

**IMPLEMENTATION OF DTW ALGORITHM FOR APPLICATION  
SECURITY**

---

A Thesis

Presented to the

Faculty of

San Diego State University

---

In Partial Fulfillment

of the Requirements for the Degree

Master of Science

in

Computer Science

---

by

Preetam Borah

Fall 2012

**SAN DIEGO STATE UNIVERSITY**

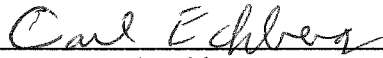
The Undersigned Faculty Committee Approves the

Thesis of Preetam Borah:

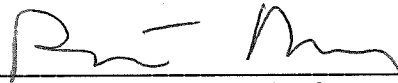
Implementation of DTW Algorithm for Application Security



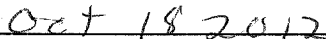
Joseph Lewis, Chair  
Department of Computer Science



Carl Eckberg  
Department of Computer Science



Robert Malouf  
Department of Linguistics and Asian/Middle Eastern Languages



Approval Date

Copyright © 2012

by

Preetam Borah

All Rights Reserved

## **DEDICATION**

I want to dedicate this Thesis to my Guru, Thesis Advisor Dr. Joseph Lewis for his constant support , advice and guidance for the successful completion of this project. A great professor and more than that a very humble and noble person who always puts priority for students first. I also want to thank my parents Mr. and Mrs. D.K. Borah whose enduring love, kindness and support have made me for what I am today. I would also like to sincerely thank my thesis committee members Dr. Carl Eckberg and Dr. Robert Malouf for their advice and guidance during this project. I would also like to thank San Diego State University. Ever since I joined SDSU I have been in the constant learning and growing path for both my professional and personal life and helping me to be a better person every day. Last but not the least I would like to thank “The Almighty” who has given me the courage and strength to pursue a second Master’s degree from SDSU.

## **ABSTRACT OF THE THESIS**

Implementation of DTW Algorithm for Application Security

by

Preetam Borah

Master of Science in Computer Science

San Diego State University, 2012

Signatures have been used to authenticate documents from a long time. It is a behavioral metric which is not based on the physical properties of a person such as the fingerprint or face. Authenticating an identity is very crucial today. With a lot of data used an individual's personal data can be easily compromised. With the advent of technology the number of smartphone users is increasing rapidly. The capability of these smartphones is almost limitless. A user stores a huge amount of personal data on his smartphone which includes his personal email accounts, social networking accounts, bank accounts etc. This data is vulnerable if the smartphone is not provided with application or a lock that provides access only to the user. It is very necessary to have some kind of application so that no one else except the user himself can access his personal accounts and data.

This thesis involves designing a mobile application that provides security to the user's personal accounts. A user will initially train the application by putting a set of signatures. For designing this application we will be using the Dynamic Time Warping algorithm. Once the application is trained one signature will be used as a reference signature. When the user tries to log in by signing a signature, this input signature will be compared to the reference signature and if it is within a certain threshold of the reference signature the user will be able to log into the application and access his personal accounts.

## TABLE OF CONTENTS

	PAGE
ABSTRACT.....	v
LIST OF TABLES .....	vii
LIST OF FIGURES .....	viii
CHAPTER	
1 INTRODUCTION .....	1
1.1 Introduction to Signature Based Verification .....	1
1.2 Advantages of Signature Based Verification Over Other Biometric Techniques .....	2
1.3 Applications of Signature Based Verification .....	2
2 BACKGROUND AND LITERATURE REVIEW .....	5
3 EXPERIMENTAL PROCEDURES .....	20
3.1 DTW Algorithm.....	20
3.2 The Framework Used for Designing the Application.....	22
3.3 Our Approach in Designing the Application .....	23
3.3.1 Training the Signature.....	24
3.3.2 Normalization .....	26
3.3.3 Authentication.....	26
3.4 UI Design.....	26
3.4.1 Signature Training Screen.....	26
3.4.2. Backup Password.....	27
3.4.3 Authentication Screen.....	28
3.4.4. Accounts Screen and Settings.....	28
4 RESULTS AND DISCUSSION .....	29
5 CONCLUSIONS.....	38
6 FUTURE WORK.....	39
REFERENCES .....	40

## LIST OF TABLES

	PAGE
Table 4.1. Table Showing the Acceptance Rate of the Two Signatures with a Threshold Factor of 0.5 .....	31
Table 4.2. Table Showing the Acceptance Rate of the Two Signatures with a Threshold Factor of 1.5 .....	32
Table 4.3. Table Showing the Acceptance Rate of the Two Signatures with a Threshold Factor of 1.0 .....	33
Table 4.4. Acceptance and Rejection Rate using Skilled Forgery by Five Different Users with Varying Threshold Factor .....	35
Table 4.5. Acceptance and Rejection Rate Using Simple Forgery by Five Different Users with Varying Threshold Factor .....	36

## LIST OF FIGURES

	PAGE
Figure 2.1. Biometric authentication system..	5
Figure 2.2. Two nose prints of the same cattle taken in a period of six months.....	10
Figure 2.3. Conceptual structure of SIDS.....	14
Figure 2.4. The voice recognition architecture proposed by the authors.....	15
Figure 3.1. Two series A and B arranged in a grid. ....	21
Figure 3.2. Architecture of the cocoa framework for iOS. ....	23
Figure 3.3. Steps involved in the design of the application. ....	24
Figure 3.4. Flowchart showing the design of the UI.....	27
Figure 4.1. Signature training screen. ....	30
Figure 4.2. Authentication screen of the application. ....	30
Figure 4.3. A signature on the screen labeled as “Signature 2”.....	31
Figure 4.4. Acceptance and rejection rate versus threshold factor for signature 1.....	33
Figure 4.5. Acceptance and rejection rate versus threshold factor for signature 2.....	34
Figure 4.6. Plot of the FAR for five different users with varying threshold for the scenario skilled forgery.....	36
Figure 4.7. Plot of the FAR for five different users with varying threshold for the scenario simple forgery.....	37



## **CHAPTER 1**

### **INTRODUCTION**

Since early ages signatures have been used to authenticate documents. The Sumerians used seals on clay tablets to authenticate their documents [1]. The Romans in 439 AD were the first to practice the authentication of documents by affixing hand written signatures. At the end of the document a short handwritten sentence called the “subscripto” was used for authenticating wills which stated that the signer has subscribed to the document. In 1677 England passed a law called “An Act for Prevention of Frauds and Prejuries” which required documents to be signed by the participating parties. In 1977 three Mathematicians from Massachusetts Institute of Technology generated a pair of numerical keys by developing an algorithm which was able to secure data transmissions electronically [2].

In today’s world authentication of identity and securing data is very crucial. Various techniques can be used to secure data or authenticate a document/identity. One of the techniques used is Biometric. A few well-known biometric methods are fingerprint, nose print, retina and facial based identification and verification [3]. These methods require special and expensive hardware to capture the image. Another important but rather less expensive biometric method used for verification is a “signature”. As stated above hand written signatures are the most commonly used technique to verify identity or documents. Earlier signature was verified by visual inspection.

#### **1.1 INTRODUCTION TO SIGNATURE BASED VERIFICATION**

Digital signature capture is used in a lot of applications nowadays for verification. A signature which pertains to an individual is captured and treated like an image containing a pattern of pixels which can be used for verification. However no two signatures of a person are precisely the same. The important factor is to differentiate between the parts of the signature and those that vary with almost every signature. There are two types of features that validates the signature. They are Static and Dynamic. Static features are extracted that are recorded as an image whereas dynamic features are extracted from signatures that are signed

in real time [4]. Signature verification is a common Biometric technique to identify human beings for purposes of verifying their identity. Basically we can classify the Signature authentication system into two types:

- Off-Line: The signature is scanned to get its digital image representation eg. Pen and paper.
- On-Line : Uses special hardware such as a digitizing tablet or a pressure sensitive pen to record the pen movements during signing.

## **1.2 ADVANTAGES OF SIGNATURE BASED VERIFICATION OVER OTHER BIOMETRIC TECHNIQUES**

As mentioned there are several other biometric techniques available for verification such as fingerprint, voice, retina etc. But these techniques have some disadvantages too. For example in case of fingerprint verification authentication may fail for a person working in a factory. Similarly a person who has retinal disorders is subjected to retinal change for which retina verification might not be a good solution for any authentication. Moreover a person's voice gets changed over time or due to flu the person's voice varies and so voice recognition may not be suitable for all situations. Last but not the least these biometric techniques are very expensive to implement which requires special hardware.

On the other hand signature verification is a socially accepted verification technique which is used in a variety of systems. It is relatively inexpensive compared to the other biometric techniques mentioned above and thirdly a signature can be changed by the user whenever he wants to unlike retina and voice pattern recognition which cannot be changed.

## **1.3 APPLICATIONS OF SIGNATURE BASED VERIFICATION**

Signature based verification is being widely used. It is used in forensic applications, banks for transactions, transactions in grocery store while using credit cards, signing tablets during receiving of shipments and more.

Another prominent technology today one can think of putting a signature based verification technique is a smartphone. The number of smartphone users is exponentially increasing. Earlier cell phones were used only for calling, messaging and playing games. But with the advent of technology the capabilities of a smart phone is limitless. A smart phone today none the less than a mini computer that fits into a pocket. A user can have the flexibility to choose smartphones with different Operating systems like the Android, iPhone,

RIM and the Windows. A user not only makes call and send messages via a smartphone but also uses it to do bank transactions, checking emails, getting traffic information via GPS and many more. In this 21<sup>st</sup> century we are surrounded by data everywhere and securing the data is extremely important to prevent forgery or malicious activity. A huge number of downloadable applications are available for the smart phones today. For example the Apple Store has more than half a million applications that can be downloaded on the phone. A user can use social networking applications, bank applications and personal email accounts in his smartphone. With a lot of personal data stored in the smartphone, this delicate data becomes vulnerable if a phone is lost or is used by a malicious person. For security reasons there are applications that provides a more secure environment to access or use the phone. For example a user can lock his phone screen by setting up his own password or a screen pattern before he or anyone can access to the phone's home screen. However one big disadvantage to it is that a user has to remember the password which could be a 5 digit complex one or he has to remember a screen pattern he has set for the screen lock. An online tech savvy user nowadays have social networking accounts, several email accounts, bank accounts and many more. With so many accounts to log in the user has to remember the password for each and every account and adding to these account will be the password for the smartphone if the user has so. Although locking the phone with a password or a screen pattern seems secured but most of the users keep an easy password to remember which is easy to crack by anyone like "12345" or repeating a digit between 0 to 9 four to five times. Even passwords are easy to guess. Another disadvantage with using passwords and screen lock is that it restricts access to the phone dialer. In case of an emergency a user may not be able to call quickly while trying to unlock his phone. During such a scenario it is extremely important that a user is able to dial an emergency number without the hassle of accessing the phone after entering the password or the screen pattern lock. This is where we can think of an application where a user need not remember a tough password or a complicated screen pattern to lock his phone. An application that uses a signature verification seems most suitable to secure a user's personal email, bank or social networking accounts and at the same time providing access to any user to dial a number in case of an emergency.

At this point we propose of designing a secure application that allows a user to put a hand written signature on an Iphone. Once the user logs in by signing his own signature on

the screen he can have access to the other web accounts he has put such as Gmail, Facebook or other bank accounts. In this way he secures the web accounts only after his signing in through the application is successful.

The algorithm we are proposing to design the signature verification application is the Dynamic Time Warping (DTW) algorithm. It is a popular algorithm which was first introduced in the 1960's [5] and have been used for speech recognition pattern [6]. Other algorithm like the Fast Fourier transform (FFT) have been used for pattern recognition like nose prints and facial recognition. This FFT algorithm is computing intensive and takes into account a lot of data. Since we are designing an application for a smartphone we need to take into account the amount of power consumed by the application while using it. Battery in smartphones today drains out quite quickly and the implementation of FFT does not seem to be the best fit. So we are designing the application using the DTW algorithm.

The structure of this Thesis is provided as follows:

Chapter 2 – Background and Literature Review.

Chapter3 – Experimental Procedures.

Chapter 4 - Results and Discussions.

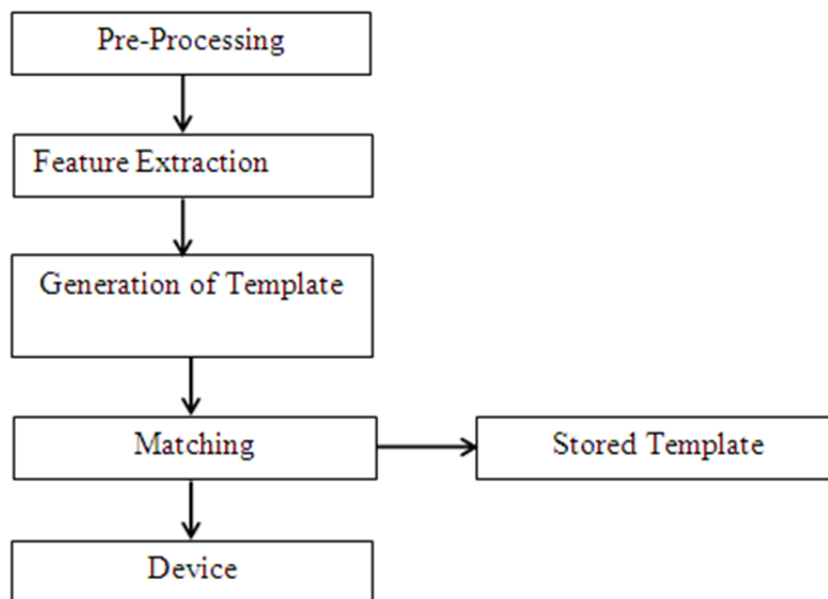
Chapter 5 - Conclusions.

Chapter 6 - Future Work.

## CHAPTER 2

### BACKGROUND AND LITERATURE REVIEW

Data security plays a very crucial role in every aspect of human life. All systems are being developed nowadays with security as the highest priority. When it comes to security of data biometric authentication for any system has been very popular and has been constantly evolving [3, 4]. Biometric authentication systems for pattern matching are easy to build, implement, secure and easy to use. There are numerous biometric authentication attributes like the hand print, finger print, nose print, retina verification. Although these attributes have been implemented in a number of systems one major drawback is that these authentication systems require special hardware and so they are expensive to implement. Another important biometric authentication attribute is the signature verification of an individual. Unlike other biometric authentication systems, signature verification is cheap. A simple biometric authentication system is shown in Figure 2.1 [7].



**Figure 2.1. Biometric authentication system. Source: T. Venkatesh, S.Balaji, and A. S. N. Charavarthy. Security evaluation of online verification system using webcams. *Inter. J. Comp. App.*, 41(15):28-33, 2012.**

Signature verification authentication can be static or dynamic. The two authentication techniques are discussed in brief below:

For static (off-line) verifications a signature of a user is written on a piece of paper and then scanned or written to a computer /digital pad and then compared with a reference signature. Quite a significant amount of work has been done for developing offline signature verification and recognition systems. In one of the papers Ramesh and Murty implemented four different types of pattern representation methods like, geometric features, enveloped characteristics, moment-based representations and tree-structured wavelet features [8]. The authors stated that the conclusions of the four methods are combined to validate the signature. The authors studied the combination of the two classifiers namely threshold-based and neighborhood. Finally the author concluded that feature-based classifiers increase the verification accuracy. In another paper authors have developed a system of two separate phases for signature verification and recognition [9]. In another research paper the authors developed an off-line signature verification system based on fusion of two machine experts of which one of them is based on global image analysis and a statistical distance measure and the second one on local image analysis and Hidden Markov Models [10]. Experimentally the authors found out that the local systems outperforms the global ones for signature verification but the best performance is obtained when fusing all the systems together.

For Dynamic verification the signature of the user is taken in real time. It uses a digitalized signature of a person which is acquired in real time. On-line signature verification is not so easy to forge. The online signature verification method extracts the features of a signature which is used to characterize the signature. There are mainly three different methods for selecting the features of a signature. The first one is based on features like velocity, pressure, acceleration etc. The second one deals with the global features like the time taken to write the signature, breakpoints while signing the signature, pen up time and direction of pen including the starting and end points. The third one takes the shape of the signature into consideration. Local features are used in signature verification and one of the most commonly used method is the Dynamic Time Warping algorithm [11]. Hidden Markov Models (HMM) approach have also been used for signature verification technique. In one of the papers the authors using the HMM approach computed a number of features for each segment which are scale and displacement invariant [12].

Fingerprint matching is one of the most important and promising methods among all the biometric pattern recognition techniques. It is a very essential part of intrusion detection systems. A lot of research has been done on fingerprint matching in improving the technique [13, 14]. Authentication of fingerprint matching is still a challenging problem because of the distortions between two impressions of the same finger. A decent number of novel approaches has been proposed by different authors to resolve this problem. In one of the papers the authors proposed a new fingerprint matching approach based on genetic algorithms. The algorithm tries to find the optimal transformation between two different fingerprints [14]. The authors designed a fitness function based on the local properties of each triplet of minutiae which included angles, triangle handedness, triangle detection, maximum side, minutiae density and ridges counts. Genetic algorithms have been used to recognize 2D or 3D objects from 2D intensity images. The authors stated that the recognition strategy using genetic algorithms is based on the theory of algebraic functions of views. The experimental results of the proposed algorithm were better than another approach based on mean-squared error estimation. The proposed algorithm achieved a very good performance even for a large portion of fingerprints in the database are of poor quality. To deal with fingerprint distortion problems a novel approach was proposed by authors that corrects the fingerprints which has already been acquired [13]. The proposed approach by the authors is completely automatic and unsupervised. The authors used two different models to handle the fingerprint distortion. The first one is a 1D model that appears to work well on the badly distorted prints while on the other hand the 2-D model could not handle the distorted prints well. The authors concluded that the 2-D model works well for lightly distorted prints. In another paper the authors studied the relationship between authentication reliability and region size. The authors applied a bank of Gabor filters orientated to different angles to clean the image from noises that could result in authentication mistakes. A function that reduced the image to a specific size and increments one by one was used to investigate which type of orientation the fingerprint has. This helped the authors to determine the core from which the vectors were traced to the minutiae for the purpose of image alignment and fingerprint matching. The proposed approach by the authors was much better than a purely minutiae based matching scheme [15].

Another biometric authentication technology is the facial-scan of a person. The technology is based on the natural means of biometric identification since human beings have the ability to distinguish between faces [16]. The hardware used includes cameras, workstations and back end processors. The technology involves using a digital video camera image which analyses the facial characteristics of a person such as measuring the distance between the mouth, eyes and the nose. These measurements are stored in a database and they are used to compare it with person standing in front of the camera. Facial recognition systems are primarily classified into two groups. The first one is called as the “controlled scene” where the variation of the scene is minimum i.e. the person to be verified for facial scan is located in a known environment. For example the person needs to stand in front of the camera within a specified distance. The second one is called the “random scene” where the person to be tested might appear anywhere in the camera scene. In one of the papers the authors did an extensive literature review on two types of face recognition techniques [17]. One of them captures from still images and the other from video. The authors identified two issues in face recognition namely illumination and pose variation. Illumination problem occurs when the same face appears different due to variation in lighting. Change in the lighting causes the system to give an error while comparing the image on the stored database. Various approaches have been proposed by researchers to solve this illumination problem. Among them are the Heuristic approach to compensate the light changes, the image comparison approach based on the image comparison using different image representations and distance measure and the model based approaches where a 3-D face model is used to synthesize the virtual image from a given image under desired illumination conditions. For a significant pose variation the performance of face recognition system drops. The authors classified the approaches for solving the pose based problem into three categories. First one is the single image based approach which includes low-level feature based methods, invariant-feature based methods and 3-D model based methods. No serious implementation for this approach has been made due to the complexity and computational cost. The other two approaches are the hybrid approach and the multiview based approach. In the multiview based approach the pose estimation and face recognition is coupled on an iterative loop. Among the three approaches the most successful one is the hybrid approach. This type of approach makes use of prior class information. Wiskott et al. proposed a robust recognition scheme based on



EBGM approach. This method is fully automatic, including face localization, landmark detection and flexible graph matching. In another paper the authors proposed a method for face recognition by elastic bunch graph matching [18]. The authors proposed this new idea for recognizing human faces from single images out of a large database containing one image per person. The authors compared their approach to other systems. The authors made several modifications to the systems which they used for face recognition. Wavelet phase information was used for accurate node localization which was previously not used and was imprecise. The improvements made by the authors in the new design made it possible to extract an image graph from a new face image in one matching process. The image graph reliably refers to the fiducial points even if the person of the new image is not included in the FBG. This accelerates the recognition from a large database as stated by the authors.

Another sophisticated and prominent biometric authentication technique is the retina scan which uses the retina of the person under scanner. This technology is based on the blood vessel pattern in the retina of the eye. The blood vessels of each person provide a unique pattern which is used for pattern recognition or identification. Infrared energy is absorbed faster by the blood vessels and it is used to illuminate the retina of the eye and then analysis of the retinal blood vessel image takes place to find the pattern [16]. Although this technology was developed in 1980's it is the least deployed technology for biometric authentication. It is usually deployed in high security organizations. In one of the papers the authors proposed a new way of person identification based on the criteria of retina matching. The authors stated that the process consists of retina image acquisition, image process, feature extraction and finally matching the patterns. The algorithm developed by the authors is based on color centroid calculation and its variation in polar grid [19]. The correlation coefficient was used to quantify the degree of matching. The authors found out that although the number of images matched is not very high, the results are robust within the tolerable limit and the method is insensitive to translational and rotational displacement. In another research the authors proposed a new retina identification based on the combination of Fourier and wavelet transform [20]. In this approach the authors stated that at first, optical disc is localized using template matching technique and then use it for rotating the retinal image of the person to reference position. For feature definition angular partitioning on magnitude spectrum of the retinal image and wavelet transform is used. Due to small feature vector and the extraction

vessel this method is simple and has low computational complexity. The feature vector generated has useful information about vessel density and vessels direction in the image as stated by the authors. For experimental purpose the proposed method was applied on a database consisting of 400 retinal images obtained from 40 persons. Noisy and rotational retinal image were used in the identification process. The authors finally concluded that 99.1% identification rate could be achieved using their proposed method.

Another method of biometric pattern matching is the nose print. This method is used to identify cattle and was first published by Dr. Petersen, a dairy researcher at the University of Minnesota in 1922. This method for nose print matching was developed to curb potential fraud such as tattooing, branding and ear tags. The cattle can be individually identified on the basis of the arrangement and distribution of ridges and valleys on the muzzle [21]. The method is simple and cheap. Ink is applied to the nose of the cattle and an impression is taken on paper just like taking an imprint of the fingers of a human being. The accuracy of the nose print matching depends on how the print is taken and whether same pressure is applied while taking an imprint. Figure 2.2 below shows the nose print of the same cattle taken in a period of six months [22].



**Figure 2.2. Two nose prints of the same cattle taken in a period of six months. Source: Oklahoma Cooperative Extension Service. Some Ways to identify Beef Cattle, n.d. <http://pods.dasnr.okstate.edu/docushare/dsweb/Get/Document-1563/N-612web.pdf>, accessed Aug. 2012.**

In 2004, an FBI Special Agent named Mr. Kozma proved that a 2D image correlation, a pattern matching method which is widely used in other context , can also be used to match cattle noseprints.

Data security today is one of the most prominent issue. With data flowing everywhere around through the internet and smartphones, security of data is a very serious concern. Computer viruses and network attacks are becoming sophisticated day by day. With regards to network security it has been managed on a local basis. Various areas of data security has been taken into consideration for research such as information leakage, privacy, fine grained access control, data encryption and secure shared computation. In one of the papers ZigBee technology, a new short distance wireless technology has been used more widely in the area of wireless network. The technology meets not only low confidentiality, but also low power, low complexity and low cost [23]. ZigBee's AES-128 encryption algorithm can ensure the data security of the wireless transmission. In another paper the authors proposed a new voice over internet protocol technique with a new hierarchical data security protection scheme [24]. The proposed HDSP scheme can maintain the voice quality degraded from packet loss and preserve high security data. The technique performs both the data inter-leaving on the inter-frame of voice for achieving better recovery of voices suffering from continuous packet loss and the data encryption on the intra frame of voice for achieving high data security. As stated by the authors the simulation and analysis results show that the proposed HDSP scheme can effectively reduce the occurrence of continuous packet loss and have good data protection.

Cloud computing has many advantages and many enterprise applications and data are migrating to public or hybrid cloud. Data security and privacy protection is very important from the perspective of a consumer. In one of the papers the authors provide a concise but all round analysis on data security and privacy protection issues associated with cloud computing [25]. The authors stated that the challenges in privacy protection are sharing data while protecting personal information. The typical systems that require privacy protection are e-commerce systems that stores information such as credit card and health care systems. The authors stated that the key to privacy information protection in cloud environment is to strictly separate the sensitive data from the non-sensitive ones followed by encryption of the sensitive data.

The number of smartphone users is growing rapidly. With many applications supported by the smartphone right from messaging to location based services the number of users has grown. It is predicted that worldwide the number of smartphone users will grow to 1 billion by 2015. A more and more people are using smartphones, mobile security concerns increase. According to a survey 24% of the users store their computer or banking password on their smartphones and 55% of the users do not use a pin lock. Some users use “jailbreak” on their Iphone to switch wireless carriers or download softwares that are not available on the App Store. In such a case a user may download third party malicious apps without any knowledge which can steal personal information from the user. Same goes the case with the Android OS. Since Android is an open source platform, a poorly written software can open up vulnerabilities on a user’s phone. Freely available wireless networks in cafeterias or coffee shops are often encrypted, thereby increasing the possibility of stealing data while transmitting sensitive data like bank transactions etc. It is quite a challenge to protect the privacy of cell phone users and the sensitive data. With the rapid growth of mobile communication the demand for secure transmission and execution of data has increased. In one of the papers the authors deals with these mobile data security technologies and aims to exhibit their potential of integrity, availability and confidentiality [26]. The authors provides a thorough analysis of the most important packet data services and technologies, which can reveal the data in secure manner. The authors addressed the security issues of the underlying mobile data network communication and proposed some important work in the areas of architecture, source and access point authentication. Health systems too depend on mobile devices for delivery of medical and health services. In another paper the authors addressed the issue of adding security to mobile data collection. They addressed security issues for a company called mHealth that has a profound and increasing impact on the delivery of medical and health services using mobile devices [27]. In their paper the authors proposed a protocol that provided end-to-end security, encrypted data storage and recovery mechanisms on mobile devices. The protocol proposed addresses security concerns regarding low-end mobile devices used for collecting sensitive and personal data.

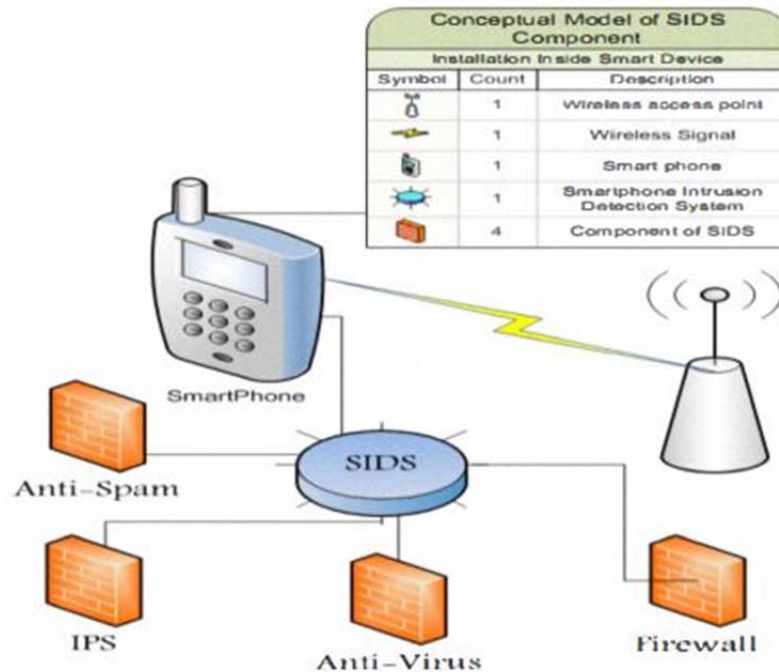
To address the security of smartphone users, the authors in one of their research papers proposed a differentiated user access control model to enhance smartphone security and user privacy [28]. This method classifies smartphone users based on certain sets of user

access privileges. The authors implemented a prototype of the proposed model on real world T Mobile GI smartphones. The prototype was developed using the Android platform. The prototype is a security model that provides smartphone users with a predefined set of access rights for different smartphone uses in different context. When a user powers on a smartphone, it will load a normal user mode where the user can configure everything except the system level settings. In order to switch to the administrative mode the user must input the correct password and the administrator can define the access rights for different user classes as stated by the authors [28]. The prototype system proposed by the authors is lightweight and flexible.

In order to enhance security on smartphones a lot of research has been on intrusion detection systems. Intrusion into a smartphone communication and transaction has significantly affected the reliability of data transfer and security [29]. In one of the papers the authors have thoroughly studied the performance analysis of intrusion detection and prevention system in smartphone communications and transactions [29]. The authors analyzed the existing Intrusion Detection System and identified its efficiency and inefficiency. Smartphone users need to be secured with regards to sensitive data like credit card information, bank accounts, phone bills etc. The authors proposed a new detection system called Smartphone Intrusion Detection System (SIDS) that acts solely to detect all types of unwanted and suspicious events that causes threats to smartphone security. As seen in the Figure 2.3 [29] the proposed system has blended the well known security mechanisms to detect, prevent, halt and discard any type of penetrations into the smartphone [29].

The authors stated that with this proposed system it can effectively and efficiently detect suspicious events, strongly prevent suspicious events, can remove viruses that reside on the smart phone and notify the user about new threats.

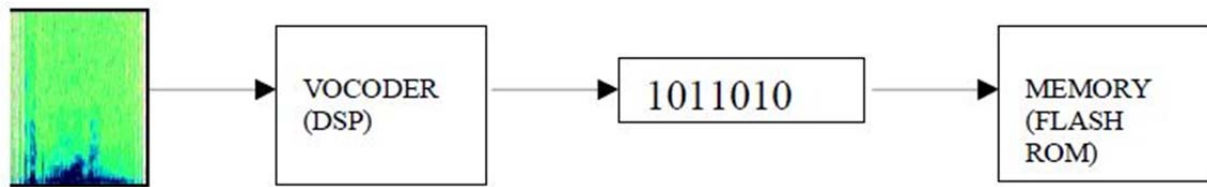
A cloud based intrusion detection and response system was proposed in one of the papers by the authors [30]. The authors implemented a working prototype of the intrusion of forensic analysis engine. The forensic engine as stated by the authors makes use of two types of information. Firstly the set of IDSes deployed and secondly system calls which are logged by a loadable kernel module. System call logs are used to create an information flow graph in which nodes are OS-level objects. The dependency graph with triggered IDS alerts generate



**Figure 2.3. Conceptual structure of SIDS. Source: S. Salah, S. A. Abdulhak, H. Sug, D. K. Kang, and H. Lee. Performance analysis of intrusion detection systems for smartphone security enhancements. *Mobile IT Convergence* 1:15-19, 2011**

system attack graph which encodes the possible attack paths according to provided information and the graph is later analyzed to identify misbehaving smartphone application.

Security of smartphones has also been addressed using biometric pattern matching techniques. Losing a cellphone exposes all the sensitive data of the user to risk. In one of the papers the authors proposed voice recognition and fingerprint matching techniques to be implemented on the cell phones as a mean of protecting it from stealing sensitive data. For the voice recognition system the authors proposed that as a user buys a smartphone, the user has to record his voice in the cellphone [31]. The frequency spectrum of speech signal is encoded by means of Vocoder and this coded signal is stored in the internal memory of the phone. As explained by the authors the microprocessor sends a control signal to the vocoder saying it to code the incoming voice input and also sends the address and a control signal to the memory dictating that it should store the coded signal in Flash ROM as shown in Figure 2.4 [31].



**Figure 2.4. The voice recognition architecture proposed by the authors. Source: H. A. Shabeer and P. Suganthi. Mobile phones security using biometrics. *International Conference on Computational Intelligence and Multimedia Applications*, Salem, 2007. IEEE.**

Whenever the user uses his cellphone the spectrum will be coded and will be compared with the coded spectrum stored in the Flash ROM and if it matches the Vocoder will send a signal to the user saying that he is authorized to use the phone.

For the fingerprint matching authors proposed that the database could be either ROM or Smart Cards. For authentication the user would press the thumb on the scanner and the impression will be stored in an EPROM. The authors stated that using the image comparator the impression of the thumb would be compared with the stored thumb impression and if both the impression matches then a signal is sent allowing the user to use the cell phone. But the main disadvantage of this technique is that it requires a lot of power consumption.

Handwriting or signature recognition has been a hot topic for research for quite a long time. The most fundamental step in understanding the handwriting or signature of a person is gathering the and analysis of the statistics related to the slant, shape, stroke number, order and direction. As discussed earlier signature verification technique has been used to authenticate the identity of the users. Quite a lot of research has been done for both the types of signature verification techniques. In one of the papers the authors provided a good overview and compared both the on-line and off-line algorithms with respect to accuracy [32]. Algorithm for preprocessing, character and word recognition was also considered by the authors. The authors stated that on-line signature verification technique was more accurate than the off-line case, however only handwriting recognition was observed instead of signature verification. The basic idea for both the on-line and off-line is the same except how much of information can be extracted from the writings. Various techniques have been used to for on-line signature verification. Among them the most popular ones are the Gaussian Mixture Model, Hidden Markov Model and the Dynamic Time Warping.

The Gaussian Mixture Model (GMM) is a statistical method which can be used for clustering of low dimensional data. It involves several multidimensional Gaussian probability distributions. Random initialization of cluster centers and their shapes are needed to construct a Gaussian Mixture Model. In the next step the parameters of the mixture is adjusted so that the correspondence between the data and the model is maximized and once the model is trained, it is used to measure the correspondence between sample data and the model [33]. Gaussian Mixture models have been used for face authentication and voice authentication. Using GMM in one of the papers the authors proposed a facial feature extraction technique which utilizes polynomial coefficients derived from 2D Discrete Cosine Transform (DCT) Coefficients obtained from horizontally and vertically neighboring blocks [34]. Experiments showed that the proposed technique by the authors is over 80 times faster to compute than features based on Gabor wavelet. The proposed technique is more robust than 2D Gabor wavelets and 2D DCT coefficients. In another paper the authors extended the proposed technique and enhanced it. The use of the new technique is useful for increasing robustness against white noise and compression artefacts. In another paper the GMM technique has been used for speaker verification system. The author presented a novel speaker verification system that generates a new feature set that captures long duration speaker identifying characteristics while taking advantage of the Gaussian mixture Model [35]. The system proposed by the author consists of a collection of independent GMMs, one for each phoneme which are built on these long duration feature vectors. The proposed system reduced both the equal error rate and the minimum value of the decision cost function on a standard verification test set.

Gaussian Mixture Model has been used for on-line signature verification. In one of the papers, the authors did a complete experimental evaluation of the Gaussian Mixture signature models and algorithmic issues were explored and compared to other commonly used on-line signature modeling techniques based on Hidden Markov Models [36]. The authors stated that the diagonal covariance matrix GMMs with more Gaussian components outperformed full-covariance matrix GMMs, at least at equal error rate. At close to EER the performance of the GMM based system is close to the performance of the HMM based system. The research trend indicated that reducing the number of states in the HMM is justified.



Another algorithm that is popularly used for pattern matching is the Hidden Markov Model. It takes the assumption that the system that needs to be modeled consists of Markov processes with unknown parameters. Basically the HMM consists of three elements. The first one is the amount of samples taken from a signature. The second one represents the X and Y coordinates and the third element is the probability of transition from one state to another. In one of the papers the authors introduced a new system known as the Handwritten Signature Verification which is an automated method of verifying a person by examining the features inherent in his/her handwritten signature [37]. HMM algorithm was used in this technique for signature verification. It extracts several local features (both static and dynamic) from the signature data and models these using the best aspects of historical HMM modeling for HSV as stated by the authors. The experimental results showed that the HSV system performed reasonably well with an overall error rate of 3.5%. In another paper the authors used HMM technique for on-line signature verification [38]. For each signature the authors constructed a HMM using a set of sample signature described by the normalized directional angle function of the distance along the signature trajectory. The Baum-Welch algorithm was used by the authors for both training and classification. Experiments were carried out by the authors based on 496 signatures from 31 subjects which showed the HMM technique is very potential for signature verification. A survey paper was published by for pattern recognition techniques in on-line hand written signature verification [39]. According to the authors the existing on-line signature verification can be classified into four classes: feature based approach, function-based approach, hybrid methods and trajectory construction methods. Feature based are statistical methods which use vector representation for a set of global features. Time sequences are used for function based approaches which describes the local properties of the signature for recognition. In regional methods, the time functions are converted to a sequence of vectors describing regional properties and trajectory construction methods produce and control complex 2D synergistic movements [39]. The survey of the techniques provided the authors a platform for development of the novel techniques to include novel integrated classifier to assist different modes of acquisition. The authors found out that the function based approach has the drawback of lumping together the local differences between signatures with its prototype and DTW alignment minimizes the natural differences between each sample of a signature.

In one of the papers the authors used a new technique for signature verification called the elliptic curve algorithm [40]. The main advantage using this algorithm as stated by the authors was that there is no sub exponential algorithm known to solve the elliptic curve discrete logarithmic problem. This proposed algorithm which is one of the variants of the Elliptic Curve Cryptography is an alternative to established public key systems such as Digital Signature Algorithm and Rivest Shamir Adleman. The authors stated that the key generated by their proposed algorithm is highly secure consumes less bandwidth due to the use of small key size by the elliptic curves. The biggest benefits of having a smaller key size is faster computing times and reduction in processing power, storage space and bandwidth which made the algorithm ideal for use in PDA's , cellphones and smart cards.

Another popular algorithm that is used for pattern matching is the Dynamic Time Warping. DTW was first introduced in the 1960s and used extensively in speech recognition, sign language recognition, data mining, online signature matching etc. Although DTW is quite popular a lot, in one of the papers the authors of research has been conducted by modifying this algorithm to produce more effective results [41]. As stated by the authors one major issue with the DTW algorithm is the behavior called “singularity”. The algorithm tries to explain the variability in the Y-axis by warping the X-axis which leads to unintuitive alignments where a single point on one time series laps onto a large subsection of another time series. This behavior is undesirable. A lot of measures have been taken to address this issue however these measures suffer from the drawback that may prevent the “correct” warping [41]. Another additional problem that the authors addressed is that the DTW algorithm may fail to find obvious natural alignments between two sequences due to variation of one feature which can be higher or lower than its corresponding feature in the other sequence. The authors stated that the considering two data points  $q_i$  and  $c_j$  which has identical values, but  $q_i$  is a part of the rising trend and  $c_j$  is part of a falling trend. DTW considers a mapping of these two points ideal but it should not be so. To address this issue the authors proposed a modification on the DTW algorithm that does not consider the Y value of the data points, but rather considers the higher level feature of “shape” and they obtained information about the shape by considering the first derivative of the sequences. The authors tested the ability of the algorithm to discover the correct warping between the two sequences. The authors approach was to take a sequence Q and make a copy of it. This

copy has a warping randomly inserted into it and then both the original sequence  $Q$  and the copy were used as input into the two algorithms and compared warping. The modified algorithm of the authors showed superior alignment between the time series. In another paper the authors modified the DTW algorithm and proposed their own new method called Continuous Dynamic Time Warping for comparing planar curves with application to the signature verification [42]. Although the proposed algorithm belongs to the general class of the DTW algorithm, they were successful to derive the structural properties that allowed the spatial complexity of the algorithm bounded without the need of heuristic approximations. The proposed algorithm when compared to the DTW algorithm provides a less noisy way of computing the reference from the training set but with regards to the computational time for performing the matching the proposed algorithm is three times slower than the original DTW.

Traditionally the use of dynamic time warping algorithm for signature verification consists some form of dissimilarity between the signature to be matched for authentication and the set of trained signatures. Taking this issue into account in one of the papers the authors proposed to replace this set of trained signatures with the hidden signature and used it to normalize errors of signature under verification [43]. The hidden signature approach proposed by the authors extends the least square approach. In the least square approach, the model approaches the expected value as the number of observations increases to infinity. The authors proposed two main directions for the hidden signature estimation. Firstly the iterative point by point averaging where this method denotes signature transformation into a time space of another signature. Secondly the evolutionary algorithm, each of which is constructed as an iterative procedure that alternates the warping steps and the averaging steps. The experiments were carried out in MCYT database as stated by the authors. The results obtained in test on the database were promising and the method proposed is mostly based on an engineering approach to error signals processing. The authors concluded that the overall simplicity allowed this method to be implemented on embedded or mobile systems. The authors used the payment terminals for signature verification during payment transactions. The proposed approach also met the memory and computational power constraints of payment terminals, as well as the time constraints.

## CHAPTER 3

### EXPERIMENTAL PROCEDURES

#### 3.1 DTW ALGORITHM

DTW algorithm has been a very popular technique for pattern matching. It has been widely used for speech recognition, sign language recognition, data mining, gesture recognition, handwriting and on-line signature authentication.

The DTW is basically a time series alignment algorithm which was developed in the 1960s. The main idea behind the algorithm is to compute the optimal alignment between the two series (feature vectors) in the form of warping path “w”.

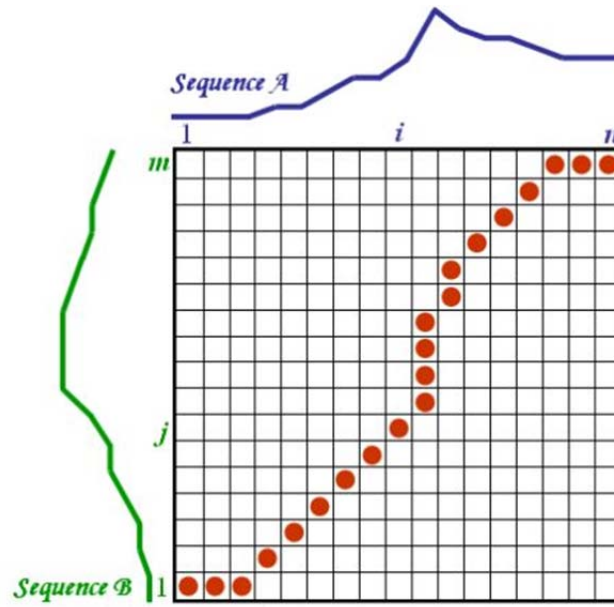
Consider two time series of feature vectors A and B below. A has “n” point and B has “m” points

$$A = a_1, a_2, \dots, a_i, \dots, a_n$$

$$B = b_1, b_2, \dots, b_j, \dots, b_m$$

In order to align the two sequences we use the DTW algorithm by constructing a “n x m” matrix where the  $i^{\text{th}}$  and the  $j^{\text{th}}$  element of the matrix contains the distance between the two points  $a_i$  and  $b_j$ . The two series A and B can be arranged in a grid with one of the series on the top of the grid and the other one on the left hand side of the grid as shown in Figure 3.1 [44].

Comparing the corresponding elements of the two series a distance measure is placed inside each cell of the grid. To find the best match between the series one has to find a path through the grid that minimizes the total distance between the two series. The step to find the overall distance will involve finding all possible paths through the grid and compute the overall distance for each one. The overall distance is the minimum of the sum of the distances between the individual elements on the path divided by the weighting function. There are two techniques to match the points between the two curves. One is called the linear matching which is simple and easy to compute because this technique is used when the two series or curves are of equal length. The second technique is a more complicated one because it is computationally more expensive. In this technique the distance between each and every



**Figure 3.1. Two series A and B arranged in a grid. Source: Universiteit Gent. DTW Algorithm, n.d.**  
<http://www.psb.ugent.be/cbd/papers/gentwarper/DTWalgorithm.htm>, accessed July 2012.

point of the first series is calculated with every point of the second series. The smallest distance for every point to the other series is decided and these smallest distances are added and divided by the number of points where each point on one series does not match more than one point of the other series. The DTW algorithm has a few constraints that need to be taken into account.

**Continuity Condition:** This first condition decides how much the matching of the two series is allowed to differ from linear matching.

**Boundary Condition:** Looking at Figure 3.1 the path starts at the bottom left and ends at the top right. This condition forces a match between the first points of the series and a match between the last points of the series.

**Monotonicity Condition:** This condition says that there is no “reversing back” while matching. In other words if the  $i^{\text{th}}$  point of the first series matches the  $j^{\text{th}}$  point of the second series then it is not allowed to match any point greater than the  $i^{\text{th}}$  point on the first series to a point on the second series which is less than  $j$ .

Warping Path Condition: The best path for matching the two series is likely to deviate very far from the diagonal of the grid.

In one of the papers the authors proposed a method for on-line handwritten signature verification [4]. The authors used digitizing tablets for the signatures, which was capable of capturing both the dynamic and spatial information of the writing. Several features were extracted after preprocessing of the sample signature. The input signature of the person is authenticated by comparing it with a stored set of three signatures referred by the authors as a template. The authors used the string matching algorithm to find the similarity between an input signature to be verified and the reference set computed. The similarity is matched by using a threshold value. The authors investigated several different approaches to find the optimal value for the threshold. A total of 1232 signatures from 102 different individuals were taken and the author found out that writer-dependent thresholds yield better results than using a common threshold. The best results as stated by the authors yielded a false reject rate of 2.8% and a false accept rate of 1.6%.

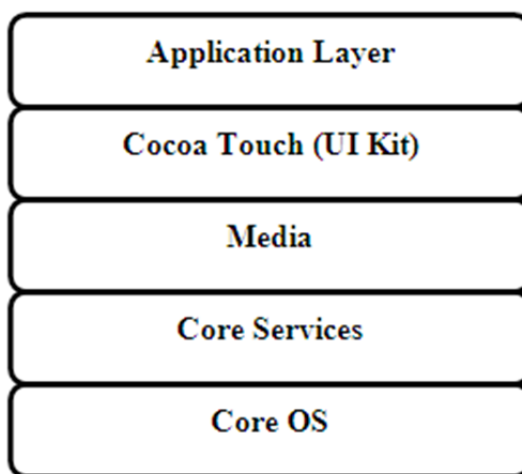
### **3.2 THE FRAMEWORK USED FOR DESIGNING THE APPLICATION**

The iPhone application was designed using the Cocoa Framework which is an application environment for both the Mac OS X and iOS operating systems. It consists of a suite of object oriented software libraries, a runtime system and an integrated development environment. The application framework that ios used for developing iOS applications is called the Cocoa Touch. The architecture of the Cocoa framework for iOS consists of the Core OS, CoreServices, Media and Cocoa Touch. The Core OS consists of the kernel, power management, device drivers file system and security. The Core Services provides services like the URL utilities, contact management, string manipulation and other services related to the hardware such as the compass, GPS, accelerometer and gyroscope. The Media provides the graphical and multimedia services which include the Graphics, Core Text, OpenGL ES, Core Audio ,video etc. The Cocoa Touch framework is the top of the architecture which supports the applications based on iOS.

Objective-C framework is used for developing applications for iOS which is used by both the Cocoa touch layer and the Core services. There are two core Cocoa frameworks in iOS:

1. **UIKit:** This framework displays applications in its user interface and also provides the classes needed for developing an application and manage its user interface.
2. **Foundation:** This framework provides objects to primitive data types and collections, provides the mechanisms for the management of the basic behavior of objects etc.

The architecture for the iOS Cocoa framework is shown in Figure 3.2 with the Application layer at the top and the Core OS at the bottom.



**Figure 3.2. Architecture of the cocoa framework for iOS.**

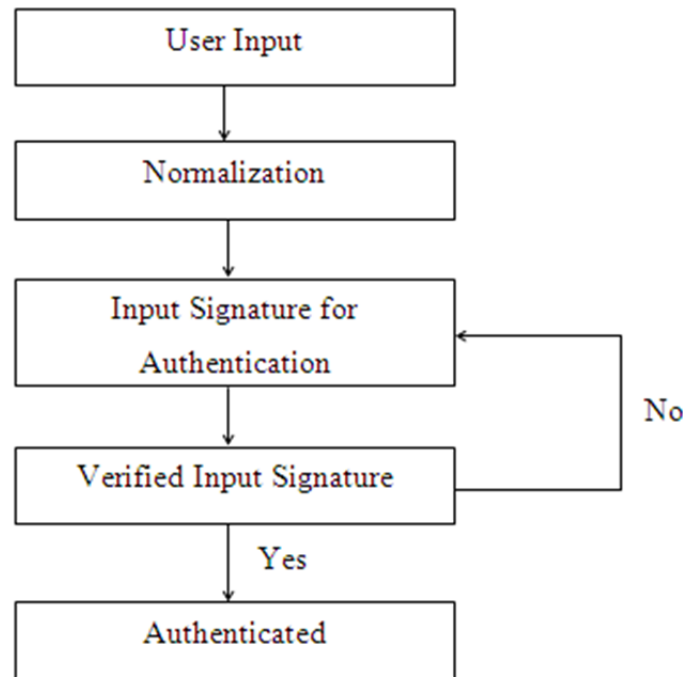
For designing the Cocoa application project XCode have been used. XCode provides the complete suite of development tools and frameworks. XCode powers the Integrated Development Environment for both the Mac OS and iOS operating systems. Using the XCode a developer can manage and create projects, write source code, build projects and debug the code. C, C++, Objective-C and Objective-C++ is used for building projects from source code in XCode. A platform SDK is chosen while creating a project which contains everything necessary for developing software for a given platform and OS. After building and running the project the XCode runs the simulator which presents our application as it would appear on the iPhone. For Graphics display we have used the OpenGL ES framework which is a low level API used for programming 2D and 3D graphics on the iPhone. The classes of the UIKit used for iOS inherit from NSObject which is the root class.

### **3.3 OUR APPROACH IN DESIGNING THE APPLICATION**

The application is designed in such a way that when a new user enrolls for the very first time he is asked to provide a set of signatures. In addition a threshold is provided for the

input signature to be authenticated with the reference set of signatures. Once the application is trained the user is taken to an authentication screen where he is asked to put a signature. If the user's input signature is within a certain threshold limit that has been set then the user will be able to log into the application and access the web accounts he has put.

A flowchart of the entire process is summarized in Figure 3.3.



**Figure 3.3. Steps involved in the design of the application.**

### 3.3.1 Training the Signature

For training the application the user is first asked to sign five times on the training screen. Each signature represents a vector point in space. Using the DTW algorithm we compute the minimum distance between all the possible signature pairs: 1-2, 1-3, 1-4, 1-5, 2-3, 2-4, 2-5, 3-4, 3-5, 4-5. After computing the distance we get ten values of distance. Each value corresponds to a pair of signatures. The signature pair that has the least value of the minimum distance is taken in consideration. For example if signature 2 and 3 has the smallest minimum distance values, we pick up signatures 2 and 3 for the stored reference set. Now from the signature pair that has been selected for reference we use the first signature and use it to compute the minimum distance using the DTW algorithm with the other three remaining signatures. The middle value is taken as the threshold value. A total of five signatures have to



be provided for training the application. No two signatures of the same user are the same. Due to fatigue or speed of the fingertip the signatures vary in size or stroke. If a user is not able to sign in due to variation of his signature it will be quite annoying for him. For this reason we have kept a set of three signatures in the stored reference set so that the input signature for authentication matches one of the three signatures within a certain threshold.

Algorithm to choose three candidate signatures:

1. Get the array of each point of the signature put on the screen.
2. Repeat the step 1 for five times for five signatures.
3. Calculate the minimum distance of each signature using the DTW algorithm.
4. Using the five minimum distance value of each of the signature we now obtain the minimum distance value using the DTW algorithm of each of the signatures with each other which will give us ten values.
5. We choose the pair of signature which has the least value of the minimum distance. This two signatures will be the two candidate/reference signatures.

Algorithm for Authenticating the Signatures:

1. Get the input signature.
2. Using the DTW algorithm find the minimum distance of the input signature with the two reference signature.
3. Take the pair that gives smaller value of minimum distance.
4. If the selected value is smaller than the threshold than the input signature is accepted.

DTW Algorithm:

1. Normalize the signature by finding out the least X coordinate and subtracting it with the other X coordinates and similarly doing the same for the Y coordinates.
2. Obtain the i and j value of the distance matrix by computing the square of the difference of each of the X values with all the Y values.
3. We now compute a second distance matrix called mat2 .The value first cell of both the matrices are the same.
4. Compute the first row and first column of the global distance matrix.
5. For the first row keeping the value of the first cell same add the previous value to get the current value of the cell.
6. Similarly for the first column we add the previous value of the cell to get the current value.
7. Obtain the other values of the mat2 matrix to find the minimum distance of the signature. Find the least of the three in the mat2 matrix i.e  $[j-1][i]$ ,  $[j-1][i-1]$  and  $[j][i-1]$  and add to  $\text{DistMat}[j][i]$ .
8. If all the three points are equal take the mid point and add to  $\text{DistMat}[j][i]$ .

9. Keep repeating steps 5 and 6 until we reach the last cell of the matrix which gives us the minimum distance.

### **3.3.2 Normalization**

This application is designed in such a way that a user can put his signature anywhere in the screen.

### **3.3.3 Authentication**

This is the most important part of the application which provides security to the accounts that have been added by the user. It is quite possible a user's signature might be forged. Forgery of on-line signatures can be classified into two types: skilled and simple forgery. A skilled forgery is one where the forger has prior knowledge about the user's signature. The forger practices the user's signature before duplicating for authentication. A simple forgery is one where the shape of the signature differs but the semantics of the signatures are the same.

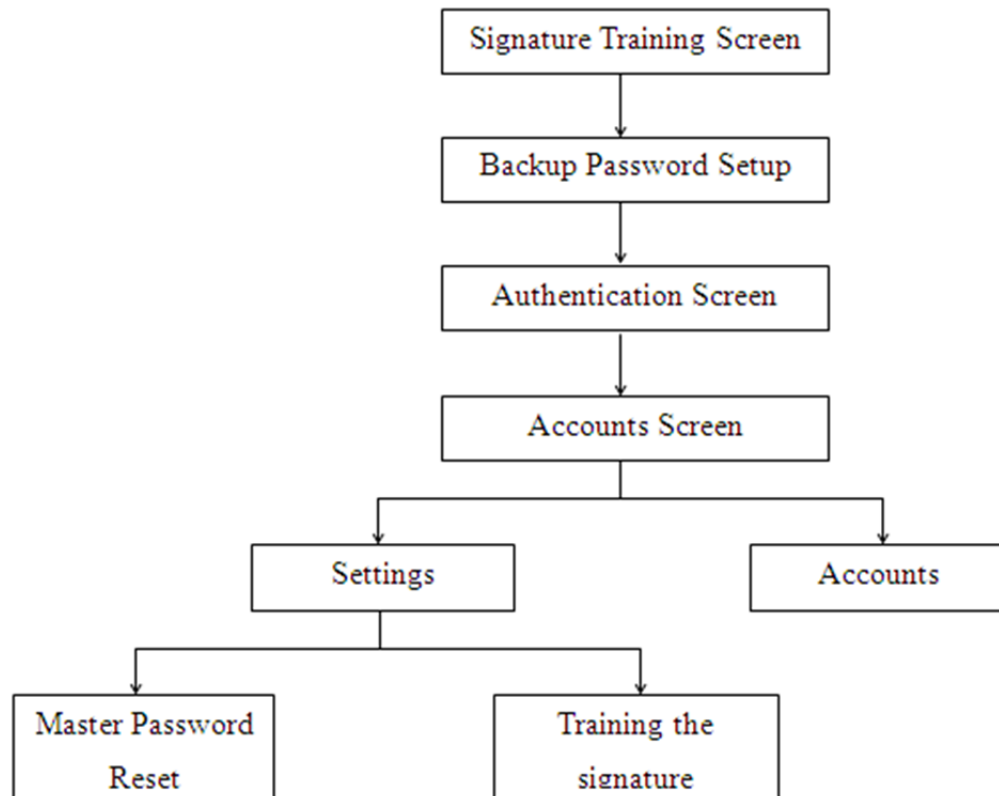
A user has variation in his signature. No two signatures of the same user are alike. For this reason a threshold value is given so that a user is able to log into his application. But the allowed variation in the signatures is a major factor in determining whether the signature is a genuine one or a forged one. This is where the concept of Confusion matrix comes in which has been used in the field of artificial intelligence. If a large variation is allowed in the threshold value, then the chances of a forged signature getting accepted is quite high. This is what we call as "False Acceptance" of the signature. But if the variation in the threshold value is decreased, there is a great possibility that the genuine signature put in by the user will not be accepted. This is what we call as "False Rejection" of the signature. A balance should be maintained between these two scenarios and an optimum threshold should be selected so both false acceptance rate and false rejection rate should be kept low.

## **3.4 UI DESIGN**

The UI for the application comprises of the following features (see Figure 3.4):

### **3.4.1 Signature Training Screen**

The signature training screen is where the user inputs five signatures one by one for training his application. Open GL ES platform which is a low level API has been used for



**Figure 3.4. Flowchart showing the design of the UI.**

designing this feature. The class defined for this feature takes five input signatures one by one and also finds out two reference signature for authentication. Write to file function (method in iPhone) has been used where we can write the strings directly to a file we named as plist. This method has been used to store each signature that the user inputs. The two reference signatures are chosen and the threshold value is set as described in the algorithm.

### **3.4.2. Backup Password**

Once the user inputs the five signatures for training he is asked to set up a backup password in case he is not able to log into the authentication screen using his signature. This feature was designed by taking into consideration that a user may not be able to put in a signature that matches the threshold due to finger fatigue or speed. The user is prompted with the strength of password too when he chooses his password. If the user enters four or less characters the strength is weak and if he enters ten or more characters the password strength is good. ViewDidLoad method was used for this backup password screen.

### **3.4.3 Authentication Screen**

In the authentication screen the user is asked to input a signature and if the signature matches the threshold value than the user logs into the application. In this class file the threshold value and the reference DTW signatures are read in first and then the DTW algorithm is applied between the input signature and the two reference signatures separately. If any of the two minimum distance value is less than the set threshold the input signature is accepted and the user logs into the screen.

### **3.4.4. Accounts Screen and Settings**

After the user is able to sign into the application next comes the account screen. Using the viewDidLoad class the user can add accounts for this application. For this purpose we have added Gmail and Facebook .Once the accounts are added then using the UIWebView class the list of accounts are listed that has been added and UIAlertView have been used for entering the account information like username and password. The Key Chain wrapper class has been used to store the username and password for each account. Using UIAlertView the user select the sites he wants if listed or added before. The accounts screen also has been provided with the Settings tab under which the user has been provided with the flexibility to reset the master password and retrain the signature if he feels like changing his signature pattern.

## CHAPTER 4

### RESULTS AND DISCUSSION

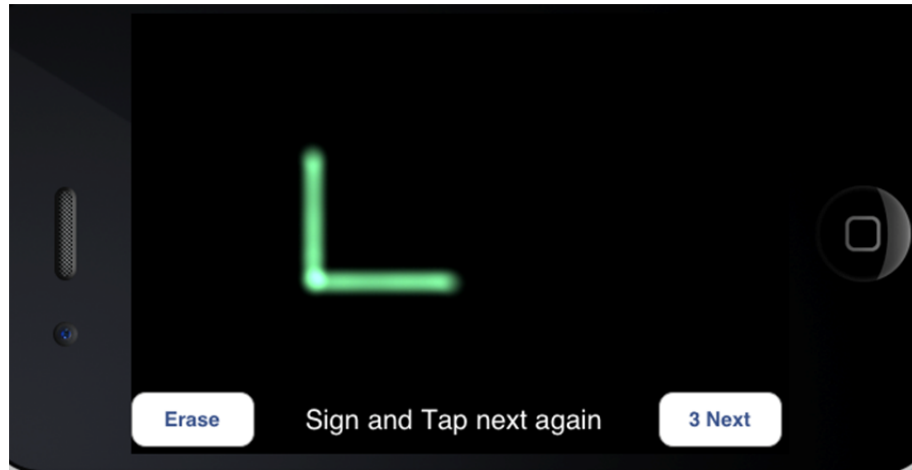
The UI is designed in a user friendly manner. The biometric characteristics of the same signature by the same person always vary. A little variance in the signatures is inevitable. This brings the need of the DTW algorithm to match the signatures and a threshold limit to accept the input signature if it matches within this set limit. While choosing the threshold limit one has to keep in mind the performance of the application with regards to False Acceptance Rate (FAR) and False Rejection Rate (FRR). A balance between the two must be maintained between the two.

The False Acceptance Rate is the ratio of the unauthorized users accepted by the application to the total number of attempts made by the user to sign in. Increasing the threshold limit by a factor will increase the False Acceptance Rate which is highly undesirable as it directly compromises the security of the application. With high False Acceptance Rate an unauthorized user can easily log into the application.

The False Rejection Rate is the ratio of the authorized users rejected by the application to the total number of attempts made by the user to sign in. Decreasing the threshold limit by a factor will increase the False Rejection Rate. This is undesirable as a genuine user may be deprived of signing into the application. So a balance between FAR and FRR is necessary in order to optimize the performance of the application. While training the signature in the authentication screen the application prompts the user to provide with a password which works as a backup in case the user is not able to log into the application due to stress or fatigue in his fingers.

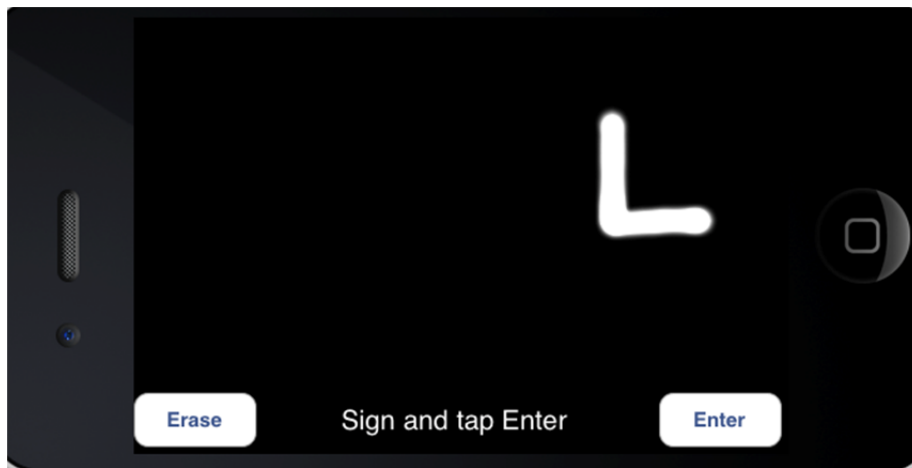
The biggest advantage of using this application is the normalization part where the user can put his signature on any part of the screen. This gives the ease of use and flexibility to the user while using the application.

Figure 4.1. shows the authentication screen of the application where a user is signing to train the application. From the Figure it is seen that the user has put his signature on the



**Figure 4.1. Signature training screen.**

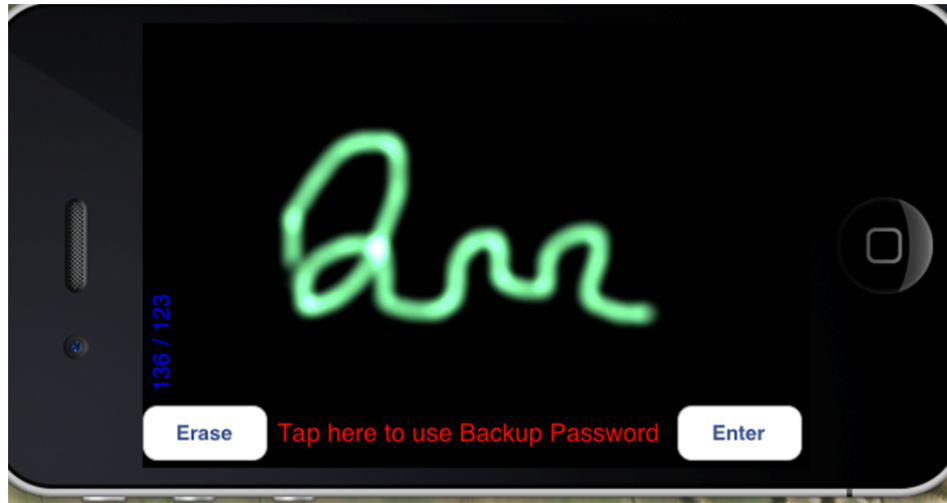
left bottom corner of the screen. Figure 4.2 shows the authentication screen and the user is trying to sign in by putting the signature at the top right of the screen.



**Figure 4.2. Authentication screen of the application.**

In order to compare the performance of the application with respect to different thresholds we have chosen two signatures one of which is an easy one and the other one is a little bit more complicated. For experimental purpose we are labeling the signature on Figure 4.1 as “Signature 1” and the signature on Figure 4.3 as “Signature 2”.


We are setting the threshold value by multiplying with a factor of 0.5 and 1.5 and comparing the performance or behavior of the application with the set threshold value.



**Figure 4.3. A signature on the screen labeled as “Signature 2”.**

From Table 4.1 it can be concluded that using a factor of 0.5 times the threshold value the application’s performance have degraded. For Signature 1 which appears to be in the most simple form the acceptance rate was found to be 20%. For this experiment 10 attempts were made for each of the signatures after training the application. Only two successful attempts were made for “Signature 1” which means the user was able to sign in only two times out of the ten attempts. On the other hand Signature 2 which looks like a more complicated one had zero successful attempts. This is really undesirable as to the much annoyance of the user he will not be able to sign into the application.

**Table 4.1. Table Showing the Acceptance Rate of the Two Signatures with a Threshold Factor of 0.5**

Signature		Threshold Factor	Acceptance rate
Signature 1		0.5x Threshold	20%
Signature 2		0.5x Threshold	0%

From Table 4.2 we can see that multiplying the threshold with a factor of 1.5 gives 100% acceptance to both the signatures. The acceptance rate was found to be 100% for which the user was able to sign into the application for the every attempt made. Even the acceptance rate for the “Signature 2” was 100%. With a factor of 1.5 it can be concluded that the user can sign in easily into the application. This seems to be much better factor of the threshold unlike the 0.5 factor where the acceptance rate was 20% for the “Signature1” and 0% for the “Signature 2”. However we need to keep in mind that there must be a balance between the false acceptance rate and the false rejection rate. A factor of 1.5 times the threshold seems appropriate to sign into the application but we do not want to threat the security of the application by allowing a bogus user to sign into the application with the slightest hint of the owner’s signature.

**Table 4.2. Table Showing the Acceptance Rate of the Two Signatures with a Threshold Factor of 1.5**



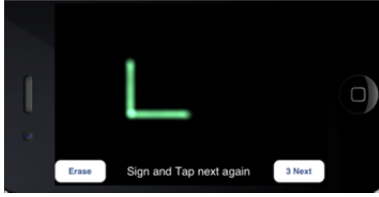
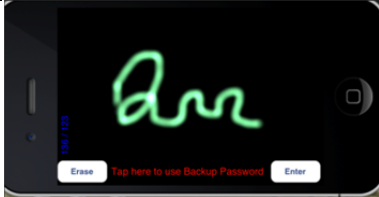
Signature		Threshold Factor	Acceptance rate
Signature 1		1.5x Threshold	100%
Signature 2		1.5x Threshold	100%

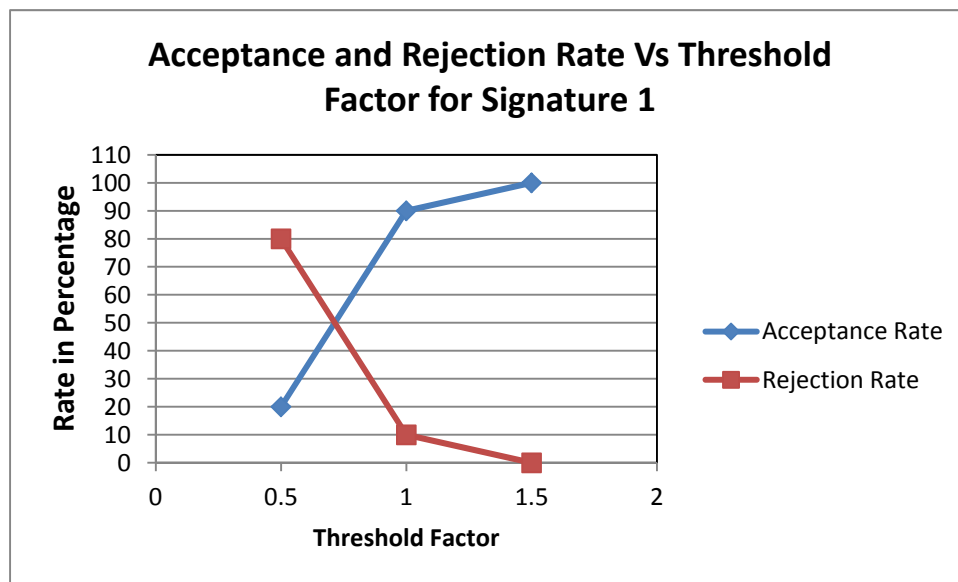
Table 4.3 represents the acceptance rate for both the signatures with a threshold factor of 1.0. We can see that the acceptance rate for “Signature 2” was 70%. This seems to be an optimum threshold value in the real world application. A balance between the FRR and FAR is maintained. Again this acceptance rate also depends on other factors like fatigue in fingers, user is trying to sign in while in motion. These factors will do have an impact on the acceptance rate of the application.

Figure 4.4 shows the plot of the acceptance rate and the rejection rate for Signature 1 with different values of the Threshold Factor. It can be easily concluded that the user can log



**Table 4.3. Table Showing the Acceptance Rate of the Two Signatures with a Threshold Factor of 1.0**

Signature		Threshold Factor	Acceptance rate
Signature 1		1.0x Threshold	90%
Signature 2		1.0x Threshold	70%



**Figure 4.4. Acceptance and rejection rate versus threshold factor for signature 1.**

into the application even when the threshold factor is 0.5 . But this experiment has been tried by the user who has trained the application himself and tries to log in with his own signature.

Being the Signature 1 pretty simple with the letter “L” it is quite obvious that even with the toughest threshold factor the user can log in. With the threshold factor set to 1.0 and 1.5 the acceptance rate is 90 and 100 percent which is not desirable for the application.

Again the plot for the acceptance and the rejection rate for signature 2 is being studied with Figure 4.5. Signature 2 being a complicated signature has lower acceptance rate and higher rejection rate compared to the Signature 1. With a threshold factor of 1.0 and 1.5 the rejection rate by the application for the same user is lower. This may not be true for a user trying to forge someone else's signature because the style of handwriting is unique for every user. One can imitate some else's signature only by practice and looking at the signature.



**Figure 4.5. Acceptance and rejection rate versus threshold factor for signature 2.**

Authenticating a document or a signature is very crucial. While designing a product for authentication one of the most important factor that one should keep in mind is forgery. A person cannot make two signatures mathematically identical. Two signatures can be identical if one is superimposed over another. This is possible only if the person is signing in a piece of paper. This application has been tested on the basis of forgery too. Basically we have tested this application for two types of forgeries scenarios. The first type is a skilled forgery where the forger is shown the user's signature and asked to practice before he tries to imitate the original user's signature. The second type can be called simple forgery where the forgers are asked to forge the original user's signature without having a look at the signature but the forgers know how to spell the signature.

In the first scenario a User X trains the application first. Five different forgers named as User 1 to User 5 are shown the User X's signature and were allowed to practice the

signature only a couple of times. Ten attempted signatures were provided by each of the forger to break into the application with different thresholds and the acceptance and rejection rate were observed. From Table 4.4 we can see that with a higher threshold factor the users are able to forge the signature. This is expected because signature is the only biometric authentication which is expected to be forged. But as the threshold factor decreases it becomes more difficult for the users to forge the signature. In one of the papers [37] the FAR for the skilled forgery was reported to be 11.25% which is comparable to the data we observed with our application. However the authors did not mentioned about the threshold they set to set the application. For our case the application performed pretty well with the threshold factor of 1.0 and 0.5.

**Table 4.4. Acceptance and Rejection Rate using Skilled Forgery by Five Different Users with Varying Threshold Factor**

Scenario 1	Skilled Forgery									
	Threshold Factor									
USERS	2.5		2		1.5		1		0.5	
	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR
User 1	70%	30%	60%	40%	40%	60%	0%	100%	0%	100%
User 2	70%	30%	50%	50%	20%	80%	20%	100%	0%	100%
User 3	80%	20%	60%	40%	40%	60%	10%	90%	0%	100%
User 4	70%	30%	60%	40%	40%	60%	10%	90%	0%	100%
User 5	80%	20%	50%	50%	20%	80%	0%	100%	0%	100%

From Figure 4.6 we can see a plot of the threshold factor versus the FAR for the scenario skilled forgery by five different users. With the threshold factor reaching to 2.0 and 2.5 the FAR is high. But with threshold factor of 1.0, one of the user was able to forge only 2 times and with a threshold factor 0.5 none of the users were able to forge the signature.

In the second scenario the forgers were asked to forge without looking at the signature. This time the FAR for all the different threshold factor were lower than the skilled forgery FAR. This is quite expected as it is very difficult to forge a person's signature without having a look at it. The style of handwriting of a person is difficult to predict when a user tries to forge it. As can be seen from Table 4.5 the FAR is the highest for threshold factor of 2.5, 2.0 and 1.5. But at the threshold factor of 1.0 and 0.5 none of the users are able

to log into the application. In this case of simple forgery the algorithm did well in protecting the application from forging in. In the same research paper the authors reported a FAR of

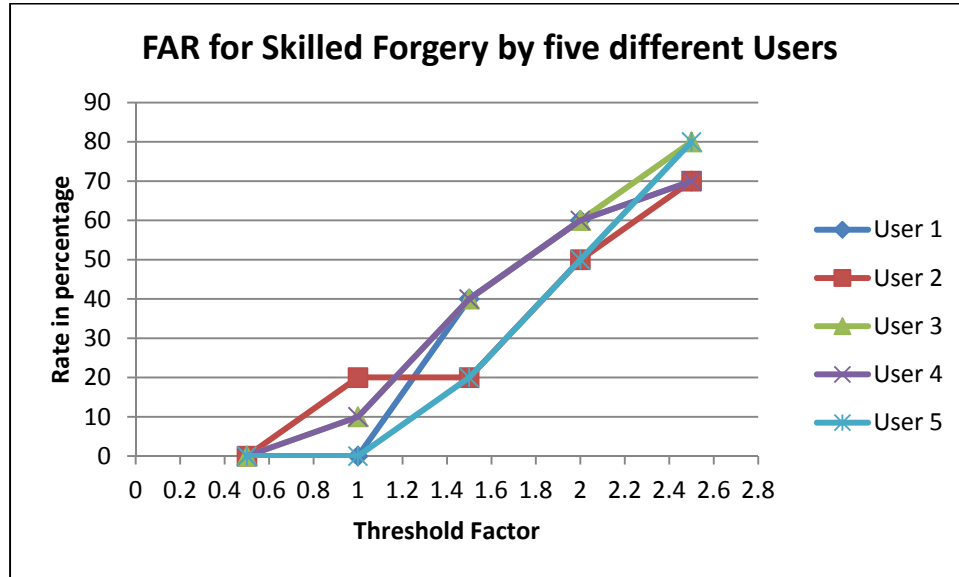


Figure 4.6. Plot of the FAR for five different users with varying threshold for the scenario skilled forgery.

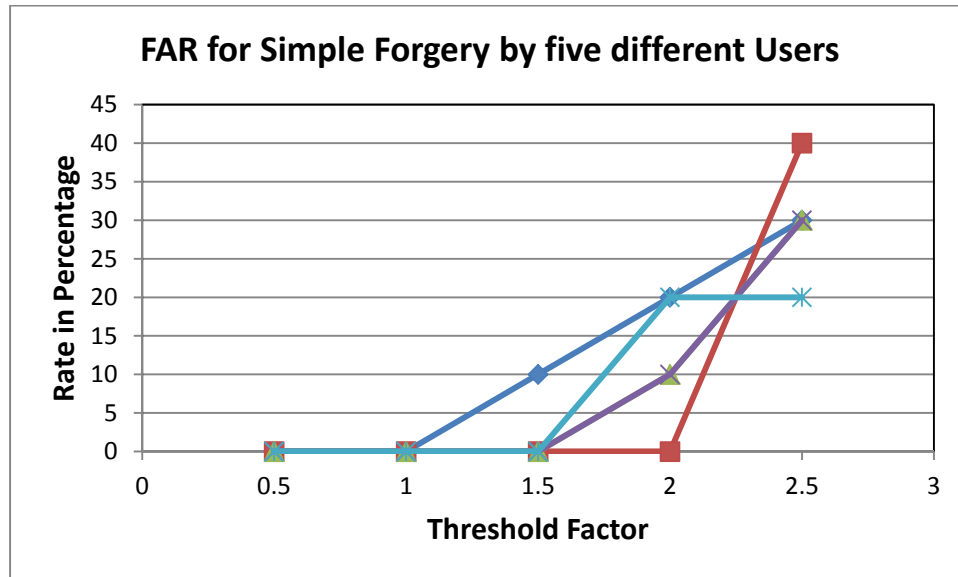
Table 4.5. Acceptance and Rejection Rate Using Simple Forgery by Five Different Users with Varying Threshold Factor

Scenario 2	Simple Forgery									
	Threshold Factor									
	2.5		2		1.5		1		0.5	
USERS	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR	FAR	FRR
User A	30%	70%	20%	80%	10%	90%	0%	100%	0%	100%
User B	40%	60%	0%	100%	0%	100%	0%	100%	0%	100%
User C	30%	70%	0%	100%	10%	90%	0%	100%	0%	100%
User D	30%	70%	10%	90%	0%	100%	0%	100%	0%	100%
User E	20%	80%	20%	80%	0%	100%	0%	100%	0%	100%

0.64 %. Comparing this data with the threshold factor of 1.0 and 0.5 our application performed better.

From the Figure 4.7 we can see the acceptance rate for simple forgery is lower compared to the skilled forgery and the results found with this algorithm is comparable with the results published by other authors. In one of the papers the authors reported a FAR of 0.64 %. In our case with the threshold factor of 0.5, 1.0 and 1.5 among 50 attempts only one

attempt was successful in forging the application. The algorithm was effective in preventing forgery with the threshold factor of 0.5, 1.0 and 1.5.



**Figure 4.7. Plot of the FAR for five different users with varying threshold for the scenario simple forgery.**

## CHAPTER 5

### CONCLUSIONS

DTW algorithm was implemented for signature pattern matching on the iPhone application. Five input signatures are taken from the user in the process of training the application. From the five signatures using the DTW algorithm two signatures are chosen as the reference signature. A threshold value is set in order to compare with input signature. The input signature is taken by the application and using the DTW algorithm the distance is computed using it with the other two signatures respectively. If the distance values computed lies within the given range of the threshold value the signature is accepted and the user is able to sign in. The design of the application was made by considering factors like finger fatigue of the user while trying to sign into the application. In order to make the application more flexible a backup password has been provided to enable the user to sign into the application in case the user is not able to sign in due to finger fatigue. Different threshold values were evaluated to find the optimum one. It was found that multiplying the threshold by a factor of 0.5 the false rejection rate was quite high and the user was not able to sign into the application with a more complicated signature. However a factor of 1.5 times the threshold seems to have a higher acceptance rate of the input signature. This is not desirable as we do not want a bogus user to sign into the application and threatening the security of the application. Some amount of balance must be maintained between the false acceptance rate and the false rejection rate. The optimum threshold was found and a balance between the false acceptance rate and false rejection rate is maintained. The application was tested with respected to two different forgery scenarios and the results found were comparable with respect to the results published by other authors with different algorithm. The acceptance rate was low in the case of simple forgery as compared to skilled forgery. With the threshold factor of 1.0 and 0.5, none of the authors were able to forge into the application which speaks about the effectiveness of the algorithm.

## **CHAPTER 6**

### **FUTURE WORK**

Pattern matching techniques have been applied in a number of systems. These include biometric techniques like retina matching, fingerprint, face recognition, signature etc. A number of algorithms have been used for pattern matching. Of all the algorithms the DTW and the Hidden Markov model algorithms are the most popular ones. For future work Hidden Markov Model algorithm can be tried and applied for pattern matching on the iPhone. The performance of both the algorithms can be compared. DTW was used primarily by keeping in mind the energy consumption of the battery. Algorithms like HMM is computing intensive and requires a lot of data which in turn takes more power from the battery.

Another enhancement to the application one can think of a tool to remind the user for training the application. The tool can keep track of the number of successful login attempts versus the rejection rate .For example if a user is trying to log into the application the tool can keep track of a record in the database regarding how many times the user made attempt for a successful login. The tool can keep track of the number of times the user accesses the application and maintain the number of attempts made by the user in each access to successfully log into the application. The tool should be developed in such a way that for a particular number of accesses by the user if the attempts made by the user the tool should prompt the user to retrain his signature again. Another alternate solution is to make the tool pop up the retraining signature screen for the user every 30 days. The user must be provided with the option of either accepting it or rejecting it.

Another upcoming technology that can be applied for security of cellphone applications is the technology of writing in air. The users can write in the air using a pen. However this technology needs to consider the battery power consumption of a cellphone while implementing it on the cellphones.

## REFERENCES

- [1] Michigan State University. Signature Verification, n.d.  
[http://www.cse.msu.edu/~cse802/Papers/802\\_Signature\\_Verification.pdf](http://www.cse.msu.edu/~cse802/Papers/802_Signature_Verification.pdf), accessed June 2012.
- [2] D. Lepre. RPM Mortgage, 2002.  
<http://www.loanmine.com/content.aspx?FileName=CustomPage113.x>, accessed June 2012.
- [3] A. K. Jain, S. Pankanti, and R. Bolle. *Biometrics: Personal identification in networked society*. Kluwer Academic Publishers, Dordrecht, Netherlands, 1999.
- [4] A. K. Jain, F. D. Griess, and S. D. Connell. On-line signature verification. *Pattern Recognition*, 35:2963-2972, 2002.
- [5] R. Bellman and R. Kalaba. On adaptive control process. *Auto. Control IRE Trans.*, 4(2):1-9, 1959.
- [6] C. Myers, L. Rabiner, and A. Rosenberg. Performance tradeoffs in dynamic time warping algorithms for isolated word recognition. *Acoustics Speech and Signal Processing, IEEE Trans.*, 28(6):623-635, 1980.
- [7] T. Venkatesh, S. Balaji, and A. S. N. Charavarthy. Security evaluation of online verification system using webcams. *Inter. J. Comp. App.*, 41(15):28-33, 2012.
- [8] V. E. Ramesh and M. N. Murty. Off-line signature verification using genetically optimized weighed features. *Pattern Recognition*, 32(2):217-233, 1999.
- [9] M. A. Ismail and S. Gad. Off-line Arabic signature recognition and verification. *Pattern Matching*, 33(10):1727-1740, 2000.
- [10] F. Alonso-Fernandez, J. Fierrez, M. Martinez-Diaz, J. *International Conference on Image Processing*, Cairo, 2009. IEEE Computer Society.
- [11] R. Martens and L. Claesen. *Proceedings of the International Conference on Pattern Recognition*. Vienna, 1996. IEEE Computer Society.
- [12] M. M. Shafiei and H. R. Rabiee. *Proceedings of the Seventh International Conference on Document Analysis and Recognition*. Edinburgh, 2003. IEEE Computer Society.
- [13] A. Senior and R. Bolle. Improved fingerprint matching by distortion removal. *IEICE Transactions Info. Syst.*, 84:1-7, 2001.
- [14] X. Tan and B. Bhanu. Fingerprint matching by genetic algorithms. *J. Pattern Recognition Society*, 39:465-477, 2006.
- [15] A. Ross, S. Prabhakar and A. Jain. *Proceedings of the International Conference on Image Processing*, Thessaloniki, 2001. IEEE Signal Processing Society.



- [16] E. Spinella. *Biometric scanning technologies: Finger, facial and retinal scanning*. SANS GSEC, San Francisco, CA, 2003.
- [17] W. Zhao, R. Chellappa, P. J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Survey*, 35(4):399-458, 2003.
- [18] L. Wiskott, J. M Fellous, N. Kruger, and C. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Tran. on Pattern anal. Mach. Intell.*, 19(7):775-779, 1997.
- [19] S. M. R. Kabir, R. Rahman, M. Habib, and M. R. Khan. *International Conference on Electrical and Computer Engineering*, Dhaka, 2004. ICECE.
- [20] M. Sabaghi, S. R. Hadianamrei, M. Fattahi, M. R. Kouchaki, and A. Zahedi. Fourier and wavelet transform. *J. Signal Info.Processing*, 3:35-38, 2012.
- [21] B. Ebert. Identification of Beef Animals, 1997.  
<http://www.aces.edu/pubs/docs/Y/YANR-0170>, accessed June 2012.
- [22] Oklahoma Cooperative Extension Service. Some ways to identify beef cattle, n.d.  
<http://pods.dasnr.okstate.edu/docushare/dsweb/Get/Document-1563/N-612web.pdf>, accessed Aug. 2012.
- [23] L. Chunqing and Z. Jiancheng. Research of ZigBee's data security and protection. *International Forum on Computer Science-Technology and Applications*, Chongqing, 2009. IEEE.
- [24] J. I. Guo, J. C. Yen, and H. F. Pai. New voice over internet protocol technique with hierarchical data security protection. *IEEE Proceedings – Visual, image and signal processing*, 149(4):237-243, 2002.
- [25] D. Chen and H. Zhao. Data security and privacy protection issues in cloud computing. *International Conference on Computer Science and Electronics Engineering*, Hangzhou, 2012. IEEE.
- [26] D. Nayak, N. Rajendran, D. B. Pathak, and V. P. Gulati. Security issues in mobile data networks. *Vehicular Technology Conference*, Los Angeles, 2004. IEEE.
- [27] F. Mancini, K. A. Mughal, S. H. Geijbo, and J. Klungsoyr. Adding security to mobile data collection. *13th International Conference on e-Health Networking, Applications and Services*, Columbia, 2011. IEEE.
- [28] X. Ni, Z. Yang, A. C. Champion, and D. Xuan. DiffUser: Differentiated user access control on smartphones, 2009. [http://www.cse.ohio-state.edu/~champion/pubs/09\\_wsns\\_nybcx.pdf](http://www.cse.ohio-state.edu/~champion/pubs/09_wsns_nybcx.pdf), accessed Oct. 2012.
- [29] S. Salah, S. A. Abdulhak, H. Sug, D. K. Kang, and H. Lee. Performance analysis of intrusiu detection systems for smartphone security enhancements. *Mobile IT Convergence*, 1:15-19, 2011.
- [30] A. Houmansadr, S. A. Zonouz, and R. Berthier. *International Conference on Dependable systems and Networks Workshops*, Urbana, 2011. IEEE.

- [31] H. A. Shabeer and P. Suganthi. Mobile phones security using biometrics. *International Conference on Computational Intelligence and Multimedia Applications*, Salem, 2007. IEEE.
- [32] R. Plamondon and S. N. Srihari. On-line and Off-line handwriting recognition: A comprehensive survey. *Pattern Anal. Mach. Intell., IEEE Trans.*, 22:63-84, 2000.
- [33] R. S. Corradin. Signature verification in consignment notes. Master's thesis, VU University, Amsterdam, Netherlands, 2008.
- [34] C. Sanderson and K. K. Paliwal. Fast features for face authentication under illumination direction changes. *Pattern Recogn. Lett.*, 24:2409-2419, 2003.
- [35] S. J. Stafford. The Sequential GMM: A Gaussian Mixture Model Based Speaker Verification System that Captures Sequential Information, 2005.  
[http://www.icsi.berkeley.edu/ftp/global/pub/speech/papers/stacked\\_GMM\\_report\\_v25.pdf](http://www.icsi.berkeley.edu/ftp/global/pub/speech/papers/stacked_GMM_report_v25.pdf), accessed Oct. 2012.
- [36] J. Richiardi and A. Drygajlo. *Proceedings of the ACM SIGMM Workshop on Biometrics Methods and Applications*, Berkley, 2003. Association for Computing Machinery.
- [37] A. McCabe and J. Trevathan. *International Conference on Embedded and Ubiquitous Computing*, Shanghai, 2008. IEEE Computer Society.
- [38] L. Yang, B. K. Widjaja, and R. Prasad. Application of hidden Markov models for signature verification. *Pattern Recogn.*, 28:161-170, 1995.
- [39] K. P. Radhika, G. N. Sekhar, and M. K. Venkatesha. International Conference on Multimedia Computing and Systems, Quarzazate, 2009. IEEE.
- [40] A. Khalique, K. Singh, and S. Sood. Implementation of elliptic curve digital signature algorithm. *Int. J. Comp. App.*, 2(2):21-27, 2010.
- [41] E. J. Keogh and M. J. Pazzani. Derivative Dynamic Time Warping, 2001.  
<http://www.ics.uci.edu/~pazzani/Publications/keogh-kdd.pdf>, accessed Oct. 2012.
- [42] M. E. Munich and P. Perona. Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. *International Conference on Computer Vision*, 1:108-115, 1999.
- [43] J. P. Leszcynska and M. Kudelski. Hidden signature for DTW signature verification in authorizing payment transactions. *J. Telecomm. Info. Tech.*, 4:59-76, 2010.
- [44] Universiteit Gent. DTW Algorithm, n.d.  
<http://www.psb.ugent.be/cbd/papers/gentxwarper/DTWalgorithm.htm>, accessed July 2012.