

**A Cryptosystem With Private Key Generation From
Dynamic Properties of Human Hand Signature**

HAO FENG

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2002

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research and has not been submitted for a higher degree to any other University or Institution.

Date

Hao Feng

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Associate Professor Chan Choong Wah. Thanks for giving me least restrictions on choosing a topic of my interest. The project won't be a success without his kind support and insightful guidance.

I want to thank the twenty-five students and teachers who kindly donated their signatures to build the signature database. Thank my buddy friend, Mr. Cao for giving me valuable comments on the thesis writing. Thank all the lab technicians in the Information System Research Laboratory, EEE, for their passionate help and technical support throughout the research.

Finally, I would like to acknowledge the School of Electrical and Electronic Engineering, NTU. Thanks for giving me an opportunity to pursue a graduate research, and use the lab facilities.

Summary

In recent years, Public Key Infrastructure (PKI) has emerged as co-existent with the increasing demand for digital security. A digital signature is created using the existing public key cryptography technology. This technology permits commercial transactions to be carried out across un-secure networks without fear of tampering or forgery. The relative strength of digital signatures relies on the access control over the individual's private key. The private key storage, which is usually password-protected, has long been a weak link in the security chain. In this project, we propose a novel and feasible system: BioPKI cryptosystem, which dynamically generates private keys from user's on-line handwritten signatures. The BioPKI cryptosystem eliminates the need of private key storage. The system is secure, reliable, convenient and non-invasive. In addition, it ensures non-repudiation to be addressed on the maker of the transaction instead of the computer where the transaction occurs.

The working mechanism of the BioPKI cryptosystem is based on the fact that certain features in the human handwritten signature are relatively consistent. From those consistent features, the same private key can be generated from each of the genuine signatures. The system comprises three processing stages: shape matching, feature coding and private key generation. In the shape matching stage, it exams the static shapes of sample signatures and rules out simple forgeries. In the feature coding stage, it extracts a common set of dynamic features and generates an all-bits-correct code string. It will further filter out skilled forgeries. Finally in the private key

generation stage, a private key is derived from the code string, by following the well-established public key algorithms, e.g. RSA and DSA.

A database comprising 25 users and 1000 signature samples, collected at one-month interval, has been built to evaluate the system performance. The Equal Error Rate (EER) for the BioPKI cryptosystem is 11.77%. The false alarm is maintained at a reasonable level. A user may need to try on average 1.4 times to derive an authentic private key. Nevertheless it is worth paying a bit more effort on signing so as to enjoy the great convenience of not bringing around any smart card or memorizing any password. Because the private key is dynamically generated from a familiar and natural way of human behavior – hand signing.

Contents

Acknowledgments	i
Summary	ii
List of Figures	ix
List of Tables	xi

Chapter 1 Introduction

1.1 Background	1
1.1.1 History of Biometrics	1
1.1.2 Major Biometrics Methods	2
1.1.3 Pros and Cons of Biometrics	3
1.1.4 Digital Signature	5
1.2 Motivation	7
1.2.1 Vulnerability of Private Key Storage	7
1.2.2 Synergy of Biometrics and PKI	8
1.3 Project Overview: BioPKI Cryptosystem	10
1.4 Summary of Contributions	10
1.5 Organization of the Thesis	11

Chapter 2 State-of-the-Art in Signature Verification

2.1 Overview	12
2.2 Data Acquisition	14
2.3 Preprocessing	15
2.3.1 Normalization	15
2.3.2 Re-sampling	16

2.3.3 Smoothing	17
2.4 Feature Extraction	17
2.4.1 Features from On-line Signature	17
2.4.2 Segmentation	18
2.4.3 Feature Selection	19
2.5 Comparison	21
2.5.1 Functional Approach	22
2.5.2 Parametric Approach	24
2.6 Performance Evaluation	25
2.7 Summary	27

Chapter 3 BioPKI Cryptosystem

3.1 Overview	28
3.2 Why Choose On-line Signature	28
3.3 An Overview of BioPKI Cryptosystem	29
3.3.1 First Processing Stage: Shape Matching	30
3.3.2 Second Processing Stage: Feature Coding	31
3.3.3 Third Processing Stage: Private Key Generation	31
3.3.4 The Role of the Template	31
3.4 Two Phases of Operation	32
3.5 Security Aspects of the BioPKI Cryptosystem	33
3.6 Data Acquisition	34
3.7 Database Collection	35
3.8 Preprocessing	36
3.8.1 Normalization	36
3.8.2 Re-sampling Shape	37

3.8.3 Speed computation	38
3.9 Template Generation	39
3.10 Summary	39

Chapter 4: Shape Matching Stage

4.1 Overview	40
4.2 Characteristic Functions of Shape	41
4.2.1 Center of Mass (CoM)	41
4.2.2 Torque	44
4.3 Dynamic Time Warping	46
4.3.1 Introduction to DTW Algorithm	46
4.3.2 Application of DTW	47
4.3.3 Problems with DTW	49
4.4 A New Matching Technique: Extreme Points Warping	50
4.4.1 EPs Marking	51
4.4.2 EPs Matching	52
4.4.3 Segment Warping	61
4.4.4 Other Examples of Using EPW	62
4.4.5 Warping Forged Signals using EPW	65
4.5 Evaluation of the New Technique	67
4.5.1 Testing Conditions	67
4.5.2 Error Rates for DTW and EPW	69
4.5.3 Improved Error Rates with Weight for EPW	70
4.5.4 Computation Time for EPW	72
4.6 Summary	74

Chapter 5 Feature Coding Stage

5.1 Overview	76
5.2 Feature Coding and Signature Identification	76
5.3 The Proposed Coding Scheme	79
5.4 Selection of Feature Parameters	82
5.4.1 A Common Feature Set	82
5.4.2 A Personalized Feature Set	84
5.5 Private Key Generation Stage: An Example of DSA	85
5.6 Summary	86

Chapter 6 Performance Evaluation

6.1 Overview	88
6.2 Analysis of Overall Performance	88
6.2.1 Performance Using 43 Features	89
6.2.2 Performance Using a Personalized Feature Set	91
6.3 Uniqueness and Security Strength of the Private Key	94
6.4 Suggested Remedies	95
6.5 Analysis of Overall Performance	96

Chapter 7 Conclusions and Recommendations

7.1 Conclusions	97
7.2 Recommendations	99

Author's Publications	101
------------------------------	-----

Bibliography	102
---------------------	-----

Appendix A	A Summary of On-line Signature Verification Projects During 1994~2002	110
Appendix B	Some Signature Examples in the Database	112
Appendix C	Execution Time Using DTW and EPW	115
Appendix D	Histograms of the 43 Features	116
Appendix E	Technical Terms Used in the Thesis	119

List of Figures

Figure 1-1: 2001 Market Shares by Biometric Technology	3
Figure 1-2: Authentication Methods – Knowledge, Possession and Biometrics	4
Figure 1-3: Digital Signature Signing and Verification	7
Figure 2-1: Five Stages in Signature Verification	14
Figure 2-2: A Typical Tablet Hardware and Data Extracted	15
Figure 2-3: Three Types of Parameters in Feature Selection	19
Figure 2-4: FRR and FAR Curves	25
Figure 3-1: A Block Diagram of the BioPKI Cryptosystem	30
Figure 3-2: Two Operating Phases – Enrolment and Testing	33
Figure 3-3: Six Sets of Data Captured by a Java Program	35
Figure 3-4: Pre- and Post- Re-sampling at Equal Distance	38
Figure 4-1: The Sliding Gaussian Window	42
Figure 4-2: A Normalized Gaussian Window with Width $8L$	43
Figure 4-3: A Sliding Coordinate Frame	44
Figure 4-4: X, Y and Torque Signals	45
Figure 4-5: The Time-Alignment Path in DTW	46
Figure 4-6: Waveforms before and after DTW	48
Figure 4-7: A Demonstration of Extreme Points and Ripples	51
Figure 4-8: A Demonstration of Extreme Points From Two Torque Signals	53

Figure 4-9: The Warping Path of EPW	54
Figure 4-10: An Example of Using Extreme Points Matching	57
Figure 4-11: Two Cases for Matching the End Extreme Points	58
Figure 4-12: Local Warping Paths for the Last Corner-Cell in EPW	59
Figure 4-13: A Demonstration of Correct Matchings of Extreme Points	60
Figure 4-14: Matching Segments Based on Extreme Points	61
Figure 4-15: Linearly Warping the Matching Segments	61
Figure 4-16: Sample Signals Before and After EPW	62
Figure 4-17: Example of EPW (1) – Showing the Correct Matching at Start	63
Figure 4-18: Example of EPW (2) – Showing the Correct Matching at End	63
Figure 4-19: Example of EPW (3) – Showing the Mismatching	64
Figure 4-20: A Comparison of Using EPW for Genuine and Forged Signals	66
Figure 4-21: The Point of Operation for Shape Matching	72
Figure 4-22: A Comparison of Speed between DTW and EPW	74
 Figure 5-1: A Demonstration of Feature Coding for Pen-Down Time	 79
Figure 5-2: Features With Relative Flat Data Distributions	83
Figure 5-3: Unselected Features with Lumped Data Distributions	84
 Figure 6-1: The Error Rate Curves for Overall Performance	 89
Figure 6-2: Trade-off Curves Using Personalized Features	91
Figure 6-3: Error Curves Using Personalized Features	92

List of Tables

Table 1-1: A Comparison for Different Biometric Technologies	5
Table 3-1: A Summary of Database Collection Activity	36
Table 3-2: A Summary of Signatures in the Database	36
Table 4-1: Equal Error Rates for EPW and DTW (%)	69
Table 4-2: Equal Error Rates for EPW with Added Weight (%)	71
Table 4-3: Computation Times Using DTW and EPW for All Users (ms)	73
Table 5-1: Concatenation of Feature Codes into a Code String	81
Table 5-2: A Common Set of 43 Features	83
Table 6-1: The Error Rates and Bit-lengths for 25 Users	91
Table 6-2: Error Rates and Bit-lengths for Different Numbers of Features	93
Table 6-3: Key Length and the Estimated Number of Features	95

Chapter 1 Introduction

1.1 Background

1.1.1 History of Biometrics

The term “biometrics” is derived from the Greek words *bio* (life) and *metric* (to measure) [1]. In the IT (Information Technology) environment, it is the technology that automates the identification of a person by analysing their physical or behavioural traits [2].

Long before the creation of automated biometric technology, human had already learnt to use biometric features under various circumstances. Some biometric features, e.g. face, voice, and figure, were often used to identify acquaintance or even unmet people. Other biometric features, such as fingerprint and handwritten signature, were commonly adopted to enforce legal binding between a person and agreements. In ancient Babylon, fingerprints were used on clay tablets for business transaction. In ancient China, thumbprints were found on clay seals [2]. Similar to fingerprint, handwritten signatures have long been regarded as unique-to-person. In many countries, the practice that people sign their names to signify their assent to business or official documents has been used for thousands of years [3]. Today it is still widely used in daily transactions across the world.

In the new electronic era, computers are becoming commonly found in households and the Internet connections are growing explosively. The e-documents have replaced traditional paper documents in many modern offices. The replacement of the

traditional ink-on-the-paper signatures (e.g. handwritten signature) by the electronic signatures for those e-documents is a hot research topic [3], which is also the focus of the work in this thesis.

1.1.2 Major Biometrics Methods

Based on the characteristics to be measured, biometrics can be grouped into two categories: physical biometrics and behavioural biometrics. Physical biometrics measures an individual's congenital physical characteristics while behavioural biometrics measures the acquired behavioural characteristics.

Physical biometrics – fingerprint, hand geometry, facial recognition, iris, retina, etc

Behavioural biometrics – handwritten signature, keystroke, etc

In addition to various types of biometrics mentioned above, voice recognition is an example that is a hybrid of the two categories [3]. Other varieties of biometrics proposed include thermal facial recognition with infrared cameras, ear or lip shape, knuckle creases, body odour, and even DNA [2]. A detailed introduction of each type of biometrics can be found in [1][2][3]. According to *Biometric Market Report 2003* [4], released by International Biometric Group (IBG), the total biometric revenue was \$523.9m in 2001 and is expected to reach \$1.9b by 2005. The report summarized the market shares of various biometric technologies in 2001 (see Figure 1-1).

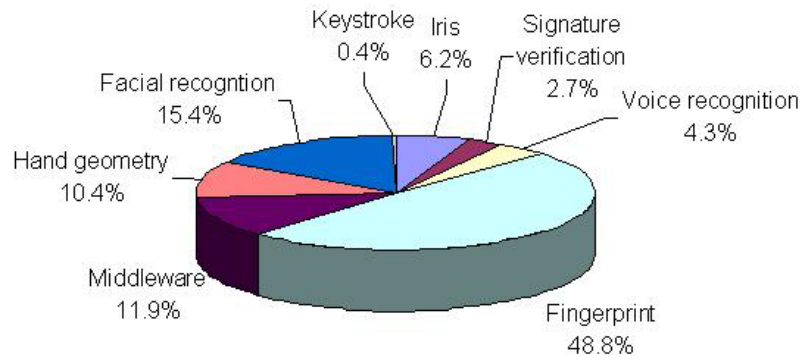


Figure 1-1: 2001 Market Shares By Biometric Technology

From Figure 1-1, it shows that fingerprint is by far the most widely adopted biometric technology and will remain very important in the coming future [4]. Facial recognition has replaced hand geometry and become the second to fingerprint in the market share. Middleware products, like smart cards, data storage servers etc, continue to grow as a critical technology [4]. One may notice that signature verification accounts for only 2.7% in the market share. However it has some distinguished advantages, which make it a suitable choice to use in our project. It will be illustrated in details in Section 3.2.

1.1.3 Pros and Cons of Biometrics

The main use of biometrics is, by far, replacing passwords and access cards in authentication. In general, there are three methods to authenticate an individual. They are through procession (what you have), knowledge (what you know) and biometrics (who you are), as shown in Figure 1-2.

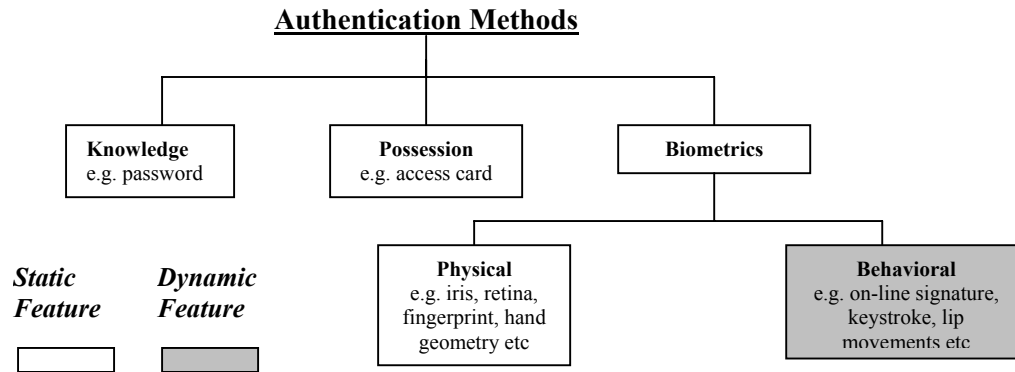


Figure 1-2: Authentication Methods - Knowledge, Possession and Biometrics

The difference between physical biometrics and behavioural biometrics is on whether the features extracted are static or dynamic. Behavioural biometrics, e.g. on-line signatures, requires the individual to be present and actively participate in the authenticating process. The features extracted are hence dynamic and not easily recorded.

As compared to traditional ways of authentication involving access cards, tokens, passwords or PIN numbers, biometrics has seductive advantages. Access cards may be lost or stolen. Passwords and PIN numbers may be forgotten or guessed. On the other hand, biometrics identifies a person's identity based on his/her physiological or behavioural characteristics. Those characteristics are inherent with the person. They are very unlikely to be lost or forgotten. Simply said, the use of biometrics is more convenient, secure and reliable than other ways [2].

However biometrics also has its drawbacks. Firstly, it is not foolproof. Though fingerprint, iris can achieve very low error rate, they are still not 100% foolproof and none of others are [3]. Secondly, there are no commonly accepted industrial standards

on biometrics system interface and data format yet [2]. Lack of standards hinders interoperability between systems from different vendors. Thirdly it is likely to compromise individual's privacy. Since the way biometrics works is to compare a biometric test sample with a stored biometric reference sample, people may regard the biometric storage as an invasion of privacy and reluctant to accept it [2].

The above explains the pros and cons of biometrics in general. A more specific comparison for different biometric technologies is given in Table 1-1. The table was compiled from [2] and [3]. Note that the public acceptance of dynamic (or on-line) signature is very high. This adds the advantage on the use of dynamic signature from social aspect.

Technology	Accuracy	Privacy invasiveness	Ease of use	Cost	Public Acceptance
Dynamic signature	Average	Very low	Average	Low	Very high
Fingerprint	High	High	Easy	Average	Average
Face recognition	Average	Average	Easy	Average	Average
Hand geometry	Average	Average	Average	Average	Average
Iris scan	Very high	Very high	Easy	High	Low
Retina scan	Very high	Very high	Average	High	Very low
Voice recognition	Low	Low	Easy	Low	High

Table 1-1: A Comparison for Different Biometric Technologies

1.1.4 Digital Signature

A digital signature is completely different from a handwritten signature [9]. It is a checksum which depends on all the bits of the transmitted e-document, and also on a secret (or private) key, but which can be checked without knowledge of the secret key

[16]. With digital signature, one would be able to authenticate the identity of the sender of a message or the signer of a document, and ensure that the original content of the message or document after being sent is not tampered with [16]. Digital signature has been with us since 1976, when Diffie and Hellman introduced the digital signature as an application of public key cryptography in their classic paper [11]. Since then, there has been a great deal of public interest shown in applying the idea to solve security problems.

In June 2000, the former US President Bill Clinton signed the Electronic Signatures in Global and National Commerce Act (E-Sign) into law [13]. Under the act, digital signatures are placed in the same legal category as pen-on-paper signatures, meaning individuals and businesses can now be legally bound to agreements verified over the web. This act induces a big boost in web transactions with the industry acceptance. Digital signature is increasingly accepted within US and in other parts of the world.

Beside public key cryptography, there are a number of techniques for generating digital signatures, e.g. symmetric cryptography, trapdoor, tamper-resistant modules [12]. Among these proposed implementations of digital signatures, public key cryptography is the most popularly used because of its better security [12]. When public key cryptography is adopted to implement digital signature, an individual has a pair of keys: a private key and a public key. The private key is for signing the documents to generate the digital signature while the public key is for verification. The signing and verification process is illustrated in Figure 1-3.

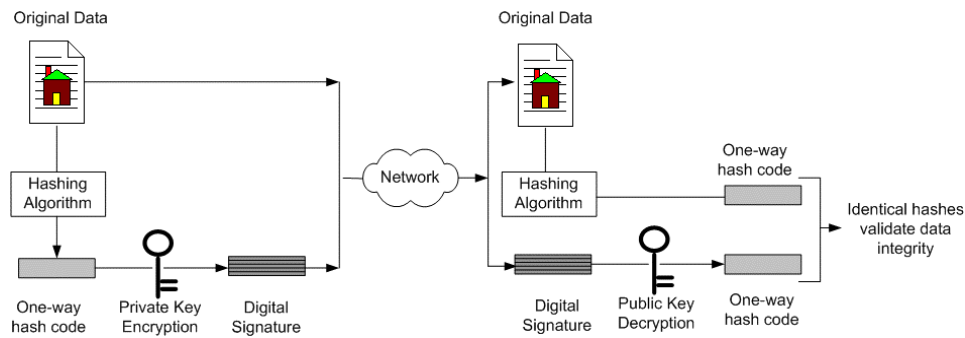


Figure 1-3: Digital Signature Signing and Verification

Because the public-key algorithms are often too inefficient to sign long documents, a one-way hashing algorithm is often implemented in digital signature generation to save time [16]. In Figure 1-3, a one-way hash code is firstly obtained by applying a hashing algorithm function, e.g. Secure Hashing Algorithm (SHA-1) [16]. Then the sender would encrypt the hash with his private key, thereby signing the document. Thus a digital signature is obtained and will be transmitted with the original data to the receiver. Upon getting the data and the digital signature, the receiver will use sender's public key to decrypt the signature and obtain the hash code. In parallel, the receiver obtains another hash code of the received data by following the same hashing algorithm (e.g. SHA-1). If the two hash codes are identical, the digital signature has been verified. The receiver can be convinced that the data are indeed originated from the sender and unchanged during transmission.

1.2 Motivation

1.2.1 Vulnerability of Private Key Storage

The most popular public key algorithms accepted by industries to implement digital signature are RSA (Rivest Shamir Adleman) and DSA (Digital Signature Algorithm)

[16]. For a private key, whose key-length is 1024-bit or above, it is considered safe for both RSA and DSA. The fact that it takes millions of years to hack the key thwarts most of the hackers [16].

However a study has shown that the vast majority of security failures are due to blunders in implementation and management, and essentially independent of the strength of the underlying cryptographic algorithms [5]. In particular, the private key storage is a vulnerable link in the security chain of digital signature applications [6]. Private keys are usually stored in a smart card or a disk and protected by passwords. The 1024-bit private key can stand against strong attacks [16]. But when it comes to storage, the security strength drastically reduces to a 6-to-8-character password. It is well known that password is an insecure way of authentication [2]. An individual may use the same passwords for his e-mail account, network logon, on-line banking, office access PIN, etc. When one is compromised, all passwords are compromised. In addition, he may write down the passwords on a piece of note, which could be peeped at. He may choose his alias or date of birth as passwords, which could be guessed by someone close to him. Hence a person authenticated to the access of a private key only means that he has the knowledge of the password but does not necessarily mean that he is the right person. In our thesis, we will solve this problem by incorporating biometrics for the storage of the private keys.

1.2.2 Synergy of Biometrics and PKI

The research works on biometrics are mainly for the purpose of authentication [2]. When complemented with Public Key Infrastructure (PKI), a straightforward way is to use biometrics to protect the private keys instead of passwords. For example, CIC

Corporation develops a system that uses signature dynamics in place of passwords to protect the stored private keys [8].

Though using biometrics to replace passwords is easy to implement and provides better security, the private key still needs to be physically stored in some mediums. The mediums, e.g. the disk or the smart card, still have to be carried with the person and the danger of these items being lost or stolen remains. In our project, we propose a different way of synergizing the two technologies, which overcomes the weakness mentioned above. Instead of using biometrics to protect the private keys, we propose to dynamically generate the private keys from biometrics. In this way, there is no need to stored the private keys in any physical mediums. Instead they are stored in a person's natural physiques or behaviors.

The concept of Generating Private Keys from Biometrics (GPKB) was first seen in Pawan's paper [7]. He proposed an idea to derive a private key from a biometric sample and used the private key to sign an e-document. The implementation of GPKB comprises two parts:

1. To obtain highly consistent biometric sample data
2. To derive a private key from the sample data

To date, there are no reports of achievements on the first part [7]. Thus no successful implementations of such application have been seen in the past literature. The difficulty for the first part is that all the bits in the biometric sample should be "exactly" correct. Pawan's paper [7] only addressed the second part. He gave a

conceptual example of using iris biometrics and presumed that a 256-byte iris sample had been obtained without a single bit error. Based on the sample, a private key could be derived by following some well-established public key algorithms, e.g. RSA or DSA. However when one's iris image is captured, it is extremely unlikely that every bit in the 256-byte sample is "correct". If it is, then it is most likely an attack [3].

Our research fills the gap in the first part. We propose a low-cost, reliable and feasible implementation based on on-line signatures, a common form of behavioral biometrics. There are reasons why we choose on-line signature and not other types of biometrics. The reasons will be explained in Section 3.2.

1.3 Project Overview: BioPKI Cryptosystem

BioPKI cryptosystem is the working system we propose in this project. It comprises three processing stages: shape matching, feature coding and private key generation. In the first stage, a check of the signature shape will be performed to filter out simple forgeries. In the second stage, the dynamic features of the on-line signature are used to generate an all-bits-correct data string. In the last stage, a private key will be derived from the all-bits-correct data string. Though our system is based on on-line handwritten signatures, the concept of the system design can be applied to other types of biometrics.

1.4 Summary of Contributions

In this research report, the followings will be the contributions of the author.

- Design and development of a novel and feasible BioPKI cryptosystem based on on-line handwritten signatures. The system introduces an innovative way to combine biometrics and PKI.
- Design of a new warping algorithm, Extreme Points Warping (EPW), to replace the conventional Dynamic Time Warping (DTW) algorithm. The new algorithm should be more suitable than DTW in the field of signature verification and yields better results.
- Design of a new coding scheme to obtain an all-bits-correct string from dynamic features of an on-line signature. The scheme can also be used for signature identification applications.

1.5 Organization of the Thesis

In the first chapter, we have introduced some background knowledge, explained the motivation of carrying out this project and provided an overview of the work done. A literature review of the research works and the techniques used in the related field is presented in Chapter 2. Some of the techniques will be applied to our project. Chapter 3 provides an overview of the system and explain the system design requirements. As mentioned before, the system will comprise three processing stages. Chapter 4 and 5 explain the first two stages of the BioPKI Cryptosystem. Details of the processing at the third stage have been covered in some papers (see [7]). It is not the emphasis of our research. However for completeness, a brief explanation of the third stage will be included in Section 5.5 in Chapter 5. Chapter 6 addresses the evaluation of system performance and the security issues. Finally Chapter 7 presents the conclusion and recommendations for future research.

Chapter 2 State-of-the-Art in Signature Verification

2.1 Overview

There is an old Chinese saying “signature reflects a person”. The way that a person signs his/her signature is unique. A person’s writing habit, literacy, even personal characters can be deduced from his signature. Because the handwritten signature is unique to a person, it has been adopted as a legal proof for the signed documents for thousands of years. The long history of practice makes handwritten signature one of the most non-invasive biometrics. Today it is still widely used in various fields, e.g. medical records, doctor prescriptions, receipt acknowledgement, legal contracts, banking agreements, credit card bills, government documents and official announcements.

Inspired by many practical applications in real life, the research in signature verification is very active. In recent years, with the advancement of hardware technology, the interest in this area has shifted from off-line to on-line signature verification. The classification of off-line and on-line is based on the data acquisition method. An off-line signature is a static signature image obtained from either a camera or a scanner. On the other hand, an on-line signature is usually obtained from a tablet. It contains not only the spatial information of a signature but also a rich set of dynamic information, e.g. pressure, speed, altitude and azimuth. Though off-line verification still has practical importance in some up-to-date applications, e.g. bank cheque automatic checking, the on-line verification is increasingly implemented in

many computer and web related applications. An important reason for the rising interest is that signature verifications based on on-line signatures have so far yielded superior results over off-line signatures [36].

For an on-line signature verification system, it not only checks what a signature looks like but also the process how it is generated. The working for such a system relies on the hypothesis that the production of a signature is a ballistic action, rather than a deliberate action [55]. It is called ballistic because when the brain sends commands through the nerve system to sign, the hand muscles execute the commands spontaneously with little visual feedback for confirmation. It is a learned, practiced and perfected action over many years. Hence, not surprisingly, it shows a high level of consistency in both shape and dynamics. Through not scientifically proven, it is an observed phenomenon and has been confirmed by many experiments results [14].

For the prior art in the area of on-line signature verification, there are several classic paper reviews that are frequently cited. In [25], Plamondon presented a comprehensive review of the state-of-the-art techniques in signature verification and writer identification before 1988. The paper [36] is a follow up to the previous paper [25]. It summarized the research activities from 1989 to 1993 and highlighted new advancements in automatic signature verification. In addition to these two, Nalwa's paper [38] provided an insightful explanation on the context of signature verification problem. However at the time of writing, no systematic reviews after 1994 are found in the past literature. Appendix A is the author's contribution, which presents a summary and a comparative review of various projects reported from 1994 to 2002. The scope of the work is limited to on-line signature verification only, without

covering off-line verification, on-line (or off-line) signature recognition and on-line (or off-line) signature identification. This is because the on-line signature verification techniques are closely relevant to the research work. In addition, the review is intent to cover only typical systems. For example, in a series of works done by Wan-Suck et al. to seek improvements [33][34][35], only the most recent one, i.e. [35], is included in Appendix A.

Plamondon summarized in [25] that an on-line signature verification system included a number of stages: data acquisition, preprocessing, feature extraction, comparison and performance evaluation. These stages are shown in Figure 2-1.

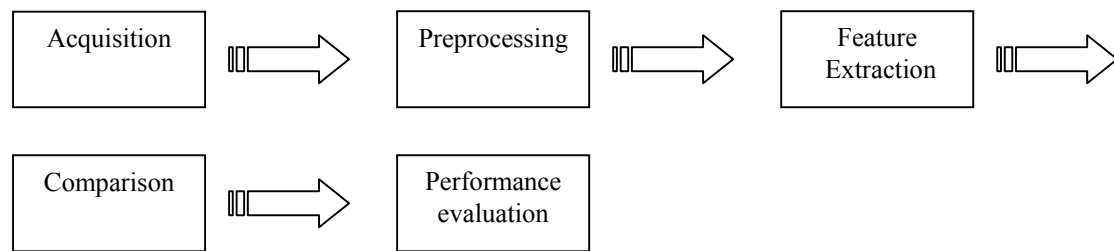


Figure 2-1: Five Stages in Signature Verification

The following sections in this chapter will be arranged in the sequence of these stages. In Appendix A, the techniques used in each stage, except the first stage, are summarized from the recently reported projects. The first stage, acquisition, is not included because it involves only the hardware set-ups.

2.2 Data Acquisition

Signature verification starts with data acquisition. However due to the lack of a standard database, most of the works listed in Appendix A required the researchers to

collect their own databases for performance evaluation. This makes direct comparison of the results very difficult, as it will be further explained in Section 2.6.

Unlike off-line signatures, on-line signatures are collected through a specialized hardware, i.e. a tablet. A typical tablet and the data extracted are shown in Figure 2-2.

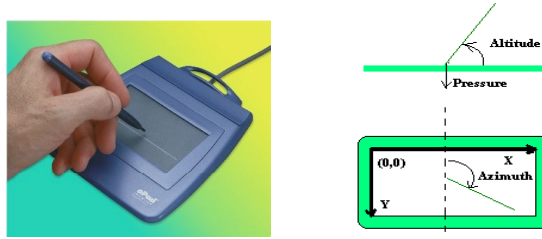


Figure 2-2: A Typical Tablet Hardware and Data Extracted

The tablet is able to capture timestamps, x , y coordinates, pressure, altitude and azimuth angles during signing (i.e. t , x , y , p , al , az). Apart from using commercial tablets, researchers in [30][31] used the self-built hardware, e.g. a SmartPen tablet. The SmartPen tablet is able to capture 3-D forces exerted at the tip of the pen.

2.3 Preprocessing

Signature data need to be preprocessed after the data acquisition. The preprocessing includes normalization, smoothing and re-sampling. In [25], Plamondon included segmentation as part of the preprocessing. However considering that the purpose of preprocessing is to trim data rather than process data, we include segmentation under the feature extraction phase in the thesis.

2.3.1 Normalization

The raw data obtained from the tablets need to be normalized firstly. The normalization can be explicit or implicit. The explicit approach performs translation,

size and orientation normalizations on raw data before feature extraction [25]. On the other hand, the implicit approach chooses those features, which are location, size or orientation invariant [25]. Translation is to re-assign the coordinate origin, e.g. to the signature centroid point. Size and orientation normalizations are to reduce the natural variations in size and orientation angle.

Among the three, orientation normalization is the most difficult one. The difficulty relies on finding a reliable reference angle. It is not easy to define the reference angle given the variant nature of the handwritten signature. In [47], all the computed look-up angles along the curve are normalized with respect to the first angle. We argue that the first angle is certainly not a reliable reference. Away from the difficulty of finding a reference angle, a simple yet reliable solution is adopted in [45][51] by drawing a horizontal reference line for users to self-adjust orientation. Thus it exempts the need of spending computing power for the orientation normalization. From the results, it seems that this method works well [45][51].

In addition to the three normalizations, some systems also perform duration normalization [29][37][48]. The duration of the signature writing is normalized to be the same for all the genuine samples. However it is not generally adopted because the variation of the signing duration is by itself an important feature of the signature.

2.3.2 Re-sampling

Two types of re-sampling are usually done in preprocessing: temporal re-sampling and spatial re-sampling. The first type usually involves uniformly re-sampling temporal signals, e.g. $x(t)$, $y(t)$, at equi-distant points by using interpolation. This is

needed because in most of the cases the tablet's sampling rate is not constant. On the contrary, Sakamoto [40] defined a *non*-uniform re-sampling technique in order to preserve sharp corners. The second type involves re-sampling signature curve at equidistant points. It is a common method used for describing the signature's static shape [35][37][38].

2.3.3 Smoothing

In [37][39], researchers applied spline-smoothing differentiation algorithm to compute velocity and acceleration. However the majority of projects adopt a simple and efficient method to compute the approximate values of velocity and acceleration [26][40][48][53][54]. The approximation is reasonably based on the closeness of the adjacent sample points, and there is not much difference from the observed results of these two methods [39].

2.4 Feature Extraction

2.4.1 Features from On-line Signature

During the preprocessing, the data are trimmed to remove some noise and variations. After trimming, the features are extracted from the signature. A typical commercial tablet, e.g. Wacom ArtZ II [32], is able to capture 5 sets of data along the time axis. They are x , y coordinates, pressure, altitude, azimuth or $x(t)$, $y(t)$, $p(t)$, $al(t)$ and $az(t)$ respectively. These data can be grouped into two types of features: static shape and dynamics. Static shape contains x , y trajectories only. Dynamics comprise the time information, pen angle and pressure - those transparent and dynamic features. These two types of features have somehow complementary roles in distinguishing the valid

signatures from the forgeries. It is because the more the forgers try to match every detail of a signature's shape, the less likely they are to match the dynamics. Hence it is suggested in [18] that a reliable dynamic signature verification system should consider shape as well as dynamics of the signature.

In signature verification, the features, either shape-related or dynamics-related, can be complete signals or derived parameters. The five basic sets of data themselves, i.e. $x(t)$, $y(t)$, $p(t)$, $al(t)$, $az(t)$, can be treated as five features. In addition, speed in either horizontal or vertical directions can be computed as new features. The look-up angle, computed from x and y , along the curve is a feature used in [47]. In recent research, Ma Mingming [26] applied Discrete Fourier Transform (DFT) to find the spectrums of x , y signals and used as features.

Besides using the complete signals as features, statistical features or parameters are frequently used. They are group into two types: local and global parameters [23]. As the name suggests, local parameters describe values at local points, e.g. maximum or minimum values of data signals, starting direction, ending speed etc. On the other hand, global parameters are computed globally, e.g. total time, number of strokes, means and standard deviations, number of zero crossings, etc.

2.4.2 Segmentation

Segmentation is part of the feature extraction process. It results in an improvement in performance. This has been reported in several papers [27][47][42][54]. The reason is that segmentation helps to extract more features and it facilitates the comparison of the two signatures based on the stable segments instead of the whole signatures. Wirtz

[42] used each natural writing stroke as a segment. Brault [49] evaluated curvatures along the length of a signature and found perceptually important points for segmentation. In [27], Schmidt searched for extreme points (i.e. peaks and valley) on the x, y signals and used those points for segmentation. In [33][34][35], segments are defined in such a way that each segment comprises fixed number of points. Recently Rhee [52] proposed a more robust segmentation technique called model-guided segmentation. The proposed method ensures the same number of segments, which easily enables segment-to-segment matching [52].

2.4.3 Feature Selection

With segmentation, more features can be defined. However the problem of selecting the right set of features is not trivial [25]. In general, three types of features can be defined, as shown in Figure 2-3. The feature selection undergoes two steps. Firstly a common set of parameters is selected from a pool of parameters. Secondly an optimum set of parameters for each individual, i.e. a personalized set, is selected from the common set.

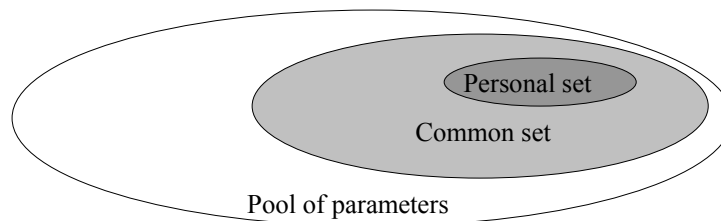


Figure 2-3: Three Types of Parameters in Feature Selection

There is a pool of parameters, which can be computed from signals in various ways. However not all the computed parameters are useful. Only those non-redundant and discriminating features are selected into the common set [41]. To select non-

redundant features, Bauer [41] performed correlation computation to remove redundant or correlated features. Kim [44] used the forward method to define a common set. He selected 23 features from 76. Several other methods, e.g. Karhunen-Loeve transform, neural nets and jackknife statistics are introduced in [45] to remove redundant features. To select discriminating features, the most common way is by evaluating the weighted Euclidean distance metric [45]. In a recent research, Ma Mingming [26] defined a Discriminating Power (DP) function. The function computes the DP value for each feature. Those features with largest DP values are to be included into the common set. Overall there seems no consensus on what is the best common set.

Among the common set of parameters, the importance of each parameter is different for each individual. Hence a personalized set comprising relatively more important parameters is needed. The set reflects the individuality of each user. Crane and Obsteam [24] first proposed to use personalized parameters for each user. The improvement in system performance is observed in the subsequent experiments [26][41][45][53]. The most common way to define a personalized set is through computing the Euclidean distance metric, as outlined in [26][41][45]. Neural Network is often applied for feature selection [25]. Wijesoma [53] proposed a new selection method based on Genetic Algorithms (GA). An initial set of features is encoded into a population of genes or chromosomes. Through a process of genetic evolution, an optimized subset of features is obtained for each individual. In a comparison to his previous work [26] using the Euclidean distance, it is found that GA selection algorithm yields an improvement of around 8% in terms of error rates.

In theory, the personalized sets of parameters obtained from the above methods are unlikely the optimum sets. The optimum set is hardly attainable practically. For example in [44], to select the optimum 23 features from 76, $1.7 * 10^{19}$ possibilities need to be evaluated. It would be very impractical to exhaustively try out every combination within a short time using a common PC. Hence usually only a suboptimal set can be achieved [26][41][45][53]. Aware of this problem, Kim [44] avoided using a personalized parameter set. Instead, he proposed to use a common set of parameters with personalized weights to reflect individuality of different users. He also tried the usual method to define a personalized feature set with the forward method. It turns out that the new method reduces Equal Error Rate (EER) from 5.5% to 4.28% [44].

2.5 Comparison

After the features extraction stage, the signature passes on to the comparison stage (see Figure 2-1). It compares the extracted features with the reference values stored in the template. The two common comparison methodologies in signature verification are: functional approach and parametric approach. In the functional approach, complete signals ($x(t)$, $y(t)$, $v(t)$, etc) directly or indirectly constitute the feature set. Signal values at a series of sampled points are compared point-to-point between the test signature and the reference signature. In the parametric approach, only the computed parametric features are compared [23]. In addition to these two approaches, Rhee [52] proposed an approach based on segment-to-segment comparison. For each segment, eleven parameters were extracted and compared [52]. It is noted that segmentation helps to extract more parameters and the comparison is still based on

the parameters. Hence we group the segment-to-segment comparison into the parametric approach in this review.

2.5.1 Functional Approach

Functional approach compares complete signals point-to-point. The complete signal can be either a spatial function, e.g. x , y along the curve [38][47], or a temporal function, e.g. x , y along the time axis [29][30][40][41][46]. To compare the two signals, a straightforward method is to use a linear correlation [23], but a direct computation of the correlation coefficient is not valid due to the following two problems:

1. Difference of overall signal durations
2. Existence of non-linear distortions within signals

For either spatially functional signal or temporally functional signal, it is unlikely that the signal duration is the same for different samples even from the same signer. In addition, for both signals, distortions occur non-linearly within the signal. To correct the distortion, a non-linear alignment needs to be performed before comparison. For a temporal signal, the most common method used is Dynamic Time Warping (DTW) [30][47]. To warp a spatial signal, Nalwa [38] used a method called Length Warping. However the two types of warping differ only in the notation, as they adopt the same approach based on the Dynamic Programming (DP) matching algorithm [38]. From a different viewpoint, we can regard the length axis of a spatial signal as the pseudo-time axis. For convenience, we will use DTW to refer to both types of warping for the rest of this thesis.

A detailed explanation of DTW will be presented in Section 4.3. The warped signal after DTW is obtained by following a defined warping path. The point-to-point Euclidean distance between the warped signal and the reference signal can be computed [29][40][46]. Alternatively the correlation coefficient can be computed, as a measure of their similarity [38].

It is evident that each point on the warped signal has different importance or weight. The more stable the value at this point is, the more important it will be. Hence by assigning different weight to each point based on its stability, the comparison will be more meaningful. An improvement in performance is reported in [27][38][46].

Besides the warped signal, the warping path can also be used as an important measure to differentiate the forgeries from the genuine signatures. The two linear-correlated signals will have a warping path of linear curve. The less linear-correlated for the two signals, the less linear the warping curve will be. Sato defined the motion measure based on the warping path [50]. The motion measure is also used in [27][30][41].

Besides the mainstream of using DTW for the functional approach, there are other warping methods. Wu [43] proposed a split-and-merge matching algorithm. Firstly the signal is split into 4 equal-length segments by 3 points. The best segment-to-segment matching between the sample and the reference is determined based on the Euclidean distance. If several segments of one signal match to one segment of the other signal, then the segments merge into one segment. The process continues to

further spit-and-merge the corresponding segments. The iteration occurs for pre-defined *depth* times. It turns out that *depth*=1 yields the optimum result [43].

Yang [47] proposed the use of Hidden Markov Model (HMM) in the functional approach. A reference HMM model is constructed from 8 prototype signatures. By defining a left-to-right transition model with 6 states, the probability that a sample signature is generated from the HMM reference model is computed. From the probability with respect to the defined threshold, the genuineness of the sample signature can be judged [47].

2.5.2 Parametric Approach

In distinguishing two sets of parameters, say one from a genuine signature and the other from a forgery, Euclidean distance is the most straightforward and commonly used measure [41][44][45][46][52]. First a reference set of parameters is defined from several prototype signatures. After the same set of parameters is extracted from a sample signature, then the authenticity of the test sample can be decided based on the Euclidean distance to the reference set.

Apart from using Euclidean distance, there are several other methodologies for parametric comparison. Kiran [51] proposed a probabilistic feature model to compute a score from a common set of ten features. The model fits a Gaussian density function to the values of each feature. To verify a test sample, a probability score (PS) is computed for each feature. The accumulation of ten probability scores will be compared to a threshold to make a decision. Dolfing [48] applied Left-to-Right

Hidden Markov Model in the parametric approach. The final decision is based on the log likelihood between a sample model and a reference model [48].

Applying Neural Network to the parametric comparison is proposed in [36]. It is also seen in [35][54] with quite impressive results (refer to Appendix A). Wijesoma [53] adopted a fuzzy logic for parametric comparison. The same logic is also used in [26]. The fuzzy logic resembles the way our brain works toward verifying a signature. The computation is based on the Degree Of Authenticity (DOA) [26][53] rather than the Boolean logic (“true” or “false”, 1 or 0) on which the modern computer is based. A number of partial authenticities, i.e. DOA, for each feature are aggregated. If the aggregation exceeds a threshold, then the signature is judged as a genuine one otherwise a forgery.

2.6 Performance Evaluation

After comparing the features, error rates are the indicators of the system performance. Two types of error rates are often used. They are False Rejection Ratio (FRR) and False Acceptance Ratio (FAR). The two curves vary with respect to the threshold settings, as shown in Figure 2-4.

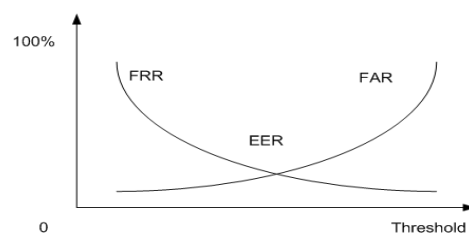


Figure 2-4: FRR and FAR Curves

Equal Error Rate (EER) is commonly used as the indicator of the error rate performance in comparison with other systems. It is the point where the FRR curve intersects with the FAR curve (see Figure 2-4). The EER values for recently proposed systems are tabulated in Appendix A.

However a direct comparison of results among different systems is of little value due to a lack of standardized database [38]. Researchers build their own database to evaluate the system performance (refer to Appendix A). The qualities of collected signatures are likely to differ due to different hardware set-up, the mood of users when signing, racial group and education background of users. For example, in [48] and [52], tablets with LCD display on board were used. It is expected to produce more “genuine” signatures than those from blind signing on normal tablets. Normally voluntary users are encouraged to mimic each other to produce forgeries, e.g. in [43][47]. In some researchers, cash awards are offered to hire forgers to produce skilled forgeries [26][53]. A different type of forgery is seen in [30][56], where other users’ genuine signatures are used as forgeries for a particular user. The forgeries are called random forgeries or zero-effort forgeries [25]. The different types of forgeries make the comparison of the system performance very difficult.

In despite of the database problem, we are able to evaluate the effectiveness of the system and the newly proposed techniques. Comparative analysis is a very important method to evaluate the performance [20]. A system or a new technique can be compared with others based on the same set of database with all other conditions held the same. This method is adopted in [44][52][53].

2.7 Summary

In this chapter, the five stages in a signature verification system are described: data acquisition, preprocessing, feature extraction, comparison and performance evaluation. We have reviewed a range of techniques used in each of the stages. They are mainly summarized from research works in the recent ten years. The main purpose of this chapter is not to identify problems in this area, but to review the useful techniques that can be applied to our research problem, i.e. generating private keys from signatures. However we are able to make contribution to this field by proposing a new warping algorithm for the functional approach at the comparison stage. The new warping algorithm is called Extreme Points Warping, which will be explained in details in Chapter 4. In the next Chapter, we will introduce the BioPKI cryptosystem and its features.

Chapter 3 BioPKI Cryptosystem

3.1 Overview

The BioPKI cryptosystem is the solution we propose to address the problem of the private key storage. It is implemented based on on-line signatures. We will explain the reasons why choose on-line signatures instead of other types of biometrics in Section 3.2. The system comprises three processing stages: shape matching, feature coding, and private key generation. We will explain the features and functions at each of the stages in details. In addition, issues like data acquisition, database collection and data preprocessing will also be introduced in this chapter.

3.2 Why Choose On-line Signature

One may ask why on-line signature is chosen instead of other types of biometrics. There are three main reasons for that. Firstly, handwritten signature is more acceptable by the public. It has been adopted as a legal proof for paper documents in business transactions for thousands of years. A natural transformation in the new e-commerce environment is to use on-line signatures to sign e-documents.

Secondly, as compared to other types of biometrics (e.g. iris, fingerprint), it is relatively easier to revoke a written signature once it is compromised. A person can easily abandon the compromised signature by changing a different signing style.

However it would be infeasible to abandon the fingerprint or iris, since they are inherent with the person.

Thirdly, it is more secure to use on-line signatures than other types of biometrics. Although fingerprint (or iris) is currently more widely used than handwritten signature (see Section 1.1.2), it is static-image based. It is possible to obtain one's fingerprint (or iris) sample even without the person's awareness. For example, a person would leave fingerprints on the keyboard while typing on it. A hidden camera is able to capture a person's iris image even one meter away [3]. With the fingerprint or iris image, an attacker knows the biometric secret of a person and is able to derive an authentic private key. In our proposed cryptosystem, the private key generation is based on the dynamic properties of a person's on-line handwritten signature. Those dynamic properties are unique-to-person, consistent and transparent. They have to be captured through a specialized hardware, i.e. the tablet. In addition, it requires the signer's intent and commitment to provide genuine signatures. Hence it is much more difficult for an attacker to "cache" the dynamic properties of a signature *without the person's awareness*. Nevertheless an attacker (e.g. a Mafia-shop owner) can still possibly deceive a user to sign on a board and cache the "hidden" dynamics. But it would be much more difficult than obtaining a fingerprint (or iris) image. It is at the user's discretion to supply his on-line signature only in a safe environment. The same argument also applies for the usage of credit card, cash card, and ATM card.

3.3 An Overview of BioPKI Cryptosystem

We propose a BioPKI cryptosystem with private key generation from the dynamic properties of on-line signature. The cryptosystem merges the merits of both biometrics and Public Key Infrastructure (PKI). Figure 3-1 shows a block diagram of the proposed BioPKI cryptosystem. This cryptosystem consists of three stages: shape matching, feature coding, and private key generation.

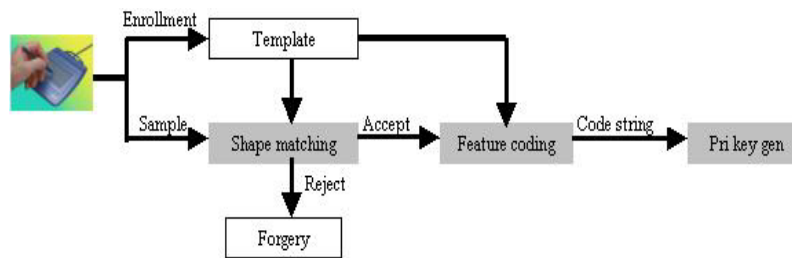


Figure 3-1: A Block Diagram of the BioPKI Cryptosystem

3.3.1 First Processing Stage: Shape Matching

The Shape Matching Stage is to filter out the simple forgeries by examining the shape of a test sample. Firstly the characteristic functions describing the signature shape are defined. Subsequently these functional signals are compared with the reference signals. A decision – genuine or forged - will be made based on the similarity between the two signals. Because of natural variations of the signature shape, a warping process need to be performed before comparison. It is to remove the positional variations of the shape function. The commonly adopted technique is Dynamic Time Warping (DTW). A detailed explanation of DTW will be presented in Chapter 4. Some problems associated with the application of DTW in signature verification will be identified and analysed. To address those problems, a new warping technique – Extreme Time Warping (ETW) – will be proposed and implemented at this stage. One may refer to Chapter 4 for more details about this processing stage.

3.3.2 Second Processing Stage: Feature Coding

The Feature Coding Stage is to obtain an *all-bits-correct* string from a set of dynamic features. Firstly a list of features is defined. Those features are unique-to-person, consistent and transparent. Because of the natural variation of human behaviour, it is impossible for a person to sign twice with identical dynamic features. At this stage, we will propose a new coding scheme to perform feature coding. Under the scheme, a feature code is computed from each of the dynamic features. Then all the feature codes are concatenated to form a code string. Despite the variations of feature values, an all-bits-correct string can be obtained. One may refer to Chapter 5 for more detailed explanations about this stage.

3.3.3 Third Processing Stage: Private Key Generation

An all-bits-correct string is obtained from the feature coding stage. Finally the Private Key Generation Stage takes the code string as the input and generates the individual's private key. The operation follows the well-established public key algorithms, e.g. DSA and RSA. In Section 5.5, we will use DSA as an example to demonstrate how the private key is generated.

3.3.4 The Role of the Template

As shown in Figure 3-1, the template is involved in the first two processing stages. It stores a reference shape, which will be retrieved to compare with a test sample. In addition, it stores information as how the feature coding will be done. One of our goals in designing the three processing stage is to diminish the sensitivity of the template. In other words, the private key can't be deduced from the template alone.

The template could be a target of attack. Here we can't presume that the template is always safe. Under some circumstances, it may be compromised, e.g. it is stolen or revealed. In that case the private key, should *still* remain secret. For this reason, we only write the static shape into the template. The shape will be used as the reference in the shape matching stage. The private key will only be derived from the dynamic features of a signature after the feature coding stage. The template stores only the guiding information as how feature coding is done for each dynamic feature. But the values of the dynamic features will not be contained or even deduced from the template. In Chapter 5 we will explain how it can be achieved by using our proposed coding scheme. To sum up, in the proposed BioPKI cryptosystem, templates are useless if they are stolen. An authentic private key would only be derived from a combination of the data from the template and a live signature sample.

3.4 Two Phases of Operation

The BioPKI cryptosystem has two phases of operation: enrolment and testing (see Figure 3-2). During the enrolment phase, a pair of keys: a private key K_1 and a public key K_2 , are derived from a reference signature. The private key is then discarded while the public key is kept. During the testing phase, the person provides a written test sample. After being processed by the three stages, another pair of keys - K_1' and K_2' , is generated. If $K_1 = K_1'$, the generated private key is authentic and will be used to digitally sign an e-document. Otherwise it will be considered invalid and rejected.

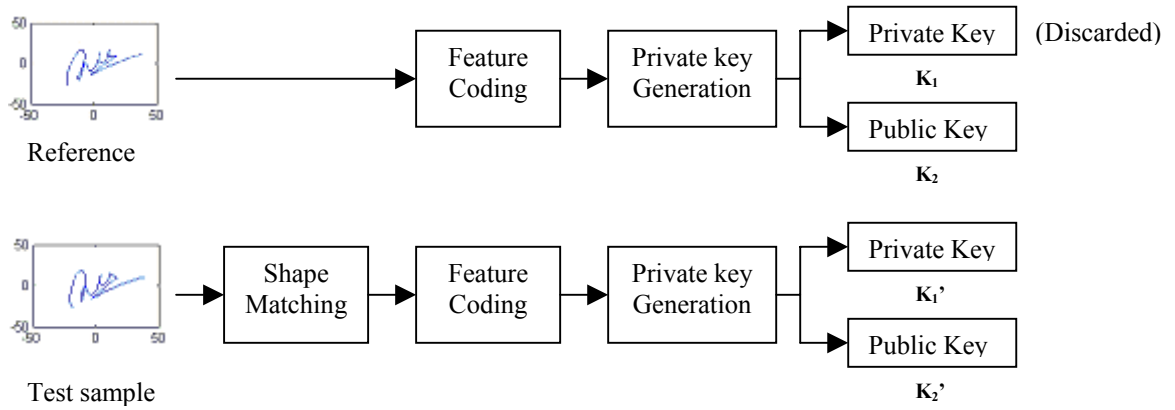


Figure 3-2: Two Operating Phases - Enrolment and Testing

Since K_1 is discarded, we can't directly compare K_1 and K_1' . However we can determine whether $K_1 = K_1'$ through some other ways. In one way, the generated private key K_1' is used to encrypt a pseudo-random message M . If the encrypted message can be decrypted successfully by the public key K_2 , namely $D_{K_2}(E_{K_1'}(M))=M$, the key K_1' is authentic. In another way, the two public keys K_2 and K_2' are compared. If $K_2 = K_2'$, then the corresponding private keys are also equal, i.e. $K_1 = K_1'$. The mathematical relationship between the public and the private key can be found in [16].

3.5 Security Aspects of the BioPKI Cryptosystem

The possible attacks of the system are mainly from two aspects: forging and hacking. An attacker mimics the genuine person's signature and provides a forged signature to deceive the system. A professional forger is likely to forge the signatures well enough to gain entry into the system. In this project, participants are asked to mimic each other's signatures. The performance of the system against forging will be measured in

terms of False Acceptance Ratio (FAR) and False Rejection Ratio (FRR), as outlined in Section 6.2.

Another form of attack is through hacking. Since the private key is derived from a code string (refer to Section 3.3.2), a brute-force attack may be launched to search for the code string bit by bit. The effort of the brute-force attack is exponentially proportional to the bit-length of the code string (see [16]). Hence the security strength against hacking will be evaluated in terms of the average bit-length of the code strings, which will be presented in Section 6.2.

3.6 Data Acquisition

We use a Wacom ArtZ II tablet [32] to capture the written signature. It has an active area of 8 x 6 inch² with an average sampling rate of 50 Hz. The hand-signature capture program was written in Java to obtain the data from the tablet and to build up the database.

As shown in Figure 3-3, as the user signs within the Java frame, six sets of data are captured in the background and displayed on the DOS console window. During the actual signing process, the console window is turned off. The data captured are timestamp, x coordinate, y coordinate, pressure, altitude and azimuth. A detailed explanation on the Hand-signature capture program can be found in [10].

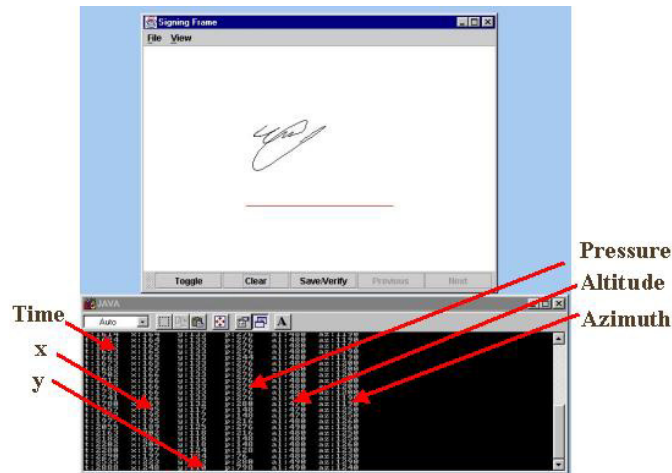


Figure 3-3: Six Sets of Data Captured by a Java Program

For the data captured, the pressure is expressed as the quantised pressure levels, ranging from 0 to 1024. For the altitude the range is from 26° to 90° , and for the azimuth is from 0° to 359° . The definitions of altitude and azimuth angles can be referred to Figure 2-2 in Chapter 2.

3.7 Database Collection

A signature database has been built, which comprises 25 users. For each user, the signatures are collected at two phases at one-month interval. The one-month interval is to take into account the short-term effect of signature evolution. Investigation on the long-term effect, which takes for years, is beyond our scope of study. During Phase I, each user signs 20 signatures. The first 10 samples are used to derive the user's template. The next 10 samples are stored as the authentic samples. During Phase II, each user provides another 10 authentic samples and 10 forged signatures. The collection activity is summarized in Table 3-1.

Phase I		Phase II	
1. Sep ~ 30. Sep 2001(Per user)		1. Nov ~ 30. Nov 2001(Per user)	
10 template samples	10 authentic samples	10 authentic samples	10 forgeries

Table 3-1: A Summary of Database Collection Activity

For the collection of the forgeries, we encourage participants to mimic each other's signature as closely as possible. The forger is allowed to view the static images of all the authentic samples and practise for several minutes before providing the forgeries. A summary of the signatures in the database is tabulated in Table 3-2. Overall 1,000 signatures have been collected and stored in the database. Some signature examples can be referred at Appendix B.

No of users	During enrolment	Genuine samples	Forgeries
25	10	20	10
<i>Total</i>	<i>250</i>	<i>500</i>	<i>250</i>

Table 3-2: A Summary of Signatures in the Database

3.8 Preprocessing

3.8.1 Normalization

As shown in Figure 3-3, a horizontal baseline is drawn in the signing area to assist users to self-adjust the orientation during signing. By studying users' signing behaviour during database collection, we notice that in most cases the users can orient their signatures consistently with reference to the baseline. However the location and

size are beyond users' control. In general, the user signs with no prior-knowledge where and what size they signed the previous signatures. Hence we apply the location and size normalizations, while the orientation normalization is not necessary in the presence of a baseline.

We assign the coordinate origin (0, 0) to the signature's centroid point. The new coordinates after the location normalization, or translation, can be obtained as:

$$x(j) = x(i) - 1/N \sum x(i) \quad (3.1)$$

$$y(j) = y(i) - 1/N \sum y(i) \quad (3.2)$$

where i ($i=1,2,\dots,N$) is the serial No before translation and j ($j=1,2, \dots,N$) is the serial No after translation. Subsequently the size normalization will be performed. The following formulas are found to be suitable for size normalization.

$$x(k) = x(j) * \sum |X(j)| / \sum |x(j)| \quad (3.3)$$

$$y(k) = y(j) * \sum |Y(j)| / \sum |y(j)| \quad (3.4)$$

where j ($j=1,2,\dots,N$) is the serial No before the size normalization while k ($k=1,2, \dots,N$) is the serial No after the size normalization. The $X(j)$ and $Y(j)$ are coordinates for the reference signature.

3.8.2 Re-sampling Shape

After normalization, the x , y signals are obtained by re-sampling the shape at equi-arc-length intervals. Mohankrishnan [35] conducted a power spectral analysis of the

signatures and concluded that 256 samples points were sufficient to represent all salient curve characteristics. Hence we choose 256 points for re-sampling in the project. Figure 3-4 (a) and (b) display the signature shape before and after re-sampling, respectively.

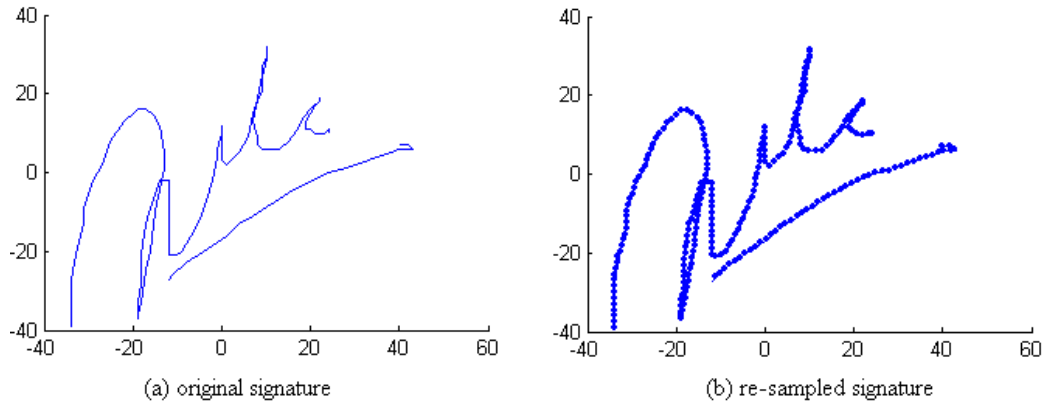


Figure 3-4: Pre- and Post- Re-sampling at Equal Distance

3.8.3 Speed Computation

Speed, either horizontal or vertical, is an important feature in signature verification [25]. It is defined as the first derivative of displacement. Since the inter-point distance is quite small, the velocities can be calculated as the followings:

$$V_{x(k)} = (x_{(k+1)} - x_{(k)}) / (t_{(k+1)} - t_{(k)}) \text{ pix/ms} \quad (3.5)$$

$$V_{y(k)} = (y_{(k+1)} - y_{(k)}) / (t_{(k+1)} - t_{(k)}) \text{ pix/ms} \quad (3.6)$$

To facilitate computation, we re-sample the speed signals in time domain at equi-time-distance of 20 ms, i.e. $t_{(k+1)} - t_{(k)} = 20 \text{ ms}$. Hence the formulas in (3.5) and (3.6) can be simplified as:

$$V_{x(k)} = (x_{(k+1)} - x_{(k)}) / 20 \text{ pix/ms} \quad (3.7)$$

$$V_{y(k)} = (y_{(k+1)} - y_{(k)}) / 20 \text{ pix/ms} \quad (3.8)$$

3.9 Template Generation

During the enrolment, a user provides ten samples. Among the ten, one sample is selected as the reference signature and its static information (i.e. the signature shape) will be written into the template. The selection process goes in this way. Firstly, we take a sample as the reference. Then we warp the x, y data ($[x(k), y(k)], k=1, 2 \dots N$) of the rest nine samples against the reference though Dynamic Time Warping (DTW). A warping cost will be obtained for each DTW operation (one may refer to Section 4.3 for a detailed description of the DTW operation). Hence a summation of the nine warping costs will be obtained. As the procedure repeats for each of the ten samples, the sample with the minimum summation of the warping costs will be selected as the reference signature and written into the template.

3.10 Summary

This chapter gives an overview of the BioPKI cryptosystem we propose in the research. This cryptosystem comprises three processing stages: shape matching, feature coding and private key generation. In this chapter we have explained matters on system design and set-up. The chapter includes defining the function of each stage, introducing the hand-signature capturing software, data collection and pre-processing. The detailed explanations on each of the three stages will be presented in the following chapters.

Chapter 4 Shape Matching Stage

4.1 Overview

Shape matching is the first processing stage in the proposed BioPKI cryptosystem. It examines the static shapes of the input sample signatures and rules out simple forgeries. To compare the shapes of two signatures, the two common methodologies are: functional approach and parametric approach. The former uses the complete signals (e.g. $x(t)$, $y(t)$) as the features while the later uses the parameters abstracted from the complete signals. We will adopt the functional approach at this stage as the functional approach usually results in better error rate performance than the parametric approach [25].

In this chapter, we will first introduce several characteristic functional signals, which describe the signature shape. As a nature of human signing behaviour, deviations of shape positions are common. Hence the functional signal needs to be warped before comparison. From a survey of the recent researches in the field (see Appendix A), the mainstream warping technique used is Dynamic Time Warping (DTW) [57]. DTW is to apply Dynamic Programming (DP) matching algorithm to non-linearly warp a discrete signal with respect to a template [57]. To address some of the limitations of DTW, we will propose a new warping technique, named as Extreme Points Warping (EPW). A detailed description of the EPW algorithm will be introduced in Section 4.4. Instead of warping every point on the signal as DTW does, EPW warps selective extreme points (EPs), i.e. the peak points and the valley points. The new technique

results in better performance than DTW in terms of error rate and speed. A comparative evaluation between EPW and DTW will be presented in Section 4.5.

4.2 Characteristic Functions of Shape

A robust characteristic function of signature should be insensitive to intra-signer difference while sensitive to inter-signer difference [38]. To describe the signature shape, the x, y trajectories along the curve are often used as characteristic functions for comparison [27][29][47][43]. Nalwa [38] proposed the use of *torque* as a robust characteristic function to describe the shape. Besides torque, Center of Mass (CoM) functions were also introduced in [38] to describe the shape. In our research, we will comparatively evaluate the five signals: x, y, CoM of x, CoM of y and torque.

4.2.1 Center of Mass (CoM)

A Gaussian window, defined in [38], is used to compute the CoM. The window function, ignoring a scale factor, is given in [38] as:

$$G(\lambda)=\exp(-\lambda^2/2\alpha^2) \quad (4.1)$$

Where the ‘ α ’ is a constant. The Gaussian window is a bell-shaped curve, as shown in Figure 4-1 (a). Figure 4-1 (b) shows how the CoM is computed, as the Gaussian window slides along the signature curve.

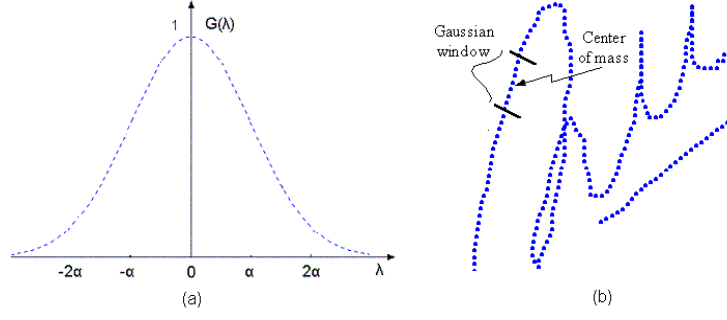


Figure 4-1: The Sliding Gaussian Window

The computation of the CoM is based on the discrete points as shown in Figure 4-1 (b). However a discrete Gaussian function expression is not explicitly given in [38]. We will derive the Gaussian window function in its discrete form in the followings. We let $n = \lambda$ and $m = 2\alpha$, then the Gaussian window function in equation (4.1) can be expressed in its discrete form:

$$G(n) = \exp(-2n^2 / m^2) \quad (4.2)$$

where m is an integer variable, indicating the window width. Nalwa [38] defined the *Gaussian window width* as the length of the signature curve, which the sliding window spans over. The window width in our case will be $2mL$, where L is the inter-point distance on the signature curve. Finally we normalize the Gaussian function to have a unit integral over the width of the window. We obtain the normalized window function as:

$$g(n) = \frac{\exp(-2n^2 / m^2)}{\sum_{k=-m}^m \exp(-2k^2 / m^2)} \quad (4.3)$$

where n is within $[-m, m]$. Outside this range, $g(n)=0$. With regard to the window's weight, a window with the Gaussian-weight curve is better than the one with a uniform-weight curve [38]. This is because the Gaussian window well serves the goal of gradually phasing in and phasing out the center of attention along the length of the signature as we slide the window along the curve [38].

The purpose of using a Gaussian window sliding along the signature curve is to smooth the signal noise. The broader the width, the more net effect of noise is suppressed. However at the same time more characteristic variations are also smoothed out. A suitable window width is found to be $8L$ through experiments and it is used in our project. A plot of the normalized Gaussian window with the width $8L$ is shown in Figure 4-2.

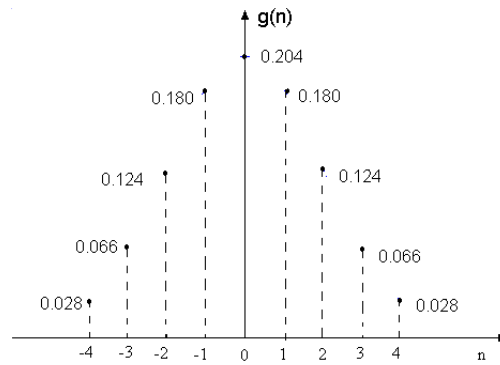


Figure 4-2: A Normalized Gaussian Window with Width $8L$

We assume the signature has unit mass per unit length. Thus for the small segment confined by the Gaussian window, the coordinates of the Center of Mass (CoM) are defined as:

$$\bar{x}(k) = \sum_{n=-m}^m g(n)x(k+n) \quad (4.4)$$

$$\bar{y}(k) = \sum_{n=-m}^m g(n)y(k+n) \quad (4.5)$$

where k is the serial No of the point. The definitions of (4.4) and (4.5) are the same as those defined in [38] except that they are now expressed in the discrete forms. One may refer to [38] for more explanations of CoM.

4.2.2 Torque

The torque is computed within a sliding coordinate frame in [38]. Figure 4-3 shows how a sliding coordinate frame is defined.

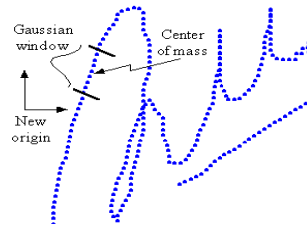


Figure 4-3: A Sliding Coordinate Frame

When the Gaussian window spans a section of the curve, the CoM can be computed in the way introduced in Section 4.2.1. After locating the CoM, as shown in Figure 4-3, the point with the position coordinate $(-x_0, -y_0)$ relative to the CoM will be defined as the origin of the new coordinate frame. The coordinates of every point on the signature will be translated with respect to the new origin. In our project, we choose $x_0=y_0=1/25$ of the signature curve length.

Within a coordinate frame, the torque \mathbf{T} exerted by a vector \mathbf{v} , which is located at position \mathbf{p} , the torque with respect to the origin is $\mathbf{T} = \mathbf{v} \times \mathbf{p}$. Nalwa has come out with an expression of torque in [38] as:

$$T(l) = \int_{-L}^{+L} g(\lambda) (y(l+\lambda)dx(l+\lambda) - x(l+\lambda)dy(l+\lambda)) \quad (4.6)$$

where $T(l)$, $x(l)$, $y(l)$ are continuous functions along the curve and $g(\lambda)$ is also continuous in the span of $\pm L$. Again we rewrite equation (4.6) in the discrete form expression.

$$T(n) = \sum_{k=-m}^{m-1} g(k) (y(n+k)(x(n+k+1) - x(n+k)) - x(n+k)(y(n+k+1) - y(n+k))) \quad (4.7)$$

Torque is a robust characteristic function to describe the shape and it is more indicative of forgeries as separated from genuine signatures. This is demonstrated in Figure 4-4.

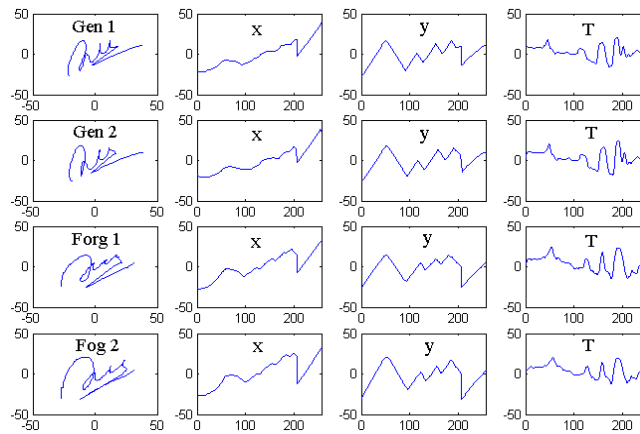


Figure 4-4: X, Y and Torque Signals

In Figure 4-4, the top two rows display the x, y and torque signals for two genuine signatures, while the bottom two rows are for two forgeries. It is noted that the torque signal is more distinguishable between the genuine signatures and the forgeries as compared to the x, y signals. Detailed explanations on the definition of the torque can be found in [38].

4.3 Dynamic Time Warping

4.3.1 Introduction to DTW Algorithm

DTW (Dynamic Time Warping) has been extensively used in speech recognition [57]. In the past decade, it has also become a major technique in signature verification to warp functional signals [38][40]. DTW applies the DP (Dynamic Programming) technique to find the best-matching path, in terms of the least global cost, between an input and a template [57]. Figure 4-5 shows the time alignment path during the DTW process.

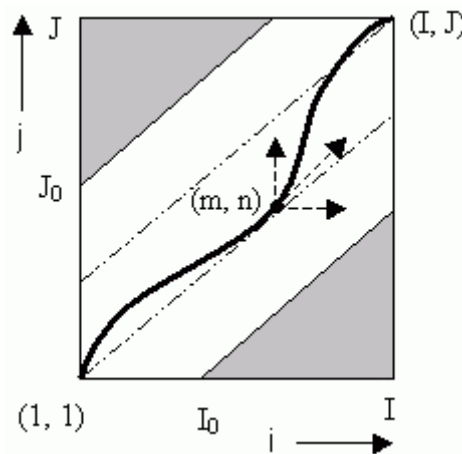


Figure 4-5: The Time-Alignment Path in DTW

In Figure 4-5, the points 1, 2 ... m ... I are from the template while the points 1, 2, ... n ... J are from the input. We define $d(m, n)$ as the local distance between the m^{th} template point and the n^{th} input point. It is usually computed as the Euclidean distance between the two points [62]. We define $D(m, n)$ as the global distance, which is the partial summation of the local distances from (1, 1) to (m, n). The warping process using DTW (refer to Figure 4-5), can be explained in the followings:

1) Initial condition

$$D(1,1) = d(1,1) \quad (4.8)$$

2) Monotonicity and continuity conditions

$$0 \leq i_k - i_{k-1} \leq 1, 0 \leq j_k - j_{k-1} \leq 1 \quad (4.9)$$

3) Boundary condition

$$i=1, j=1, i=I, j=I-I_0, j=I+J_0 \quad (4.10)$$

4) Global cost calculation

$$D(i, j) = \min \begin{bmatrix} D(i, j-1) + d(i, j) \\ D(i-1, j-1) + 2d(i, j) \\ D(i-1, j) + d(i, j) \end{bmatrix} \quad (4.11)$$

The warping process stops when $i=I$ and $j=J$. As a result, the warping cost is obtained by $S = D(I, J) / I$. For more details about the DTW process, one may refer to [57].

4.3.2 Application of DTW

In signature verification, researchers apply DTW to align a waveform from a test sample with the respective reference one [40][46]. The warping process is non-linear and changes the sample waveform in two aspects:

1. The end of the test waveform will be aligned with that of the reference one.
2. Peaks and valleys will be shifted to align with those of the reference one.

When the x, y signals are captured from a tablet, the duration of the signals is unlikely the same for different signings. Figure 4-6 shows the signal waveforms from two different signings, where x, y data are sampled from the tablet every 20 ms.

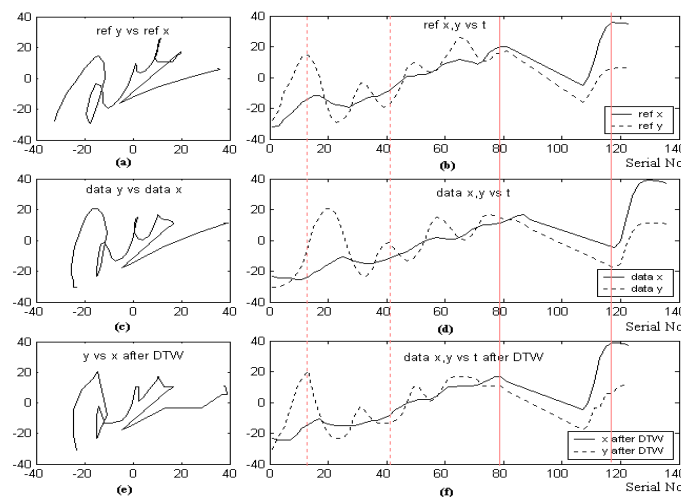


Figure 4-6: Waveforms before and after DTW

In Figure 4-6, the top two graphs (a, b) are drawn from the reference signature. The middle two graphs (c, d) are from the sample signature before DTW, while the bottom two graphs (e, f) are from the DTW-warped sample signature. Both x and y are independently warped through DTW. The graphs on the left panel (a, c, e) show signatures in x - y coordinates while the graphs on the right panel (b, d, f) show the x, y data along the point serial No. From graphs (b), (d) and (f), one may notice that the peaks and valleys of the sample waveforms are shifted to align with those of the reference waveforms. Some of such shifts of peaks and valleys have been highlighted in graphs (b), (d) and (f) of Figure 4-6.

4.3.3 Problems with DTW

DTW was originally used in speech recognition in the 1980s [58]. Since it was applied into the field of signature verification, few adaptations have been made [30]. DTW has two main drawbacks when applied in signature verification, including: i) heavy computational load, ii) warping on forgeries.

The former drawback is a known problem in speech recognition. DTW performs non-linear warping on the signal. As compared to linear warping, the computational load of DTW is heavy and the execution time is relatively long [58]. The execution time is proportional to the square of the signal size (refer to Figure 4-5). There are some measures to deal with this problem. One is by defining boundaries to reduce computations as in equation (4.10). With the definition of boundaries, computations outside the boundaries (i.e. in the shaded areas in Figure 4-5) can be skipped. However the resultant computation time is still relatively long. It takes on average around 0.4s, as we will explain it in Section 4.5.4. Other measures include adopting the parallel processing algorithm and specially designed DSP hardware as to optimise the warping computation [20].

The second drawback, however, is not well documented in the past literature. When used in speech recognition, DTW searches a best way to trim the voice signal to be more recognizable. However in signature verification, in the presence of forgeries, forgery signals also undergo DTW to be trimmed so as to become more “authentic”. Hence some adaptations of the algorithm in the field of signature verification need to be made. The problem can be implicitly addressed by the use of a motion measure

defined by Sato in [50]. The motion measure accounts for the deviations of the warping path. If the warping path is straight, it means few trimming changes are done to the signal, and hence the signal is more likely to be authentic [50]. That is to say, both the warped output and the warping path need to be involved in the comparison. Sato defined the former as the shape measure and the later as the motion measure [50]. However the inclusion of the motion measures adds to the complexity of data classification and decision-making, and it is not used in many recent researches [38][40][46].

In our project, we will introduce a simple solution, without using the motion measure, to adequately address this problem. Considering the fact that the DTW process warps every point on the signal, we propose a new warping technique to warp only selective important points on the signal. Section 4.4 will give a detailed explanation of the new technique.

4.4 A New Matching Technique : Extreme Points Warping

During the DTW process, every point is involved in the matching process [57]. The proposed new matching technique is called Extreme Points Warping (EPW). As the name suggests, the new technique warps only the extreme points (EPs) instead of the whole signal. The EPW warping process comprises three steps: EPs marking, EPs matching and segment warping. Firstly, the EPs, i.e. the peaks and the valleys, are marked on the signal. Secondly, the Dynamic Programming technique is applied to non-linearly align those EPs on the sample signal to the EPs on the reference signal. Lastly, segments between the consecutive EPs will be warped linearly.

4.4.1 EPs Marking

Along a signal, the peaks and valleys can be identified alternating. Here we define the Extreme Points (EPs) as those important peaks and valleys, precluding the small ripples. An example of ripples, peaks and valleys is shown in Figure 4-7. A ripple is formed when a peak and a nearby valley are close to each other. Small ripples are not considered as the EPs, because they are unreliable most of the time. In our matching algorithm, too many ripples will increase the chance of error matching, as we will explain in Section 4.4.4.

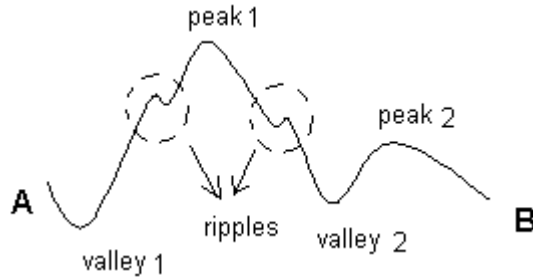


Figure 4-7: A Demonstration of Extreme Points and Ripples

In the following, we will define an EPs marking algorithm to identify the EPs. We first define a rise-distance, denoted by ‘ r ’, as the amplitude from a valley to the following peak. In addition, we define a drop-distance, denoted by ‘ d ’, as the amplitude from a peak to the following valley. For any peak (or valley), a rise-distance can be computed at one side of the curve while a drop-distance can be computed at the other side of the curve. A peak or valley is marked as an EP only if the rise-distance and the drop-distance are both larger than a defined threshold, h_0 .

$$r \geq h_0, d \geq h_0 \quad (4-12)$$

Choosing a large value of h_0 may miss out some of the import peaks and valleys. On the other hand, a too small value of h_0 introduces unwanted ripples and makes the subsequent matching difficult. Through experiments, an appropriate trade-off value is chosen as $h_0=1$ pixel in our project. Our simple EPs marking algorithm can filter small ripples in the definition of the EPs. However the two consecutive EPs may still form a ripple when the two EPs are close to each other. In Section 4.4.2, we will demonstrate how to match the two sets of EPs correspondingly, given the presence of ripples.

4.4.2 EPs Matching

After marking the EPs along the sample and the reference signals, the corresponding EPs need to be matched. The matching of the EPs helps to remove the position variations. Through studying the variation phenomena from the collected database, we can summarize the three types of variations:

1. Non-synchronicity for the start point – the first EPs of two signals may not synchronously start from a peak (or a valley)
2. Existence of ripples – one or more ripples may be found at the start, between a consecutive peak/valley pair, and at the end
3. Non-synchronicity for the end point – the last EPs of two signals may not synchronously end up with a peak (or a valley).

Figure 4-8 shows an example of two sets of EPs. In Figure 4-8, the graph (a) displays the torque signal from a reference signature while the graph (b) displays the torque signal from a genuine signature. The EPs on both signals are marked with ‘*’. Non-

synchronicities are observed both at the start and the end. In addition, an extra ripple appears on the signal in Figure 4-8 (a).

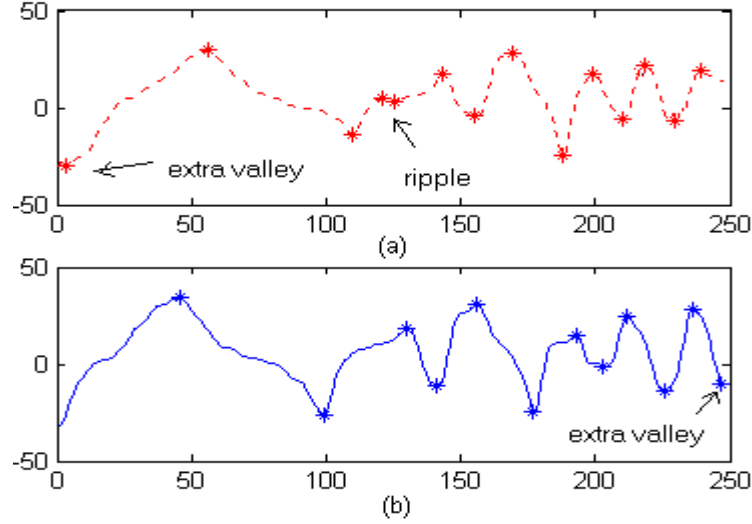


Figure 4-8: A Demonstration of Extremes Points From Two Torque Signals

We will define a matching algorithm to identify the matching pairs of the corresponding EPs despite the variations mentioned above. The EPs matching algorithm is based on the Dynamic Programming (DP) matching technique [57]. In a typical DP matching process [29], one point on one signal can be matched to any point on the other signal. However in our case, the EPs are peaks and valleys alternating. Hence the corresponding matching pairs of EPs have to be peak-peak or valley-valley matching. We need to introduce some new rules in the DP algorithm to suit the application.

In the EPs matching process, an EP-EP matrix is first established as in Figure 4-9 (a). In the matrix, the EPs on the reference signal form the horizontal axis and the EPs on the sample signal form the vertical axis. Note in the matrix, the two sets of the EPs need to synchronously start with a peak (or a valley). The global costs at the elements

within the unshaded region in Figure 4-9 (a) are to be computed. The warping path is defined by following the least-global-cost path from $(1, 1)$ to (I, J) .

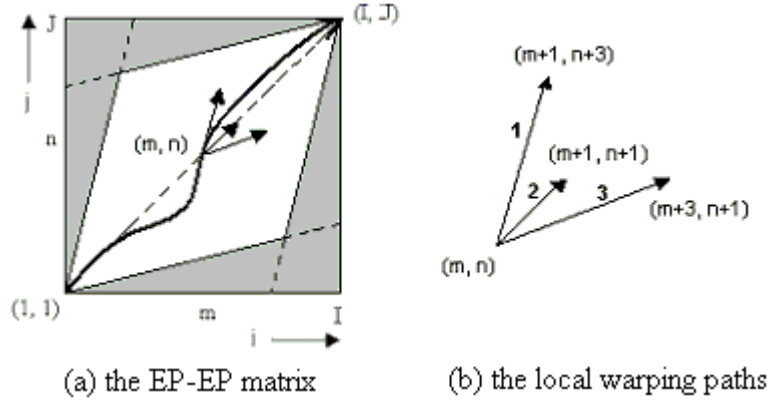


Figure 4-9: The Warping Path of EPW

In Figure 4-9 (b), it shows three local warping paths at the element (m, n) . That is to say, if we assume that (m, n) is a matching pair (i.e. the m^{th} EP on the reference signal is matched to the n^{th} EP on the sample signal), the next matching pair can be one of the three: $(m+1, n+1)$, $(m+1, n+3)$ and $(m+3, n+1)$. The $(m+1, n+3)$ means that the two EPs right after the n^{th} EP on the sample signal are regarded as a ripple, and hence skipped in the matching. Similarly the matching at $(m+3, n+1)$ means that the two EPs after the m^{th} EP on the reference signal are regarded as a ripple and hence skipped in the matching. Whenever a pair of EPs is skipped, a skipping cost $S(k, k+1)$ is incurred. It is defined as the City Block distance [62] between the k^{th} and the $(k+1)^{\text{th}}$ EPs on the signal.

The EPs on the reference signal can be expressed as two-dimensional data (x_i, y_i) , where the ' x_i ' is the horizontal position of the EP and the ' y_i ' is the vertical amplitude of the EP. Similarly the EPs on the sample signal are expressed as (x_j, y_j) . The values for both x_i and x_j range from 1 to N , where N is the total number of points on the

signal. Since we choose to re-sample shape with 256 points in the project, $N = 256$. The amplitude value, y_i or y_j , are expressed in terms of pixels. A matching pair of EPs on the input signal and the reference signal should have close values in both the position and the amplitude. The position and the amplitude are considered equally important to determine the matching between the two EPs. Hence we normalize the reference signal EP position (x_i) and the input signal EP position (x_j) in the equations (4.14) and (4.15) respectively. The normalization factor ρ_0 is obtained from equation (4.13).

$$\rho_0 = \frac{\max(y_i) - \min(y_i)}{N} \quad (4.13)$$

$$x_i' = x_i \times \rho_0 \quad (4.14)$$

$$x_j' = x_j \times \rho_0 \quad (4.15)$$

After normalization, the position and amplitude values will have the same unit (i.e. pixel) and roughly the same range. They will be equally important to influence the decision of matching between the two EPs. In the process of EPW, the local distance $d(i, j)$ is defined as the City Block distance [62] between the two EPs.

$$d(i, j) = |x_i - x_j| + |y_i - y_j| \quad (4.16)$$

Note the City Block distance is adopted instead of the Euclidean distance [62]. This is to avoid the situation when a big difference in position or amplitude may over-influence the final decision. If we assume synchronicity for the first EPs of the two signals, the matching process of ETW can be explained in the followings:

1) Initial condition

$$D(1, 1)=d(1, 1), D(1, 3)=d(1, 3), D(3, 1)=d(3, 1) \quad (4.17)$$

2) Monotonicity and continuity conditions

$$1 \leq i_k - i_{k-1} \leq 3, 1 \leq j_k - j_{k-1} \leq 3, i_k - i_{k-1} \neq 2 \text{ and } j_k - j_{k-1} \neq 2 \quad (4.18)$$

3) Boundary condition

$$\text{If synchronicity at end: } j=3i, j=1/3 i, j-J=3(i-I) - 2, j-J=-1/3 (i-I+2) \quad (4.19)$$

$$\text{Else: } j=3i, j=1/3 i, j-J=3(i-I) - 3, j-J=-1/3 (i-I+3) \quad (4.20)$$

4) Global cost calculation

$$D(i, j) = \min \begin{bmatrix} D(i-1, j-3) + d(i, j) + \rho_s \times S(j-2, j-1) \\ D(i-1, j-1) + \frac{1}{2}d(i, j) \\ D(i-3, j-1) + d(i, j) + \rho_s \times S(i-2, i-1) \end{bmatrix} \quad (4.21)$$

In equation (4.21), the ‘ ρ_s ’ is defined as the skipping factor. As different from the normal DTW process [29], we introduce a skipping cost into the global cost computation. The skipping cost is usually very small when skipping a ripple in the matching, however it would be much larger if a pair of important peak and valley is misinterpreted as a ripple and skipped. A skipping factor ρ_s , as shown in equation (4.21) is to adjust the influence of the skipping cost on the decision of matching. Through fine-tuning, $\rho_s = 2$ is found to be appropriate.

We will show an example by applying the algorithm to match correspondingly the two sets of the EPs (see Figure 4-8). Firstly an EP-EP matrix is established in Figure 4-10, where the EPs on the reference signal form the horizontal axis while the EPs on the sample signal form the vertical axis.

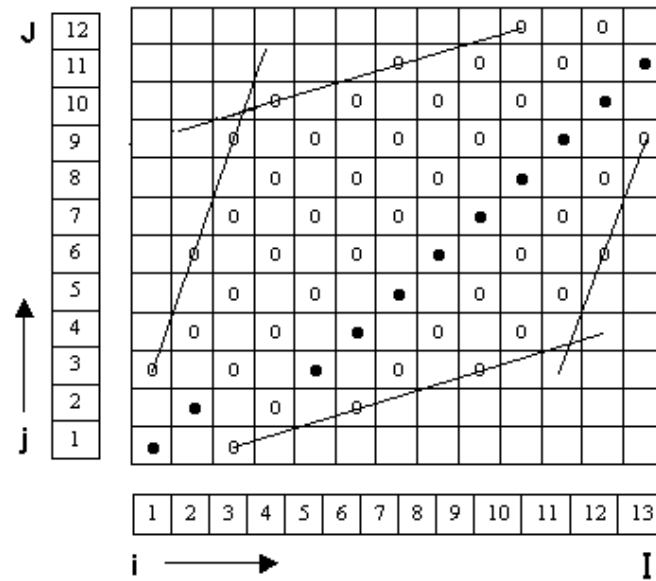


Figure 4-10: An Example of Using Extreme Points Matching

As we have explained, the matrix assumes synchronicity of the two sets of EPs. Hence the first EP (an extra valley) on the reference signal is removed, so that two set of EPs start synchronously with a peak point. One may also remove the first EP (a peak) on the sample signal to make two sets of EPs synchronously start with a valley. Though it is not proper from the visual inspection (see Figure 4-8), it will still form a valid EP-EP matrix. But the much higher costs incurred at elements of the second matrix will indicate that it is not the correct matrix.

In Figure 4-10, the circled cells (include the dotted cells) within the defined boundaries indicate possible matchings. The global costs are to be computed for all the circled cells. The dotted cells indicate the correct matchings, which follow the least-global-cost path. A dotted cell at (i, j) will indicate that the i^{th} EP on the reference signal is matched to the j^{th} EP on the input signal.

During initialization, the global costs at (1, 1), (1, 3) and (3, 1) are computed as equal to the corresponding local costs, i.e. $d(1, 1)$, $d(1, 3)$ and $d(3, 1)$. The circle at (1, 1) indicates that the 1st EP of the reference is matched with the 1st EP of the input. On the other hand, the circle at (1, 3) indicates that the 1st EP of the reference is matched with the 3rd EP of the input. It indicates that the first two EPs of the input are a pair of ripple so they are skipped in the matching process. Similarly, the circle at (3, 1) means that the first two EPs of the reference are a pair of ripple and are hence skipped.

As explained in Section 4.4.2, one common variation in EPs matching is non-synchronicity for the end EPs between the two signals. Figure 4-11 shows how synchronicity and non-synchronicity at end will make a difference in the EP-EP matrix.

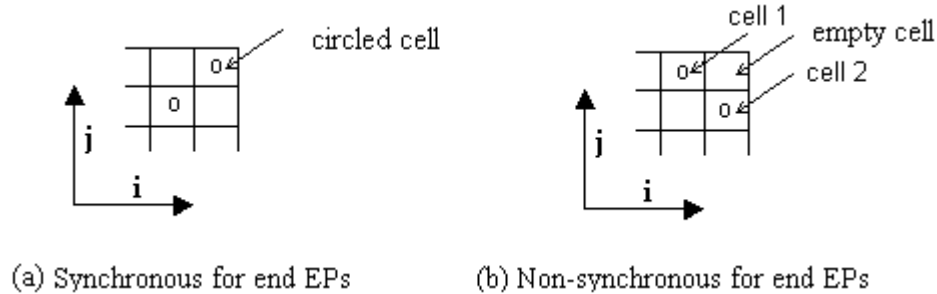


Figure 4-11: Two Cases for Matching the End Extreme Points

In Figure 4-11 (a), the end EPs of the reference and input signals are synchronous and hence matched. However there is a possibility that they are not matched if a ripple exists at the end of either signal, which will be explained later in this section. Figure 4-11 (b) shows the case when the end EPs are not synchronous. In such a case, either the reference signal has an extra end EP or the input signal has an extra end EP. The two possibilities are reflected in the two immediate adjacent circled cells respectively:

cell 1 and cell 2. The correct matching will be at one of the cells with the lower global cost. Similarly there is a possibility that neither cell indicates the correct matching if a ripple exists at the end of either signal. We will explain this later in this section.

After establishing the EP-EP matrix as shown in Figure 4-10, we will compute the global cost at each of the circled cells progressively by following equation (4.21). The warping process starts from (1, 1) and continues until the last corner-cell. The corner-cell is defined as the cell at the corner of the matrix in the end of the matching process. Referring to Figure 4-11, one may find that there are two such corner-cells if non-synchronous for end EPs while only one corner cell if synchronous for end EPs. Figure 4-12 is an example, showing the case of the non-synchronicity at end EPs. We will illustrate how the global cost is computed at one of the two corner-cells (i.e. cell 1). The global cost at the other corner-cell (i.e. cell 2) is computed in the same way as at cell 1.

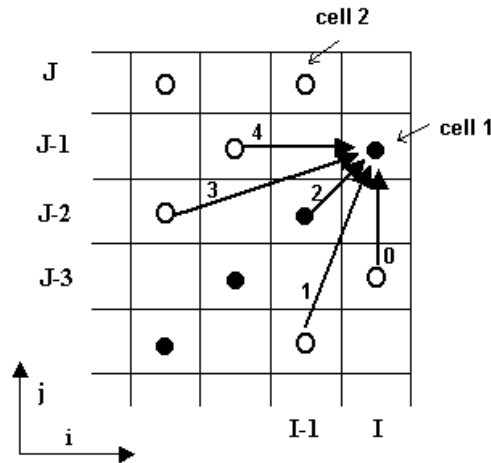


Figure 4-12: Local Warping Paths for the Last Corner-Cell in EPW

The computation of the global cost at the corner-cell 1 is given as:

$$D(i, j) = \min \begin{bmatrix} D(i, j-2) + d(i, j) + \rho_s \times S(j-1, j) \\ D(i-1, j-3) + d(i, j) + \rho_s \times S(j-2, j-1) \\ D(i-1, j-1) + \frac{1}{2}d(i, j) \\ D(i-3, j-1) + d(i, j) + \rho_s \times S(i-2, i-1) \\ D(i-2, j) + d(i, j) + \rho_s \times S(i-1, i) \end{bmatrix} \quad (4.22)$$

As shown in Figure 4-12, there are additional two warping paths, i.e. path 0 and path 4, to reach the corner cell. These two paths cater for the two cases respectively: i) a ripple exists at the end of the input signal, and ii) a ripple exists at the end of the reference signal. We will use an example to illustrate this. Assume the cell 1 is reached via path 0 following equation (4.22). It means that after removing the extra EP at the end of an input signal, the last peak/valley pair is a ripple on the input signal and is subsequently skipped in matching. Hence the last matching pair of EPs will be the I^{th} EP on the reference signal and the $(J-3)^{\text{th}}$ EP on the input signal (see Figure 4-12).

Finally by following the least-global-cost path, we will be able to determine the correct matching of EPs between two signals. Figure 4-13 shows the result after the matching process, where the matching pairs are ordered in sequence.

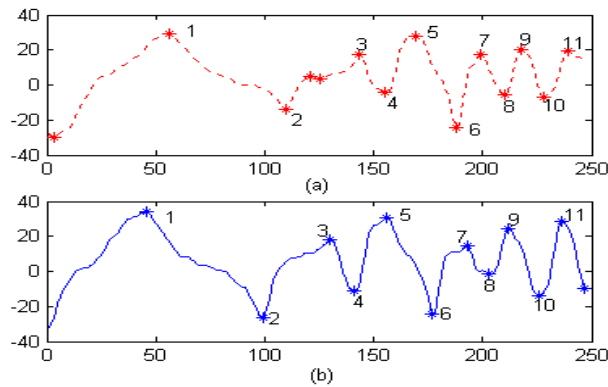


Figure 4-13: A Demonstration of Correct Matchings of Extreme Points

4.4.3 Segment Warping

After determining the correct matching pairs of the two EPs sets, we will warp the segments between the EPs linearly. Figure 4-14 shows the segments of the two signals.

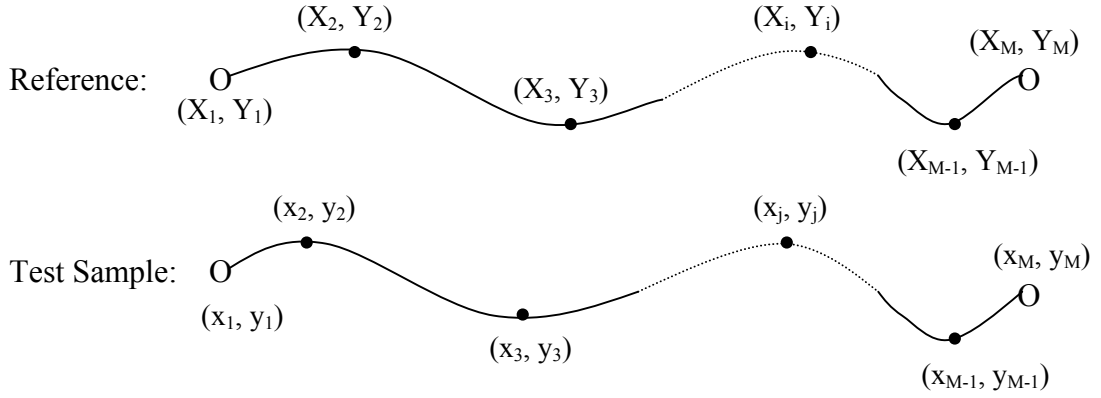


Figure 4-14: Matching Segments Based on Extreme Points

Figure 4-15 shows two corresponding segments. The point (x_j, y_j) is an arbitrary point on the segment of a sample signal.

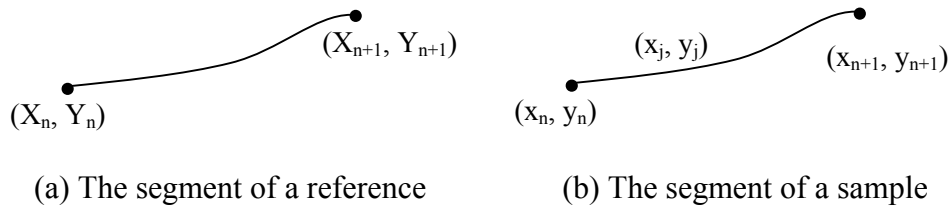


Figure 4-15: Linearly Warping the Matching Segments

In the segment warping process, the sample segment will be linearly stretched to align with the reference segment. The stretching only changes the position of a point (i.e. x), without changing the magnitude (i.e. y). After segment warping, we have:

$$x_n' = X_n, x_{n+1}' = X_{n+1} \quad (4.23)$$

$$x_j' = X_n + (x_j - x_n) \times \frac{X_{n+1} - X_n}{x_{n+1} - x_n} \quad (4.24)$$

Referring to the two torque signals in Figure 4-8, we have applied a new warping technique – Extreme Points Warping – to warp the sample signal against the reference signal. Figure 4-16 shows the result of the warping using EPW. Figure 4-16 (a) shows the sample signal before EPW while Figure 4-16 (b) shows after EPW. It is observed that by warping a set of selective extreme points, we are able to achieve the goal of warping the whole signal. The correlation coefficient between the sample signal and the reference signal is increased from 34.8% to 91.7% after the warping process.

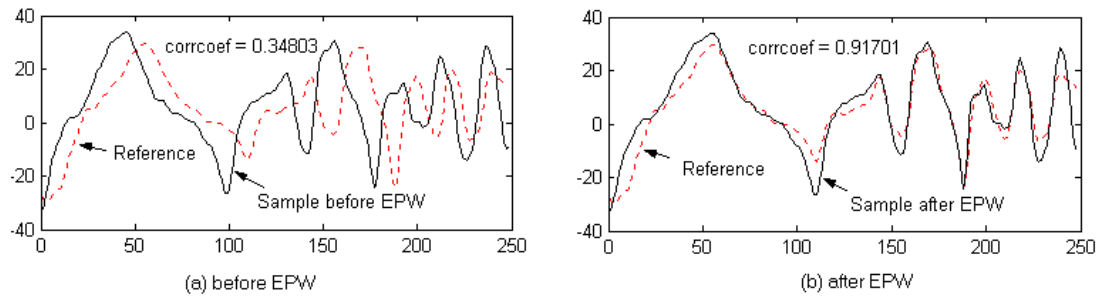


Figure 4-16: Sample Signals Before and After EPW

4.4.4 Other Examples of Using EPW

Figure 4-17 and Figure 4-18 are two more examples to demonstrate the warping results using EPW.

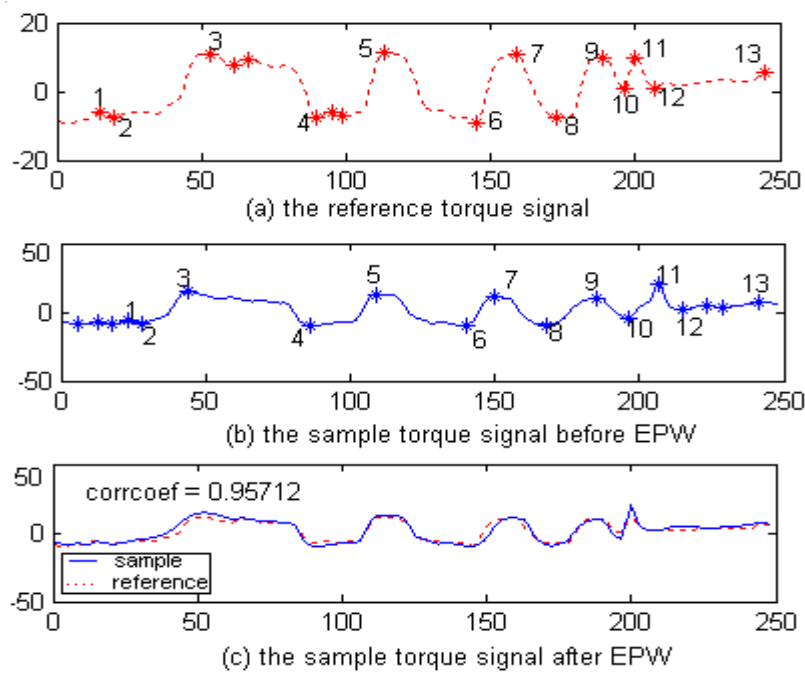


Figure 4-17: Example of EPW (1) - Showing the Correct Matching at Start

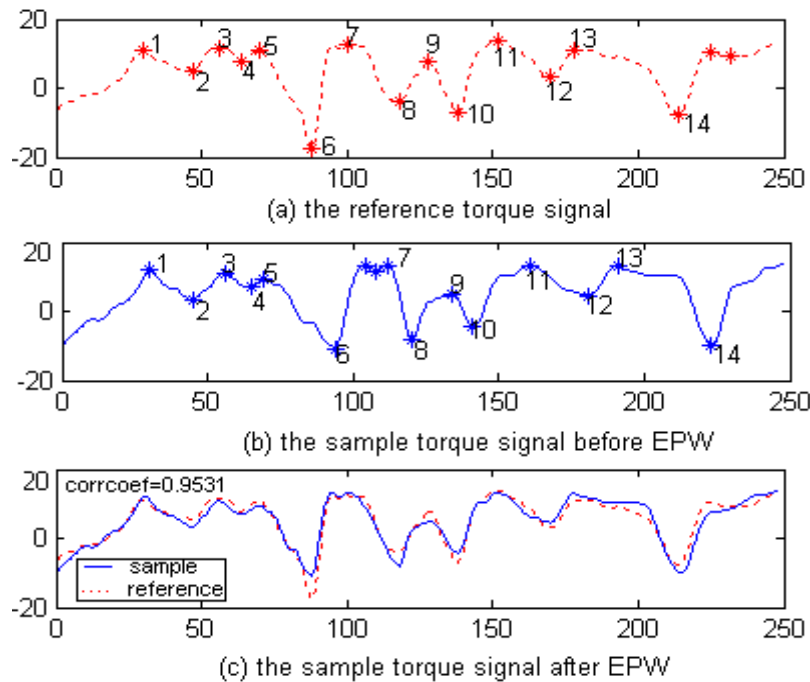


Figure 4-18: Example of EPW (2) - Showing Correct Matching at End

In Figure 4-17, the sample signal is non-synchronous with the reference signal at start as it has an extra valley and a ripple. Our matching algorithm is able to remove the

extra valley and the ripple in the matching process (see Figure 4-17). Similarly in Figure 4-18, the ripple at the end of the sample signal is removed.

The two figures, Figure 4-17 and 4-18, demonstrate the correct matchings when ripples exist at the start or the end of the signal. Besides, it is also noted from the two figures that the ripples within the signals are correctly handled as well. The three types of the variations described in Section 4.4.2 are adequately addressed and handled in our matching algorithm.

However the EPs matching algorithm is not without limitations. Figure 4-19 shows an example of mismatching using EPW. While most of the EPs are matched correctly, the two EPs, both marked with '3', are mismatched.

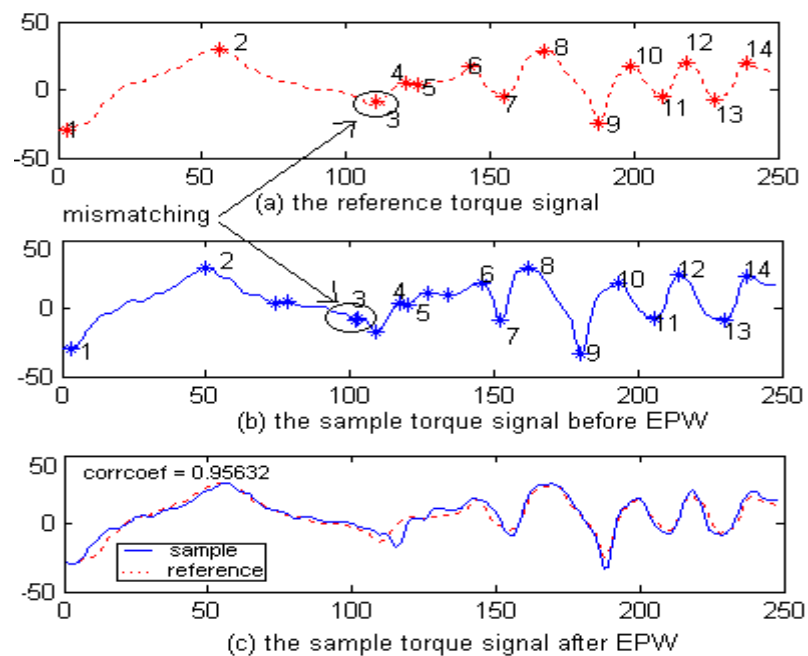


Figure 4-19: Example of EPW (3) – Showing the Mismatching

In Figure 4-19, the EP '3' (a valley) on the sample signal is matched with the EP '3' (a valley) on the reference signal. However from visual inspection, the correct matching should be the next valley after EP '3' on the sample signal that matches with the EP '3' on the reference signal. The mismatching reflects the limitation of the EPs matching algorithm. Referring to Figure 4-10, if we use (i, j) to represent the i^{th} point on the reference signal matching to the j^{th} point on the sample signal, our algorithm will find the next matching pair from $(i+1, j+1)$, $(i+1, j+3)$ or $(i+3, j+1)$. While the algorithm enjoys the simplicity and reliability, it can identify and hence remove at maximum only one ripple between EPs. Mismatching occurs when there are more than one ripples between the EPs, which is the case shown in Figure 4-19 (b).

This problem can be alleviated, by choosing an appropriate value of h_0 (see equation 4.12). Though there is no fixed range for h_0 except $h_0 > 0$, an appropriate value of h_0 is very important. The value of h_0 controls the number of ripples. A too small value of h_0 will introduce many unwanted ripples, while a large value of h_0 will miss some important EPs. Through experiments, a suitable trade-off value, $h_0 = 1$ pixel, is found to be appropriate. With this value, the occurrence of two or more ripples is rare between the genuine signal and the reference signal. In fact, the occasional mismatching may not have a devastating effect on the judgement of the signature genuineness. In Figure 4-19, in the presence of mismatching, the resultant warped signal still achieves a correlation coefficient with the reference signal as high as 95.6%.

4.4.5 Warping Forged Signals using EPW

To evaluate the performance of the EPW process, we will warp the forged signals exactly in the same way as the genuine signals. The difference only lies on the results from warping. Most of the time, the EPs of the genuine signal are matched correctly to the corresponding EPs of the reference signal. However because of ripples and deviation, mismatch occurs more frequently for forged signals. For a genuine signal, a high correlation coefficient can be obtained after EPW. For a forged signal, a much lower correlation coefficient is obtained even after all its EPs are aligned correctly to the reference positions. Figure 4-20 shows examples of a genuine torque and a forged torque before and after the EPW process.

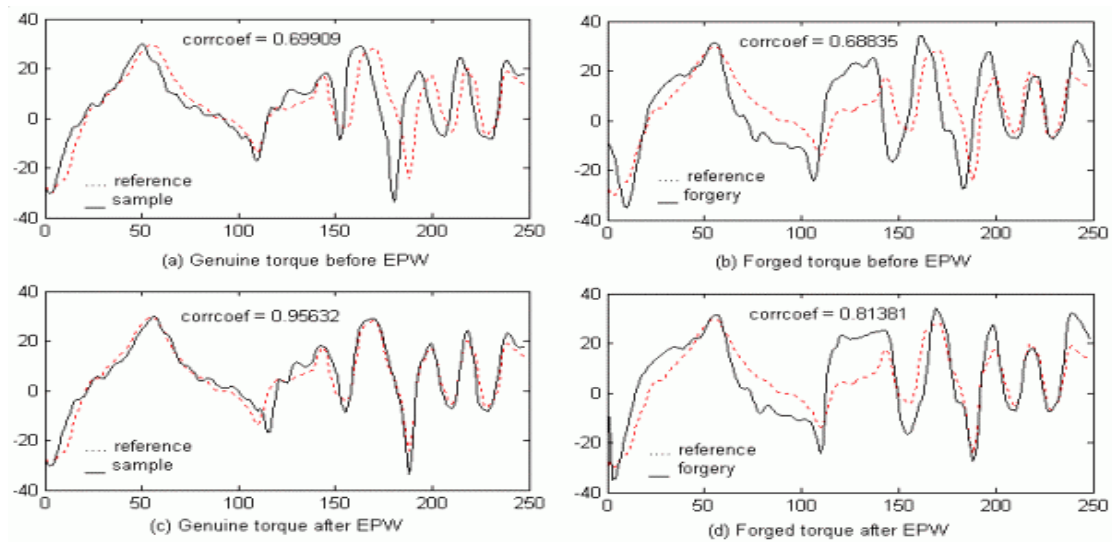


Figure 4-20: A Comparison of Using EPW for Genuine and Forged Signals

Figure 4-20 (a) and (b) show a genuine torque signal and a forged torque signal before EPW, respectively. The correlation coefficients in the two cases are approximately the same, around 70%. Figure 4-20 (c) and (d) show the genuine torque signal and the forged torque signal after EPW, respectively. After EPW, the correlation coefficient for the former is about 95.6%, while for the later is only 81.4%. This is because EPW

only warps extreme points while preserving local curvatures between the extreme points. It is different from DTW, which warps every point on the signal and hence destroys the local curvatures. A systematic comparison of performance between DTW and EPW will be introduced in Section 4.5.

4.5 Evaluation of the New Technique

To evaluate the proposed new technique, we perform a comparative analysis between EPW and DTW. We will evaluate the performance in two aspects: error rate and speed. The results from EPW will be compared to those from DTW, based on the same database and under the same test conditions. The test conditions include threshold definition and classification decision. These will be introduced in Section 4.5.1. The result of error rate will be expressed in terms of Equal Error Rate (EER), which is a key indicator of system performance in signature verification. The result of speed will be expressed in terms of the computation time in milliseconds.

4.5.1 Testing Conditions

To measure the similarity of the two signals, computing the correlation coefficient is the most straightforward way [26][38]. The correlation coefficients for the genuine signatures and the forged signatures form two clusters of data. Classification of the two clusters is based on a threshold. Besides the correlation coefficient, some researchers use the Euclidean distance between two signals [27][29][31][46]. We will present the comparative analysis using these two measures. If Γ_{xy} is used to denote the similarity of the two signals x and y , it can be computed in the following ways:

Correlation coefficient:

$$\Gamma_{xy} = \frac{\sum (xy - \bar{x}\bar{y})}{\sqrt{\sum (x^2 - \bar{x}^2) \times \sum (y^2 - \bar{y}^2)}} \quad (4.25)$$

Euclidean distance:

$$\Gamma_{xy} = -\sum |x - y| \quad (4.26)$$

Note that the Euclidean distance is a measure of dissimilarity. Hence in equation (4.26), a minus sign is added to make the result be consistent with the definition of similarity.

During the enrolment phase, the user provides ten samples for registration. One of the samples is selected as the reference. The similarities between the remaining nine samples and the reference can be obtained as $[\Gamma_1, \Gamma_2, \dots, \Gamma_9]$. The unbiased estimate of standard deviation for these values is denoted as std_Γ , which is expressed as:

$$std_\Gamma = \sqrt{\frac{\sum_{i=1}^M (\Gamma_i - \bar{\Gamma})^2}{M - 1}} \quad (4.27)$$

where $M=9$, which indicates the No of samples. When a test sample is to be verified, Γ_t is obtained by either equation (4.25) or equation (4.26). The decision of acceptance is based on the condition below:

$$\Gamma_t \geq (\bar{\Gamma} - a \times std_\Gamma) \quad (4.28)$$

where ‘a’ is a variable, which is used to adjust the threshold in the shape matching stage. The FRR and the FAR curves will be plotted by changing the value of ‘a’ and the EER can be determined by finding the intersection of the two curves.

4.5.2 Error Rates for DTW and EPW

To compare the error rates between DTW and EPW, we will use the five characteristic functional signals and the two similarity measures. The results in terms of EER are obtained by adopting the two warping techniques: DTW and EPW. Table 4-1 presents the results of the comparison.

	Euclidean distance						Correlation coefficient					
Signal	X	Y	X_C	Y_C	Torque	Ave	X	Y	X_C	Y_C	Torque	Ave
DTW	30.5	34.4	29.5	35.1	35.2	33.0	33.5	36.0	33.2	34.8	37.4	35.0
EPW	23.6	26.3	25.5	26.4	25.4	25.4	27.3	27.8	30.6	28.4	24.2	27.7

Table 4-1: Equal Error Rates for EPW and DTW (%)

It is noted from Table 4-1 that under the same test conditions, the equal error rate of EPW shows improvement over DTW. When the Euclidean distance is used, the average EER of the five signals is 33.0% for DTW and 25.4% for EPW, *an improvement of 22.8%*. When the correlation coefficient is used, the average EER is 35.0% for DTW and 27.7% for EPW, *an improvement of 20.8%*. Overall, using EPW shows an improvement of around 20% in terms of equal error rate as compared with, using DTW.

4.5.3 Improved Error Rates with Weight for EPW

A shape signal is constituted by a series of discrete points. In the above, we compute the correlation coefficient and the Euclidean distance between the discrete points using the equations (4.25) and (4.26) respectively. In the two equations, each of the signal points is treated equally, hence carries the same weight. In [31][37][38], researchers find that assigning different weights over the signal helps to improve the equal error rate.

Different weights are assigned to the points along the signal so as to reflect the stability at that point [37]. Those points, which are highly stable among genuine samples, should be more important and carry more information. On the other hand, those unstable points are less important and carry less information. So the weight for each of the points on the signal can be computed based on the stability at that position among several prototype signatures. By assigning different weights over the signal, the comparison between signals will be more meaningful, and an improvement in EER is expected [37]. In the following, we will show an improvement of the EER obtained from EPW with the added weights. The weights for the points along the signal are computed from the following:

$$w(i) = 1/\text{std}^n[s_1(i), s_2(i), \dots, s_{10}(i)] \quad (4.29)$$

where $i = 1, 2, \dots, N$. The N is the length of the signal. In equation (4.29), the weight for the i^{th} point is computed as the inverse of the n -ordered standard deviation among the corresponding points of the ten enrolled signature samples. With the definition of the weights, the two similarity measures can be re-defined as:

Weighted correlation coefficient:

$$\Gamma_{xy} = \frac{\sum (w^2 xy - \overline{wx} \overline{wy})}{\sqrt{\sum (w^2 x^2 - \overline{wx}^2) \times \sum (w^2 y^2 - \overline{wy}^2)}} \quad (4.30)$$

Weighted Euclidean distance:

$$\Gamma_{xy} = -\frac{\sum |wx - wy|}{\sum w} \quad (4.31)$$

In the researches at [26][38], the order ‘n’ in equation (4.29), is taken as n=1. In [42], Wirtz defined several values for n and the best result was obtained when n=2. In our research, we will set n, ranging from 1 to 5. Table 4-2 shows the experiment results. Note that n=0 (i.e. no weights) is also added in the table for the comparison purpose.

Order	Euclidean distance					Correlation coefficient				
	X	Y	X_C	Y_C	Torque	X	Y	X_C	Y_C	Torque
0	23.55	26.3	25.53	26.36	25.43	27.31	27.83	30.6	28.43	24.23
1	22.59	27.27	26.40	27.33	22.33	24.51	30.35	25.51	28.13	20.23
2	22.29	28.67	25.73	28.56	17.87	24.25	32.37	27.66	31.85	22.03
3	23.51	27.02	29.08	27.73	20.80	26.96	30.68	26.66	33.20	23.68
4	25.12	26.14	29.96	29.94	23.81	28.30	32.59	30.00	34.16	27.60
5	26.10	28.59	30.06	31.16	26.36	29.04	34.57	32.26	34.50	29.66

Table 4-2: Equal Error Rates for EPW with Added Weight (%)

From Table 4-2 it is noted that in general with the added weights, it shows an improvement of EER. In particular, the improvement is most obvious for the torque signals. However for some signals, e.g. Y, Y_C (y of Centre of Mass), a slight setback of the error rate is observed. The cause of the phenomenon is mainly because those signals are not as robust as the torque signals. As a result, the stability information

derived from the prototype samples may not be consistent with that among the test samples. An illustration of the robustness of the torque signals, as compared to others, can be found in [38].

In addition, it is observed from Table 4-2 that, an over-ordered (i.e. $n > 2$) weighting function will in general worsen the result. The best result obtained for the torque signal is by using the Euclidean distance measure, and setting the weight order 'n' equal to 2. In this case, the Equal Error Rate (EER) achieved is 17.87%. Figure 4-21 shows the error rate curves. The operating points on the curves are chosen at $a=10$. When $a=10$, the FRR (False Rejection Ratio) is 7.4% while the FAR (False Acceptance Ratio) is 47.6%. In other words, in the shape matching stage, over half of the forged signatures (52.4%) are successfully rejected. It is done at a small cost that less than one among ten genuine signatures are falsely rejected.

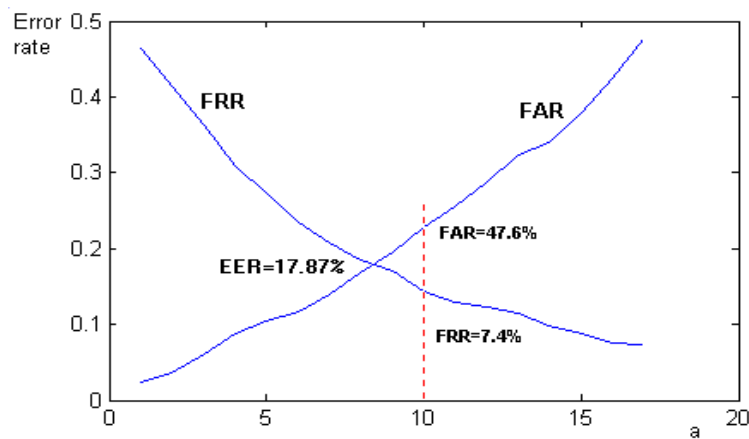


Figure 4-21: The Point of Operation for Shape Matching

4.5.4 Computation Time for EPW

Besides error rate, speed is another attribute used to compare the performance between DTW and EPW. The speed will be expressed in terms of computation time in

milliseconds. The simulation is done using Matlab 6.1 on a Pentium IV 1.9 GHz PC with 256 MB RAM, running Windows 98.

The computation time recorded for DTW is the time of the non-linear warping process for a signal. The computation time recorded for EPW includes the three steps: marking the extreme points, matching the extreme points and linearly warping the segments. The simulation is performed for all the 25 users, using the five characteristic signals outlined in Section 4.2. The simulation results are tabulated in Appendix C. Table 4-3 summarizes the averaged computation time of the five signals for each user.

Users	1	2	3	4	5	6	7	8	9	10
DTW	415.9	414.5	414.5	416.0	416.8	415.2	415.2	416.3	416.3	416.0
EPW	15.8	17.9	17.2	44.0	50.8	28.7	71.1	41.2	30.4	60.8

Users	11	12	13	14	15	16	17	18	19	20
DTW	415.1	416.6	414.2	416.5	416	417	417	415.1	416.4	414.7
EPW	51.5	23.3	13.4	39.6	18.2	49.8	49.8	15.5	50.9	24.7

Users	21	22	23	24	25	Ave
DTW	414.5	415.5	415.8	415.2	416.7	415.7
EPW	22.4	21.0	48.2	23.6	95.1	37.0

Table 4-3: Computation Times using DTW and EPW for All Users (ms)

The data in Table 4-3 are plotted in Figure 4-22. It is noted from Figure 4-22 that the computation time using DTW is quite constant. That is because the number of points involved in each of the DTW process is the same. The number of points is fixed at 256 points. On the other hand, the computation time using EPW is variant among users. This is because the computation time is mainly determined by the number of

the extreme points involved in the EPs matching process. Because of the natural difference of the signature complexity, the number of extreme points identified is different for different users. The more complex a signature, the more extreme points identified and hence the longer the computation time. In fact, for the same user, the warping of the torque signal using EPW requires longer time, as the torque signal is more complex than other signals like x, y, etc (see Appendix C). In Table 4-3, the average computation time among the 25 users using DTW is 415.7 ms, while using EPW is only 37.0 ms, with an improvement of the factor of 11. From 0.4s to 0.037s, one may not be able to perceive the difference in the real-time applications. However the improvement would be most evident if it runs on a slower PC and deals with multiple users' requests simultaneously.

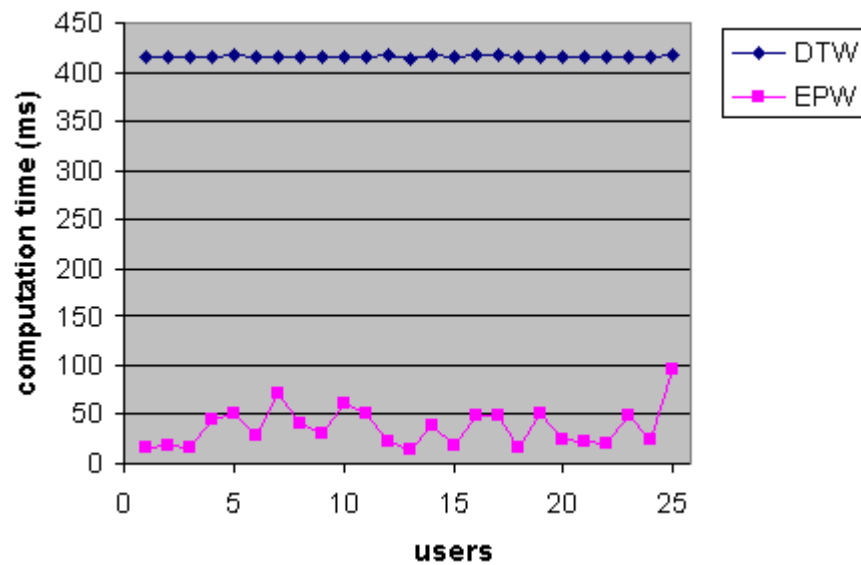


Figure 4-22: A Comparison of Speed between DTW and EPW

4.6 Summary

In this chapter, we propose a new warping technique called Extreme Points Warping (EPW). We compare EPW with the conventional technique DTW (Dynamic Time

Warping), using five characteristic shape signals and two similarity measures. The comparison has been done in two aspects: equal error rate and speed. It can be concluded that with the adoption of EPW, the equal error rate has improved by 20% and the speed has improved by a factor of 11.

Chapter 5: Feature Coding Stage

5.1 Overview

In the previous shape matching stage, we have examined the static shapes of the sample signatures to ensure that only good-quality signatures proceed to the feature-coding stage. The purpose of the feature-coding stage is to generate a 1-D bit-string from the dynamic features of the handwritten signature. The string should be unique and able to identify a person. Based on the bit-string, a private key is derived. As we have explained in Section 3.3, the bit-string has to be *all-bits-correct*. In this chapter, we will first illustrate that the existing signature identification schemes fail to achieve the requirement of all-bits-correctness. Hence we will define a new coding scheme to meet the requirement. We will then explain how to select the dynamic features for use in the system. Finally we will introduce how a private key can be generated. The private key generation is at the final processing stage in the BioPKI cryptosystem. We will include a brief explanation on the private key generation process in Section 5.5.

5.2 Feature Coding and Signature Identification

There are two different types of signature applications: signature verification and signature identification [61]. In simple terms, the former determines one-to-one matching while the later determines one-to-many matching [61]. We have explained the signature verification in Chapter 4. For signature identification, the system has no prior knowledge about who the user is. The system identifies the user by finding the

best match of the signature to a group of signatures. Detailed explanations on the difference between signature verification and signature identification can be found in [61].

The purpose of the feature coding stage is to generate a string from a signature. The string should be unique to the user. In other words, it should be able to identify the user from a group of users. So the idea of feature coding is very similar to signature identification except it has two additional requirements. Firstly the features used for identification must be dynamic. It is one of the system design requirements, which were described in Section 3.3. Secondly, the identifiable string has to be *all-bits-correct*. As we will explain the private key generation process in Section 5.5, the string will first be hashed using a hashing algorithm, e.g. SHA-1 [16]. One of the properties of a good hashing algorithm states that a single bit error at the input will cause the hashed output to be completely different (see [16]). So in order to generate the same private keys for the different genuine samples, the string obtained from each sample has to be all-bits-correct.

Signature identification is not as commonly used as compared to signature verification [61]. In the following, we will present a brief review of the signature identification systems proposed in recent researches. It should be noted that none of the existing signature identification schemes fulfil the two requirements mentioned above, especially the all-bits-correctness.

Pottier developed an automatic system in [59] to identify off-line handwritten signatures, using a connectionist approach. He first applied the image processing

techniques to extract the significant parameters from the signature images. Based on the extracted parameters, a signature was identified to belong to one individual by a 2-layer perceptron neural network. The neural network was trained to recognize the parameters with tolerance [59]. The extracted parameters don't have to be constant as the neural network can tolerate the variations.

Ke Han proposed an identification system in [60], which used a set of geometric and topologic features to characterize each signature. The system mapped each signature into two strings of finite symbols. Because the strings varied for different signatures, a local associate indexing scheme was then used to match part of the string to that of the reference [60]. The features are static and the string of symbols is not constant.

In Paulik's work [61], he transformed a signature into a 1-D spatial stochastic sequence. An Autoregressive Hidden Markov Model was then employed to describe the evolution of sequence changes.

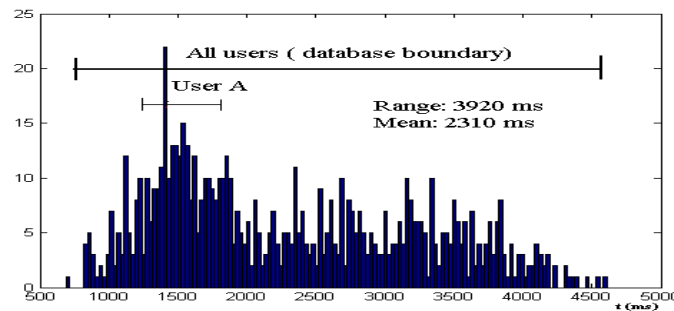
In [18], Gupta proposed to represent a waveform by a complete tree. A single text string was obtained from two positional profiles, which were x, y signals versus time. The method could be extended to other dynamic signals, like velocity, acceleration etc. However the text string can hardly be exactly the same as variations are unavoidable in the tree representation [18].

In view of the existing signature identification techniques outlined above, none of them extract features that are all-bits-correct. They are not meant to design in this way. Hence the existing identification techniques can't be directly applied to our

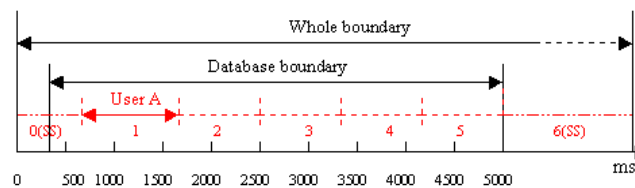
feature coding stage. Instead, we propose a new coding scheme in Section 5.3. The new scheme provides a mechanism to achieve all-bits-correctness with a trade-off of the error rates.

5.3 The Proposed Coding Scheme

The coding scheme encodes a dynamic feature value into a decimal number, which is defined as the feature code for that feature. A set of dynamic features will be defined in Section 5.4. Later the feature codes of all the defined features will be concatenated together to form a code string. We will use one feature, i.e. pen-down time, as an example to demonstrate how the scheme works.



(a) The histogram for all pen-down time values



(b) A skeleton view of the histogram

Figure 5-1: A Demonstration of Feature Coding for Pen-Down Time

Figure 5-1 (a) shows the histogram for the pen-down time values of 750 authentic samples in our database. Figure 5-1 (b) is a skeleton view of Figure 5-1 (a). In Figure

5-1 (b), three boundaries are defined: *whole boundary*, *database boundary* and *user boundary*. The *whole boundary* includes all possible values for a feature. For example, the whole boundary range for the pen-down time is between 0 and infinity. The *database boundary* includes values collected from the database. The *user boundary* includes values for a particular user. The *user boundary* is defined as:

$$User\ boundary = (\bar{T} - b \times std_T, \bar{T} + b \times std_T) \quad (5.1)$$

where ' \bar{T} ' is the mean of the ten feature values computed from the ten enrolled signatures samples. The ' std_T ' is the standard deviation of those ten values. The user boundary is flexible and its range is adjusted by the variable 'b'. A bigger value of 'b' corresponds to more error tolerance and on the other hand, easier barrier for forgeries.

In the defined scheme, the whole boundary will be divided into several segments as shown in Figure 5-1 (b). Each segment will be assigned with a decimal number starting from 0. The segmentation takes place in the following order. First a user boundary is defined with a chosen value 'b'. Then the same boundary is unfolded to both ends before exceeding the database boundary. Finally the superfluous portion at either end would be extended into the whole boundary and becomes one segment. The two Superfluous Segments in Figure 5-1 (b) are No 0 and 6. The boundaries for all the segments will be defined in a template. The system would first extract a particular feature value, fit it into a segment and obtain the feature code, i.e. the sequence No. After processing all the features, the feature codes are concatenated to output a binary code string, as shown in Table 5-1.

Feature No	1	2	3	4	...	n-1	N
Feature Code	101	100	01	11	...	101	010

Table 5-1: Concatenation of Feature Codes into a Code String

We will use μ to denote the number of segments defined over the database boundary for a feature. For the feature of pen-down time, $\mu = 5$ (see Figure 5-1 (b)), which excludes the 0th and 6th segments as they are superfluous. Hence the bit-information of this feature can be obtained as below:

$$\lambda = \log_2 \mu \quad (5.2)$$

The total bit-information, κ , for one person's signature is the summation of λ for each of the defined features.

$$\kappa = \sum \lambda \quad (5.3)$$

The total number of bits varies from person to person. In general, the more consistent the user's signature is, the bigger value of λ will be obtained and hence the more bit-information will be added. The average bit-information for 25 users in the database is around 40 bits, which will be explained in Section 6.2.

The template includes only the boundary definitions, without containing any hint on which particular segment a feature will fit in. It doesn't release any information about the feature code nor arise any privacy concern as usually people have for the biometric storage. In the following section, we will explain the features used in the scheme.

5.4 Selection of Feature Parameters

5.4.1 A Common Feature Set

Ideally, for each feature used in the scheme, the histogram of the feature should be flat. A flat histogram is important because it will generate diversified feature codes for users. A diversified feature code for each feature helps to ensure the uniqueness of the concatenated feature string. Hence in the feature selection process, the features with relatively flat histograms are preferred. In addition, only the dynamic features should be included for feature coding. The rationale for this is that the dynamic features, unlike the static features, are transparent to users. Visually it does not give any hint about the feature codes even with the knowledge of all boundary definitions.

Table 5-2 summarise a common set of 43 features. Most of these features are cited from [45]. Note that the shape-related static features defined in [45] are not included in the table. The defined 43 dynamic features can be grouped into five types: horizontal velocity related, vertical velocity related, altitude related, azimuth related and pen-down time related. In the table, peaks and valleys of a signal are identified by the EPs marking algorithm, as we have explained in Section 4.4.1. The Al+ and Az+ refer to part of the signal above the mean of the altitude and azimuth respectively, while the Al- and Az- are below the mean. The pressure-related features are not used, as the pressure data are quite vibrant and not consistent. In our project, we use the pressure signal only as the on-off signal to detect pen-down and pen-up.

1	Std of $ V_x $	12	RMS of V_y -	23	Max Al	34	Mean of Az
2	RMS of V_{x+}	13	Skewness of $ V_y $	24	Min Al	35	Median of Az
3	Skewness of $ V_x $	14	Max forward V_y	25	Mean of Al peaks	36	Mean of Az peaks
4	Mean of V_x	15	Max backward V_y	26	Mean of Al valleys	37	Mean of Az valleys
5	Std of V_x	16	Time of V_y last peak	27	Amplitude of Al first peak	38	Amplitude of Az first peak
6	RMS of V_x	17	Time of V_y last valley	28	Amplitude of Al first valley	39	Amplitude of Az first valley
7	Max forward V_x	18	Skewness of V_y	29	Amplitude of Al last valley	40	Amplitude of Az last valley
8	Max backward V_x	19	RMS of Al+	30	Time of Al last peak	41	Time of Az last peak
9	Time of V_x last peak	20	RMS of Al-	31	Time of Al last valley	42	Time of Az last valley
10	Time of V_x last valley	21	Mean of Al	32	RMS of Az+	43	Pen down time
11	Skewness of V_x	22	Median of Al	33	RMS of Az-		

Table 5-2: A Common Set of 43 Features

The histogram for each of the 43 features can be referred in Appendix D. As uniformly distributed features can hardly be found, we choose features with relatively flat data distributions. Figure 5-2 shows two such examples. Figure 5-2 (a) shows the histogram of the time of the V_x signal's last peak, while Figure 5-2 (b) shows the histogram of the time of the V_y signal's last peak.

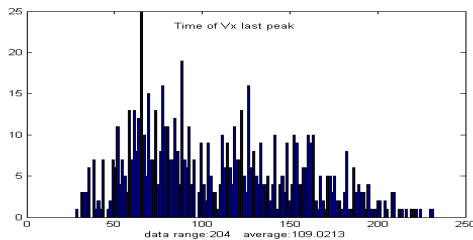
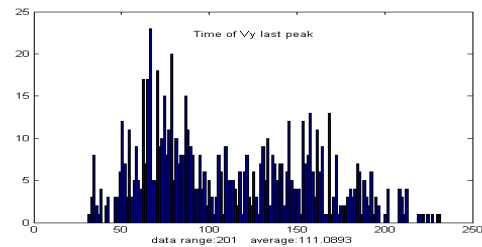
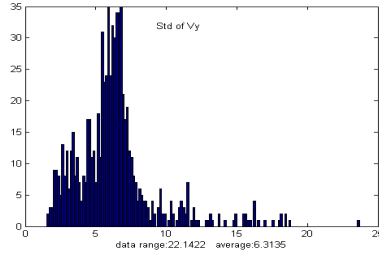
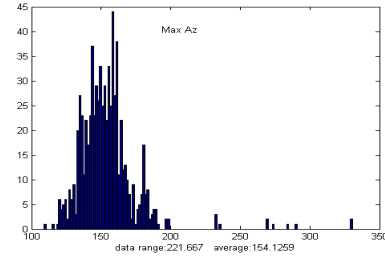
(a) Time of V_x last peak(b) Time of V_y last peak

Figure 5-2: Features with Relatively flat Data Distributions

However, some features are not selected into the common feature set because of the data distribution lumped (much higher probability) in certain regions. Figure 5-3 shows the data distributions of the two unselected features: standard deviation of Vy and max Az.



(a) Std of Vy



(b) Max Az

Figure 5-3: Unselected Features with Lumped Data Distributions

After the definition of the common set of features, we will introduce how to select a personalized feature set in Section 5.4.2.

5.4.2 A Personalized Feature Set

A personalized feature set reflects the individuality of each user [45]. In our project, we adopt a simple and effective selection algorithm used in [45] to select the personalized features for each user. In [45], the selection is based on the distance measure between the genuine signature and the forged signature. The distance measure for the i^{th} feature defined in [45] is expressed as:

$$d_i = \frac{|m(a,i) - m(f,i)|}{\sqrt{\sigma^2(a,i) + \sigma^2(f,i)}} \quad (5.4)$$

where $m(a, i)$ and $\sigma^2(a, i)$ are the sample mean and the sample variance of the i^{th} feature computed from the genuine signatures, respectively. Similarly, the $m(f, i)$ and $\sigma^2(f, i)$ are the sample mean and sample variance computed from the forged signatures. The most important feature for a user will be the feature with the maximum distance measure. A top-down list of the features with importance in descending order can be obtained for each user. A personalized n -feature set will be the top n features on the list [45]. In Chapter 6, we will compare the overall performance using the entire 43 features and personalized feature sets.

5.5 Private Key Generation Stage: An Example of DSA

The Private Key Generation Stage is part of our BioPKI cryptosystem, but not the focus of the research work. We will, however, give a brief explanation of the operation at this stage.

Pawan explained in details in [7] on how to generate a private key from a string. He used a conceptual example of iris sample and presumed that an all-bits-correct string had been obtained from the biometric sample. In our project, we propose a feasible and realistic real-time application to output an all-bits-correct string from an on-line signature. Reader can refer to [7] for the private key generation process in details.

To derive a private key from the code string, Digital Signature Algorithm (DSA) is preferred over Rivest Shamir Adleman (RSA) [7]. With DSA, the private key and the public key can be computed in the steps below.

1. Computation of p, q and g

$p=512\sim 1024$ bit prime number

$q=160$ bit prime factor of $(p-1)$

$g=h^{(p-1)/q} \bmod p$, where $h < (p-1)$ and $h^{(p-1)/q} \bmod p > 1$

2. Generation of the private key

Compute SHA1-hash [16] of code string obtained. The hash value is a 160-bit private key, denoted by x .

3. Generation of the public key

Compute $y=g^x \bmod p$. ' y ' is a p -bit public key.

From step 2, one may appreciate the importance of all-bits-correctness since a hash function is involved. A single bit difference at the input would results in very big difference at the hashed output [16]. More explanations on DSA and the mathematical proof can be found in [17].

5.6 Summary

In this chapter, we have explained the process at the feature coding stage of the BioPKI cryptosystem. In order to obtain an all-bits-correct code string from a set of dynamic features, we define a new coding scheme. In the scheme, each feature will output a feature code. The code string is formed, by concatenating all the feature codes. We have defined 43 dynamic features at this stage. The algorithm to select a personalized subset is presented. In addition, we have briefly summarized the steps

involved in the third processing stage: private key generation. An example of using DSA to derive a private key from the code string is described. The following chapter will describe the system performance for the BioPKI cryptosystem.

Chapter 6 Performance Evaluations

6.1 Overview

In the previous two chapters, we have covered all the processing stages of the proposed BioPKI cryptosystem. In this chapter, we will present an analysis of the system performance. The performance will be evaluated in two aspects: error rate and the average bit-length of the coding string. The average bit-length measures the robustness against brute-force attack [16]. The performance using the common set of features will be evaluated first. It will then be compared with using a subset of features. In addition, the uniqueness of the private key, its security strength and suggested remedies for practical applications are also introduced in this chapter.

6.2 Analysis of Overall Performance

The overall performance combines the results from both the shape matching and the feature coding. The accepted signatures are those, which not only pass the shape matching stage, but also achieve all-bits-correctness in the feature coding stage. The error rates will be expressed in terms of False Rejection Ratio (FRR) and False Acceptance Ratio (FAR) versus threshold. The intersection between FRR and FAR is defined as the Equal Error Rate (EER).

We will evaluate the system performance in two aspects: EER and average bit-length. The EER is an important indicator of the error rate performance. The average bi-

length indicates the length of the generated code string. A large bit-length value ensures the uniqueness of the code string and robustness against the brute-force attack. We will first show the performance using the entire 43 features. Then we will show how a personalized feature set can affect the performance of the BioPKI cryptosystem.

6.2.1 Performance Using 43 Features

When the entire 43 features are used for feature coding, the overall error rates are shown in Figure 6-1. The FRR and FAR are plotted versus the variable 'b'. The variable 'b' has been defined in equation (5.1) to adjust the user boundary.

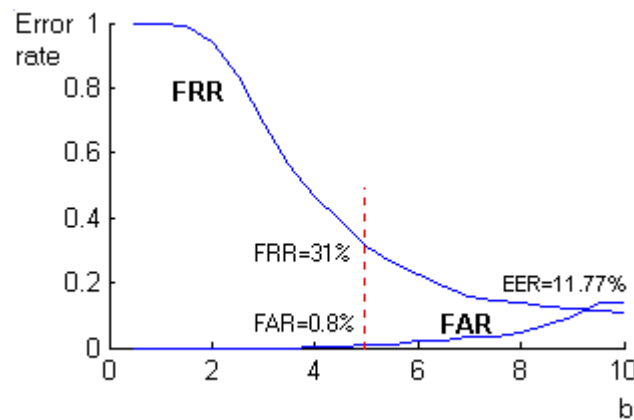


Figure 6-1: The Error Curves for Overall Performance

The initial results are encouraging. The EER is only 11.77%. The FAR indicates the percentage that a legal private key would be generated from a forgery. A low FAR is desired because it would give people high confidence about the legal validity of the private key generated from the BioPKI cryptosystem. As an example, we choose $b=5$ as the operating point for practical applications. When $b=5$, the FRR is 31% and the FAR is only 0.8%.

An interesting note is on the 31% FRR, which may alleviate some people's concern on the false alarm. From common perception, a string of all-bits-correct biometric data is difficult to obtain as the biometric features vary from time to time. It may be done at the expense of a high false alarm. However from our implementation using on-line signatures, the false alarm is at a reasonable level. Each user may try 1.4 times on average to get the correct private key from the signature, which is not too annoying to most of the people. It is worth paying a bit more effort on signing so as to enjoy the great convenience of not bringing around any smart card or remembering any passwords.

When 'b' is chosen as 5, the bit-length of the code string for each user can be obtained from equation (5.3). The average bit-length for 25 users is 42.5 bits. The error rates and bit-length for each of the 25 users are tabulated in Table 6-1.

User No.	FRR (%)	FAR (%)	Bit-length κ
1	35	0	50.09
2	0	0	63.31
3	15	10	56.96
4	25	0	47.30
5	40	0	51.30
6	10	0	45.71
7	50	0	44.21
8	65	0	36.02
9	10	0	2.58
10	35	0	26.92
11	20	0	26.81
12	65	0	28.08
13	70	0	53.13
14	50	0	47.85
15	15	0	38.58
16	15	0	15.34
17	15	10	32.54
18	15	0	48.49
19	45	0	25.92
20	15	0	65.38
21	30	0	74.00
22	35	0	39.45

23	40	0	55.07
24	35	0	47.75
25	25	0	38.79
Ave	31.0	0.8	42.5

Table 6-1: The Error Rates and Bit-lengths for 25 Users

In Table 6-1, it is noted that the bit-length for User 9 is only 2.58. This is undesirable although his FRR and FAR are quite low, which also suggests the big difference from conventional signature verification systems. In such a case, it remains secure against ‘forging’, however appears very vulnerable to brute-force attack. An attacker may easily find out the code string by trying out bit-by-bit. This happens due to the large deviations in the enrolled ten samples. This shows that the system is not suitable for those, whose signatures are not consistent.

6.2.2 Performance Using a Personalized Feature Set

In the second experiment, instead of using the entire 43 features, we choose a personalized subset for feature coding. The algorithm in selecting the best n features into the subset has been introduced in Section 5.4.2. We selected $n=30, 20, 10$ to form the personalized feature subset. The trade-off curve, i.e. FAR Vs FRR, for each of the ‘ n ’ values is shown in Figure 6-2.

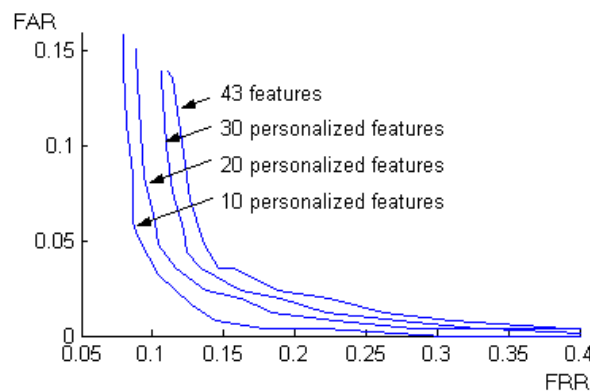


Figure 6-2: Trade-off Curves Using Personalized Features

In Figure 6-2, an improvement in error rate with the use of personalized features is observed, which is also evident in [26][45][53]. Figure 6-3 shows the error curves and the operating points for the different three subsets. The operating point for each case is chosen on the curve, where the FAR is close to 1%.

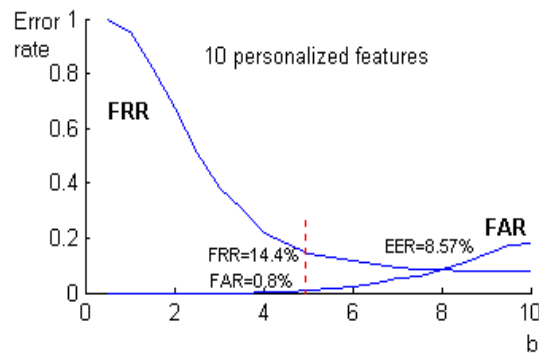
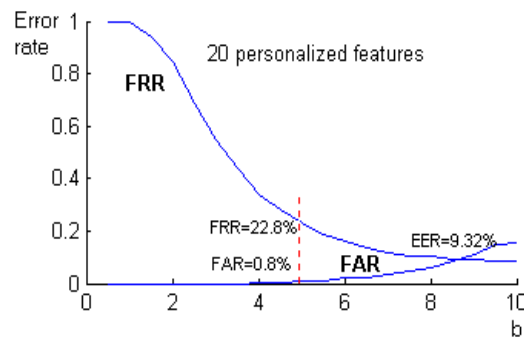
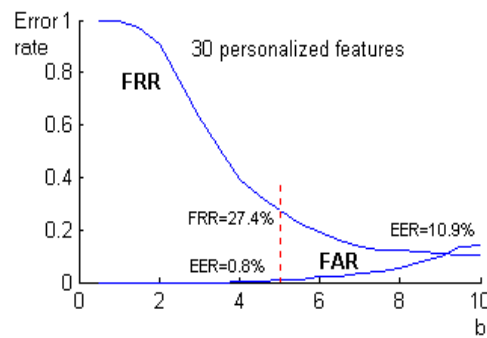
(a) $n=10$ (b) $n=20$ (c) $n=30$

Figure 6-3: Error Curves Using Personalized Features

Corresponding to the operating points on the curves, the values of 'b' are all equal to 5. By following equation (5.3), the bit-length for each user is computed. Table 6-2 summarizes the error rates and average bit-length when different numbers of features are defined.

No of Features	EER (%)	Operating Point				Average bit-length
		b	FRR (%)	FAR (%)	Ave tries	
43	11.77	5	31.0	0.8	1.44	42.5
30	10.9	5	27.4	0.8	1.37	30.5
20	9.32	5	22.8	0.8	1.29	20.9
10	8.57	5	14.4	0.8	1.16	10.4

Table 6-2: Error Rates and Bit-lengths for Different Numbers of Features

When different numbers of features are defined, i.e. $n = 43, 30, 20, 10$, the EER improves gradually. The EER is reduced from 11.77% to 8.57%, ***an improvement of 27.2%***. At the operating points, while the FAR remains constant at 0.8%, the FRR is reduced from 31.0% to 14.4%. Because of the reduced percentage of the FRR, the average successful tries to generate an authentic private key from handwritten signature are reduced from 1.44 to 1.16 times.

However, in contrast to the improvement of the error rates, the average bit-length decreases dramatically from 42.5 bits to 10.4 bits, ***a 75.5% drop***. This is because as fewer features involved in the feature coding, fewer feature codes are concatenated to form the code string. In conclusion, with the use of the personalized feature subset, the improvement of 27.2% in the EER is achieved at the expense of 75.5% drop in the average bit-length. Hence the personalized feature set is not recommended in the proposed BioPKI cryptosystem. In fact, we recommend to include more features into

the system to make the bit-length even longer, as we will explain this in details in Section 6.4.

6.3 Uniqueness and Security Strength of the Private Key

By nature, handwritten signatures may not be as unique as other types of biometrics, i.e. fingerprint, iris, and retina. When all the 43 features are used, the bit-information for an individual's signature is on average around 40 bits. Ideally if the data distributions of all the defined features are uniform, the uniqueness of a signature is 1 in 2^{40} . However the ideal uniformity is impossible and we could only choose features with relatively flat data distributions. Hence the actual uniqueness may be far less than 1 in 2^{40} .

Besides the problem of uniqueness, a 40-bit string may not be strong enough against brute-force attack if someone tries different combinations, i.e. one bit by bit. A binary string or a key with 40 bit key length is considered weak in cryptography. For a comparison, the symmetric encryption algorithm - DES (Digital Encryption Standard), defines a key with 56-bit. But a 56-bit DES key is no longer regarded as secure against brute-force attack [16]. The latest encryption standard, AES (Advanced Encryption Standard) defines a key with 128-bit. A 128-bit key is commonly regarded as robust against any brute-force attack for some years ahead [16]. To enhance the security strength and improve the uniqueness, some suggestions are presented in Section 6.4.

6.4 Suggested Remedies

In order to address the limitations on the uniqueness and the security strength, two remedies are suggested below:

1. Include more features into the system

The more features that are included, the more bit information will be added to the key length. In Table 6-2, we have used different numbers of features and achieved different key lengths. On average, the feature-to-bit-length ratio is about 1:1.01. If this ratio is preserved for the future selected features, we can tabulate the key lengths and the estimated numbers of features in Table 6-3, which will give us a guideline for future improvements.

Key length (bit)	10.4	20.9	30.5	42.5	...	56	112	128
No of features	10	20	30	43	...	55	111	127

Table 6-3: Key Length and the Estimated Number of Features

As shown in Table 6-3, to achieve the 128-bit strong security, a total of 127 features are required. However one may expect the False Rejection Ratio (FRR) to be higher since it would be more stringent to achieve all-bits-correctness.

2. Add padding information to the code string

A more feasible solution is by adding padding information. Before the 40-bit code string is hashed with SHA-1, it is attached with some padding information. The

padding information could be obtained from two methods: i) user's keying in, or ii) the template. For the former, the padding information will be the user name and pass phase. For the later, the template may save the timestamp (in milliseconds) of user's first-time registration, which is unique to each user. By either method, the uniqueness of the private key can be guaranteed. A more secure and reliable solution is the adoption of both methods.

6.5 Summary

In this chapter, we evaluate the system performance of the BioPKI cryptosystem. As far as the bit-length of the code string is concerned, the use of the entire 43 features instead of a personalized subset is recommended. The EER of the system is only 11.77%. The false alarm is at a reasonable level. A user may try on average 1.4 times to generate a correct private key. However, the system is relatively vulnerable to brute-force attack, because the average bit-length of the code string is only about 40 bits. To improve the uniqueness and robustness against the brute-force attack, we have suggested two remedies of the system for practical applications.

Chapter 7 Conclusions and Recommendations

7.1 Conclusions

A novel BioPKI cryptosystem, which dynamically generates the private keys from on-line handwritten signatures, has been implemented and evaluated. The system is an innovative way to combine biometrics and public key infrastructure (PKI). One particular application under the PKI is the use of digital signature. A sender needs a private key to sign an e-document and obtain a digital signature. The recipients will use the sender's public key to verify the digital signature. The private key has to be kept securely with the sender. Usually it is stored in a smart card or a PC and protected by a password. The proposed BioPKI cryptosystem offers a more secure and reliable way for the private key storage. In addition, it facilitates a user with great convenience for the use of the private key. The user doesn't need to bring around any smart card or memorize any password, since the private key can be derived dynamically from the hand signature. The success of the system is based on the fact that certain features for human handwritten signature are consistent.

The BioPKI cryptosystem comprises three processing stages: shape matching, feature coding and private key generation. In the shape matching stage, it exams the signature shapes and rules out poor-quality signatures. A new warping technique, Extreme Points Warping (EPW), has been proposed at this stage. A comparative evaluation shows that it is more effective than the conventional warping technique, Dynamic

Time Warping (DTW). With the use of EPW, the Equal Error Rate (EER) has been improved by 20% and the computation time has been reduced by a factor of 11.

In the second processing stage: feature coding, it extracts dynamic features to generate an all-bits-correct string. To achieve all-bits-correctness, we have proposed a new feature-coding scheme. Under the scheme, a dynamic feature is transformed into a feature code. A code string is formed, by concatenating the feature codes for all the defined dynamic features. We have defined 43 dynamic features at this stage. The data distribution is an important factor when considering the feature selection. Features with lumped data distributions are carefully excluded in the feature set. At this stage, a personalized feature subset proves to improve the EER. However, it is not recommended as it reduces the bit-length of the code string. A code string with short bit-length is vulnerable to brute-force attack. In the third processing stage: private key generation, it generates a private key based on the code string. The key generation follows some well-established public algorithms, e.g. RSA and DSA. The DSA has been used as an example to illustrate the generation process.

The overall system performance is encouraging. The EER is only 11.77%. At a chosen operating point on the error curve, the False Acceptance Ratio (FAR) is 0.8% and the False Rejection Ratio (FRR) is 31.0%. The 31.0% FRR shows that the false alarm is at a reasonable level. A user may need to try on average 1.4 times in order to get the authentic private key. The average bit-length of the code string is about 40 bits. A 40-bit code string may not be long enough to ensure uniqueness for a large number of users. In addition, the relatively short code string is not strong enough against the brute-force attack, in which an attacker tries out the code string bit by bit.

Hence for practical applications, we have proposed two remedies to improve the uniqueness and enhance the security strength against the brute-force attack.

7.2 Recommendations for Future Research

Three possible aspects of future research are suggested:

1. Inclusion of more dynamic features

To improve the uniqueness and the security strength, more dynamic features can be included at the feature coding stage. In our project, we didn't make use of the pressure signal. This is mainly because the tablet used in this project is too responsive to the pressure changes. As a result, the captured pressure signal is quite vibrant, which makes it unsuitable for use in the feature coding. With the advancement of the tablet hardware technology, the new-model tablet is likely to capture the stable pressure signals. Thus the features extracted from the pressure signal can be included into the feature coding, e.g. the mean, standard deviation, maximum, number of peaks and valleys of the pressure signal.

2. Segmentation of the signature signals

In the feature coding stage, the dynamic features are abstracted from the complete signature signals. In fact, the stabilities vary for the sampled points along the signals. Some parts of the signal may have large intra-personal variability, which will have some effect on the global features extracted from the signal. Hence it is suggested that a signature signal can be segmented based on the stabilities. The features will be extracted from the stable segments on the signal. In this way, the global features should be more consistent.

3. Investigation of signing in different environments

In the research, we had the users provide their signatures in the same environment. In other words, a user provides his/her signature samples in the same place using the same tablet. To deploy the BioPKI cryptosystem for an on-line application, e.g. e-commerce, one may expect that users sign from different places and use different tablets. In the future research, some additional preprocessing steps need to be investigated to deal with the difference in tablet size, resolution, data sampling rate, capture screen size and monitor resolution.

4. Adoption of other types of biometrics

In this project, we have mainly explored one type of biometrics: on-line signatures. There are also other types of biometrics, e.g. fingerprint, iris, keystroke, speech etc. Each type has the potential to implement Generating Private Keys from Biometrics (GPKB) applications. An enhanced BioPKI cryptosystem is likely based on the multimodal biometrics [63], which involves more than one type of biometrics.

Author's Publications

- Hao Feng, Chan Choong Wah, “Private Key Generation from On-line Handwritten Signatures”, *Information Management & Computer Security*, Vol. 10, no. 4, pp. 159-164, 2002.
- Hao Feng, Chan Choong Wah, “A Novel Usage of On-line Handwritten Signatures”, *Proceeding of the 6th WSEAS International Multi-conference on Communication, System, Computer and Circuit*, CSCC, Greece, pp. 4701-4705, July 2002.

The above paper is also included as one chapter of the following book:

- Hao Feng, Chan Choong Wah, “A Novel Usage of On-line Handwritten Signatures”, *Recent Advances in Circuits, Systems and Signal Processing*, WSEAS Press, pp. 344-348, 2002.

Bibliography

- [1] H. David, "An overview of biometrics support in NetWare Through NMAS",
[Online] Available at:
<http://developer.novell.com/research/appnotes/2001/july/01/a010701.pdf>, July
01 2001.
- [2] G. M. Johe, "Biometrics authentication", INSS 690 Professional Seminar,
[Online] Available at:
<http://faculty.ed.umuc.edu/~meinkej/inss690/messer/Paper.htm>, Oct 01 2000.
- [3] Ross J. Anderson, "Biometrics", *Security Engineering*, Wiley Computer
Publishing, pp. 272 - 286, 2001.
- [4] Biometric Market Report 2001, [Online] Available at:
http://www.ibgweb.com/reports/public/market_report.html, Sep 2002.
- [5] Ross J. Anderson, "Whither cryptography", *Information Management and
Computer Security*, Vol. 2, No 5, pp. 13-20, 1994.
- [6] Jeff, Stapleton, "PKI forum: biometrics", [Online] Available at:
<http://www.pkiforum.org/pdfs/biometricsweb.pdf>, May 2001.
- [7] K. J. Pawan, M. Y. Siyal, "Novel biometric digital signatures for internet
based applications", *Information Management and Computer Security*, Vol. 9,
No. 5, pp. 205 - 212, 2001.
- [8] White paper from CIC Corp, "Understanding Electronic Signature", [Online]
Available at:
<http://www.penop.com/enterprise/whitepapers/whitepaper1.asp>, 2001.
- [9] Whiter paper from CIC Corp, "On the distinction between biometric and
digital signatures", [Online] Available at:

- <http://www.penop.com/enterprise/whitepapers/whitepaper5.asp>, 2001.
- [10] Hao Feng, Xitao Cai, "Developing Java applets for security tools", Final Year Project in School of Electrical and Electronic Engineering, Nanyang Technological University 2001.
- [11] W. Diffie, M. Hellman, "New directions in cryptography", IEEE Transactions On Information Theory, vol. 22, pp. 644-654, 1976.
- [12] Zuoqin Wang, "Digital signature and its applications", Technical report, [Online] Available at:
<http://www.sims.berkeley.edu/courses/is290-1/f96/Student-reports/zwang/>, Dec 02 1996.
- [13] An introduction to E-Sign Act, [Online] Available at:
http://www.cybersign.com/news_news.htm#June2000esigact, June 2000.
- [14] R. Plamondon, *Process in Automatic Signature Verification*, World Scientific Publishing Co. Pte. Ltd, 1994.
- [15] L. L. Lee, T. Berger, "Reliable on-line human signature verification system for point-of-sales applications", Proceedings of the 12th IAPR International Conference on Computer Vision & Image Processing, vol. 2, pp. 19-23, 1994.
- [16] Bruce Schneier, *Applied Cryptography*, John Wiley & Sons, 2nd Edition, 1996.
- [17] NIST, "The digital signature standard proposed by NIST", Communication ACM, Vol. 35, No. 7, pp. 120-126, July 1992.
- [18] G. K. Gupta and R. C. Joyce. "A study of shape in dynamic handwritten signature verification", Technical Report, Computer Science Dept, James Cook University of North Queensland, 1997.

- [19] Quen-Zong Wu, Suh-Yin Lee and I-Chang Jou, "On-line signature verification based on logarithmic spectrum", *Pattern Recognition*, Vol. 31, No. 12, pp 1865-1871, Dec 1998.
- [20] M. Parizeau and R. Plamondon, "A comparative analysis of regional correlation, Dynamic Time Warping, and skeletal tree matching for signature verification," *IEEE Transactions On Pattern Analysis and Man*, Vol. 12, No. 7, pp. 710-717, July 1990.
- [21] J. H. Austin, C. Y. David, "Biometric authentication: assuring access to information", *Information Management & Computer Security*, Vol. 10, no. 1, pp. 12 – 19, 2000.
- [22] T. K. Worthington, T. J. Chainer, J. D. Williford, and S. C. Gundersen, "IBM dynamic signature verification", *Computer Security*, The Netherlands: North-Holland press, pp. 129-154, 1985.
- [23] R. Plamondon, G. Lorette, "On-line signature verification: how many countries are in the race", *Proceedings of the 1989 International Carnahan Conference on Security Technology*, pp. 183 –191, 1998.
- [24] H.D. Crane, J. S. Obstream, "Automatic signature verification using a three-axis-force-sensitive pen," *IEEE Transaction On Systems, Machine Intelligence And Cybernetics*, Vol. 13, no 2, pp 329-337, 1983.
- [25] R. Plamondon, G. Lorette, "Automatic Signature Verification and Writer Identification- The State of the Art", *Pattern Recognition*, Vol. 22, no. 2, pp. 107-131, 1989.
- [26] Ma Mingming, W. S. Wijesoma, "Automatic on-line signature verification based on multiple models", *Proceedings of the IEEE/IAFE/INFORMS 2000*

- Conference on Computational Intelligence for Financial Engineering, pp. 30 – 33, 2000.
- [27] C. Schmidt, K. Kraiss, “Establishment of personalized templates for automatic signature verification”, Proceedings of the 4th International Conference Document Analysis and Recognition, Vol. 1, pp. 263 –267, 1997.
 - [28] Kim Jaihie, J. R. Yu, S. H. Kim, "Learning of prototypes and decision boundaries for a verification problem having only positive samples," Pattern Recognition Letters, Vol.17, No. 7, pp.691-697, June 1996.
 - [29] S. Hangai, S. Yamanaka, T. Hammamoto, “Writer verification using altitude and direction of pen movement”, Proceedings of the 15th International Conference on Pattern Recognition, Vol. 3, pp. 479 –482, 2000.
 - [30] R. Martens, L. Claesen, “Dynamic programming optimisation for on-line signature verification”, Proceedings of the 4th International Conference on Document Analysis and Recognition, Vol. 2 , pp. 653 –656, 1997.
 - [31] R. Martens, L. Claesen, “On-line signature verification by dynamic time-warping”, Proceedings of the 13th International Conference on Pattern Recognition, Vol. 3, pp. 38 –42, 1996.
 - [32] Wacom Technology Co. Ltd.
<http://www.wacom.com>
 - [33] N. Mohankrishnan, Wan-Suck Lee, M.J. Paulik, “Multi-layer neural network classification of on-line signatures”, IEEE 39th Midwest Symposium on Circuits and Systems, Vol. 2, pp. 831 –834, 1996.
 - [34] Wan-Suck Lee, N. Mohankrishnan, M. J. Paulik, “Improved segmentation through dynamic time warping for signature verification using a neural

- network classifier”, Proceedings of International Conference on Image Processing, Vol. 2, pp. 929 –933, 1998.
- [35] N. Mohankrishnan, Wan-Suck Lee, M. J. Paulik, “A performance evaluation of a new signature verification algorithm using realistic forgeries”, Proceedings of International Conference on Image Processing, Vol. 1, pp. 575 –579, 1999.
 - [36] F. Leclerc, R. Plamondon, “Automatic signature verification: the state of the art – 1989-1993,” International Journal of Pattern Recognition and Artificial Intelligence, Vol. 8, No. 3, pp. 643-660, 1994.
 - [37] P. Zhao, A. Higashi, and Y. Sato, “On-line signature verification by Adaptively Weighted DP Matching”, IEICE Transaction on Information System, vol. E79-D, no. 5, pp. 535-541, May 1996.
 - [38] V. S. Nalwa, “Automatic on-line signature verification”, Proceedings of the IEEE, Vol. 85, no. 2, pp. 215-239, Feb 1997.
 - [39] T. Hastie, E. Kishon, M. Clark, J. Fan, “A model for signature verification”, Proceedings of IEEE International Conference on System, Man and Cybernetics, vol. 1 pp. 191-196, 1991.
 - [40] D. Sakamoto, H. Morita, T. Ohishi, Y. Komiya, T. Matsumoto, “On-line signature verification algorithm incorporating pen position, pen pressure and pen inclination trajectories”, Proceedings of 2001 IEEE International conference on Acoustics, Speech, and Signal Processing, vol. 2, pp. 993-996, 2001.
 - [41] F. Bauer, B. Wirtz, “Parameter reduction and personalized parameter selection for automatic signature verification”, Proceedings of the Third International

- Conference on Document Analysis and Recognition, Vol. 1, pp. 183-186, 1995.
- [42] B. Wirtz, "Stroke-based time warping for signature verification", Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 179-182, 1995.
 - [43] Quen-Zong Wu, Suh-Yin Lee and I-Chang Jou, "On-line signature verification based on split-and-merge matching mechanism", Pattern Recognition Letters, Volume 18, no. 7, pp. 665-673, July 1997.
 - [44] Seong Hoon Kim, Myoung Soo Park, Jaihie Kim, "Applying personalized weights to a feature set for on-line signature verification", Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 2, pp. 882-885, 1995.
 - [45] L. L. Lee, T. Berger, E. Aviczer, "Reliable online human signature verification systems", IEEE Transactions on Pattern Analysis and Man, Vol. 18, no. 6, pp. 643 – 647, June 1996.
 - [46] Anil K. Jain, Friederike D. Griess and Scott D. Connell, "On-line signature verification", Pattern Recognition, Vol. 35, no. 12, pp. 2963-2972, Dec 2002.
 - [47] L. Yang, B. K. Widjaja and R. Prasad, "Application of hidden Markov models for signature verification", Pattern Recognition, Vol. 28, No. 2, pp. 161-170, Feb 1995.
 - [48] J. G. A. Dolfing, E. H. L. Aarts, J. J. G. M. Van Oosterhout, "On-line signature verification with hidden Markov models", Proceedings of the Fourteenth International Conference on Pattern Recognition, Vol. 2, pp. 1309 –1312, 1998.

- [49] J. Brault, R. Plamondon, "Segmenting handwritten signatures at their perceptually important points", IEEE Transactions on Pattern Analysis and Man, Vol. 15, No. 9, pp. 953 –957, Sep 1993.
- [50] Y. Sato, K. Kogure, "Online signature verification based on shape, motion and writing pressure", Proceedings of the 6th ICPR, pp.823-826, Oct 1982.
- [51] G. V. Kiran, R. S. R. Kunte, S. Samuel, "On-line signature verification system using probabilistic feature modelling", International Symposium on Signal Processing and its Applications, pp.355-358, Aug 2000.
- [52] T. Rhee, S. Cho and J. Kim, "On-line signature verification using model-guided segmentation and discriminative feature selection for skilled forgeries", 6th International Conference on Document Analysis and Recognition, Seattle, WA, pp. 645-649, Sep 2001.
- [53] W. Wijesoma, Sardha, M. M. Ma, E. Sung, "Selecting optimal personalized features for on-line signature verification using GA", Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Nashville, TN, USA, pp. 2740-2745, Oct 2000.
- [54] D. Z. Lejtman, S. E. George, "On-line handwritten signature verification using wavelets and back-propagation neural networks", Proceedings of Sixth International Conference on Document Analysis and Recognition, pp. 992 – 996, 2001.
- [55] G. Lorette, R. Plamondon, "Dynamic approaches to handwritten signature verification", Computer Processing of Handwriting, World Scientific Press, pp. 21-47, 1990.

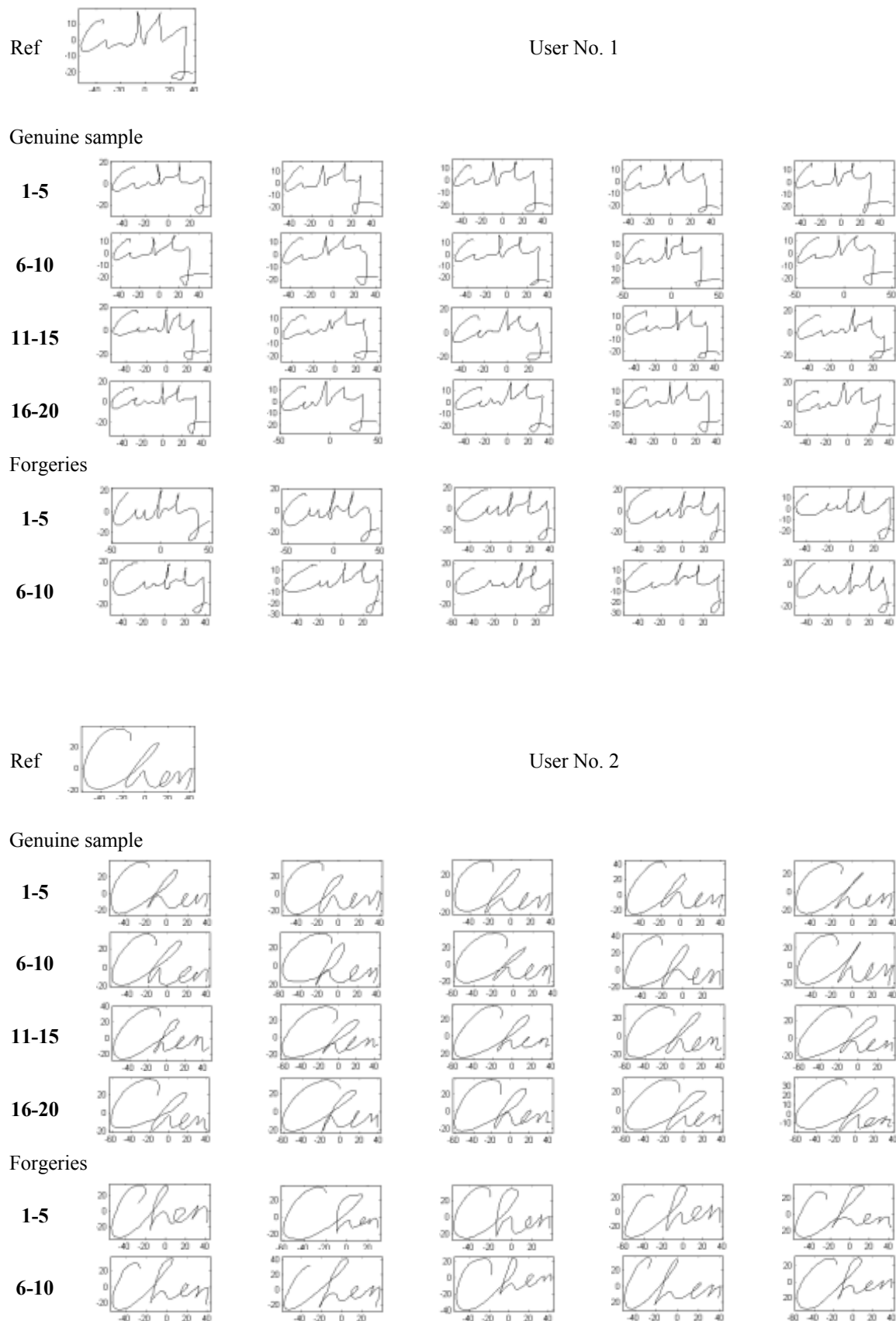
- [56] M. J. Paulik, N. Mohankrishnan, M. Nikiforuk, "A time varying vector autoregressive model for signature verification", Proceedings of the 37th Midwest Symposium on Circuits and Systems, Vol. 2, pp. 1395 –1398, 1994.
- [57] D. Sankoff and J. B. Kruskal, *Time Warps, String Edits, and Macromolecules: The Theory and Practice of Sequence Comparison*, Addison-Wesley Publishing, pp. 125-160, 1983.
- [58] Douglas O'Shaughnessy, *Speech Communications: Human and Machine*, IEEE Press, pp. 389-399, 2000.
- [59] I. Pottier, G. Burel, "Identification and authentication of handwritten signatures with a connectionist approach", 1994 IEEE International Conference on Neural Networks, Vol. 5, pp. 2948 –2951, 1994.
- [60] Ke Han, I. K. Sethi, "Signature identification via local association of features", Proceedings of the Third International Conference on Document Analysis and Recognition, Vol. 1, pp. 187 –190, 1995.
- [61] M. J. Paulik, N. Mohankrishnan, "A 1-D, sequence decomposition based, autoregressive hidden Markov model for dynamic signature identification and verification", Proceedings of the 36th Midwest Symposium on Circuits and Systems, vol. 1, pp. 138 –141, 1993.
- [62] An introduction of the distance measures, [Online] Available at: http://www.clustan.com/general_distances.html, 2001.
- [63] R. W. Frischholz, U. Diechmann, "BioID: a multimodal biometrics identification system", Computer, Vol. 33, no. 2, pp. 64-68, Feb 2000.

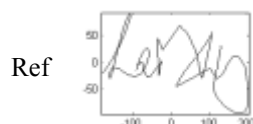
Appendix A: A Summary of On-line Signature Verification Projects During 1994~2002

Authors	Preprocess	Feature Extraction	Database		Functional Comp	Parametric Comp	Error Rate	Ref
			Gen.	For.				
A.K. Jain (2002)	Smoothing, Re-sampling	F: $\delta x, \delta y, \sin \alpha, \cos \alpha$ v P: stroke No.	520	60	Weighted Eucli-dist after DTW	Eucli-dist	FRR=2.8% FAR=1.6%	[46]
Z. D. Lejtman (2001)	Re-sampling, Size	P: 80 coefs from Wavelet transform	922/2	922/2	–	3-layer back-propagation neural network	FRR=0.0% FAR=0.08%	[54]
G.V. Kiran (2001)	NA	P: 10 features	NA	NA	–	Accumulation of probabilistic scores	FRR=0% FAR=5%	[51]
D. Sakamoto (2001)	Corner-preserving Re-sampling	F: θ , length, p, Al, Az	861	1921	Eucli-dist etc after DTW	–	EER=3.0%	[40]
T.H. Rhee (2001)	Re-sampling, Size	P: 6 and 5 features for skilled and random forgery in each segment	1000	1000	–	Eucli-dist	EER=3.4%	[52]
W.S. Wijesoma (2000)	Translation, Orientation	P: Use GA to select a personal set from 10 static and 14 dynamic features	1230	410	–	Fuzzy logic classifier	EER=4.45%	[53]
M.M Ma (2000)	Translation, Orientation	F: DFT of x, y P: Use DP function to select 10 personal features from 24	1230	410	Weighted correlation	Fuzzy logic classifier	EER=4.6%	[26]
S. Hangai (2000)	Translation, Duration	F: 3D-v X, y, p	600	480	Eucli-dist after DTW	–	EER=0%	[29]
N. Mohan-krishnan (1999)	256-point spatial re-sampling	P: segmentation by DTW, 16 AR coeffs + 2 features for each segment	2400	1920	–	16-neuron neural network	FRR=0.78% FAR=1.6%	[35]
Q.Z. Wu (1998)	Size, Re-sampling	P: Log FFT coefs for each of frames	270 560	650	–	Dynamic similarity function as classifier	FRR=1.4% FAR=2.8%	[19]
J.G.A Dolfing (1998)	Duration	P: segmentation by $V_y=0$, 32-feature vector for each segment	1530	240 1530 1470	–	Log likelihood from HMM	EER=1.0~1.9%	[48]

Authors	Preprocess	Feature Extraction	Database		Functional Comp	Parametric Comp	Error Rate	Ref
			Gen.	For.				
R. Martens (1997)	Smoothing, Re-sampling, Orientation,	F: 3 forces, Al, Az	360	Ran.	Shape and motion measures after asymmetric DTW	–	EER=8%	[30]
V.S Nalwa (1997)	Size, Orientation	F: x, y, torque and moments of inertia	1452	1150	Weighted correlation after DTW	–	EER=3.6%	[38]
Q.Z. Wu (1997)	NA	F: x, y, Vx, Vy	200	246	Eucli-dist after split-merge matching	–	FRR=13.5% FAR=2.8%	[43]
P. Zhao (1996)	Duration, Translation, Orientation, Size and 256-point Re-sampling	F: x, y, p	300	300	Shape and motion measures with weight after DTW	–	EER=1.3%	[37]
L.L. Lee (1996)	NA	P: use Euclid-dist and K-L to select personal features from 49	5603	1148	–	Majority classifier based on Eucli-dist	EER=2.5%	[45]
B. Wirtz (1995)	Re-sampling	F: x, y	6000	6000	Shape measure after stroke-based DTW	–	EER=9.9%	[42]
F. Bauer (1995)	NA	F: x, y, p P: Redundancy and Fisher's F test to select 84 common features from 300	644	669	Shape and motion measures after DTW	–	EER=6.6%	[41]
L. Yang (1995)	Orientation, Size	F: directional angle along length	496	Ran	Probability that a sample is from a ref HMM	–	FRR=1.75% FAR=4.44%	[47]
S.H. Kim (1995)	NA	P: User forward method to select 23 common features from 76	1080	1080	–	Eucli-dist with personal weights	EER=4.28%	[44]
M. J. Paulik (1994)	Translation, 512-point re-sampling, Remove end points	P: Vector AR model coefs for each of 8 segments	100	Ran	–	Distance between coef matrices	Best EER=2.87%	[56]
L.L. Lee (1994)	Duration, Size	P: Use Eucli-dist to select 15 personal features from 42 or 49	5603	4762	–	Majority classifier based on Eucli-dist	EER=10%	[15]

Appendix B: Some Signature Examples in the Database

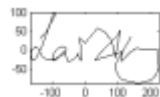
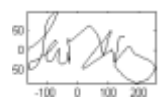
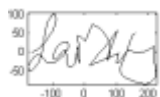
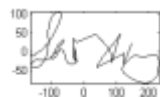




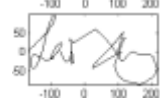
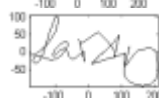
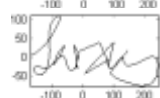
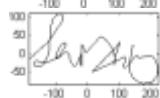
User No. 9

Genuine sample

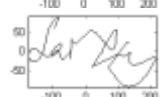
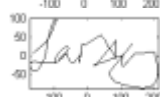
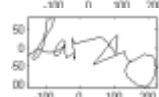
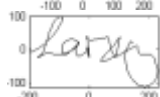
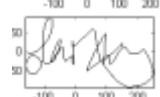
1-5



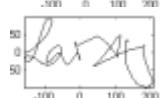
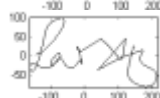
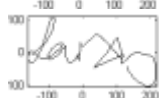
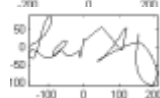
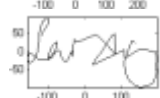
6-10



11-15

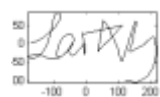
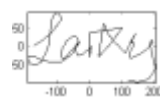
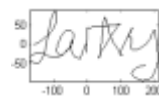
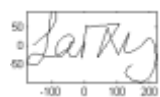


16-20

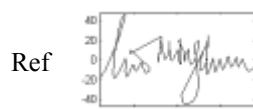
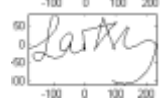
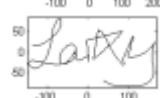
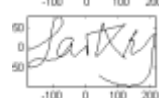
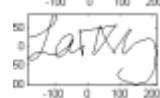
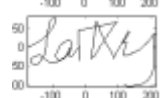


Forgeries

1-5



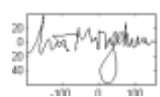
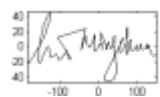
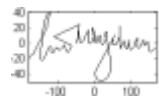
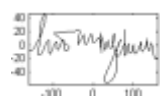
6-10



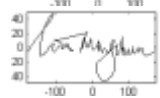
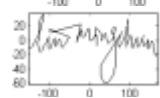
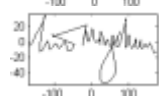
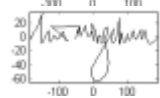
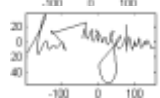
User No. 10

Genuine sample

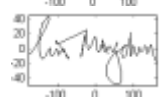
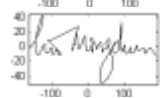
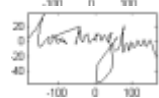
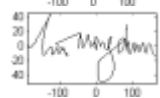
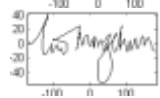
1-5



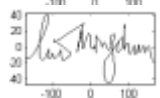
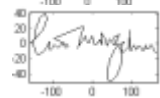
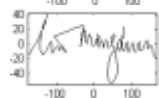
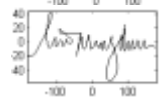
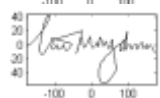
6-10



11-15

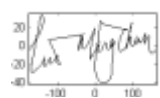
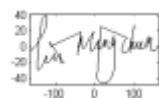
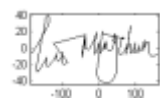
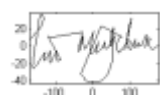


16-20

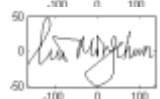
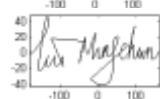
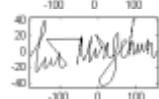
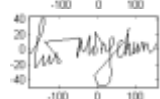
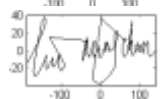


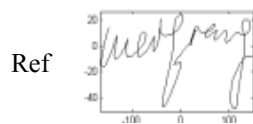
Forgeries

1-5



6-10

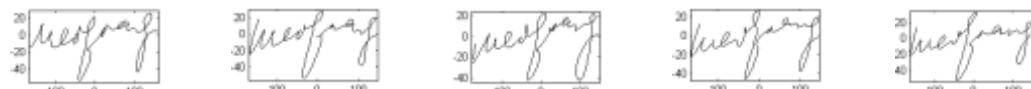




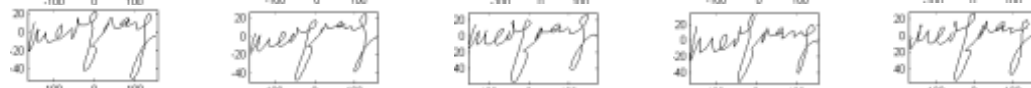
User No. 19

Genuine sample

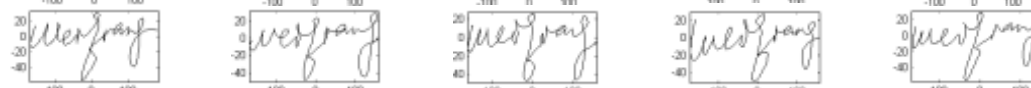
1-5



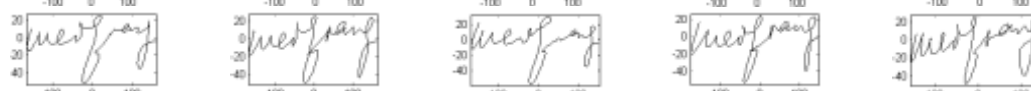
6-10



11-15



16-20



Forgeries

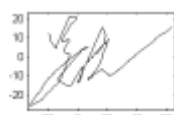
1-5



6-10



Ref



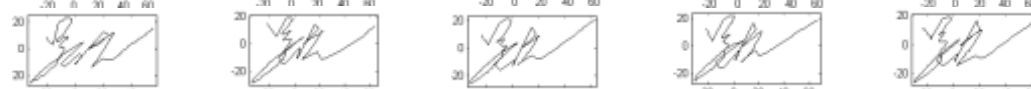
User No. 20

Genuine sample

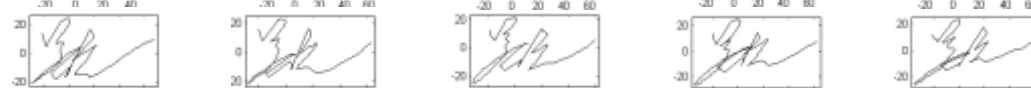
1-5



6-10



11-15



16-20



Forgeries

1-5



6-10



Appendix C: Execution time using DTW and EPW

DTW (ms)

Users	1	2	3	4	5	6	7	8	9	10
X	420.6	418.1	420.7	429.7	424.2	422.6	419.4	428.6	428	420.6
Y	424.7	421.7	420.3	423.7	427.6	425.3	427.7	425.9	420.6	426.4
X_C	409.4	408.7	411.3	403.7	411.1	408.3	406.7	408.1	410.1	412.1
Y_C	411.7	405.8	407.3	413.8	407.9	410.2	412.2	410.6	405.8	405.4
Torque	412.9	418.4	413.1	409.2	413	409.6	410.2	408.2	416.8	415.4
Ave	415.9	414.5	414.5	416.0	416.8	415.2	415.2	416.3	416.3	416.0

Users	11	12	13	14	15	16	17	18	19	20
X	419.6	426.2	423.4	425.4	428.1	422.9	421.8	425.9	427	422.9
Y	418	422.9	417.8	422.6	418.8	424.9	429.2	426.7	427.6	423.6
X_C	417.3	412.3	410.6	413.1	413.4	412.4	412.1	402.3	405.4	404.4
Y_C	407.1	407	407.6	408.3	403.7	409	413.6	415.2	412.4	413.6
Torque	413.4	414.8	411.7	413	416	415.6	408.2	405.3	409.6	408.9
Ave	415.1	416.6	414.2	416.5	416	417	417	415.1	416.4	414.7

Users	21	22	23	24	25
X	422.9	423.2	427.3	422.1	420.8
Y	423.3	422.2	424.4	425.9	428.3
X_C	406.3	412.1	409.7	407.9	412.9
Y_C	408.4	409.9	406.6	412.8	409.2
Torque	411.7	410	410.8	407.1	412.3
Ave	414.5	415.5	415.8	415.2	416.7

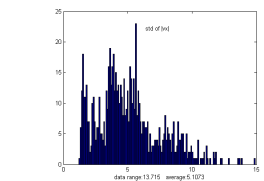
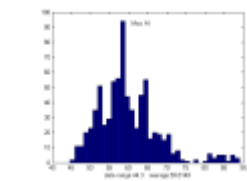
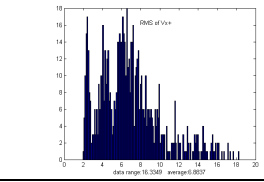
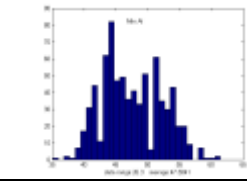
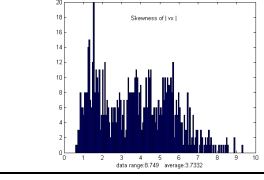
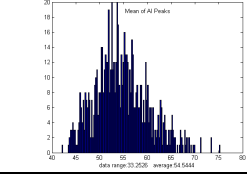
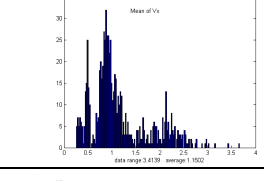
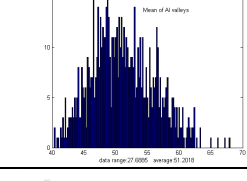
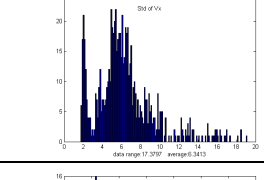
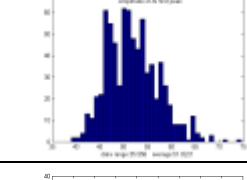
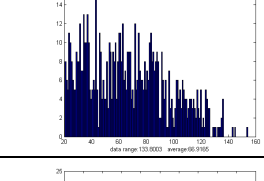
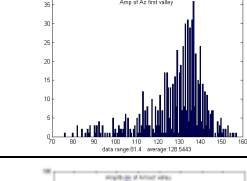
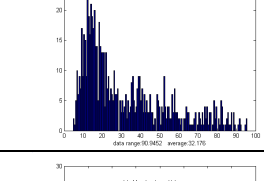
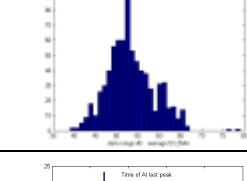
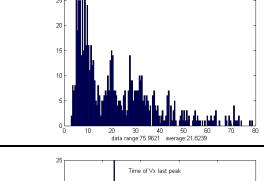
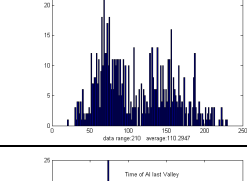
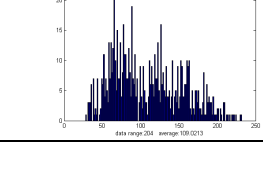
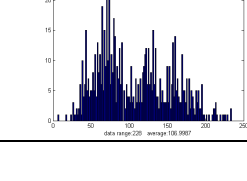
EPW (ms)

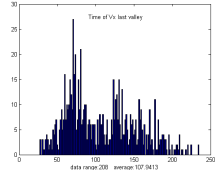
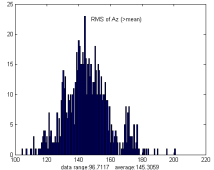
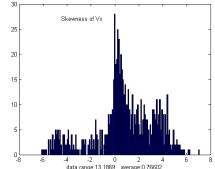
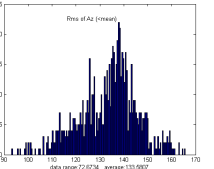
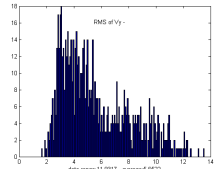
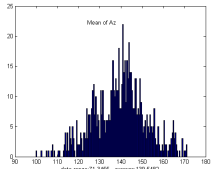
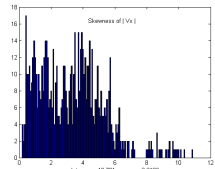
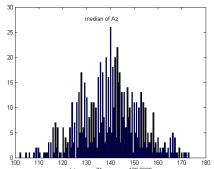
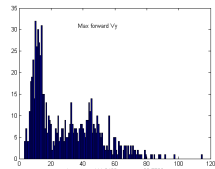
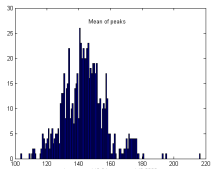
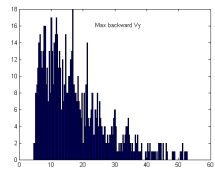
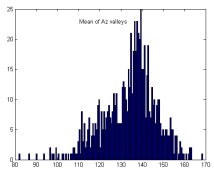
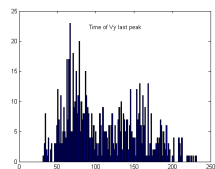
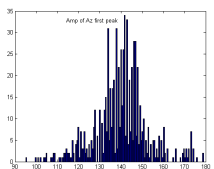
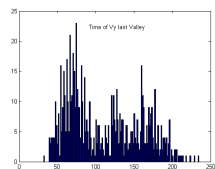
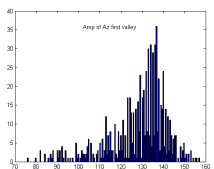
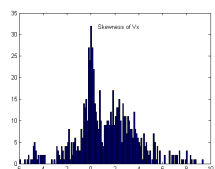
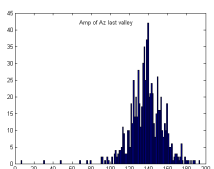
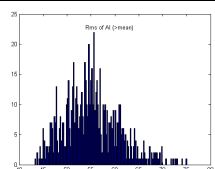
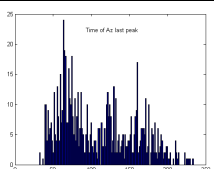
Users	1	2	3	4	5	6	7	8	9	10
X	10	19.1	16.4	46.8	48	24.3	74.3	37.4	19.9	22.6
Y	18.3	12.2	14	38.6	36.3	30.1	78.1	37.4	54.2	104.6
X_C	7.9	8.4	13.8	26.8	37.6	19.6	41.2	30	13	16.7
Y_C	16.2	12.7	14.7	36.6	26.2	27.7	53.7	21.8	26	66.2
Torque	26.8	36.9	27.3	71.4	106	41.7	108.1	79.4	38.9	93.9
Ave	15.8	17.9	17.2	44.0	50.8	28.7	71.1	41.2	30.4	60.8

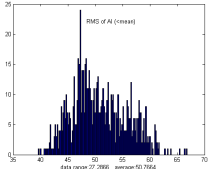
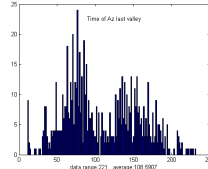
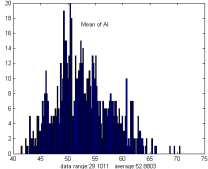
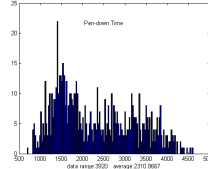
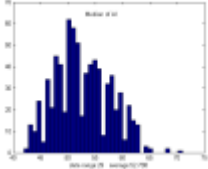
Users	11	12	13	14	15	16	17	18	19	20
X	44.7	17.1	13.7	43.7	14.2	37.9	27	14.3	51.3	25.9
Y	47.6	26.3	8.7	31.1	20.2	69.7	90.8	13.6	55.3	17.8
X_C	19.6	10.4	8.3	37.2	9.1	18.9	18.6	10.9	24.4	16.4
Y_C	41.9	19.9	12.3	18.6	15.9	52	24.9	12.7	48.1	19.3
Torque	103.6	42.6	24	67.6	31.4	70.6	87.7	26.2	75.3	44.1
Ave	51.5	23.3	13.4	39.6	18.2	49.8	49.8	15.5	50.9	24.7

Users	21	22	23	24	25
X	18.9	20.6	42.7	21.8	112.1
Y	10.9	19.9	55	16.2	88.1
X_C	12.8	12.6	29.9	20.4	50.2
Y_C	14	23.3	32.9	19.2	45
Torque	55.2	28.7	80.4	40.4	180.3
Ave	22.4	21.0	48.2	23.6	95.1

Appendix D: Histograms of the 43 Features

1	Std of $ V_x $		23	Max A1	
2	RMS of V_{x+}		24	Min A1	
3	Skewness of $ V_x $		25	Mean of A1 peaks	
4	Mean of V_x		26	Mean of A1 valleys	
5	Std of V_x		27	Amplitude of A1 first peak	
6	RMS of V_x		28	Amplitude of A1 first valley	
7	Max forward V_x		29	Amplitude of A1 last valley	
8	Max backward V_x		30	Time of A1 last peak	
9	Time of V_x last peak		31	Time of A1 last valley	

10	Time of Vx last valley		32	RMS of Az+	
11	Skewness of Vx		33	RMS of Az-	
12	RMS of Vy-		34	Mean of Az	
13	Skewness of Vy		35	Median of Az	
14	Max forward Vy		36	Mean of Az peaks	
15	Max backward Vy		37	Mean of Az valleys	
16	Time of Vy last peak		38	Amplitude of Az first peak	
17	Time of Vy last valley		39	Amplitude of Az first valley	
18	Skewness of Vy		40	Amplitude of Az last valley	
19	RMS of A1+		41	Time of Az last peak	

20	RMS of A1-		42	Time of Az last valley	
21	Mean of A1		43	Pen down time	
22	Median of A1				

Appendix E: Technical Terms Used in the Thesis

Biometrics:	A technology that automates the identification of a person by analysing their physical or behavioural traits
BioPKI:	The cryptosystem we proposed in the research, which dynamically generate private keys from the on-line handwritten signature
Digital signature:	A checksum which depends on all the bits of transmitted e-document, and also on a secret (or private) key, but which can be checked without knowledge of the secret key
DSA:	Digital Signature Algorithm
DTW:	Dynamic Time Warping
EER:	Equal Error Rate
EPW:	Extreme Points Warping – a new warping technique proposed in the research to replace Dynamic Time Warping
EP:	Extreme Point
EPs:	Extreme Points
FAR:	False Acceptance Ratio
FRR:	False Rejection Ratio
PKI:	Private Key Infrastructure
SHA-1:	Secure Hashing Algorithm