An Analysis of Minutiae Matching Strength

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 exponential growth 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 biometrics-based authentication systems be designed to withstand different sources of attacks on the system when employed in security-critical applications. This is even more important for unattended remote applications such as e-commerce. In this paper we outline the potential security holes in a biometrics-based authentication scheme, quantify the numerical strength of one method of fingerprint matching, then discuss how to combat some of the remaining weaknesses.

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. These could also benefit from enhanced security. Automated biometrics technology in general, and fingerprints in particular, can provide a much more accurate and reliable user authentication method.

In this paper, we present in more detail the problems unique to biometric authentication systems and propose solutions to several of them. Though our analysis is very general and can be extended to other biometrics, we will focus on fingerprint recognition as an example throughout. In Section 2 we use a pattern recognition model of a generic biometrics system to help identify the possible attack points. In Section 3 we analyze the power of a minutia-based fingerprint system in terms of probability of a brute force attack being successful. In Section 4 we propose several techniques to ameliorate problems with "replay attacks", the most likely source of attempted fraud.

2 Security of biometrics

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 the it at those points. Unlike password systems, which are prone to brute-force dictionary attacks, biometrics systems require substantially more effort to crack. Although standard encryption techniques are useful in many ways to prevent a breach of security, there are several new type of attacks are possible in the biometrics domain. If biometrics are used as a supervised authentication tool, this may not be a concern. But in remote unattended applications, such as web-based e-commerce applications, hackers may have enough time to make numerous attempts before being noticed, or may even be able to physically violate the remote client.

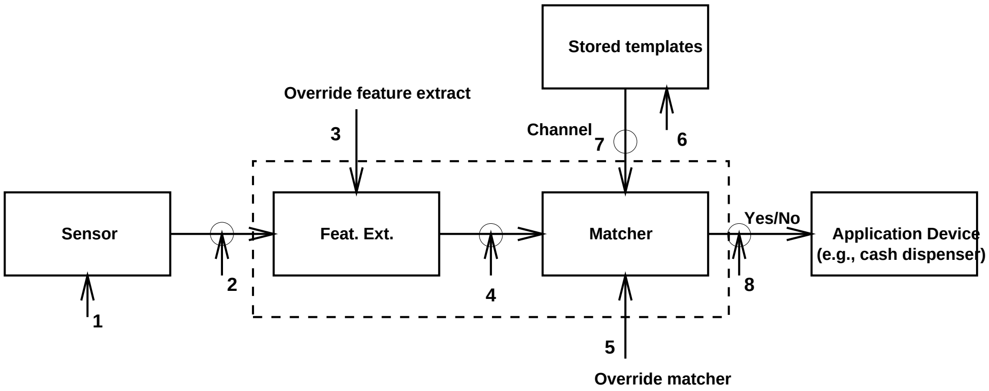

Figure 1: Possible attack points in a generic biometrics-based system.

2.1 Pattern recognition based Threat model

A generic biometric system can be cast in the framework of a pattern recognition system. The stages of such a generic system are shown in Figure 1. Excellent introductions to such automated biometrics can be found in [5, 4]. Note that the password based authentication systems can also be put in this framework. The keyboard becomes the input device. The password encryptor becomes the feature extractor and the comparator becomes the matcher. The template database is equivalent to the encrypted password database.

There are in total eight basic sources of attack on such systems as described below. In addition, Schneier describes many other types of abuses of biometrics in [3].

1. Fake biometric at the sensor: In this mode of attack, a possible reproduction of the biometric being used will be presented to the system. Examples include a fake finger, a copy of a signature, a face mask.

2. Resubmission of old digitally stored biometrics signal: In this mode of attack, an old recorded signal is replayed into the system bypassing the sensor. Examples include presentation of an old copy of fingerprint image or recorded audio signal of a speaker (a "replay" attack).

3. Override feature extract: The feature extractor could be attacked with a Trojan horse so that it would produce feature sets chosen by the hacker.

4. Tampering with the feature representation: After the features have been extracted from the input signal they are 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) than this threat is very real. One could snoop on the TCP/IP stack inside the computer and alter certain packets.

5. Override matcher: The matcher is attacked to always directly produce the an artifically high or low match score.

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 tries to modify one or more templates in the database which could result in authorization for a fraudulent individual, or at least denial of service for the person associated with the corrupted template.

7. Channel attack between stored templates and the matcher: The templates from the stored database are sent to the matcher through a channel which could be attacked to change the contents of the templates before they reach the matcher.

8. Decision override: If the final result can be overridden with the choice of result from the hacker, the final outcome is very dangerous. Even if the actual pattern recognition system had an excellent performance characteristics, it has been rendered useless by the simple exercise of overriding the result.

There exist several techniques to thwart attacks at various points. For instance, sensing finger conductivity or pulse can stop simple attacks at point 1. Encrypted communication channels [2] can eliminate at least remote attacks at points 4 and 8. The simplest way to stop attacks at points 5, 6 and 7 is to have the matcher and database reside in a secure location. Storing templates in a smartcard that a user brings with them to the point of service can also eliminate some attacks of type 6 [6]. Of course even this cannot prevent attacks in which there is collusion between a hacker and security personnel.

2.2 Comparison to passwords

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 an 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 in such a phrase anyhow (especially with no errors). Fortunately, automated biometrics can provide the security advantages of long passwords while still retaining the speed and simplicity of short passwords.

We also observe that the threats outlined in Figure 1 are quite similar in a password-based authentication system. For instance, all the channel attacks remain the same. However, in general fewer of them are typically covered. One such difference is that there is no "fake password" input detector equivalent to the fake biometrics described in 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.

Another important difference concerns the matching subsystem. A password based method always provides a crisp result: if the passwords match, it grants access and otherwise refuses access. However the performance of a pattern recognition system in general is dependent on several factor such as the quality of input and enroll data along with the basic characteristics of the underlying algorithm. This is typically reflected in a graded overall match "score" between the submitted biometric and a stored reference. In a biometrics-based system, we can purposely set a threshold on the score to directly control the false accept and false reject rates. Inverting this, given a good matching score the system can guarantee that the probability of signal coming from a genuine person is significantly high. Such a calibrated confidence measure can be used to tackle non-repudiation support – something that passwords cannot provide.

3 Brute force attacks

In this section we show the relationship between the number of brute force attack attempts (point 4 in Figure 1) as a function of number of minutiae that are expected to match in the matcher subsystem. Generating all possible images (point 2) to guess the matching fingerprint image has a much larger search space and hence would be an even harder problem.

3.1 Naive model

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

- The system uses a minutia-based method and the number of paired minutiae reflect the degree of match
- Image size $S = 300 \times 300$
- A ridge plus valley spread T = 15 pixels
- Total number of possible minutia sites $K = (S/T)^2$ = 20 × 20 = 400.
- Number of orientations allowed for the ridge angle at the minutia point d = 4, 8, 16
- Minimum number of common minutiae in a query template N_q = 10, 12, 14, 16, 18

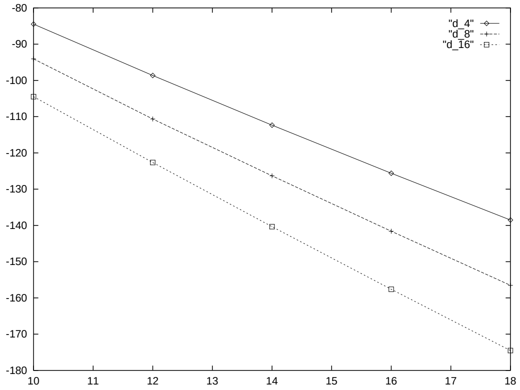

Figure 2: Probability (log_2) of a successful brute force attack versus number of minutiae under the naive model.

Possible ways to place N_q minutiae in a possible K locations = $\binom{K}{N_q}$

Possible ways to assign directions to each minutia = d^{N_q}

Hence, the total number of possible minutia combinations is:

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

Plugging the values into equation (1), for d=4, N_q=10, the probability of randomly guessing a correct feature set is $2^{-84.5}$, or one chance in 2.7×10^{25}. The log_2 of probability of randomly guessing a correct feature set through brute force attack for different values of d and N_q is plotted in Figure 2. 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 the system will require a lot of effort.

Note that the matcher has been assumed to tolerate shifts in minutia points at most by a ridge and valley pixel width. We did not constrain the angles at every minutia. A more powerful model is described below. If only those minutia patterns are generated that represent “true” fingers by modeling priors and dependence between minutiae, this number could be lower than shown here. The probability of break-ins when good quality fingers are enrolled is of course much smaller than that for poor quality fingerprint images and may be near this theoretical upper bound. The forgoing analysis assumed each fingerprint had exactly N_q minutiae, only N_q minutiae were generated and that all of these had to match.

Yet a realistic number is much lower because one can generate more than N_q minutiae say N_{total} and some N_q of them must match some N_q 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 $N_{total} = 50$ (but still only one chance in a million).

3.2 Complex model

In the naive approach, we made several simplistic assumptions. In this model, we will make more realistic assumptions and analyze the brute force attack model.

If reference print has N_r minutiae and each minutiae has d possible directions and one of K possible sites, then the probability that a randomly generated minutia will match one of the minutiae in the reference print in both site and direction is given by

$$p_{est} = \frac{N_r}{K_d} \qquad (2)$$

However, when generating random minutiae it is not desirable to generate two minutiae with the same site. So after j-1 minutia have been generated, the probability that the j^{th} minutiae will match could be as high as the following

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

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

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

The chance of getting exactly t of N_q generated minutiae to match is therefore given by

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

This break down for small K because the minutia matching probability changes depending on how many other minutiae have already been generated as well as on how many of those minutiae have matched.

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 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 a Poisson approximation to the binomial PDF.

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

This summation is usually dominated by its first term. So neglecting all but the first term we find:

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

Because m is large, we can use Stirling's approximation and the equation can be written as

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

The P_{ver} is plotted in Figure 3 for N=40 and d=4, K = 400 with m going from 10 to 35. For a value of m = 25, we roughly have 82 bits of information content in this representation. This is equivalent to a nonsense password which is 16 characters long (like "m4yus78xpmks3bc9").

We make several important observations. It can be seen in the simplistic and the complex model computations that if we have other local characteristics that can be attached to a minutia, then the probability of a brute force attack can be much lower through a brute force method (d is larger so p is smaller). And if the extent of spatial domain is increased (K is larger so p is smaller), the strength also increases. 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 – spurious minutiae from poor images are particularly detrimental.

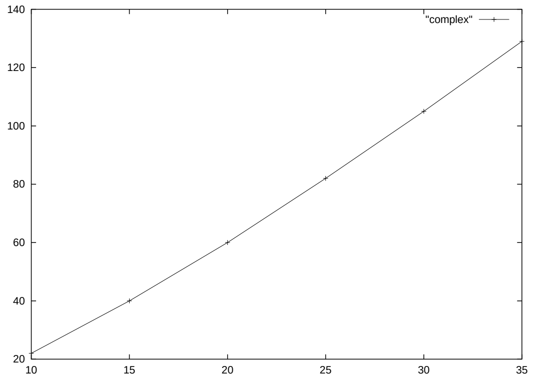

Figure 3: Bit strength (y) versus minutiae matched (x) in the complex model.

4 Replay attacks

As discussed earlier, one source of attack is the fraudulent resubmission of previously intercepted biometrics or biometric feature sets. Fortunately, several image-based techniques can be applied to address this problem. We describe two solutions based on our earlier work to thwart such replay attacks.

4.1 Image based challenge/response method

Standard cryptographic techniques, though mathematically strong, are computationally very intensive and require maintaining a secret key base for a large number of sensors. Moreover, encryption techniques cannot check for liveliness of a signal. An encryptor will accept an old stored image as readily as a fresh one. Similarly a hash or digital signature of a signal does not check for its liveliness, only its integrity.

The motivation of our approach is based instead on a challenge/response systems. Conventional challenge/response systems are based on challenges to the user. Our approach is based on challenges *to the sensor*. The sensor is assumed to have enough intelligence to respond to the challenges. We assure liveliness by providing the sensor with a different challenge each time and exploit the availability of a large number of image pixels to produce image-dependent response functions.

Our proposed solution works as follows. At the user terminal or system, the transaction gets initiated. The transaction server then generates a pseudo-random challenge for the transaction and sends it to the intelligent sensor. Note that we assume that the transaction server is assumed to be secure. The sensor acquires a signal at this point of time and computes a response to the challenge based on the new biometric signal. For instance, a typical challenge might be “3, 10, 50”. The integrated processor might then select the 3rd, 10th and 50th pixel values from the new image to generate an output response such as ”133, 92, 176”. This could be checked by the server to make sure that the client not only knew the correct response function, but also that the client was using the same image as received by the server.

By integrating the responder onto the same chip as the sensor it is just about impossible to inject a fake image (point 2 attack). Many silicon fingerprint scanners [1] will be able to exploit the proposed method as they can integrate a processor without much effort. More details are available in [7].

4.2 WSQ-based data hiding

Fingerprints are typically compressed with a wavelet technique called WSQ. Such compressed fingerprint images are then 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 bit-stream can be intercepted (and decrypted), it can then be easily decompressed using readily available software. This potentially allows the signal to be saved and fraudulently re-used.

One way to enhance security is to use data-hiding techniques to embed additional information directly in compressed fingerprint images. For instance, assuming that the embedding algorithm remains inviolate, the service provider can look for an appropriate watermark to check that the submitted image was indeed generated by a trusted machine. Or the server might look for the response to a challenge as proposed in Section 3. The method proposed here (see [8] for more details) 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.

The data-hiding algorithm works on the quantized WSQ indices before the final entropy coding stage. It first finds DWT coefficients that can be changed slightly without substantially changing the image appearance (typically high frequencies with large magnitudes). A subset of these sites is chosen by a pseudo-random number generator which has been suitably seeded (e.g. based on some of low frequency coefficients). The system then alters the least significant bit of the selected values to reflect the desired message bits. Note that we assume the message size is very small compared to the image size (or, equivalently, the number of DWT coefficients).

Any decoder can still reconstruct a reasonably faithful image, even with these unknown altered coefficients. However, only the right decoder can locate and extract the message from the compressed image during the decoding process. Many versions of the same algorithm are possible by using different random number generators or seeding strategies. 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 further advantage that cracking one version will not necessarily compromise another.

5 Conclusions

The weakest link in secure system design is user authentication. Biometrics can demonstrably improve this in terms of raw strength. And, for the same level of security, biometrics are preferable to passwords on the basis of user convenience alone. However, care must still be taken to prevent break-ins and special biometric-related issues must be understood. In particular, replay attacks must be guarded against. We proposed several methods, including an intelligent sensor challenge/response method and a data hiding technique for compressed signals, to bolster this aspect of such systems.

References

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

[2] B. Schneir, "Security pitfalls in cryptography", Proc. of CardTech/SecureTech, Washington D.C., 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] A. Jain, L. Hong and S. Pankanti, "Biometrics Identification", Communications of the ACM, February 2000, pp. 90–98.

[5] B. Miller, "Vital signs of Identity", IEEE Spectrum, February 1994, pp. 22–30.

[6] N. K. Ratha and R. M. Bolle, "Smartcard based authentication", in Biometrics: Personal Identification in Networked Society (Eds. A. Jain, R. Bolle and S. Pankanti), Kluwer, 1999, pp. 369–384.

[7] N. K. Ratha, J. H. Connell and R. M. Bolle, "A biometrics-based secure authentication System", Proc. of the AutoID 99, Oct. 1999, pp. 70–73.

[8] N. K. Ratha, J. H. Connell and R. M. Bolle, "Secure data hiding in wavelet compressed fingerprint images", Proc. of the ACM Multimedia Workshop on Multimedia and Security, Nov. 2000, pp. 127–130.