the error handling program provided by Verifi. This program also protects against Electro-Static Discharge damage and safety management, with respect to the USB connection.

5. Discussion

5.1 Design Performance

Overall system performs adequately. It simplifies secure access and password management and allows access for multiple users to various system components protected by complex passwords. There is no noticeable delay during fingerprint processing time and software does not generate false positive matches. We discovered only two problems during our testing of the system. First problem is that incorrect positioning of the finger on the pad will generate false rejection, this problem becomes noticeable only during scanning of the very large fingerprints, i.e. thumb of a large person. Second problem is related to software, the sensor's USB drivers conflict with WinXP power management resulting in an unresponsive sensor after system goes into hibernation mode. This issue is platform specific and can be resolved by system reboot or disabling of a hibernation feature.

5.1.1 Sensor Performance

The Verifi sensor seemed to experience difficulties when the user does not apply enough pressure to the sensor. This is unexpected behavior since the sensor uses RF to scan a fingerprint. Another difficulty with this sensor is the size. Due to its small size, the entire fingerprint of a large finger cannot be captured. Therefore, it may take a number of tries to correctly position the finger in order for the sensor to capture the accurate portion a fingerprint. This might be a problem for users who have larger or rounder fingers. Again, this is not a significant issue and can be resolved with some training.

5.1.2 Software Testing

In the scope of the project following software packages were tested – Bioscrypt SDK, Sourceforge, Verisoft Access Manager, Authentec SDK. We had access only to sourceforge's framework for fingerprint authentication code, rest are compiled products or standalone modules that are available free of charge for evaluation but only as a compiled programs. It is worth noting that Sourceforge is not a complete and working product rather set of modules, some are written for BSD or Linux.
In order to test modules following steps were performed – download the module, adapt module's interface to comply FVC testing application, compile if necessary and run the module, compare and not testing results.

5.2 Cost

The cost of the major components used in implementation is $130.00. We decided to use the Verifi P4000 sensor instead of going with the originally purchased Atmel FingerChip Biometric Module. Therefore, the total cost of the system implemented is $130.00.

Total cost of implemented system:

Items	**Estimated Cost**
Verifi P4000 Sensor	$130.00
Total	**$130.00**

Total cost of original system with development kit:

Items	**Estimated Cost**
FingerChip Evaluation and Development Kit	$760.00
FingerChip Biometric Module	$150.00
Total	**$910.00**

The total amount of money spent for this project is approximately $280.00 which is the cost of the Verifi P4000 sensor and FingerChip Biometric Module.

5.3 Unexpected Problems and Resolutions

5.3.1 System Interface

After we purchased the Atmel FingerChip Biometric Module we discovered that there was no easy way to interface to the module. To be more specific we discovered the manufacturer introduced additional security safeguards making it very problematic to interface with the sensor without access to specific drivers that are only available as a part of the development kit. We contacted the support team members from Atmel to see if they can provide an alternative method of interfacing without further purchase, or even supply the documentation that is included in the evaluation and development kit. Atmel stated we had to purchase the evaluation and development kit and regrettably the support team could not provide any assistance other than what was provided on the website. Regrettably purchasing the evaluation and development kit was no longer an option and we decided to explore our alterative design.

5.3.2 System Cost

One of the major problems encountered during the design phase of this project was funding. We purchased the FingerChip Biometric module that was approximately $150.00. In order to interface with this module we required the purchase of a development kit, which cost approximately $760.00. The necessity to purchase development kit pushed our Design Project over the budget. We believed that if we could secure funding from at least three sources that it would be possible to purchase the development kit. Since we have already purchased the module at that time we decided it would be best to seek additional funding.

Our first source was University of Guelph Undergraduate Equipment Fund. From which we believed we would receive full funding, but a few days later the committee retracted their offer. SSSC submitted a PRD (Petitions, Delegations, and Representations) form to request funding from the College of Physical and Engineering Science Student Council. We were approved for an amount of $158.00. However, funding was never released to us - the College of Physical and Engineering Science did not receive their funding until late November.

While we were making every effort to seek additional funding we kept running into the same problems; we would be approved and at the last minute the approval was retracted. Even with an alternate embedded sensor we would still need to purchase a development kit and the cost of such kit would exceed available funding. The alternative to work without development kit appeared to be difficult due to other limiting constraints such as time and size of this design team. Most of the time would be dedicated to developing this evaluation and development kit rather than fulfilling the objectives of this project.

In the end we decided to purchase the Verifi P4000 fingerprint reader for $119.00 port from available funding. This sensor is designed to work with a personal computer connected through USB. With this method the fingerprint is captured by a front-end sensor and then sent to the server for processing, matching and storage. We wanted to

avoid standalone implementation due to its vulnerability to identity theft during data transmission.

6. Conclusion and Recommendations

The objective of this project was to research a method of implementing a fingerprint based authentication system (FBAS) on an embedded computing device. This is important because a biometric method can ensure the security of information on embedded devices.

An embedded biometric system that incorporates the signal processing and matching from the server to the embedded device is more secure than standalone sensor device. We recommend continuing work on Atmel's FingerChip fingerprint module. This module uses FingerChip sensor directly interfaced to ARM9-based microprocessor via 8-bit bus.

While there might be a way to directly interface Atmel's module we recommend securing funds from 41X fund, IEEE or LabFund in order to purchase $800 evaluation kit. This kit already has an implemented interface and comes with source codes for identification, image processing and matching algorithms. After repeated unsuccessful attempts to interface with this module via USB, we came to conclusion that the module's security features prevent access or modification of any of its code via direct interface and found no way to unlock it without development kit. We had very limited success with FC_OpenDevice(),FC_IsFinger() and FC_GetSlice(). In our attempts to access FingerChip module we used software and driver definitions found in the Software Development Kit [15] along with modified USB schematics [16] connected to a PC.

7. References

[1] S. Yang and I. M. Verbauwhede, "A secure fingerprint matching technique", in Proc. Wkshp. Biometrics Applications & Methods, Nov. 2003, pp. 89–94.

[2] P.Gupta, S.Ravi, A. Raghunathan, N. Jha, "Efficient Fingerprint Based User Authentication for Embedded Systems", ACM Press, 2005, pp. 244 - 247

[3] "Authentec website, Technology", http://www.authentec.com,October 2005

[4] S. Yang, K. Sakiyama, I. M. Verbauwhede, "A Compact and Efficient Fingerprint Verification System for Secure Embedded Devices", http://www.emsec.ee.ucla.edu/pdf/2003asilomar_yang.pdf, October 2005

[5] "FVC2004", http://bias.csr.unibo.it/fvc2004/perfeval.asp, October 2005

[6] "Bioscrypt Core", http://www.bioscrypt.com/products/bioscrypt_core.shtml, October 2005

[7] "D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition. New York, NY: Springer, 2003.

[8] "Sourceforge website." http://ffpis.sourceforge.net , October 2005

[9] "Softex Inc", http://www.softexinc.com, November 2005

[10] "Overview of Biometric System", http://www.idteck.com/technology/biometrics.jsp, November 2005

[11] L. Hong, Y. Wan, and A. Jain, "Fingerprint Image Enhancement: Algorithm and Performance evaluation," *IEEE Trans Pattern Analysis and Machine Intelligence*, vol. 20, no.8, pp.777-789, 1998

[12] S. Greenberg, M. Aladjem, D. Kogan and I. Dimitrov , "Fingerprint Image Enhancement using Filtering Techniques", *Electrical and Computer Engineering Department*, Ben-Gurion University of the Negev, Beer-Sheva, Israel

[13] "VeriFinger 4.2 SDK", Neurotechnologija Ltd., http://www.takumivision.jp/pdf/VF_42_SDK.pdf, December 2005

[14] “Zvetico Biometrics”, http://www.zvetcobiometrics.com/pdf/OneTouch.pdf, December 2005

[15] “FingerChip Software Development Kit”, http://www.atmel.com/dyn/resources/prod_documents/doc2179.pdf, December 2005

[16]“FingerChip Demonstrator”, http://www.atmel.com/dyn/resources/prod_documents/doc2176.pdf, December 2005

Appendix A: Project Development Timetable

The current Project Development Timetable reflects the amendments made during the course of the project due to allotted dates for deliverables and unforeseen delays or problems encountered.

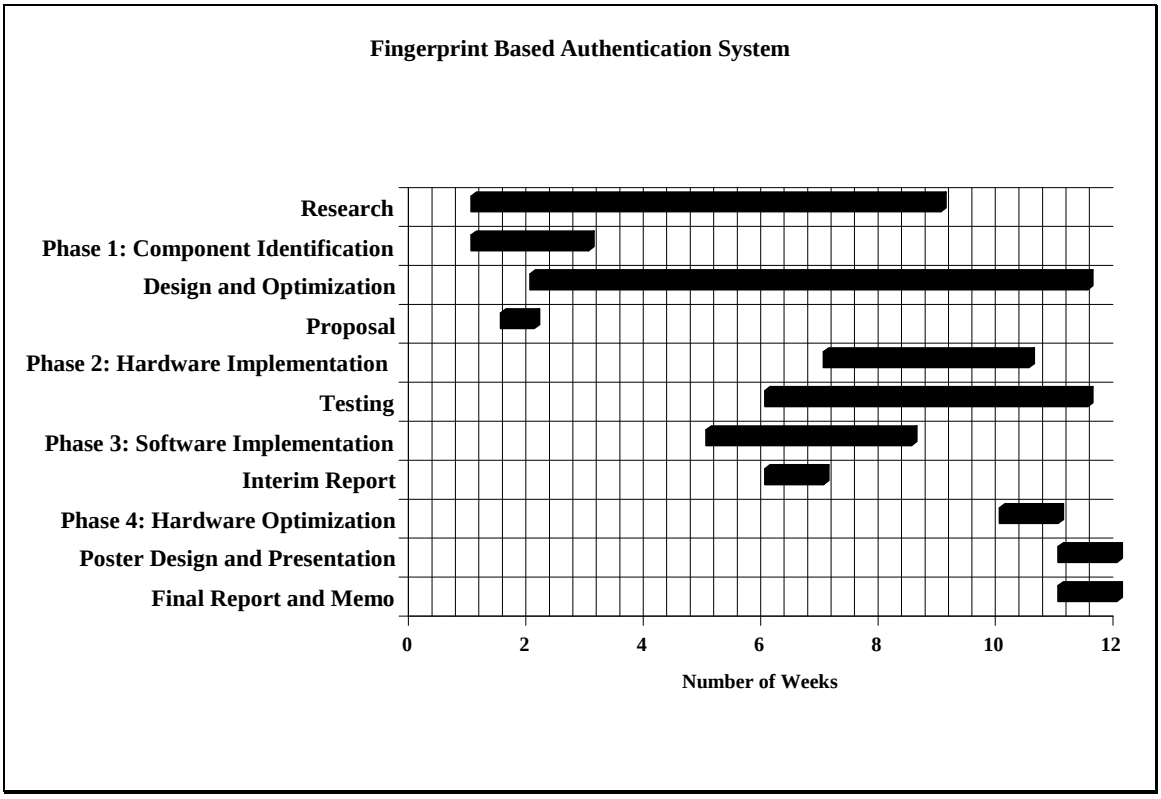

Figure 11: Project Development Timetable