RC 21977 (98775) 2 March 2001 Computer Science

IBM Research Report

Secure User Authentication Using Automated Biometrics

N. K. Ratha, J. Connell, R. M. Bolle
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598

Research Division
Almaden - Austin - Beijing - Haifa - India - T. J. Watson - Tokyo - Zurich

Secure User Authentication using Automated Biometrics

Nalini K. Ratha, Jonathan H. Connell, and Ruud M. Bolle
IBM Thomas J. Watson Research Center
30 Saw Mill River Road
Hawthorne, NY 10532
{ratha, jconnell, bolle}@us.ibm.com

Abstract

In recent years, there has been a significant surge in the use of biometrics for user authentication applications because biometrics-based authentication offers several advantages over knowledge and *possession-based methods such as password/PIN-based systems. However, it is important that such biometrics-based authentication systems are designed to withstand different sources of attacks on the system when employed in security-critical applications, and more so in unattended remote applications such as e-commerce applications. In this paper we outline the inherent strengths of* a *biometrics-bared authentication scheme and then discuss the security holes in these systems. Finally, we present new solutions for overcoming some of the remaining weak links in such systems.*

1 Introduction

Reliable user authentication is becoming an increasingly important task in the web-enabled world. The consequences of an insecure authentication method in a corporate or enterprise environment can be catastrophic, often leading to loss of confidential information, service denials, and issues with integrity of data and information contents. The value of a reliable user authentication is not limited to just computer access. Many other applications in everyday life also require user authentication, e.g., banking, immigration, and physical access control, and could benefit from enhanced security.

The prevailing techniques of user authentication that involve passwords and user ids, or identification cards with PINs, suffer from several limitations. One of the main problems with such approaches is that the authentication subsystem can be fooled relatively easily. Passwords and PINs can be illicitly acquired relatively easily by direct covert observation. Once an intruder has the password, he has total access to the associated resources. The other major problem is that there is no way to positively link the usage of the system or service to the actual user, i.e., the issue of “repudiation”. For example, a user id and password can easily be shared

Method	Examples	Properties
What you know	Userid Password PIN	Shared Many passwords are easy to guess Forgotten
What you have	Cards Badges Keys	Shared Can be duplicated Lost or stolen
What you know and what you have	ATM card + PIN	Shared PIN is a weak link (Writing the PIN on the card)
Something unique about the user	Fingerprint Face Iris Voice print	Not possible to share Repudiation unlikely Forging is difficult Can't be lost or stolen

Table 1: Existing user authentication techniques.

with a colleague or a secretary. When this happens there is no way for the system to know who is actually logged in. Similarly, the critical information about the transaction such as the credit card number and the amount are sent over the web using secure encryption methods. However, the present practice is not capable of assuring that the transaction was initiated by the rightful owner of the credit card. To summarize, in the modern networked system environment, an authentication policy based on a simple combination of user id and password has become inadequate.

Fortunately automated biometrics technology in general, and fingerprints in particular, can provide a much more accurate and reliable user authentication method. Biometrics is a rapidly advancing field that is concerned with identifying a person based on their physiological or behavioral characteristics. Examples of automated biometrics include fingerprints, faces, iris and speech. User identification and authentication methods can be broadly classified into three categories [12] as shown in Table 1. Because a biometric is an intrinsic property of some individual, they are difficult to surreptitiously duplicate and nearly impossible to share.

A big advantage of biometrics signals are that they are much longer in size than a password or pass phrase. They range from several hundred bytes to over a megabyte. Typically, the information content of such signals is correspondingly higher as well. Simply extending the length of passwords to get equivalent bit strength presents significant usability problems. It is nearly impossible to remember a 2K phrase and it would take an annoyingly long time to type such a phrase in (especially without errors). Fortunately, automated biometrics can provide the security advantages of long passwords while retaining the speed and simplicity of short passwords.

While automated biometrics can help to alleviate the problems associated with the existing methods of user authentication, hackers will still find the weak points in the system and attack it at those points. Password systems are prone to brute-force dictionary attacks. Biometrics systems, on the other hand, require substantially more effort to attack in a brute-force manner. Although standard encryption techniques are useful in many ways to prevent a breach of security, there are several new types of attacks possible in the biometrics domain. If biometrics is used as a supervised authentication tool, this may not be a concern. But in remote unattended application, such as web-based e-commerce applications, hackers may have the opportunity and enough time to make several attempts before being noticed or even physically violate the remote client.

Another new problem with biometric authentication systems concerns the re-issuance of identity tokens. Once the user authentication systems start using private details of users, there is always a privacy concern about how that information can be misused. For authentication systems based on physical possessions, like keys and badges, a previous token can be easily canceled and the user can be reassigned a new identification object. Similarly, logical entities, such as user id and passwords, can be changed as often as required. Yet, the user only has limited number of biometrics such as one face, ten fingers, and two eyes. If these are compromised, the user may quickly run out of biometrics for authentication.

In this paper, we will discuss in more detail the problems unique to biometric authentication systems and propose solutions to several of the problems. Though our analysis is very general and can be extended to other biometrics, we will focus on fingerprint recognition as an example throughout this paper. In Section 2, we detail the stages of fingerprint authentication and machine representations of fingerprint. This forms the basis for the following discussions. In Section 3, we use a pattern recognition model of a generic biometrics system to help identify the possible attack points. Section 4 analyzes the power of a minutia-based fingerprint system in terms of probability of a brute force attack being successful. Section 5 proposes several techniques to alleviate some of the other threats described in Section 3. Section 6 introduces the concept of "cancelable biometrics" and discusses their application. Finally, Section 7 recapitulates the issues discussed and summarizes the new approaches suggested

2 Fingerprint recognition

A brief introduction to fingerprint authentication is provided as background to the material presented in subsequent sections. Readers familiar with fingerprint recognition systems can skip to the next section.

Fingerprints are unique to a person and remain invariant over the lifetime of a subject. As the first step in the process, a fingerprint impression is acquired, typically using an inkless scanner. Several such scanning technologies are available [15]. Figure 1 (a) shows a scanned fingerprint obtained using an optical sensor. A typical scanner digitizes the fingerprint impression at 500 dpi with 256 gray levels per pixel. The digital image of the fingerprint consists of several unique features in terms of ridge bifurcations and ridge endings collectively referred to as *minutiae.*

The next step is to locate these minutiae features in the fingerprint image, as shown in Figure 1(b), using an automatic feature extraction algorithm. Minutiae features are commonly represented by their location (X, Y) and the ridge direction at the location of the minutiae (θ).

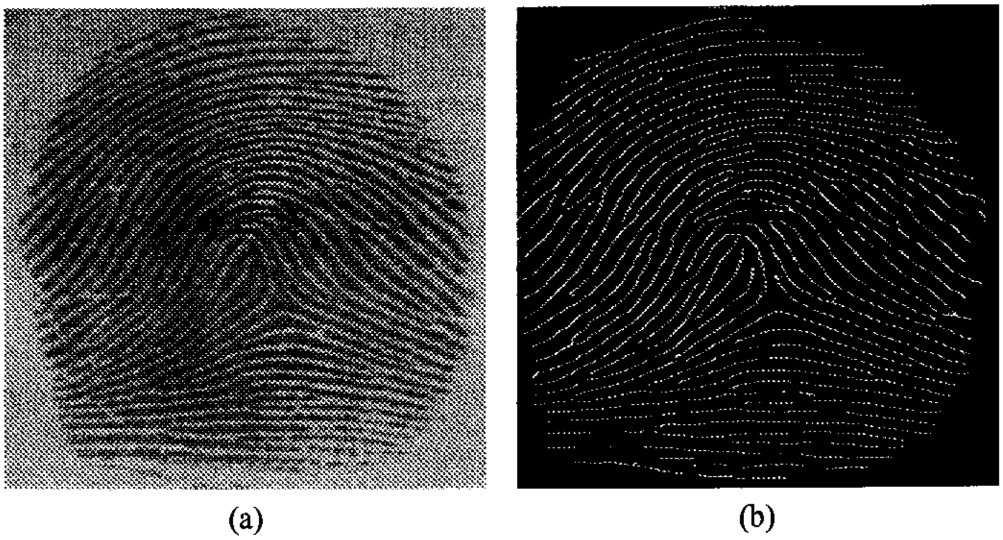


Figure 1: Fingerprint recognition. (a) input image; (b) features.

However, due to sensor noise and other variability in the imaging process, the feature extraction stage may miss some minutiae and may generate spurious minutiae. Due to the elasticity of the human skin, the minutiae may be randomly distorted of the finger impression.

In the final stage, the matcher subsystem attempts to arrive at a degree of similarity between two sets of features after compensating for the rotation, translation and scale. This similarity is ofter expressed as a score. Based on this score, a final decision of match or no-match is made. Often the score is simply a count of the number of the minutiae that are in correspondence. In a number of countries, 15 to 17 correspondences (performed by a human expert) are considered legally binding evidence of identity.

The operational issues in an automated fingerprint identification system (AFIS) are somewhat different from those in a more traditional password based system. First, there is a system performance issue known as the "fail to enroll" rate to be considered. For instance, some people have very faint fingerprints (or no fingers at all) which makes the system unusable for them. This has no analog in a password system. Then there is the fact that in a biometrics-based system the matching decision is not clear-cut. A password system always provides a correct response – if the passwords match, it grants access but otherwise refuses access. However, in a biometrics system, the overall accuracy depends on the quality of input and enroll data along with the basic characteristics of the underlying feature extraction and matching algorithm.

For fingerprints and biometrics in general, there are two basic types of errors, namely false accept (FAR), and false reject (FRR). If a non-matching pair of fingerprints is accepted as a match, it is called a false accept. On the other hand, if a mated pair of fingerprints is rejected by the system, it is called a false reject. The error rates are a function of the threshold as shown in the Figure 2. Often the interplay of the two errors is presented by plotting FAR against FRR with the decision threshold as the free variable. This plot is called as the ROC (Receiver Operator Curve). The two errors are complimentary in the sense that if one makes an effort to lower one of the errors by varying the threshold, the other error rate automatically increases.

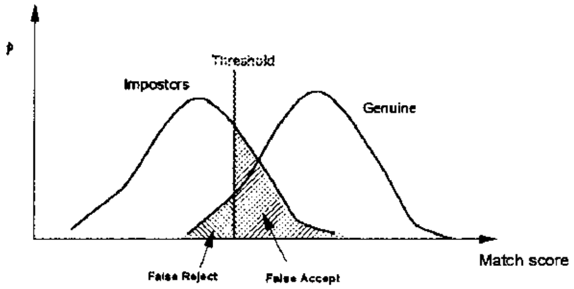


Figure 2: Error tradeoff in a biometrics system.

In a biometrics-based system, the relative false accept and false reject rates can be set by choosing a particular operating point (i.e., matching threshold). To provide high security, biometrics systems are usually operated at a low FAR (vs. an equal error rate). Typical error rates for a fingerprint system are in the range of 10^{-6} for false accept and 10^{-4} for false reject [14]. Thus, the probability that the fingerprint signal is supplied by a genuine person given a good matching score is significantly high. This confidence generally provides better non-repudiation support than passwords do.

3 Pattern recognition based threat model

A generic biometrics system can be cast in the framework of a pattern recognition system. The stages of such a generic system are shown in Figure 3. Excellent introductions to such automated biometrics systems can be found in [12, 13].

In general, the first stage involves biometrics signal acquisition from the user (e.g., the inkless fingerprint scan). The acquired signal typically varies significantly from presentation to presentation; hence, pure pixel-based matching techniques do not work reliably. For this reason, the second signal processing stage attempts to construct a more invariant representation of this basic input signal (e.g., in terms of fingerprint minutiae). The invariant representation is often a spatial domain characteristic or a transform domain (frequency) domain characteristic, depending on the particular biometric.

During enrollment of a subject in a biometrics authentication system, an invariant template is stored in a database in order to represent the particular individual. To authenticate the user against a given ID, the corresponding template is retrieved from the database and matched against a new template derived from a newly acquired input signal. The matcher arrives at a decision based on the closeness of these two templates while taking into account geometric, lighting and other signal acquisition variables.

Note that password-based authentication systems can also be put in this framework. The keyboard becomes the input device. The password encryptor can be viewed as the feature extractor and the comparator as the matcher. The template database is equivalent to the encrypted password database.

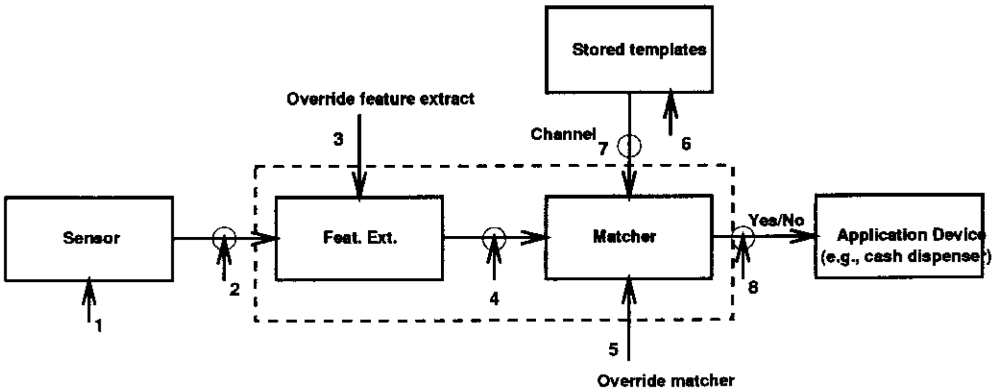


Figure 3: Possible attack points in a generic biometrics-based system.

As described in the following, such systems can be attacked at eight sources. In addition, Schneier describes many other types of abuses of biometrics in [3].

1. Presenting fake biometrics at the sensor: In this mode of attack, a possible reproduction of the biometrics is presented as input to the system. Examples include a fake finger, a copy of a signature, a face mask.

2. Resubmitting of old digitally stored biometrics signals: In this mode of attack, a recorded signal is replayed to the system bypassing the sensor. Examples include presentation of an old copy of fingerprint image or recorded audio signal of a speaker.

3. Overriding the feature extraction process: The feature extractor could be attacked with a Trojan horse so that it would produce feature sets preselected by the intruder.

4. Tampering with the biometrics feature representation: After the features have been extracted from the input signal, these could be replaced with a different synthesized feature set (assuming the representation is known). Often the two stages of feature extraction and matcher are inseparable and this mode of attack is extremely difficult. However, if minutiae are transmitted to a remote matcher (say, over the Internet), this threat is very real. One could snoop on the TCP/IP stack inside the computer and alter certain packets.

5. Overriding the matcher output: The matcher is attacked so that it always directly produces artificially high or low match scores.

6. Tampering with stored templates: The database of enrolled templates is available locally or remotely. This database might also be distributed over several servers. The stored template attacker could try to modify one or more templates in the database, which could result in authorization of a fraudulent individual or, at least, denial of service for the person associated with the corrupted template.

7. Attacking the channel between the stored templates and the matcher: The templates from the stored database are sent to the matcher through a channel. This channel could

be attacked to change the contents of the templates before these are received by the matcher.

8. Overriding the matcher: If the final match decision can be overridden with the choice of result from the hacker, this could become quite dangerous. Even if the actual pattern recognition system has excellent performance characteristics, it has been rendered useless by the simple exercise of overriding the match result.

There exist several security techniques to thwart attacks at these various points. For instance, finger conductivity or fingerprint pulse at the sensor can stop simple attacks of point 1. Encrypted communication channels [2] can eliminate at least remote attacks at point 4. However, even if the hacker cannot penetrate the feature extraction machine, the system is still vulnerable. The simplest way to stop attacks at points 5, 6 and 7 is to have the matcher and database reside at a secure location. Of course, even this cannot prevent attacks in which there is collusion. Cryptography again brings a solution to point 8.

We observe that the threats outlined in Figure 3 are quite similar to the threats to password-based authentication systems. For instance, all the channel attacks remain the same. One difference is that there is no "fake password" input detector equivalent to the fake biometrics detection processes to counter threat 1 (although, perhaps if the password was in some standard dictionary it could be deemed "fake"). Furthermore, in a password or token-based authentication system, no attempt is made thwart replay attacks (since there is no variation of the "signal" from one presentation to another). However, in an automated biometrics-based authentication system, one can go the extent of checking liveliness of the input signal.

4 Brute force strength analysis

In this section we attempt to analyze the probability that a brute force attack consisting of a set of synthetic fingerprint minutiae (attack point 4) will succeed in matching a given stored template. Note that generating all possible images (attack point 2) to guess the original matching fingerprint image would have an even larger search space and consequently would be much more difficult.

4.1 Naive model

For the purpose of analyzing the "naive" minutiae brute force dictionary attack, we assume the following.

- The system uses a minutia-based matching method and the number of paired minutiae reflects the degree of match.
- The image size, $S = 300 \times 300$.
- A ridge plus valley spread, $T = 15$ pixels.

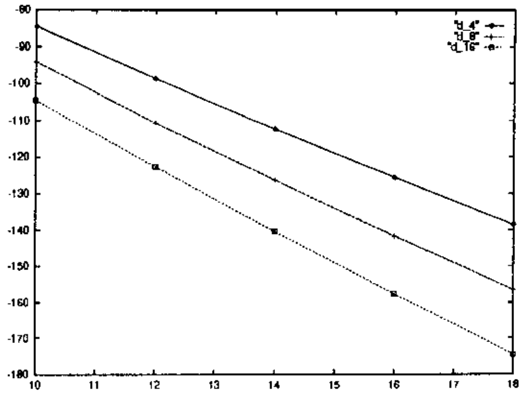


Figure 4: Probability of a successful brute force attack.

- The total number of possible minutiae sites, ($K = (S/T)^2$) = $20 \times 20 = 400$.
- The number of orientations allowed for the ridge angle at a minutiae point, $d = 4, 8, 16$.
- The minimum number of corresponding minutiae in query and reference template, $(N_q) = 10, 12, 14, 16, 18$.

Then, possible ways to place N_q minutiae in K possible locations is $\binom{K}{N_q}$;

and, possible ways to assign directions to each minutiae is d^{N_q} .

Hence, the total number of possible minutiae combinations equals

$$\binom{K}{N_q} \times (d^{N_q}) \tag{1}$$

Note that it is assumed that the matcher will tolerate shifts between query and reference minutiae of at most a ridge and valley pixel width, and of an angle by up to half a quantization bin (±45 deg for $d = 4$).

Plugging these values into Expression (1), for $d = 4$ and $N_q = 10$, the probability of randomly guessing a correct feature set is $3.6 \times 10^{-26} = 2^{-84.5}$. The log_2 of the probability of randomly guessing a correct feature set through a brute force attack for different values of d and N_q is plotted in Figure 4. This is the equivalent number of bits in a fingerprint when considered as a password. This should convince the readers that a brute force attack in the form of a random image or a random template to impersonate an individual will, on average, require a very large number of attempts before succeeding.

The forgoing analysis assumes that each fingerprint has exactly N_q minutiae, that only N_q minutiae are generated and that all of these minutiae have to match. A realistic number is much lower because one can generate more than N_q query minutiae, say, N_{total} and only some fraction N_q of these must match N_q minutiae of the reference fingerprint. This leads to a factor of about $\binom{N_{total}}{N_q}^2$ or a loss of nearly 64 bits in strength for $N_q = 10$ with $N_{total} = 50$. The equivalent strength thus is closer to 20 bits for this parameter set. A more sophisticated model, which carefully incorporates this effect is described below.

4.2 Complex model

In the naive approach, we made several simplistic assumptions. In the complex model, we will make assumptions that are more realistic and analyze the brute force attack model in a more realistic fashion.

Let the reference print have N_r minutiae, each minutiae have d possible directions and one of K possible location. The probability then that a randomly generated minutiae will match one of the minutiae in the reference print in both location and direction can be approximated by:

$$p_{est} = \frac{N_r}{K \times d} \tag{2}$$

In a comprehensive model one would really need to model the probability distribution of minutiae locations relative to the center of the print (more likely in the middle). In addition, the directional proclivities based on position (tend to swirl around the core) need to be modeled. In this model, however, we will ignore such statistical correlation between minutiae and use this somewhat simpler formulation.

While the expression above is valid for the first generated minutiae, when creating the full synthetic set it is undesirable to generate two minutiae with the same location. So after $j - 1$ minutiae have been generated, the probability that the j^{th} minutiae will match could be as high as the following (assuming the previous $j - 1$ all fail):

$$p \leq \frac{N_r}{(k - j + 1)d} \tag{3}$$

So to be conservative, while generating N_q random minutiae we can assume each of the minutiae has matching probability:

$$p = p_{hi} = \frac{N_r}{(K - N_q + 1)d} \tag{4}$$

For typically parameters values like $K = 400$, $N_q = N_r = 50$ and $d = 4$ we find $p_{est} = 0.03125$ while $p_{hi} = 0.03561$ (14% higher). This is a relatively small effect in itself, but important in the overall calculation.

Therefore, the probability of getting exactly t of N_q generated minutiae to match is about:

$$P_{thresh} = p^t(1 - p)^{N_q - t} \tag{5}$$

This derivation breaks down for small K because the minutiae matching probability changes depending on how many other minutiae have already been generated as well as on how many of those minutiae have matched. However, for the large K's typically encountered (e.g., 400) it is reasonably close.

Now there are a number of ways of selecting which t out of the N_r minutiae in the reference print are the ones that match. Thus, the total match probability becomes:

$$P_{exact} = \binom{N_r}{t} p^t (1-p)^{N_q - t} \tag{6}$$

But matches of *m or more* minutiae typically count as a verification, so we get:

$$P_{ver} = \sum_{t=m}^{N_q} \binom{N_r}{t} p^t (1-p)^{N_q - t} \tag{7}$$

For convenience, let us assume that $N_q = N_r = N$, so the above equation can be rewritten as:

$$P_{ver} = \sum_{t=m}^{N_q} \binom{N}{t} p^t (1-p)^{N-t}. \tag{8}$$

Since p is fairly small in our case, we can use the Poisson approximation to the above binomial probability density function:

$$P_{ver} = \sum_{t=m}^{N} \frac{(Np)^t e^{-Np}}{t!} \tag{9}$$

This summation is usually dominated by its first term ($t = m$). For typical parameter values the second term is 10 to 20 times smaller than the first. Neglecting all but the first term may make the overall estimate approximately 20calculations this is fine. Thus, we rewrite the expression as simply:

$$P_{ver} = \frac{(Np)^m e^{-Np}}{m!} \tag{10}$$

Because m is moderately large, we can use Stirling's approximation for the factorial and further rewrite the equation as:

$$P_{ver} = \frac{(Np)^m e^{-Np}}{\sqrt{(2\pi m)} e^{-m} m^m} \tag{11}$$

and regrouping to emphasize the exponential dependency:

$$Pver = \frac{e^{-Np}}{\sqrt{2\pi m}} \left(\frac{eNp}{m}\right)^m \tag{12}$$

This P_{ver} is plotted in Figure 5 for $N = 40$, $d = 4$, $K = 400$ with m (the number of minutiae required to match) between 10 and 35. For a value of $m = 10$, we have about 22 bits of information (close to the prediction of the revised naive model). For the legal threshold of

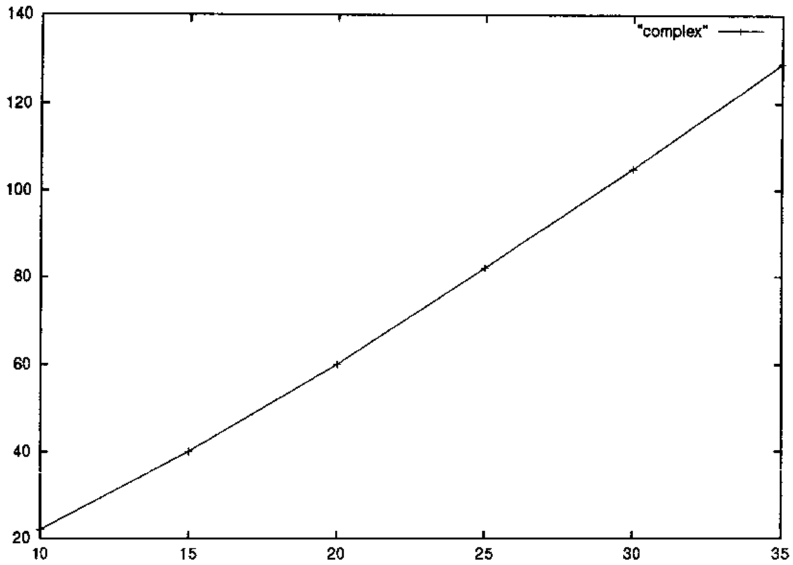


Figure 5: Bit strength in the complex model.

$m = 15$, we have around 40 bits of information (about 140 times the population of the earth). For a more typical value of $m = 25$, we have roughly 82 bits of information content in this representation. This is equivalent to a nonsense password which is 16 characters long (like "m4yus78xpmks3bc9").

We make two important observations. First, in both the naive and more sophisticated models, it can be seen that adding extra feature information at every minutiae (e.g., raising d) increases the strength of the system significantly. Similarly, if the spatial domain extent is increased or the location tolerance decreased (e.g., by raising K), the strength also increases. Both these factors directly affect p, the single minutiae matching probability, which shows up inside the exponential term of P_{ver}. Second, there is also a strong dependence on N, the overall number of minutiae in a fingerprint. For the best security, this number needs to be kept as low as possible (i.e., spurious minutiae are bad). This is one reason why the probability of break-ins is much smaller when good quality fingers are enrolled than when poor quality images are used.

5 New security enhancements

Automated biometrics can play an important role in secure user authentication. However, it comes with a new set of problems as pointed out in Section 3. We try to alleviate some of these problems with novel solutions described in this section. In particular, we will address the issue of replay attacks, i.e., the detection of stale signals (attack points 2 and 4 in Figure 3).

5.1 WSQ-based data hiding

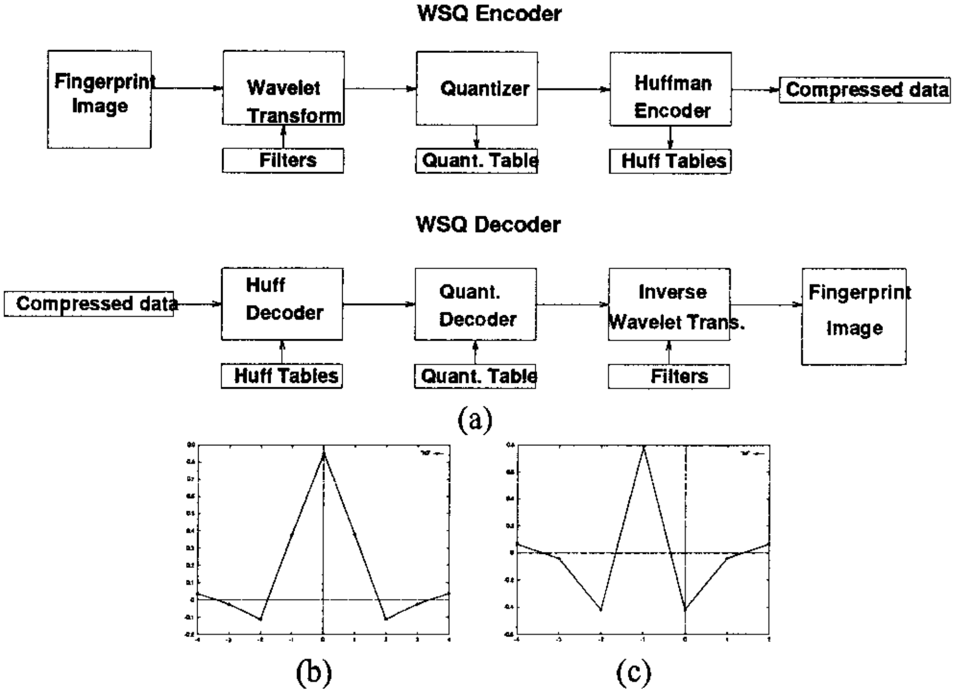
Often, a biometrics signal acquired in the field is compressed and transmitted to the authentication server. The authentication procedure is carried out at this server. In both Web-based and other on-line transaction processing systems, it is undesirable to send uncompressed fingerprint images to the server due to bandwidth limitations. A typical fingerprint image is in the order of 512 × 512 pixels with 256 gray levels, resulting in an image size of 256 Kbytes. This would take nearly 40 seconds to transmit at 53 Kbaud. Unfortunately, many standard compression methods have a tendency to distort the high-frequency spatial structural ridge features of a fingerprint image. This has lead to several research proposals regarding domain-specific compression methods. As a result, an open wavelet-based image compression scheme (WSQ) proposed by the FBI [4] has become the *de facto* standard in the industry because of its low image distortion even at very high compression ratios (over 10 : 1).

Typically, the compressed image is transmitted over a standard encrypted channel as a replacement for (or in addition to) the user's PIN. Yet, because of the open compression standard, transmitting a WSQ compressed image over the Internet is not particularly secure. If a compressed fingerprint image bitstream can be freely intercepted (and decrypted), it can be decompressed using readily available software. This potentially allows the signal to be saved and fraudulently reused (attack point 4 in Figure 3).

One way to enhance security is to use data-hiding techniques to embed additional information directly in compressed fingerprint images. For instance, if the embedding algorithm remains unknown, the service provider can look for the appropriate standard watermark to check that a submitted image was indeed generated by a trusted machine (sensor). Several techniques have been proposed in the literature for hiding digital watermarks in images. Bender et al. [7] and Swanson et al. [10] present excellent surveys of data-hiding techniques. Petitcolas et al. [11] provide a nice survey and taxonomy of information hiding techniques. Hsu and Wu [6] describe a method for hiding watermarks in JPEG compressed images. Most of the research, however, addresses issues involved in resolving piracy or copyright issues, not authentication.

Our approach is motivated by the desire to create online fingerprint authentication systems for commercial transactions that are in particular secure against replay attacks. To achieve this, the service provider issues a different verification string for each transaction. The string is mixed in with the fingerprint image before transmission. When the provider receives the image back it can be decompressed and the image can be checked for the presence of the correct one-time verification string. This guards against resubmission of stored images. The method proposed here hides such messages with minimal impact on the decompressed appearance of the image. Moreover, the message is not hidden in a fixed location (which would make it more vulnerable to discovery) but is, instead, deposited in different places *based on the structure of the image itself*. Although our approach is presented in the framework of fingerprint image compression, it can be easily extended to other biometrics.

Our information hiding scheme works in conjunction with the WSQ (Wavelet Scalar Quantization) fingerprint image encoder and decoder, which are shown in Figure 6(a). In the first step of the WSQ compression, the input image is decomposed into 64 spatial frequency subbands using perfect reconstruction multirate filter banks. The filters are implemented as a pair



(a)

(b) (c)

Figure 6: WSQ algorithm. (a) overview; (b) and (c) analysis filters.

of separable 1D filters. The two filters specified for encoder 1 of the FBI standard are plotted in Figure 6(b) and (c). The subbands are the filter outputs obtained after a desired level of cascading of the filters as described in the standard (see Figure 7(b)). For example, subband 25 corresponds to the cascading path of '00,10,00,11' through the filter bank. The first digit in each binary pair represents the row operation index. A zero specifies lowpass filtering on the row (column) while a one specifies highpass filtering on the row (column).

There are two more stages to WSQ compression. The second stage is a quantization process where the discrete wavelet transform (DWT) coefficients are transformed to integers with a small number of discrete values. This is accomplished by uniform scalar quantization for each subband. There are two characteristics for each band: zero of the band (Z_k) and width of the bins (Q_k). These parameters must be chosen carefully to achieve a good compression ratio without introducing significant information loss that will result in distortions of the decompressed images. The Z_k and Q_k for each band are transmitted directly to the decoder. The final stage is Huffman coding of the integer indices for the DWT coefficients. For this purpose, the bands are grouped into three blocks. In each block, the integer coefficients are re-mapped to numbers between 0-255 prescribed by the translation table described in the standard. This translation table encodes run lengths of zeros and large values. Negative coefficients are translated in a similar way by this table.

Our data-hiding algorithm works on the quantized indices before this final translation (i.e., between stages 2 and 3). We assume the message size is very small compared to the image size (or, equivalently, the number of DWT coefficients). The Huffman coding characteristics and tables are not changed; the tables are computed as for the original coefficients, not after the coefficient altering steps described next.

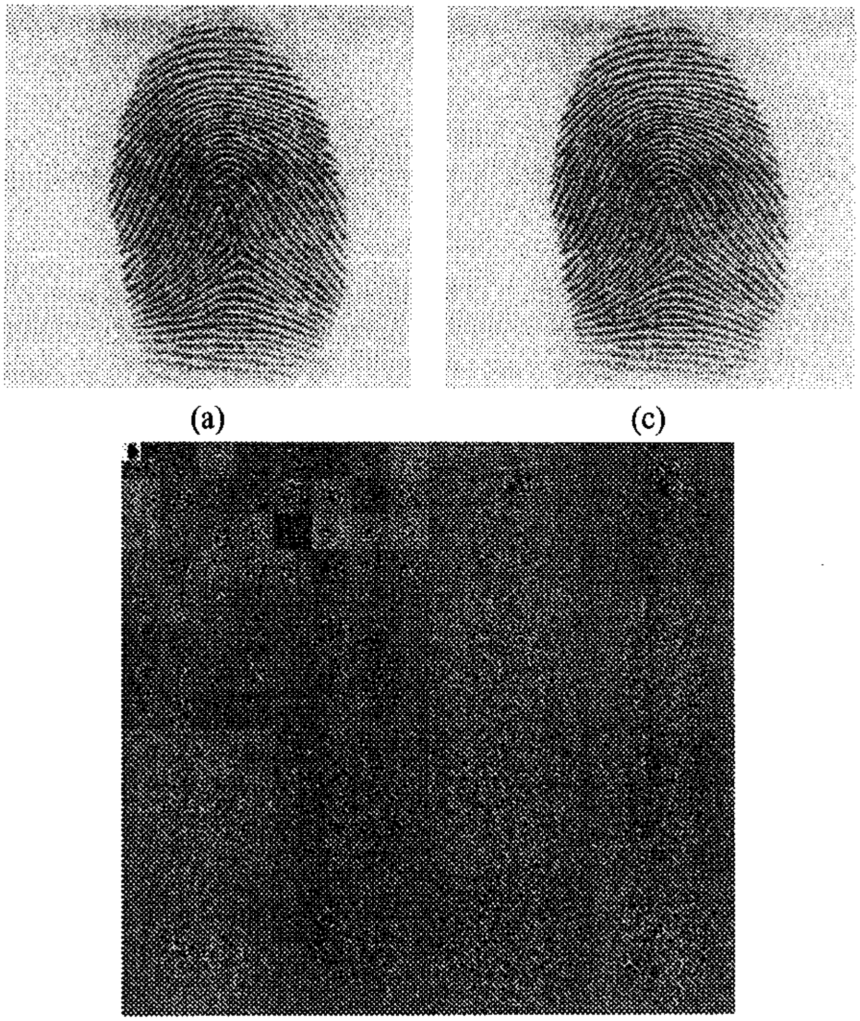
As mentioned, our method is intended for messages which are very small (in terms of bits) compared to the number of pixels in the image. The basic principle is to find and slightly alter certain of the DWT coefficients. However, care must be taken to avoid corrupting the reconstructed image. To hide a message during the image encoding process, we perform three (or, optionally, four steps) basic steps:

- The selection of a set of sites S: Given the partially converted quantized integer indices, the role of this stage is to collect the indices of all possible coefficient sites where a change in the least significant bit is tolerable. Typically, all sites in the low frequency bands are excluded. Even small changes in these coefficients can affect large regions of the image because of the low frequencies. Subsequently, candidate sites are selected with coefficient of large magnitude. Making small changes to the larger coefficients leads to relatively small percentage changes in the values and hence minimal degradation of the image. Note that among the quantizer indices there are special codes to represent run lengths of zeroes and large integer values, as well as other control sequences. All coefficient sites incorporated into these values are avoided. In our implementation, we only select sites with translated indices ranging from 107 to 254, but excluding 180 (an invalid code).

- Generating a seed for random number generator for selecting sites for modification: Sites from the candidate set S, which will be modified, are selected in a pseudo-random fashion. To retain predictability in encoder and decoder, the seed for the random number generator is based on the subbands that are not considered for alteration. For example, in the selection process the contents of sub-bands 0-6 are left unchanged in order to minimize distortion. Typically, fixed sites within these bands are selected, although in principle any statistic from these bands may be computed as seed. Selecting the seed in this way ensures that the message is embedded at varying locations (based on the image content). It further ensures that the embedded message can only be read if the proper seed selection algorithm is known by the decoder.

- Hiding the message at selected sites by bit setting: The message to be hidden is translated into a sequence of bits. Each bit will be incorporated into a site chosen pseudo-randomly by a random number generator seeded as described above. That is, for each bit a site is selected from the set S based on the next output of the seeded pseudo-random number generator. If the selected site has already been used, the next randomly generated site is chosen instead. The low order bit of the value at the selected site is changed to be identical to the current message bit. On average, half the time this results in no change at all of the coefficient value.

- Appending the bits to the coded image: Optionally, all the original low order bits can be saved and appended to the compressed bit stream as a user comment field (an appendix). The appended bits are a product of randomly selected low-order coefficient bits and the message, and hence these bits are uncorrelated with the hidden message.



(a)

(c)

(b)

Figure 7: WSQ results. (a) fingerprint image; (b) its 64 subbands; (b) reconstructed image with embedded message.

The decoder also performs three steps (optionally four steps). The first two steps are identical to the first steps of the encoder. These steps construct the set S and compute the seed for the random number generator. The third step uses the pseudo-random number generator to select specific sites in S in a particular order. The least significant bits of the values at these sites are extracted and concatenated to recover the original message.

If a restoration appendix is included, the decoder can optionally restore the original low-order bits while reconstructing the message. This allows perfect reconstruction of the image (up to the original compression) despite the embedded message. Because the modification sites S are carefully selected, the restored decompressed image will be nearly the same as the decompressed image with the message still embedded. In practice, the error due to the embedded message is not perceptually significant, and does not affect subsequent processing and authentication.

Using this process only a specialized decoder can locate and extract the message from the compressed image during the decoding process. This message might be a fixed authentication stamp, personal ID information which must match some other part of the record (which might have been sent in the clear), or some time stamp. Thus, if the bit stream does not contain an embedded message or the bit stream is improperly coded, a specialized decoder will fail to extract the expected message and hence can reject the image.

Many versions of the same algorithm are possible by using different random number generators or partial seeds. This means it is possible to make every implementation unique without much effort; the output of one encoder need not be compatible with another version of the decoder. This has the advantage that cracking one version will not compromise any other version.

5.2 Image based challenge/response method

Besides interception of network traffic, more insidious attacks might be perpetrated against an automated biometrics authentication system. One of these is a replay attack directly on the input signal (attack point 2 in Figure 3). We propose a new method to thwart such attempts based on a modified challenge/response system. Conventional challenge/response systems are based on challenges to the user, such as requesting a mother's maiden name, or challenges to a physical device, like a special-purpose calculator that computes a numerical response. Our approach is based on challenges to the sensor that is assumed to have enough intelligence to respond to the challenges. Many silicon fingerprint scanners [1] are able to exploit the proposed method as a processor can be integrated without much effort.

Note that standard cryptographic techniques are not a suitable substitute. While they are mathematically strong, they are also very computationally intensive and would require maintaining secret keys for a large number of sensors. Moreover, the encryption techniques cannot check for liveliness of a signal. An old stored image could be given to the encryptor that will happily encrypt it. Similarly, a digital signature of a signal checks only for its integrity, not its liveliness.

Our system computes a response string, which depends not only on the challenge string, but also on the content of the returned image. The changing challenges ensure that the image

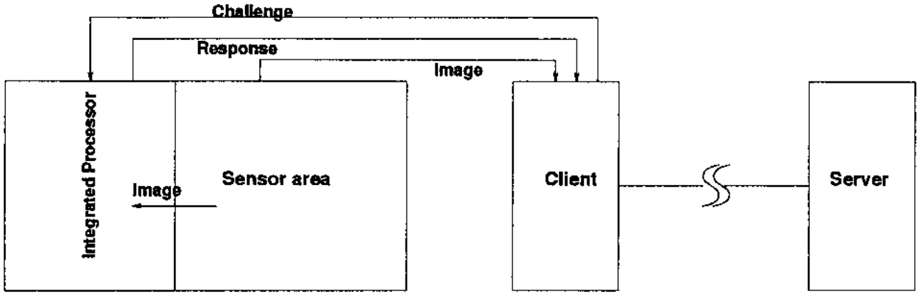


Figure 8: Signal authentication based on challenge/response.

was acquired after the challenge was issued. The dependence on image pixel values guards against substitution of data after the response has been generated.

Our proposed solution works as shown in Figure 8. A transaction is initiated at the user terminal or system. First, the server generates a pseudo-random challenge for the transaction and the sensor. Note that we assume that the transaction server itself is secure. The client system then passes the challenge on to the intelligent sensor. Now, the sensor acquires a new signal and computes the response to the challenge that is based in part on the newly acquired signal. Because the response processor is tightly integrated with the sensor (preferable on the same chip), the signal channel into the response processor *is assumed ironclad and inviolable.* It is just about impossible to inject a fake image under such circumstances.

As an example of an image-based response, consider the function “x1+” which operates by appending pixel values of the image (in scan order) to the end of the challenge string. A typical challenge might be “3, 10, 50”. In response to this, the integrated processor then selects the 3rd, 10th and 50th pixel value from this sequence to generate an output response such as “133, 92, 176”. The complete image as well as the response is then transmitted to the server where the response can be verified and checked against the image.

Other examples of responder functions include computing a checksum of a segment of the signal, a set of pseudo-random samples, a block of contiguous samples starting at a specified location and with a given size, a hash of signal values and a specified known function of selected samples of the signal. A combination of these functions can be used to achieve arbitrarily complex responder functions. The important point is that the response depends on the challenge and the image itself.

The responder can also incorporate several different response functions from among the challenger could select one. For instance, the integrated processor might be able to compute either of two selectable functions, “x1+” and “x10+”. Financial institution A might use function “x1+” in all its units, while institution B might use “x10+” in all of its units. Alternatively, for even numbered transactions, function “x10+” might be used, and for odd numbered transactions “x1+” might be used. This variability makes it even harder to reconstruct the structure and parameters of the response function.

6 Cancelable biometrics

Deploying biometrics in a mass market, like credit card authorization or bank ATM access, raises additional concerns beyond the security of the transactions. One is the public's perception of invasion of privacy. In addition to private information such as name and date of birth, the user is asked to give images of their body parts, such as fingers, faces and iris. These images, or other biometrics signals, will be stored in digital form in databases. A concern is the possible sharing of databases of biometrics signals with law enforcement agencies, or sharing of these databases among commercial organizations. These privacy concerns can be summarized as follows:

1. Much data about customers and customer behavior is stored. The public is concerned about every bit of additional information that is known about them.

2. The public is, in general, suspicious of central storage of information that is associated with individuals. This type of data ranges from medical records to biometrics. These databases can be used and misused for all sorts of purposes, and the databases can be shared among organizations.

3. The public is, rightfully or wrongfully so, worried about giving out biometrics because these could be used for matching against databases used by law enforcement agencies. They could be, for example, be matched against the FBI or INS fingerprint databases to obtain criminal records.

Hence, biometrics being coupled with other personal parametric data is a concern, as is the potential use of stored biometrics for searching other databases.

These concerns are aggravated by the fact that a biometrics cannot be changed. One of the properties that make biometrics so attractive for authentication purposes, their invariance over time, is also one of their liabilities. When a credit card number is somehow compromised, the issuing bank can just assign the customer a new credit card number. When a biometrics is compromised, however, a new one cannot be issued.

As an answer to these issues, we propose a novel concept to deal with these issues called a "cancelable biometric". This is an intentional, repeatable distortion of a biometrics signal based on a chosen transform. The biometrics signal is distorted in the same fashion at each presentation, for enrollment and for every authentication. With this approach, every instance of enrollment can use a different transform thus rendering cross-matching impossible. Furthermore, if one representation is compromised, then the transformation can simply be changed to create a new representation for re-enrollment.

Cancelable transforms can be applied in either the signal domain or the feature domain. That is the biometrics signal can be transformed directly after acquisition or the signal can be processed is usual and the extracted features can be transformed. Moreover, extending a template to a larger representation space via a suitable a transform can further increase the brute force strength of the system. Several example transforms are described below.

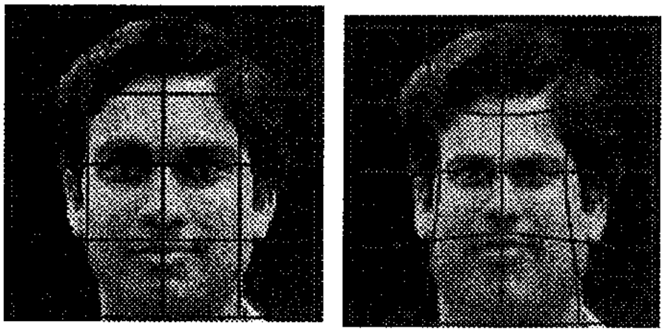


Figure 9: Distortion transform based on morphing.

- Transform of the biometrics at the signal level: This category includes grid morphing and block permutation as suitable transforms. Our claim is that the transformed images cannot be successfully matched against the original images, or against similar transforms of the same image using different parameters. In Figure 9, the original image is shown with an overlaid grid aligned with the features of the face. In the adjacent image, we show the morphed grid and the resulting distortion of the original face. In Figure 10, a block structure is imposed on the image aligned with characteristic points. The blocks in the original image are subsequently scrambled randomly but repeatable.

- Transform of the biometrics in the feature domain: One transforms in this category is the random, repeatable perturbations of the feature points. This can be done within the same physical space as the original, or while increasing the range of the axes. The second case provides more brute force strength as was noted in Section 4 (this effectively increases the value of K). Examples of these transforms are shown in Figure 11. These transform are non-invertible hence the original feature sets cannot be recovered from the distorted versions. For instance, it is impossible to tell which of the two blocks the points in composite block B,D originally came from. Consequently, the owner of the biometrics cannot be identified except through the information associated with that particular enrollment.

Note that for the transform to be repeatable, we need to have the biometrics signals properly registered before the transformation. Fortunately, this problem has been partially answered by a number of techniques available in the literature (such as finding the "core" and "delta" points in a fingerprint). Ideally the transform should be non-invertible so that the true biometric of a user cannot be recovered from one or more of the distorted versions stored by various agencies.

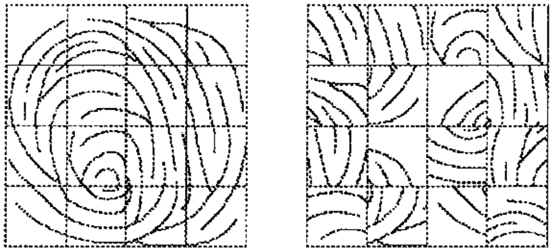


Figure 10: Distortion transform based on block scrambling.

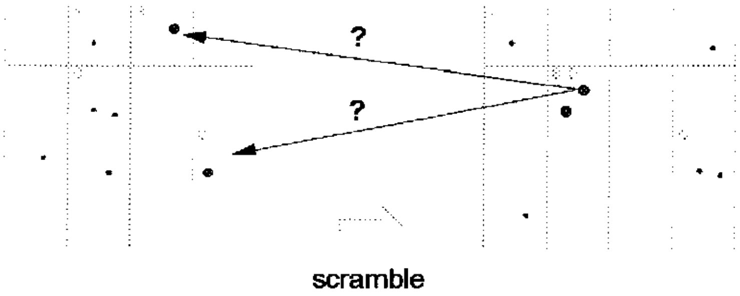


Figure 11: Distortion transform based on feature perturbation.

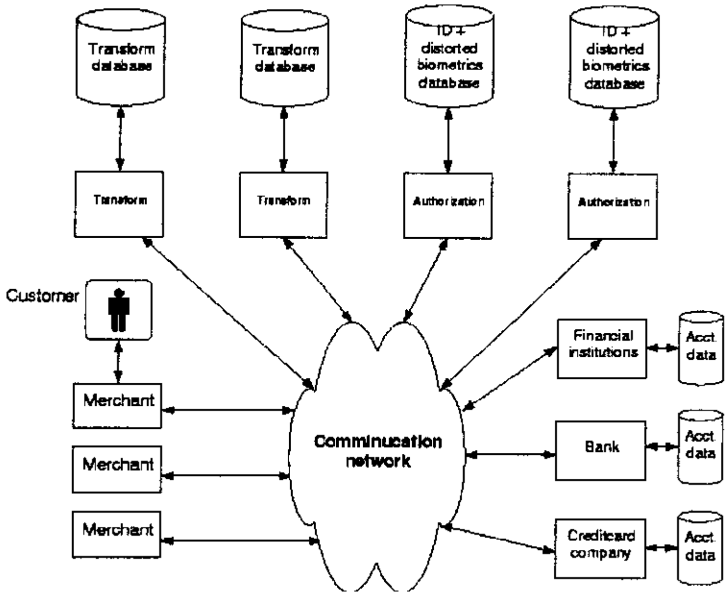


Figure 12: Business process based on cancelable biometrics.

The techniques presented here for transforming the biometrics signal differs from simple compression using signal or image processing techniques. While compression of the signal causes it to lose some of its spatial domain characteristics, it strives to preserve the overall geometry. That is, two points in a biometrics signal before compression are likely to remain at comparable distance when decompressed. Our technique also differs from encryption. The purpose of encryption is to allow a legitimate party to regenerate the original signal. Encryption of signals does not permanently obscure the signal in a non-invertible manner as cancelable transforms do.

When employing cancelable biometrics, there are several places where the transforms, the parameters of the transform, and identification templates could be stored. This leads to a possible distributed business process model as shown in Figure 12. An individual user has subscribed to multiple services (like e-commerce merchants or banks). The authentication for each transaction might be performed either by the service provider itself, or by an independent third party. Similarly, the distortion transform might be managed either by the authenticator or by still another independent agency. Alternatively, for the best privacy the transform might remain solely in the possession of the user, embedded within something like a smart card.

7 Conclusions

Biometrics based authentication has many usability advantages over older systems such as passwords. Among these are the facts that the user can never lose his biometrics and the fact that the biometrics signals is difficult to steal or forge. We have shown that the intrinsic strength of a biometric signal is quite good, especially for fingerprints, when compared to conventional passwords.

Yet, any system, including biometrics systems, is vulnerable when hackers are determined enough. We have highlighted eight particular weak points in a generic biometrics system and have discussed possible attacks. We suggested several ways to alleviate some of these particular security threats. Replay attacks have been addressed using data hiding techniques to secretly embed a telltale mark directly in the compressed fingerprint image. A challenge/response method has been proposed to check the liveliness of the signal acquired from an intelligent sensor.

Finally, we have touched on the often-neglected problems of privacy and revocation of biometrics. It is somewhat ironic that the greatest strength of biometrics, the fact that the biometrics does not change over time, is at the same time its greatest liability. Once a biometrics has been compromised, it is compromised forever. To address this issue, we have proposed intentionally applying repeatable non-invertible distortions to a biometrics signal in order. Reissuance simply requires the specification of a new distortion. Privacy is enhanced because different distortions can be used for different services and the true biometrics never has to be revealed to the authentication server. The intentionally distorted biometrics, in addition, cannot be matched to legacy databases.

References

[1] T. Rowley, "Silicon Fingerprint Readers: A solid state approach to biometrics", in *Proc. of the CardTech/SecureTech*, May 97, Vol. 1, pp. 152–159.

[2] B. Schneir, "Security pitfalls in cryptography", in *Proc. of CardTech/SecureTech*, April 98, Vol. 1, pp. 621–626.

[3] B. Schneier, "The uses and abuses of biometrics", *Communications of the ACM*, August 1999, Vol. 42, No. 8, pp. 136.

[4] "WSQ Gray-scale Fingerprint Image Compression Specification", US Federal Bureau of Investigation, 1993.

[5] C. M. Brislawn, J. N. Bradley, R. J. Onyshczak, and T. Hopper. "The FBI compression standard for digitized fingerprint images", in *Proc. of SPIE, Vol. 2847, Denver*, Aug. 1996, pages 344–355.

[6] C. T. Hsu and J. L. Wu, "Hidden digital watermarks in images", *IEEE Trans. on Image Processing*, Vol. 8, No. 1, Jan. 1999, pp. 58–68.

[7] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding", *IBM Systems Journal*, Vol. 35, No. 3 & 4, 1996, pp. 313–335.

[8] N. Memon and P. W. Wong, "Protecting digital media content", *Communication of the ACM*, Vol. 41, No. 7, July 1998, pp. 35–43.

[9] S. Mallat, "Wavelets for vision", *Proc. of the IEEE*, Vol. 84, No. 4, April 1996, pp. 604–614.

[10] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multi-media data embedding and watermarking technologies", *Proc. of the IEEE*, Vol. 86, No. 6, June 1998, pp. 1064–1087.

[11] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information Hiding – A survey", *Proc. of the IEEE*, Vol. 87, No. 7, July 1999, pp. 1062–1078.

[12] B. Miller, "Vital signs of identity",IEEE Spectrum, Vol. 31, No. 2, Feb. 1994, pp. 22–30.

[13] A. K. Jain, L. Hong and S. Pankanti, "Biometrics Identification", Communications of the ACM, Vol. 43, No. 2, February 2000, pp. 91–98.

[14] A. K. Jain, L. Hong, S. Pankanti and R. Bolle, "An identity-authentication system using fingerprints", *Proc. of the IEEE*, Vol. 85, No. 9, September 1997, pp. 1365–1388.

[15] L. Ogorman, "Practical systems for personal fingerprint authentication", IEEE Computer, Vol. 33, No. 2, February 2000, pp. 58–60.

[16] R. M. Bolle, N. K. Ratha and S. Pankanti, "Research issues in biometrics", Proc. of ACCV '98, Hong Kong, Jan. 1998, pp. 2–9.

[17] T. Ebringer, P. Thorne and Y. Zheng, "Parasitic authentication to protect your E-wallet", IEEE Computer, Vol. 33, No. 10, October 2000, pp. 54–60.