MONASH UNIVERSITY

Fingerprint based Biometric Cryptosystem using the Fuzzy Embedder

Minor Thesis

Dylan Field - 21477795

Masters of Information Technology (Honours)

April, 2014

Supervisors:

Nandita Bhattacharjee

Balasubramaniam Srinivasan

Abstract

The *Fuzzy Embedder* is a theoretical tool for embedding cryptographic keys into noisy sources of data. This extends work of the *Fuzzy Extractor* by adding renewability and support for continuous data input by use of *Quantization Index Modulation* (QIM). This method has been adopted for use with fingerprint biometrics as a template protection scheme to securely store and match samples under the encrypted domain.

To create a workable environment to test the implementation a number of components were required. The segmentation, enhancement and alignment algorithms were sourced and consolidation and QIM were constructed by the author. This includes the construction of the honeycomb lattice, indexing scheme and modified embed and reproduce functions. Fingerprint samples used were extracted by the author due to the purposely difficult competition fingerprints databases (FVC). This is not a result of QIM itself, but a trickle effect of errors through all components of the system.

The Fuzzy Embedder has been modified to accept ISO-compliant minutiae templates for interoperability. A limitation of the Fuzzy Embedder is it only supports fixed-length, ordered data sets. The solution has been considered to create an indexing scheme that is both reliable and secure. Results over 80-fingerprint samples have shown preliminary results of QIM to be satisfying, however requires more work for practical consideration.

All images in this document belong to the author.

Declaration of Originality

I declare that this thesis is my own work and has not been submitted in any form for another degree or diploma at any university or other institute of tertiary education. Information derived from the work of others has been acknowledged.

Signed,

Dylan George Field
November 15th, 2013

Table of Contents

1. Introduction

Biometrics refers to the automated measurement of individual's behavioural or physiological traits for identification. Voice, signature, face, iris, fingerprint and recently palm-vein are the commonly used and researched traits. Each carries strengths and weaknesses based on their: uniqueness, permanence, collectability, performance, acceptability and circumvention [1]. The selection for a biometric system is highly dependent on the application requirements. An identification system, comparison of one-to-many, requires highly distinctive traits. They must differ substantially between each user to reduce error. Performance of such a system may also require low resource costs and robustness due to environmental factors.

Interest in biometrics is due to the high level of assurance provided to identify an individual. In contrast to traditional credential mechanisms such as passphrases and PIN numbers, biometrics cannot be; lost, forgotten or stolen. They are highly immutable and can guarantee a level of repudiation. An example of biometric password-replacement is evident in recent smartphone technology and is adapted to various logical, physical accesses and identification systems such as; door locks, bank cards, passports and national voting systems.

Fingerprint biometrics is the focus of study due to their growing popularity, balanced properties and low cost. The International Biometric Group, 2009 report projects market growth in biometrics of $3.42 billion to $9.37 billion in 2014, driven by government identity management and border protection programs. Fingerprint biometrics comprising of 45.9% of market share, followed by facial (18.5%) and iris (8.3%) [2]. Universally, each person has at most ten different fingerprints that are highly distinctive to each individual. Fingerprint sensors are inexpensive, unobtrusive and well accepted by users.

Unlike a passphrase, biometric traits cannot be revoked or renewed. A compromised biometric sample results in its permanent loss and can have severe repercussions of identity theft and cross-organisational matching. A stolen fingerprint could be used in any application it is enrolled or registered for new ones. This is common over all biometric systems such as forgery of signatures or stolen photo identification cards. Passphrases, however, can be changed per application. A user can choose the password "1234" for one application and "5678" for another. Biometrics is the equivalent of always choosing the same password but of much higher difficulty. The fast adaptation of biometric technologies compounds the risk of exploitation due to the infancy of the technology.

This research focuses on the protection of fingerprint templates for both matching and cryptographic applications. The aim of the project is to investigate the theoretical application of the Fuzzy Embedder proposed by Buhan et al using a practical implementation. The Fuzzy Embedder is a generalised scheme that can be applied to a number of biometric traits and/or representations. This implementation will focus on standard ISO fingerprint minutiae templates and will be compared to similar template protection schemes. The goals of the project are to assess its security, performance and identify its capability, limitations and future direction for research.

2. Biometric Concepts and Challenges

A typical system involves two steps; *enrolment* and *identification/verification*. Enrolment requires the capture of the biometric trait and extraction of features - stored as a template. Verification involves the same process and a comparison to the enrolled template. In a fingerprint system, the most common feature extraction technique is based on *minutiae*. Fingerprints are made up of unique patterns of ridges and valleys. The minutiae points are the ridge discontinuation are called: endings and bifurcations. This method, reviewed in detail by Bansal et al. [24] and practically implemented by Jain et al. [27], is the supported method of feature extraction by the International Organisation for Standardisation (ISO) for their common interchange format 19794-2:2005 [3]. It was assumed the templates did not contain enough information to reconstruct fingerprint images, Cappelli et al. proved this incorrect with an average successful attack on nine different systems to be 81% [4]. This, coupled the possibility to create gelatine fingerprints [5] creates serious consequences if biometric templates can compromised. Templates therefore need to be stored in encrypted format. However, it's impossible to compare them within a standard encrypted domain.

Modern cryptosystems (AES or RSA) require bit-precision. A change in a single-bit results in drastically changed output [9]. The problem results from biometrics being naturally unreliable. It is extremely rare to produce two samples of the same fingerprint precisely. Uludag et al. explain this to be caused by the; acquisition method, environment and users interaction with the device. They explain the finger is not a rigid surface and therefore cannot be precisely controlled [8]. Thus, encrypting and matching biometric queries will yield wildly different results on each presentation. Templates can be encrypted and decrypted prior to matching. However, this adds a layer of complexity and inconvenience to store and retrieve the encryption key.

Passphrases are the traditional mechanism to unlock encryption keys. Such a system would inherit their weaknesses. Studies show users constantly use simplistic, easily predictable practices when constructing passwords [7]. It is difficult for users to retain high strength passwords and thereby increasing administrative costs to change them. Securing a biometric template using a key and passphrase should be avoided as template will become exposed when decrypted for verification or identification. If the passphrase is compromised, so is the template. This violates the goal of biometrics for non-repudiation as anyone can input the password and use the fingerprint template; even if it's not their own. In comparison, a biometric cannot is very difficult to fraud if the template can be stored and verified securely. The solution in literature is the design of template protection schemes.

3. Template Protection Schemes

A template protection scheme is a mechanism to secure a template and posses the following four properties [6].

- Diversity: the secure template must not allow cross-matching across databases, thereby ensuring user privacy.
- Revocability: it should be straightforward to revoke a compromised template and reissue a new one based on the same biometric data.
- Security: it must be computationally hard to obtain the original biometric template from the secure template. This property prevents an adversary from creating a physical spoof of the biometric trait from a stolen template.
- Performance: the biometric template protection scheme should not degrade the recognition performance (FAR and FRR) of the biometric system.

3.1 Cancellable Biometrics

In literature, the schemes fall under the categories; *cancellable biometrics* and *biometric cryptosystems*. Cancellable biometrics is the intentional, repeatable distortion of a biometric signal based on a chosen transform [11]. They are a one-way function, similar to a cryptographic hash. Biometric inputs are transformed and compared with a template using the same transformation. During enrolment, the transformation parameters are chosen. These must be stored securely and are required for each verification. These non-invertible transforms are shown for fingerprints by Ratha et al. [12] by conducting Cartesian, polar and surface folding transformations of the minutiae positions with good performance. Similar results can be seen for iris [13][14]. Biometric Salting is a similar technique; however the transform is based on a user-specified password. The password helps reduce false acceptance rates by increasing the entropy of the template shown in PalmHash [18]. It also allows for revocability. However, if the password is compromised, the template is vulnerable.

3.2 Biometric Cryptosystems

Biometric cryptosystems secure cryptographic keys using biometrics. Biometric variance does not make it feasible to extract keys directly. The techniques rely on the storage of public information known as helper data [6]. This data does not (should not) reveal enough information to reconstruct the biometric template or cryptographic key. It should be computationally infeasible to decode the key or template without the correct biometric input. The system provides matching indirectly by validating the extracted key. This is a form of template security as matching is conducted within the encrypted domain.

Two types of biometric cryptosystems exist in literature; *key binding* and *key generation*. In a key binding system, the biometric template is combined with a key. Helper data is the result of this embedding process. This is typically the association of an error correcting code, selected using the key, and biometric template. If a biometric query differs within a specified threshold, the associated codeword with similar error can be recovered. This is decoded to obtain the exact codeword to produce the key [6]. The use of error correction schemes is common and allows tolerance of biometric variance. Generally, the biometric cryptosystems do not provide revocability. Password hardening technique [19] is a measure to alleviate this. However, is more inconvenient. The two most popular primitives are Juels & Wattenberg's Fuzzy Commitment Scheme [16] and Fuzzy Vault by Juels & Sudan [17].

3.4 Key Binding Schemes

A commitment scheme is both *concealing* and *binding*. That is, once a value is committed by the committee it is computationally infeasible to view or change. The committed value can only be unlocked using a witness value (biometric). For instance, playing 'rock, paper, scissors' over email. Alice chooses 'rock' and sends her choice locked in the commitment scheme. Bob receives the choice but cannot view it. He replies with 'scissors'. Alice sends a witness value to Bob for him to unlock her choice. Without the scheme, one could simply change their result based on the first reply and win the game. If Alice sent 'rock', knowing that information, Bob can reply 'paper'.

The *Fuzzy Commitment Scheme* is motivated by Davida et al's problem of secure storage of biometric data [20] and error correcting codes. It's termed "fuzzy" as the witness value to unlock the committed value needs only be similar to that which locked it. This is to account for variations in biometric input and is treated as a corrupted codeword. The robustness depends on a distance vector (Hamming distance) between codewords and is provided by public helper data. A cryptographic hash is used to validate successful decoding (or decommitment). In this system, a user can commit a secret key with a fingerprint witness used to unlock it. Several methods of the commitment scheme have been implemented for; iris [21], signature [22] and face [23]. However, the commitment scheme requires ordered datasets. This is inappropriate for fingerprints as it's difficult to introduce order of minutiae measurements due to capture difficulties.

The *Fuzzy Vault* is the by Juels & Sudan [17] improves upon this by being order invariant and improves security over non-uniform distributions. This scheme can be thought of as an error-tolerant encryption operation where keys consist of sets, opposed to sequences. The vault locks a key under a set A. A polynomial encodes the key by embedding it as its coefficients. Security rests on the polynomial reconstruction problem, based on Shamir's secret sharing [24] and similar to Monrose et al's hardening of passwords using keystroke dynamics [25]. The elements of *A* are projected as points of the polynomial. Random 'chaff points' are created that do not lie on p. The entire collection of points is R. By selecting the correct points concealed in R by chaff points will reveal the polynomial and thus, the key. The approach does not outline the affects of fingerprint alignment. A fingerprint minutiae based fuzzy vault by Nandakumar et al. [26] use high curvature points derived from orientation fields as helper data to assist alignment. Nagar et al. improved upon this work to include a hybrid scheme by using minutiae descriptors by Feng [29]. This information includes orientation and ridge frequency information in a minutia's neighbourhood to secure the polynomial evaluation using the commitment scheme increasing security and performance. The fuzzy vault has drawbacks in its construction. If the same biometric data is used with different polynomial and chaff points, the genuine points can be easily identified by correlating the values [32]. The non-uniformity of biometric features makes it possible to identify the genuine set from the chaff point set using statistical analysis [33].

3.5 Key Generation Schemes

An alternate method to key binding is to generate a key directly from biometrics. The method was first introduced by Bodo [30] however, unreliability of biometrics at the time hindered this. As discussed by Janbandhu and Siyal [31] a generation scheme would not require storage of either key or biometric as they are generated at the time of presentation. This can have applications in PKI environment such as authentication or digital signatures using only a biometric. Generating keys requires biometrics to be accurately repeatable. This is not the case. Research into the use of error-correcting codes has significantly improved reliability. *The Fuzzy Extractor* by Dodis et al. [15] is a scheme that can reliably extract nearly uniform randomness from a non-uniform source in an error-tolerant way. This has application to biometrics and functions similar to a cryptographic one-way hash. These functions take high-entropy non-uniform sources and yield smaller uniform output based on concepts of *min-entropy* and *statistical distance*. With a zero level of tolerance, a fuzzy extractor can be viewed as a one-way hash function. To achieve this with noisy input, public helper data is generated to aid reconstruction of the repeatable string R. This string can be used for symmetric encryption, generating a public-secret key pair or other applications that use uniformly random secrets. A second primitive derived from their work is the secure sketch. Given a biometric sample, a sketch is made. This sketch does not reveal any information about the original sample (entropy loss). Given the same input with noise and the sketch, the original can be reproduced exactly. This concept is based on previous “fuzzy” commitment (Hamming Distance) and vault (Set Difference) that are seen also as sketches. Those schemes fuse information about the biometric and key together. This scheme ‘extracts’ a key from a biometric at presentation given a sketch that contains only metric distances. The more entropy available the larger key extracted. However, given the same biometric, the same string will be extracted. Although this provides no information about the biometric sample, it defies the requirement of reusability. Boyen [34] address the case of fuzzy secret reuse and suggests attacks known as adaptive chosen secret. If this string is ever compromised it cannot be revoked.

Sources of biometric data can be either discrete or continuous. Fuzzy Vault, Commitment and Extractors are discrete. Linnartz and Tuyls’s *Shielding Function* [35], Tuyls et al’s *Reliable Component Scheme* [36] and Chang et al’s multi-bit scheme are examples of the latter. The previous framework of Fuzzy Extactors is based on discrete data input. Buhan et al. [37] extend this work to unify these two models called the *CS-Fuzzy Extractor*. However, many of the definitions of fuzzy extractors (statistical distance, min-entropy) are assumed for discrete data. These no longer apply in a continuous domain and are therefore modified. In order to prove these definitions, they demonstrate the CS-Fuzzy Extractor on the three schemes mentioned previously. These methods are based on methods of quantization.

Quantization is the ability to convert continuous data (real numbers such as π) and convert them into discrete data (integers or precision such as 3.14). Quantization is the method used to convert analog-to-digital signals as described by Bennet [38] as ‘quantizing of time’. Vector quantization for image compression by Gray [39] is constructed by moving points to the nearest Voronoi region based on a distance metric. This use is seen in many compression algorithms. Chen and Wornell [40] extended quantization for ‘Digital Watermarking’ to embed a signal within another to form a composite to hide data (steganography). They propose a method of *Quantization Index Modulation* embedding and used by Linnartz et al.’s [35] shielding functions in one-dimension.

Buhan et al. [41] propose to extend the CS-Fuzzy Extractor theory with a practical implementation using QIM later termed the *Fuzzy Embedder* [42]. A quantizer is a function used to map continuous points to a discrete reconstruction point in a set. Given a number of quantizers (called an ensemble), a reconstruction point is chosen and by an input value. Each reconstruction point has a voronoi or decision region that accounts for variations in noise. Each quantizer has a minimum distance between each other throughout metric space. Based on the Fuzzy Extractor, the input can be derived from the biometric itself or in combination with another independently random generator. The concept of an independent source is used to achieve the goal of revocability. An example of the embed procedure, the random input *r* selects one of the quantizers of an ensemble {*Q*o*, Q+, Q**} and finds the nearest reconstruction point for the feature coordinate *x*. The embedder returns the distance between the point *x* and reconstruction point as *p* (sketch). For instance, if uniformly random $r \epsilon \{1,2,3\}$, it will select the corresponding quantizer {*Q*o*, Q+, Q**} closest to *x*. The distance between quantizers is λ_{max} and represents the threshold for noise in the biometric input. This distance is represented as sphere assuming noise is not directional. Equiprobably distance between quantizers, to create uniformity, causes these spheres to have undefined spaces. The solution is to ‘dither’ the space using a hexagon polytope lattice.

4. Fingerprint Based Fuzzy Embedder

The Fuzzy Embedder implementation is constructed to accept standard ISO/IEC 19794-2 templates providing interoperability between varieties of popular compliant feature extractors. To avoid assumptions on vendor implementation, only the required data fields should be considered as there is no guarantee they are supported. This is the minutia triple defined as fingerprint ridge-endings and bifurcations (fork). A template consists of an unordered set *T* being the template.

$$T = \{ m_1, m_2, \dots m_n \} \quad \text{where,} \quad m_i = (x_i, y_i, \theta_i)$$

The unordered set creates a substantial limitation in the adaptation of the Fuzzy Embedder. It is not detailed how the helper data is matched to the corresponding minutiae in the reference sample fingerprint (section 4.4). Helper data and minutia triples require a 1:1 match and become difficult given the limitations on descriptor information and noise introduced in the system. While ordered fingerprint representations exist, such as directional fields, standard templates are preferred due to their high adaptation. To address the limitation of unordered data sets an indexing scheme is proposed in section 4.4.

Each triplet contains: x_m and y_m co-ordinates representing the minutia location based on image size and resolution. θ_m is the orientation of the minutia along the x-axis counter-clockwise beginning from the right. Triplet data is represented by two-bytes each. Data is scaled to fit the granularity. For instance, orientation is 1.40625 (360/256) degrees per least significant bit. Orientations are directed inwards for endings and outwards for bifurcations.

Standard templates contain optional fields: minutia type and quality. The extended data fields; core/delta position and ridge count. *Core* and *delta* are points of high curvature and ridge count the number of ridges between neighbouring minutiae. Use of additional descriptors should also be avoided. Despite this, the best participants of the Fingerprint Verification Competition (FVC) are based on both global minutia positions and alternative/hybrid techniques using additional descriptors of: singular points, ridges counts, orientation field, local ridge frequency, textures and pattern geometry [43]. Until these are standardised, the use of additional descriptors is used sparingly.

The Fingerprint based Fuzzy Embedder is a system consisting of a number of components shown in figure 4.1. Each module is independent from another and is interchangeable with different standards, representations, mechanisms. Some can be skipped entirely. For instance, the consolidation step is not required. The Fuzzy Embedder can work directly with only a single reference sample and query. However, the consolidation will significantly improve results. As a generalisation, not only can a cryptographic key be embedded, but messages, passwords or binary data. This applies not only to fingerprints, but other biometrics traits or non-uniform sources of entropy. Each module in the Fingerprint based system is considered individually in the following sections.

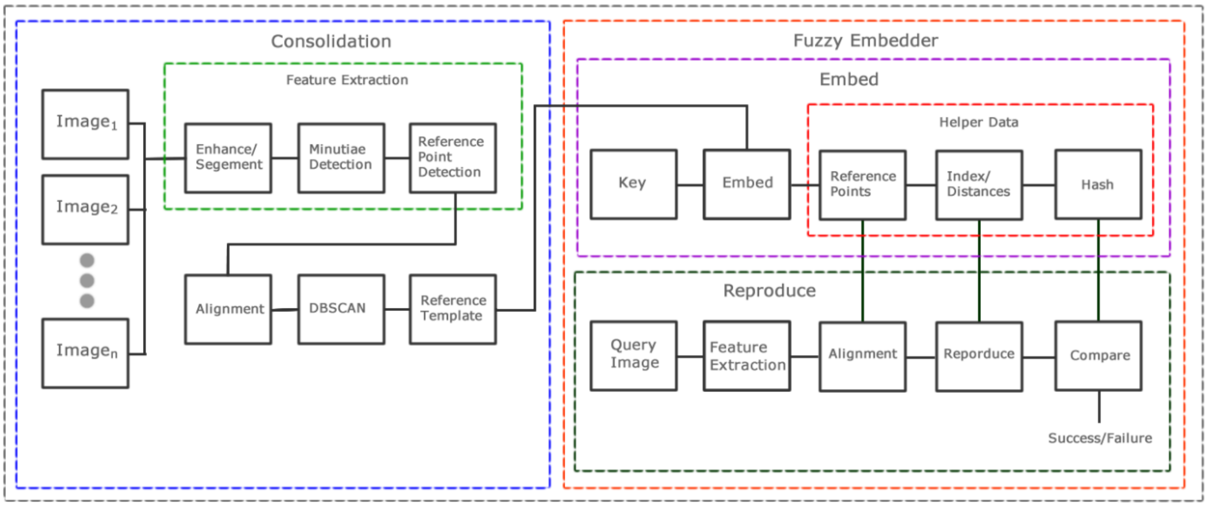

Figure 4.1, a high-level view of a fingerprint based biometric cryptosystem using the Fuzzy Embedder including consolidation step.

4.1 Alignment

The Fuzzy Embedder requires that sample template minutia match pair-wise on a two-dimensional plane. *Rotation* and *displacement* of two samples can differ greatly. Rotation being the direction the fingerprint is placed in relation to the sensor and displacement is the position it is placed in relation to the sensor. Alignment is the method of positioning two to achieve the most accurate overlap of reference and sample. Alignment must be the performed prior to; template consolidation, embedding and reproduce procedures. This is known as pre-alignment.

A number of methods have been suggested for this purpose and can be classified as; absolute or relative. Absolute is more appropriate for large identification systems, in which each sample is considered individually. The Fuzzy Embedder is a verification system and can make use of relative alignment where one sample is aligned with another for greater effectiveness.

A common method for fingerprint alignment is to determine two common reference points across all samples and globally align each by determining the angle difference, offset and transforming the reference points together. This requires an accurate method to determine two points common between fingerprint samples. In identification systems, fingerprints are classified by global features such as singularities. Common classes include; loops, whorls or delta and can be used as reference points.

The detection of singularity reference points is detected using the *Poincare Index* approach by Wang et al. [51]. The method uses a smoothed to determine regions of high curvature and shape. To remove spurious points a combination of; smoothing, segmentation and rules are used for a total performance rate of 91.54%. However, missing or inaccurate reference points due to poor quality images and distortion will affect the accuracy of the Fuzzy Embedder.

4.2 Feature Extraction

Feature extraction is the combination of image processing to determine minutia points of a fingerprint. It's extremely difficult to extract minutiae given noisy images. The security and reliability of the Fuzzy Embedder relies heavily on the ability to detect minutia points correctly as missing or spurious minutia are the main cause of error in the system. Prior to extraction, the fingerprint images are enhanced and segmented to increase accuracy.

The enhancement algorithm used is based on Sharat et al. *Short Time Fourier Transform* (STFT). The algorithm estimates probalistic ridge orientation and frequency to recover ridge discontinuities. The algorithm also smooths and binarises images to remove small areas of noise. The enhancement method reports a 17% improvement in recognition rate over the FVC2002 database [48].

Segmentation determines the foreground and background of the image known as, *region of interest* (ROI). The ROI is any part of the image that contains easily distinguished ridges up to the edges of a fingerprint. Backgrounds can contain impurities, left on a scanner from previous samples, which can disrupt enhancement and extraction. The algorithm used is block-based segmentation by Peter Kovesi [49].

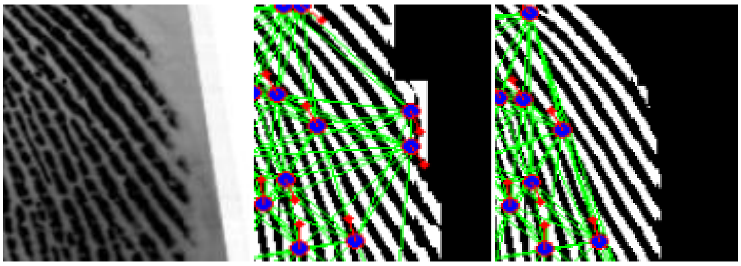

Figure 4.2.1, poor segmentation can results in incorrect feature extraction.

After image alignment, enhancement and segmentation, minute features can be extracted with higher accuracy. This has been viewed by visual comparison of extraction between grey-scale images and enhanced/segmented images. Impurity errors, clustering, ridge-errors are reduced and number of detected minutia increased. The ISO-compliant Fingerprint SDK by Griaule Biometrics [60] is used. Details of the feature extraction process are not provided, however results suggest good detection and removal of false minutia such those produced by false breaks or islands. These post-processing techniques on skeleton images are reviewed in high detail by Zhao & Tang [52]. Although the extractor falsely detects ridge-endings at the edge of a fingerprint, this can be avoided using correct segmentation (figure 4.2.1). The result of feature extraction is the compliant template.

4.3 Template Consolidation

A reference template combined with multiple registrations can significantly improve accuracy of the Fuzzy Embedder. This is known as *consolidation* and reduces the size of the reference template to only the most reliable points. Prior to consolidation, a number of sample images are aligned. The more samples selected guarantee a higher reliability. Features of each sample are used to distinguish overlapping data clusters using a *density-based scan with noise* (DBSCAN) [45]. The DBSCAN detects clustered points based on the number of points within its neighbourhood given a suitable distance metric. The scan also detects those points that do not belong to a cluster. This property is valuable as feature extraction can yield incorrectly detected/missed minutia. Figure 4.3.1 shows an incorrect minutia during feature extraction. This was caused by the placement of the enrolled finger (or noise) causing it to become segmented and removed.

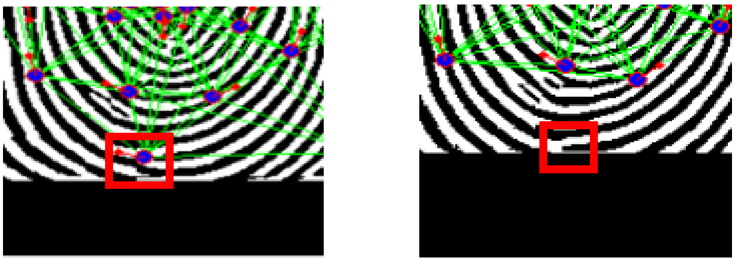

Figure 4.3.1, a missing minutia point due from two samples (red).

The consolidation process of six samples can be seen in figure 4.3.2. 1. All feature points from all sample images. 2. The result of the DBSCAN given a minimum number of points as 4 and distance of 10. The distance is directly relational to the step-size of the Fuzzy Embedder. *Crosses* (x) represent the core cluster points, *addition* (+) cluster border points, and *circle* (o) are noise. 3. Removal of noise reveals only the reliable points best used for embedding. 4. Subtractive clustering estimates the cluster centers [47]. The subtraction of noise, determination of clusters, centroids and their distances can be optimised however, is not considered in this paper.

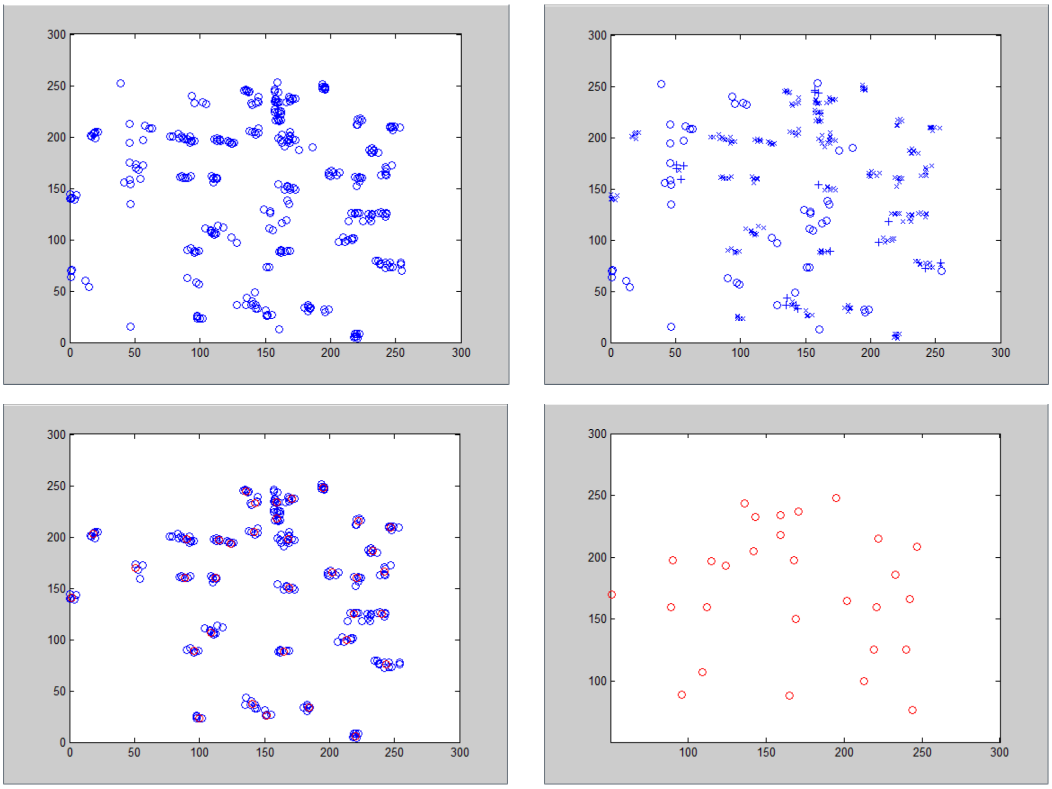

Figure 4.3.2, a) consolidation of 6 templates of the same finger. b) DBSCAN, Cross (x) core points, Addition (+) boundary points and circle (o) noise. c) removal of noise points. d) result points of sub-clustering.

4.4 QIM Construction

The practical construction of a dithered QIM defined as an ensemble of 7-quantizers suggested by Buhan et al. [42]. Ideally, each quantizer should be of equal distance from one another to provide *equiprobablility*. This is to provide maximum uniformity of output as required by cryptographic keys. The ideal shape of the regions is a circle. To fill space, the equal distance between them is known as the 'kissing number'. In 2D space, one circle can at most touch 6 others of equal shape and size without overlapping. However, this produces undefined regions. The dithered lattice removes this by using a tessellated hexagonal polytope lattice (or honeycomb).

The construction of the lattice created by the author is shown in figure 4.5.1 and is identical for both embed and reproduce functions. The lattice tiles space with quantizers from the starting point *S* given a scaling factor *q* (step-size) and is the basis of the new QIM indexing scheme.

Given starting point *S* in Euclidean space the lattice is created first by constructing the hexagonal lattice that defines its voronoi region. Shown in figure 4.5.1a, the scaling factor *q* is given as the apothem *a*, given a regular hexagon of 6 sides.

$r = a/(\cos(\pi/6))$ *– formula for a regular hexagon*

where, *r* is the circumradius

The quantizer ensemble, the group of seven quantizers, is defined by shifting the starting point (the ensemble centroid) by the dither vectors determined by the step-size (apothem) shown in figure 4.5.1b of the regular hexagon.

$\longrightarrow V_0 = (0, 0)$

$\longrightarrow V_1 = (2a, 0)$

$\longrightarrow V_2 = (a, (3/2)r)$

$\longrightarrow V_3 = (-a, (3/2)r)$

$\longrightarrow V_4 = (-2a, 0)$

$\longrightarrow V_5 = (-a, -(3/2)r)$

$\longrightarrow V_6 = (a, -(3/2)r)$

For a *quantizer ensemble*, a neighbour quantizer ensemble exists. The group of seven quantizer ensembles is named the *centroid* ensemble as their position is determined by its centroid. These shift vectors are similarly shown in figure 4.5.1c.

$\longrightarrow Vs_0 = (0, 0)$

$\longrightarrow Vs_1 = (5a, (3/2)r)$

$\longrightarrow Vs_2 = (-4a, 3r)$

$\longrightarrow Vs_3 = (-5a, -(3/2)r)$

$\longrightarrow Vs_4 = (-a, -(9/2)r)$

$\longrightarrow Vs_5 = (4a, -3r)$

$\longrightarrow Vs_6 = (5a, (3/2)r)$

4.4.1 Indexing

Each quantizer has an associated index value allocated incrementally from the starting point following the spiral mapping scheme in figure 4.5.1d. Altering the position of the starting point will shift all reconstruction points. This causes feature vectors to fall into regions of different indexes. The same occurs when altering the step-size. Therefore, the starting point and step-size must be known for both embed and reconstruction to produce the same indexing scheme.

The indexing scheme is required to find errors in matching feature-pairs between reference and query samples. Minutiae templates are sets of unordered tuples. The main limitation of the Fuzzy Embedder is it requires an ordered data set. The helper data, output by the embed procedure, has a one-to-one mapping with its input. The same mapping is required for reproduction.

Prior to embedding and reproduction, each feature vector in *X* is mapped to the nearest quantizer and is assigned its index value. Each vector *x* is then sorted descending from highest to lowest index. These indexes are stored as helper data to maintain order during reproduction as well as distinguishing set overlap of two samples *X* and *X'*.

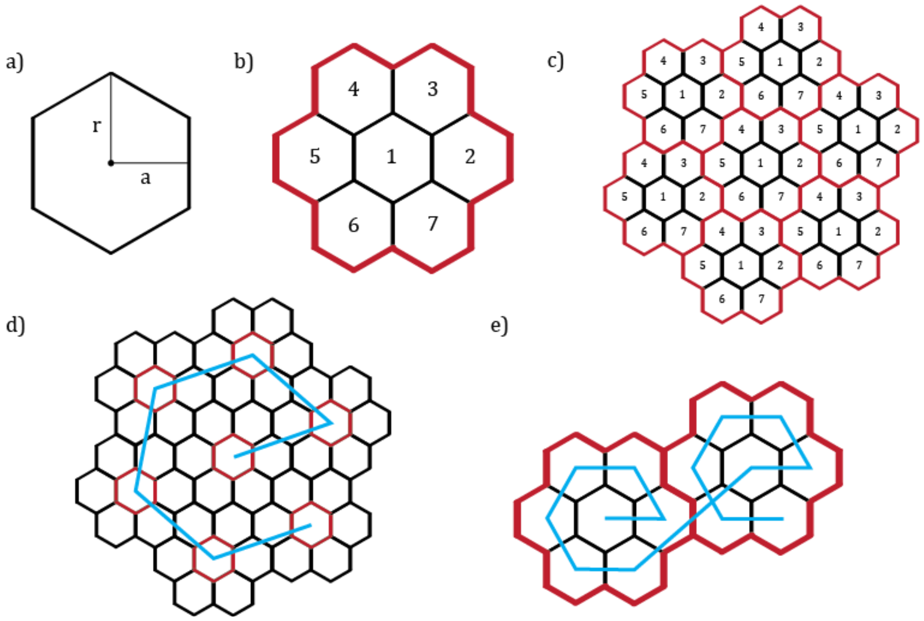

Figure 4.4.1, the construction of the hexgonal lattice. a) single tile: center, circumradius and apothem. b) single quantizer ensemble, center tile is the centroid. c) centroid ensemble (seven quantizer ensembles). d) mapping of quantizer centroid indexes from the starting point. e) mapping of individual quantizer indexes.

4.4.2 Embedding

The embed procedure takes input *X* and embeds it to independently random variable *K*. In this implementation using seven quantizers, K must be split into 3-bit vectors called *k*. This vector represents the index number (eg, 001 = index #1, 111 = index #7). Thus the value *k* selects the quantizer that *x* (minutia triple) maps to. Defined by Buhan et al as:

$$\text{Embed}(x, k) = \text{QIM}(x, k) - x$$

The embedder returns the difference between the two as *p* (helper data) where $p \leq \lambda_{\max}$. $\lambda_{\max}$ is the maximum distance between two corresponding quantizers. Where, *n* is the number of

dimensions, N the number of quantizers, σ_{min} the step-size as defined by Buhan et al.
$n = 2$, $N = 7$, $\sigma_{\text{min}} = 4$

$$\lambda_{\text{max}} = (\sqrt[n]{N})\sigma_{\text{min}}/2$$
$$\lambda_{\text{max}} = 10.6$$

This is explicit as the embed procedure always selects the nearest quantizer a centroid ensemble (figure 4.4.1a). This is achieved by first determining the nearest centroid to point *x*, then comparatively searching the minimum distance of all quantizers of *k* in the neighbourhood ensembles. The distance and index, prior to embedding, is stored as helper data.

4.4.3 Reproduce

The reproduce procedure takes the helper data *p* and input *X'* to reproduce the original random variable *K*, defined by Buhan et al as:

$$\text{Reproduce}(x', p) = Q(x' + p)$$

where, $Q(y)$ = minimum distance to a quantizer

If *x'* is sufficiently close to *x* it will return the original quantizer of the embed procedure and its associated value *k*. The reliability of this procedure is based on the step-size determining the bounding box that *x'* must fall within to correctly reproduce *k*, shown in figure 4.8.1b. The larger the bounding box, the larger tolerance of error. However, increasing step-size reduces the number of possible quantizers and increases the difficulty to order data reliably.

4.4.4 Pseudocode

Pseudocode for Embed and reproduce functions of QIM only:

```
1. Input Fuzzy Embedder Parameters (key, fingerprint)
2. Hash key
3. Create hexagonal lattice (starting point, step-size)
        Store index values of all quantizers given indexing scheme
4. For each triple
        Map triple x/y to nearest quantizer
        Store index value with triple
5. Order indexes by decending order
6. For each triple
        Map triple to nearest quantizer based on corresponding 3-bit key
            Store distance (helper data) with index
            Increment next 3-bit and triple
7. Store distances, angle, index, reference point, step-size (from feature extraction step)
   and hash
```

Pseudocode for Reproduce Procedure:

```
1. Input Fuzzy Embedder Parameters (helper data, query fingerprint)
2. Create hexagonal lattice (starting point, step-size)
    a. Store index values of all quantizers given indexing scheme
3. For each triple
    a. Map triple to nearest quantizer
    b. Store index value with triple
4. Order indexes by descending order
5. For each triple
    a. order by descending, if two or more matching indexes (points falling within the
       same voronoi region)
        i. for each triple (of matching index)
            1. Find minimum orientation distance of all triples given threshold
               (eg, 20 degrees)
            2. If conflict, remove conflicting triple
6. For each triple
    a. If has matching index
        i. add distance vector
        ii. Find nearest quantizer
        iii. Store quantizer value
    b. Else
        i. Skip, but store position of error
7. If number of errors is < 7 (or other value)
    a. For each error
        i. Increment value by one for all possible combinations
        ii. Compute and compare hash
        iii. Return result
8. Display pass or fail.
```

5. Test Results

The Fuzzy Embedder has been constructed using C# and incorporates the feature extraction method using Griaule Biometrics fingerprint SDK 2009. Alignment, consolidation and enhancement algorithms have been implemented in MATLAB. Using the FVC 2006 database 1, only 26 fingerprints had two singular points required for alignment. Of these, only one set contained enough images to be considered for template consolidation. This is due to missed delta singularities from cut-off fingerprint samples. The set that did contain two singular points, minutiae did not form dense enough clusters to output a consolidated temple. This violates the requirement of template protection schemes to provide reliable performance. This is tied both to the alignment method and the Fuzzy Embedder not tolerating rotation or displacement on its own. The solution can be found local minutiae matching techniques that are invariant to global translations such as those originally proposed by Ratha et al. [54], Jaing & Yau [55]. The most appropriate mechanism for representation of fingerprints using this scheme is Minutia Cylinder-Code by Capelli et al. [55] and will be considered in section 5.1.

The Fuzzy Embedder requires a tight level of precision. The FVC databases is designed for matching, not cryptosystems. There is little constraint on the capture of samples such as quality. The competition databases are purposely difficult to mimic practical environment or 'worst-case' scenarios. For instance, exaggerated; distortion, rotation and displacement. Practically, the Fuzzy Embedder, alignment algorithm chosen cannot cope with such constraints. Thus, a minimum level of quality is required in the test data. This includes the ability to detect two singular points and similar rotation and displacement. Fingerprints will be captured using an optical reader with a resolution of 500dpi.

Testing of the Fuzzy Embedder contains a set of 80 fingerprints captured using an optical sensor. There are 10-sets of fingerprints with 8-presentations of each with minimal rotation and distortion as possible. This was achieved using a reference point to visually locate fingerprint samples such as core/delta position. Average minutia extracted from enhanced images is 46 with minimum values of 30 and a maximum of 64. 5 samples were used for construction of the reference template. The remaining samples are used for queries. DBSCAN parameters varied depending on the amount of noise and the accuracy of the templates.

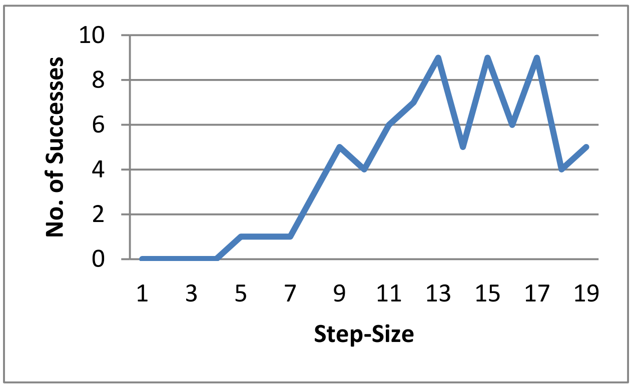

Figure 5.1, the number of successful key reproductions at different key-sizes

The DBSCAN results vary between accuracy of samples and their variance. Average reference points for a scan of neighbourhood clusters of 4 (out of 5) and a distance of 10 result in 27 points. This number has not been optimised and changes to both the DBSCAN and sub-clustering variables alter the dataset significantly. Given a range of 10 the results of the success and failure of the Fuzzy Embedder is shown in figure 5.1 to center around 13 to 17. This correlates with the average error between points of samples and is maintained through DBSCAN. All queries had a success of 100% over varying step-sizes and small bounds DBSCAN. However, this could not be obtained if it were constant. For instance, a step-size of 13 resulted in 9 successes over 30 query samples - 70% False-Rejection. If errors can be corrected (or not) by varying the step size, this suggests an error in the construction of the Fuzzy Embedder as increasing the step-size should always result in higher reliability.

5.1 Sources of Error

Two presentations of the same fingerprint are rarely the same. A combination of; alignment, pressure, impurities, scaring or moisture causes variation in feature extraction. The alignment, extraction and pre-processing (enhancement and segmentation) algorithms themselves can also exacerbate the problem.

Quantization tolerates linear distortion however, cannot correct missing or spurious minutia. The Fuzzy Embedder relies on the intersection of the two sets *X* ∩ *X'*. The indexing scheme distinguishes the intersection between the two sets. All points in set *X'* must have a corresponding matching triple in *X.* Given that errors can be corrected by the step-size, only points of *x'* within a bounding box of *x* can be mapped to the correct quantizer.

This bounding box represents the tolerance of error between two samples, shown in figure 5.1.1b. It creates an issue of defining the position of *x* without revealing too much information about it. The indexing scheme is a naïve approach as is assumed the reference point *x* falls at the center of a voronoi region. Practically, this is not the case as the bounding box can overlap at most 4-tiles. When the value of *x* falls on the boundary of a voronoi region, it's highly likely the corresponding point *x'* will fall into the neighbouring region and thus, be allocated a different index and excluded from the set. An alternative approach is to quantize the original *x* to be at the center of a region. However, this would cause public distances *P* to be of equal lengths. Simple patterns will allow an adversary to find the value of *K.*

The issue of boundary points can be corrected (crudely) by changing the step-size. For small step-sizes, each feature vector is generally allocated a different index. However, for higher step-sizes, two or more points can fall within the same voronoi region. In this case, the smallest $\Delta\theta$ of points is taken (followed by x then y) given a tolerance value. Ordering a large number of points of the same index can result in the incorrect matching of helper data *p*.

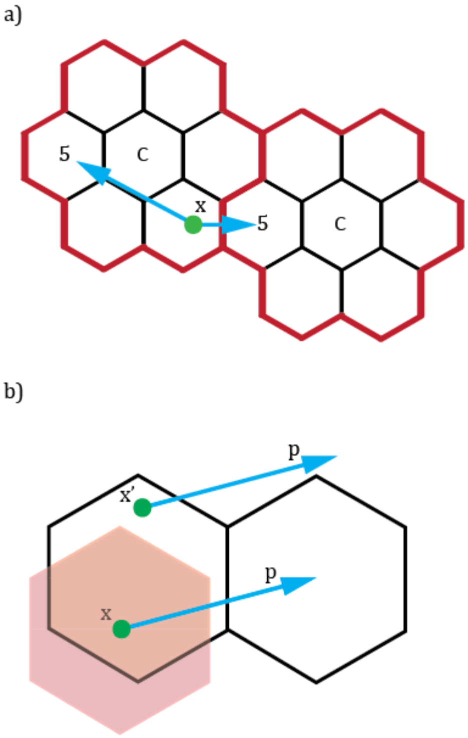

Figure 5.1.1, a) selection of the nearest quantizer based on k. b) The bounding box of error (shaded)

The incorrect overlap of reference and query sets is caused by missing pair-wise matches where there is no value in set *X'* that corresponds to a point in set *X*. For every miss-matched or missing point, the key will differ in the position that it occurred. The advantage of using the indexing scheme allows this position of error to be known by comparison of two resulting hashes.

The hash of the output R is used to determine success or failure of match attempt of the reference and sample. It is possible to correct a number of points by exhaustively testing every combination of missing key-bits (1-7) before finding a matching hash becomes computationally expensive (tested on a 2.2Ghz Intel core-duo) of around 10 seconds. The missing key-bits are identified by unmatched indexes. Each value is incremented by 1, hash computed and compared and loops until either a match occurs or all combinations are exhausted.

5.2 Key-Length Security

Security of the system is primarily based on its ability to produce key-bits. The key length required depends on the desired level of protection for an application. The ECRYPT II annual report [44] suggests recommended key-sizes based on current attack models and resources on cryptographic algorithms. For instance, in 2008, a 58-bit DES key would provide around 200-days of protections against a $400 FPGA system or 3-days with a $7,500 system. The report suggests minimum key lengths for symmetric ciphers to be 80-bits. This protects against the most reasonable and threatening attacks. This length provides short term protection against large-agencies (high resources) and long term protection from small of less than 4-years. In comparison, 128-bit keys provides general long-term protection for up to 30-years – 256-bit for the 'foreseeable future' (quantum computing).

The key-length is given by the number of feature vectors and number of quantizers. Given seven quantizers contains 3-bits of entropy an 80-bit key requires 27 features and equivalent 43 for 128-bits. This provides minimum benchmark for the number of vectors for embedding. Given the reproduce function can correct up to seven missing vectors, the query sample must overlap over 20 vectors of the reference template. The average number of features extracted from the DBSCAN was 27 with a minimum of 17 and maximum of 41. On average this is sufficient for the minimum key-size requirement, however varies highly on the fingerprint sample, DBSCAN result and parameters.

5.3 Information Leakage

The preceding table is sample data of the Fingerprint based Fuzzy Embedder. The two tables represent the data required for both embed and reproduce from a reference template *X* and query sample *X'*. The data has been reduced to show only the first seven variables. The embedder was set with a step-size of 5. The reference template contained 21 feature vectors (63-bit key) and was correctly reproduced with 7 missing points.

Input message:	123456712345671234567
Result:	133456616327674234567
Post-Correction:	123456712345671234567

Public data includes: the angle (θ), index, distance, reference points and hash result (red shade).

Data Set X			Key	Index	Quantizer	Distance	Reference
X:	**Y:**	**Θ:**	**Bits:**				Reference 1:
191	238	176	000 (1)	3378	(186, 234.83)	(5, 3.17)	(116, 153)
162	213	76	001 (2)	2878	(161, 226.17)	(1, -13.17)	Reference 2:
236	153	200	010 (3)	2821	(241, 156.88)	(-5, -3.88)	(201, 162)
138	225	216	100 (4)	2412	(136, 217.51)	(2, 7.49)	
162	188	72	101 (5)	2402	(156, 200.19)	(6, -12.19)	
215	144	76	110 (6)	2354	(216, 148.22)	(-1, -4.22)	
139	194	212	111 (7)	1977	(131, 191.53)	(-6, 1.38)	
Hash:		8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f01665370476383ea776df414					

Data Set X’			**Key**	**Index**	**Quantizer**	**Distance**	**Reference**
X:	**Y:**	**Θ:**	**Bits:**				Reference 1:
191	238	172	000 (1)	3378	(186, 234.83)	(5, 3.17)	(116, 153)
0	0	0	001 (2)	0	(-4, 9.66)	(1, -13.17)	Reference 2:
0	0	0	010 (3)	0	(1, 1)	(-5, -3.88)	(201, 162)
140	226	212	100 (4)	2412	(136, 217.51)	(2, 7.49)	
160	191	68	101 (5)	2402	(156, 200.19)	(6, -12.19)	
215	144	76	110 (6)	2354	(216, 148.22)	(-1, -4.22)	
0	0	0	111 (7)	0	(-9, 0)	(-6, 1.38)	
Hash:		8bb0cf6eb9b17d0f7d22b456f121257dc1254e1f01665370476383ea776df414					

The main security vulnerability exists in an adversary gaining knowledge of either *X* or *K* using the knowledge of helper data *P*. Although both values are important, *K* can be revoked if compromised. However, because *X* is a biometric template, should never be revealed. This is the main requirement and motivation of a template protection scheme.

Given the worst case, an adversary has knowledge of; *P*, *K, S* and step-size. This insinuates the adversary can correctly reproduce the Fuzzy Embedder and indexing scheme. Using the public indexes, it is possible to find the region which potential points *x* can fall within. Given the true value of *x* can take any value within the voronoi region it is not feasible to find the exact value of *x*. For a step-size of 5 the number of integer values that *x* can take is the area of a regular hexagon.

Area = ½(apothem)(perimeter)

Perimeter = 6(side)

Side = (2a)tan(180/n) where n = 6

Area = 87

Any feature vector of *X* can potentially take any of the 87 values given a step-size 5. A step-size of 1, provides 3.5 possible values. This is equivalent to a one-to-one mapping of index to feature and should be avoided.

There are two solutions to prevent this. Fuzzy Embedder variables *S* and step-size should be stored securely on a token, smart-card or remote database. The public indexes can be combined with chaff points, such as those used in a Fuzzy Vault implementation. Chaff points would be significantly different from genuine points. Given an adversary has *K* the task is simplified by exhaustively checking which points map to the key by reversing the process. Split the key into 3-bit blocks and find all the reconstruction points associated with the value. Given a step-size 5 and templates range of 256 x 256.

Diameter = 2a = 10

No. of quantizers x = 260/10 = 26

No. of quantizers y = 260/12.5 = 20

No. of quantizers (xy) = 20 * 26 = 520

520 / no. of quantizers = 74

Therefore, there are 74 quantizers of the same type. Incrementally trying each, with the addition of public distance will reveal the index and thus *x* – ignoring chaff points. Incrementally working a 27-feature key, would require under 2000 cycles, easily calculated on modern processors.

Given the key is never stored, finding the value of *K* is difficult. However, based on recommended key lengths in section 4.9, with enough time, the key can be revealed. Using an old instance of a Fuzzy Embedder (even if parameters are changed regularly) it is possible to compute *X* offline with enough time and resources.

Without the parameters *S* and step-size this task is made more difficult. The step-size is proportional to the indexing scheme, given the same starting point. Given a step-size of 1, key value of 1 and starting point (1, 1), the index value of a point (0, 256) is 7453. An increased step-size of 10, results in the index value of 502. Higher step-sizes result in lowered indexes. However, chaff points can be used to obscure this.

Altering the starting point *S* also alters the indexing scheme. Using the previous data, where *S* = (1.1, 1.1) the index changes to 7467. Small changes have a dramatic effect on the indexes as well as the step-size. This ensures that no two Fuzzy Embedders are the same, somewhat like a password salt.

6. Conclusion

Biometric cryptosystems aim at either binding or generating keys long enough to be applied to standard encryption mechanisms. To reduce the probability of being guessed in brute-force attacks, they must exhibit sufficient entropy and randomness (eg, 128-bit key). The main challenge when combining biometric cryptosystems and fingerprints is the issue of reliability. The Fuzzy Embedder is a generalised and simplistic scheme that has the potential to satisfy the goals of; diversity, revocability, security and performance with the added benefit of applying extractor theory to continuous data.

The Fingerprint based Fuzzy Embedder currently does not show sufficient evidence to satisfy these requirements. Further analysis and testing is required to solve the main issue of the bounding box problem. Improvements to a number of components can also further improve its reliability. This includes the DBSCAN, alignment, enhancement, segmentation and feature extraction. The restriction of good quality fingerprint images and failure to produce results on FVC data sets is an example of this. Further, the security benefits need to be analysed more carefully to determine the exact repercussions of information leakage, starting point and step-size. The Fingerprint based Fuzzy Embedder is likely to achieve better results using additional descriptor information, 'state-of-the-art' mechanisms such as *Minutia Cylinder-Code* [55] or other 3D data structures.

Minutia Cylinder-Code uses local minutia structures of fixed-radius and nearest-neighbours originally proposed by Jiang & Yau [56] and Ratha et al. [57]. The two local structures provide a higher tolerance to rotation and displacement transformations using a concept of cylinders. Smoothing and saturation effects are implemented to limit spurious and missing minutia caused by feature extraction. In comparison to previous implementations and Feng [58], MCC shows significantly higher accuracy over benchmark fingerprint database FVC2006 [59]. The issues addressed by MCC are equivalent those that affect the Fuzzy Embedder. It removes the need to pre-align images and use orientation information in helper data. Such implementation will require a complete reconstruction of the Fuzzy Embedder to accept MCC input.

An alternative construction of the Fuzzy Embedder could simply incorporate a 3rd dimension. Like MCC, this extra dimension would represent the minutia orientation. The same requirements of QIM size and shape remain, however the hexagonal lattice is no longer appropriate as it does not tessellate equally in 3-dimenstions. Alternative shapes would include platonic solids such as a cube (used in MCC) or dodecahedron. However, distances between adjacent cubes in an ensemble will not be of equal distance and requires careful planning to achieve the best equiprobability in a 3D lattice.

Bibliography

1. Maio, D., & Jain, A. K. (2009). *Handbook of fingerprint recognition*. Springer.
2. Market, B. (2008). Industry Report 2009-2014. *International Biometric Group*.
3. ISO/IEC 19794-2:2005, Information Technology—Biometric Data Interchange Formats—Part 2: Finger Minutiae Data, 2005.
4. Cappelli, R., Lumini, A., Maio, D., & Maltoni, D. (2007). Fingerprint image reconstruction from standard templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *29*(9), 1489-1503.
5. Matsumoto, T., Matsumoto, H., Yamada, K., & Hoshino, S. (2002, April). Impact of artificial gummy fingers on fingerprint systems. In *Electronic Imaging 2002* (pp. 275-289). International Society for Optics and Photonics.
6. Anil K, J., Karthik, N., & Abhishek, N. (2008). Biometric template security. *EURASIP Journal on Advances in Signal Processing*, *2008*.
7. Riley, S. (2006). Password security: What users know and what they actually do. *Usability News*, *8*(1).
8. Uludag, U., Pankanti, S., Prabhakar, S., & Jain, A. K. (2004). Biometric cryptosystems: issues and challenges. *Proceedings of the IEEE*, *92*(6), 948-960.
9. Katz, J., & Lindell, Y. (2008). *Introduction to modern cryptography*. Chapman & Hall.
10. Rathgeb, C., & Uhl, A. (2011). A survey on biometric cryptosystems and cancelable biometrics. *EURASIP Journal on Information Security*, *2011*(1), 1-25.
11. Ratha, N. K., Connell, J. H., & Bolle, R. M. (2001). Enhancing security and privacy in biometrics-based authentication systems. *IBM systems journal*, *40*(3), 614-634.
12. Ratha, N. K., Chikkerur, S., Connell, J. H., & Bolle, R. M. (2007). Generating cancelable fingerprint templates. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *29*(4), 561-572.
13. Zuo, J., Ratha, N. K., & Connell, J. H. (2008, December). Cancelable iris biometric. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (pp. 1-4). IEEE.
14. Hämmerle-Uhl, J., Pschernig, E., & Uhl, A. (2009). Cancelable iris biometrics using block re-mapping and image warping. In *Information Security* (pp. 135-142). Springer Berlin Heidelberg.
15. Dodis, Y., Reyzin, L., & Smith, A. (2004, January). Fuzzy extractors: How to generate strong keys from biometrics and other noisy data. In *Advances in cryptology-Eurocrypt 2004* (pp. 523-540). Springer Berlin Heidelberg.
16. Juels, A., & Wattenberg, M. (1999, November). A fuzzy commitment scheme. In *Proceedings of the 6th ACM conference on Computer and communications security* (pp. 28-36). ACM.
17. Juels, A., & Sudan, M. (2006). A fuzzy vault scheme. *Designs, Codes and Cryptography*, *38*(2), 237-257.
18. Connie, T., Teoh, A., Goh, M., & Ngo, D. (2005). Palmhashing: a novel approach for cancelable biometrics. *Information Processing Letters*, *93*(1), 1-5.
19. Nandakumar, K., Nagar, A., & Jain, A. K. (2007). Hardening fingerprint fuzzy vault using password. In *Advances in biometrics* (pp. 927-937). Springer Berlin Heidelberg.
20. Davida, G. I., Frankel, Y., & Matt, B. J. (1998, May). On enabling secure applications through off-line biometric identification. In *Security and Privacy, 1998. Proceedings. 1998 IEEE Symposium on* (pp. 148-157). IEEE.
21. Hao, F., Anderson, R., & Daugman, J. (2006). Combining crypto with biometrics effectively. *Computers, IEEE Transactions on*, *55*(9), 1081-1088.
22. Campisi, P., Maiorana, E., Gonzalez, M., & Neri, A. (2007). Adaptive and distributed cryptography for signature biometrics protection. *SPIE Proc. Security, Steganography, and Watermarking of Multimedia Contents IX*, *6505*.

23. Kevenaar, T. A., Schrijen, G. J., van der Veen, M., Akkermans, A. H., & Zuo, F. (2005, October). Face recognition with renewable and privacy preserving binary templates. In *Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on* (pp. 21-26). IEEE.
24. Shamir, A. (1979). How to share a secret. *Communications of the ACM*, *22*(11), 612-613.
25. Monrose, F., Reiter, M. K., & Wetzel, S. (2002). Password hardening based on keystroke dynamics. *International Journal of Information Security*, *1*(2), 69-83.
26. Nandakumar, K., Jain, A. K., & Pankanti, S. (2007). Fingerprint-based fuzzy vault: Implementation and performance. *Information Forensics and Security, IEEE Transactions on*, *2*(4), 744-757.
27. Jain, A., Hong, L., & Bolle, R. (1997). On-line fingerprint verification. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *19*(4), 302-314.
28. Nagar, A., Nandakumar, K., & Jain, A. K. (2008, December). Securing fingerprint template: Fuzzy vault with minutiae descriptors. In *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on* (pp. 1-4). IEEE.
29. Feng, J. (2008). Combining minutiae descriptors for fingerprint matching. *Pattern Recognition*, *41*(1), 342-352.
30. Bodo, A. (1994). Method for producing a digital signature with aid of a biometric feature. *German patent DE*, *42*(43), 908.
31. Janbandhu, P. K., & Siyal, M. Y. (2001). Novel biometric digital signatures for Internet-based applications. *Information Management & Computer Security*, *9*(5), 205-212.
32. Hernández Álvarez, F., Hernández Encinas, L., & Sánchez Ávila, C. (2009). Biometric Fuzzy Extractor Scheme for Iris Templates.
33. Chang, E. C., & Li, Q. (2006). Hiding secret points amidst chaff. In *Advances in Cryptology-EUROCRYPT 2006* (pp. 59-72). Springer Berlin Heidelberg.
34. Boyen, X. (2004, October). Reusable cryptographic fuzzy extractors. In *Proceedings of the 11th ACM conference on Computer and communications security* (pp. 82-91). ACM.
35. Linnartz, J. P., & Tuyls, P. (2003, January). New shielding functions to enhance privacy and prevent misuse of biometric templates. In *Audio-and Video-Based Biometric Person Authentication* (pp. 393-402). Springer Berlin Heidelberg.
36. Tuyls, P., Akkermans, A. H., Kevenaar, T. A., Schrijen, G. J., Bazen, A. M., & Veldhuis, R. N. (2005, January). Practical biometric authentication with template protection. In *Audio-and Video-Based Biometric Person Authentication* (pp. 436-446). Springer Berlin Heidelberg.
37. Buhan, I., Doumen, J., Hartel, P., & Veldhuis, R. (2007, March). Fuzzy extractors for continuous distributions. In *Proceedings of the 2nd ACM symposium on Information, computer and communications security* (pp. 353-355). ACM.
38. Bennett, W. R. (1948). Spectra of quantized signals. *Bell syst. tech. J*, *27*(3), 446-472.
39. Gray, R. (1984). Vector quantization. *ASSP Magazine, IEEE*, *1*(2), 4-29.
40. Chen, B., & Wornell, G. W. (2001). Quantization index modulation methods for digital watermarking and information embedding of multimedia. *Journal of VLSI signal processing systems for signal, image and video technology*, *27*(1-2), 7-33.
41. Buhan, I. R., Doumen, J. M., Hartel, P. H., & Veldhuis, R. N. J. (2007). Constructing practical fuzzy extractors using qim.
42. Buhan, I., Doumen, J., Hartel, P., Tang, Q., & Veldhuis, R. (2008). Embedding renewable cryptographic keys into continuous noisy data. In *Information and Communications Security* (pp. 294-310). Springer Berlin Heidelberg.
43. Cappelli, R., Maio, D., Maltoni, D., Wayman, J. L., & Jain, A. K. (2006). Performance evaluation of fingerprint verification systems. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *28*(1), 3-18.
44. Yearly Report on Algorithms and Keysizes (2012), D.SPA.20 Rev. 1.0, ICT-2007-216676 ECRYPT II, 09/2012
45. Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996, August). A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD* (Vol. 96, pp. 226-231).

46. Ohtsuka, T., Watanabe, D., Tomizawa, D., Hasegawa, Y., & Aoki, H. (2008, June). Reliable detection of core and delta in fingerprints by using singular candidate method. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW'08. IEEE Computer Society Conference on* (pp. 1-6). IEEE.
47. Chiu, S. L. (1994). Fuzzy model identification based on cluster estimation. *Journal of intelligent and Fuzzy systems*, *2*(3), 267-278.
48. Chikkerur, S., Govindaraju, V., & Cartwright, A. N. (2005). Fingerprint image enhancement using STFT analysis. In *Pattern Recognition and Image Analysis*(pp. 20-29). Springer Berlin Heidelberg.
49. Peter Kovesi (2005) Ridge Segment [MATLAB]. The University of Western Australia: School of Computer Science & Software Engineering.
50. Wang, L., Bhattacharjee, N., & Srinivasan, B. (2011, December). A novel technique for singular point detection based on Poincaré index. In *Proceedings of the 9th International Conference on Advances in Mobile Computing and Multimedia* (pp. 12-18). ACM.
51. Yager, N., & Amin, A. (2006). Fingerprint alignment using a two stage optimization. *Pattern Recognition Letters*, *27*(5), 317-324.
52. Zhao, F., & Tang, X. (2007). Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction. *Pattern Recognition*, *40*(4), 1270-1281.
53. Ratha, N. K., Bolle, R. M., Pandit, V. D., & Vaish, V. (2000). Robust fingerprint authentication using local structural similarity. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.* (pp. 29-34). IEEE.
54. Jiang, X., & Yau, W. Y. (2000). Fingerprint minutiae matching based on the local and global structures. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on* (Vol. 2, pp. 1038-1041). IEEE.
55. Cappelli, R., Ferrara, M., & Maltoni, D. (2010). Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *32*(12), 2128-2141.
56. Jiang, X., & Yau, W. Y. (2000). Fingerprint minutiae matching based on the local and global structures. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on* (Vol. 2, pp. 1038-1041). IEEE.
57. Ratha, N. K., Bolle, R. M., Pandit, V. D., & Vaish, V. (2000). Robust fingerprint authentication using local structural similarity. In *Applications of Computer Vision, 2000, Fifth IEEE Workshop on.* (pp. 29-34). IEEE.
58. Feng, J. (2008). Combining minutiae descriptors for fingerprint matching.*Pattern Recognition*, *41*(1), 342-352.
59. FVC2006. (n.d.). *Fingerprint Verification Competition*. Retrieved September 30, 2013, from http://bias.csr.unibo.it/fvc2006/
60. Homepage. (n.d.). *Griaule Biometrics*. Retrieved September 30, 2013, from http://www.griaulebiometrics.com/