BIOMETRICS, ACCESS CONTROL, SMART CARDS: A NOT SO SIMPLE COMBINATION*

Gaël Hachez
UCL Crypto Group
Université Catholique de Louvain
Place du Levant,3 B-1348 Louvain-la-Neuve, Belgium
hachez@dice.ucl.ac.be

Francois Koeune
UCL Crypto Group
Université Catholique de Louvain
Place du Levant,3 B-1348 Louvain-la-Neuve, Belgium
fkoeune@dice.ucl.ac.be

Jean-Jacques Quisquater
UCL Crypto Group
Université Catholique de Louvain
Place du Levant,3 B-1348 Louvain-la-Neuve, Belgium
jjq@dice.ucl.ac.be

Abstract

Although biometrics can be an useful component for access control, the security they procure is often overestimated, as if they were a magic tool whose simple use will automatically prevent each and every type of attack. Biometrics are not secure unless they are embedded in a strong cryptographic protocol, whose design pays special attention to their specificities. In particular, smart card reveals to be an useful and efficient partner of biometrics for such a protocol. This paper reviews and discusses the most important issues raised by biometrics and presents a secure authentication protocol skeleton.

*Work partially done within the European IST project BANCA

1. INTRODUCTION

The great advantage of biometrics is that it relieves the user from the difficult task of choosing and remembering a good key. A fingerprint or iris is probably complex enough to be turned into a strong 128-bit key, and the user does not need to memorise it, as he always carries it with him.

On the other hand, the possibility of fraud with biometric data should not be underestimated. Many designers seem to ignore this, believing that biometrics is the final and perfect solution to each and every identification problem[1].

Finally, biometric data present some specificities that make them fundamentally different from a classical (basically, a random bit string) key. Once again, many designers seem to ignore this point and think that, to obtain a biometrics-based security protocol, it suffices to take a well-known security protocol and replace the key by biometric data. We believe many important issues are ignored by not taking their specificities into account. The goal of this paper is to overview them.

After reviewing some of the characteristics of biometric keys, we will identify the basic components of a biometrics-based authentication protocol and the properties they must satisfy. We will then build a general-purpose authentication protocol skeleton based on biometrics. We finish by considerations about the possible future of biometrics, as technology (and therefore pirates' skills) will evolve.

2. SIMILARITIES BETWEEN BIOMETRIC DATA AND KEYS

In many respects, biometrics data are just like keys. This is often ignored by designers, who seem to believe that, as the real-world object which gave birth to this data is difficult to counterfeit, the system must be "more secure". But from the system's point of view, biometrics is just a bit string, not different from any other key.

For example, it is known for a long time that the naive password-based identification scheme is insecure, due to the risk of *replay attack*: an adversary can spy the login process of a valid user and then simply reuse the password he observed to enter the system. However, even if no

[1]For example, it appears that some manufacturers do not even try to protect communications between the measure device and the central unit, thus leaving the system completely open to anyone having access to that line. See e.g. (Calabrese, 1999) for a review of several commercial biometrics-based products and potentially dangerous errors in their protocols.

two persons have exactly the same biometric profile[2], this does not mean that a biometrics-based system is immune to replay attacks. An attacker could simply pass over the biometrics measure phase and directly send a bit string to the system.

A first thing to have in mind is thus that biometrics are like any other key: they must be protected.

3. DIFFERENCES BETWEEN BIOMETRIC DATA AND KEYS

On the other hand, biometrics present several specificities that make them different from standard keys.

3.1. EQUALITY VS. SIMILITUDE

To avoid clear transmission of biometric data (the equivalent of the naive password-based model), (Calabrese, 1999) proposes a solution based on hash functions: store only a secure hash of the biometric data in a database; to perform authentication, hash the data from the biometric reader and compare it with the stored value; if the values match, allow access, otherwise refuse it.

Independently of a number of important security issues[3] that are not too difficult to solve but must be taken into account, we claim this scheme just cannot work.

Biometric data are not perfect constants. Due to imprecision in measures, temperature variations, biological factors, ..., the same data measured from the same person at different times will not give perfectly equal results: the difference will be small, but probably not zero. Biometrics-based authentication takes this into account using something we could define as *loose equality*: two samples are considered equal if their difference is so small that they cannot possibly come from different persons.

This loose equality induces some problems with biometrics. A biometrics system does not give a basic authentication mechanism with a categorical (yes or no) answer, but rather something like: with a confidence level of 99%, I have authenticated the user. More formally, the performance of a biometrics system is commonly given with his False Rejection Rate (FRR) and his False Acceptation Rate (FAR). Those rates indicate the probability of rejecting a correct user (FRR) and the

[2]This may be correct but even in that case, the system can wrongly identify a user, see next section for further details.

[3]An attacker could for example simply replay the hash without any need to know the original value.

probability of accepting a false user (FAR). The latest is important because even without pirates, the system will sometimes wrongly identify somebody.

But the introduction of a hash function completely destroys loose equality: a hash function will put the stress on small differences between similar messages, producing completely different digests. The question "do these two digests correspond to similar messages?" is impossible to answer.

This specificity of biometrics seems to rule out hash functions from the candidate building blocks: to be able to perform biometrics-based authentication, it must be possible to recover, at some time in the protocol, the input and reference values "in clear".

3.2. KEY MANAGEMENT

Another important issue is the key-management problem: if someone discovers that his password has been compromised, all he has to do is to choose a new password. Biometrics does not allow such type of key renewal (Schneier, 1998).

4. PROPOSED MODEL

4.1. THE COMPONENTS

A biometrics-based identification scheme will at least contain the following components:

measure device: this component is in charge of measuring the user's biometric data when he tries to identify himself and submitting this data to the feature extraction unit;

feature extraction unit: This component takes as input the raw information from the measure device and extracts the features from the data. The biometric features are sent to the comparison unit.

comparison unit: this component is in charge of comparing the data provided by the measure device and that stored in a reference database; even with a genuine user, these two values will never be exactly equal, and the unit has to decide if the features measured come from the same person as the reference data stored in the database. An important consequence of this is that the two inputs must be available *in clear* to the device.

reference database: to make comparison possible, a reference value of each user's biometric data has to be stored somewhere ; the

reference database can sometimes be located at the same place as the comparison unit, but not always: in a smart card based environment, for example, it could be better to store each user's data on his own smart card rather than in a centralised, highly sensitive database.

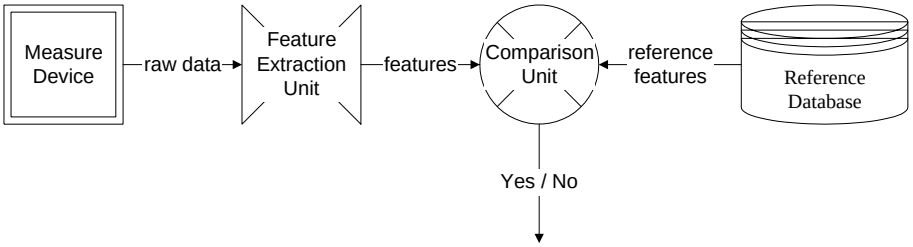

Figure 1 Components of the model

In some environments, some of these components can be integrated. However, we believe them to be, in essence, distinct, as reflects the example in figure 2.

measure device: a sensor that will take 'a picture' of the fingerprint with a CCD or with a capacity detector. The data consists in a bitmap picture (about 100 KB).

feature extraction unit: The whole picture of the fingerprint is not really useful. Only some characteristics are needed. The commonly used characteristics are the minutiae. Minutiae consist in particular points of the fingerprint: the end of ridges or the bifurcations of ridges on the fingerprint. In USA, a minimum of 8 minutiae are required (in Europe, 12 are required). Now the features consisting of the minutiae have a typical size of 500 - 1000 bytes.

comparison unit: it is a match device that will compare the minutiae coming from the feature extraction device and the minutiae coming from the database.

reference database : a smart card holding the minutiae of the user.

Figure 2 Example of a classical fingerprint system (STMicroelectronics, 1999)

We will therefore deal with the general case, in which the components are physically distant and must communicate through insecure channels. Note that 'physically distant' does not mean the distance must be long. On a smart card with integrated fingerprint reader (e.g. planned by Infineon (Infineon, 2000)), for example, the measure device and the processor *are* physically distant, since the reader is located outside the chip and connected to it by wires, that could probably be eavesdropped or reconnected.

4.2. IDENTIFICATION VS. AUTHENTICATION

Classically, a biometrics-based system can be used for two main purposes.

authentication: the system will verify the matching between a user and a stored biometric profile of this user.

identification: the system will find, based only on the biometrics of the user, which user it is.

Those two schemes imply some differences for the comparison unit and the reference database.

In the identification case, the reference database must be a central database where every biometric profile must be stored. In the authentication case, the database can also be distributed among smart cards, one per user.

The comparison unit must be much more efficient in the identification case because the purpose is no more to verify a loose equality but answer two questions: are these biometric data coming from a valid user and which user it is? The side effect is that the FAR and the FRR will both increase and lead to a weakening of the system. In the authentication case with the database distributed among smart cards, the FAR will decrease sharply because two things must happen at the same time: Carol's biometric profile must be close to that of a given user, so that she has some chance to be identified as him, and she must possess the smart card of that specific user.

The link between these two components is also affected. In the identification case, the comparison unit needs to access to a lot of biometrics profiles stored in the database. This high volume of messages can also increase the practicality of an attack.

For the rest of the paper, we will consider the authentication case, while keeping in mind that some attacks can be amplified in an identification case.

4.3. IDENTIFICATION OF WEAKNESSES AND PROTOTYPE OF A MORE SECURE MODEL

In this section we will gradually build our identification model, by listing the possible attacks and taking adequate measures to counter them.

4.3.1 Attacking the measure device. Consider for now that biometrics is safe, in the sense that it is impossible to reproduce somebody's biometric profile and successfully present it to the measure device[4]. Even with this assumption, an attacker can still open the device and reconnect its input wires to a recorder containing a copy of Alice's biometric data. To defeat this attack, the measure device has to be tamper-proof: it must detect intrusion and refuse to work if improperly used. It must also detect misleading submissions (e.g. fingerprint photocopy, dead finger, ...)!

4.3.2 Attacking the link between measure device and feature extraction unit. Now the attacker still has the possibility to simply remove the tamper-resistant device and directly send fake data to the comparison unit. It is therefore necessary for the device to sign the data it sends, to ensure that only genuine, correctly-measured biometric data are presented to the feature extraction unit.

This is still insufficient, as the attacker could simply observe an honest identification procedure and replay the signed data later. To prevent this, (Calabrese, 1999) suggests to encrypt and sign data, we do not believe this to be effective (a replay of the encrypted and signed data is possible). A better solution is to concatenate a time indication or sequence number before having the device sign the message, so as to make each message different. Note that freshness must be checked at reception, in order to prevent an attacker from absorbing the message (i.e. intercept it in such a way that the receiver will not receive it) and using it later, which would not be detected as a replay. This problem of sequenced transmission has been widely studied, and several robust solutions are known (see e.g. (Lamport, 1981; Vandenwauver, 1998)).

It is interesting to note that, with the above assumption, it is not necessary to encrypt the communication: knowledge of Alice's biometric data would not help the attacker, as there is no other input channel than

[4]This assumption may seem reasonable at first sight, but must not be underestimated. In section 5, we will discuss it in more details and show how its removal would impact biometrics.

through the secure device. One must however take this assumption with great care. See section 5 for further details.

4.3.3 Attacking the feature extraction unit. Besides checking its input's integrity, it is also necessary for the feature extraction unit to be tamper-proof, to make sure its internal computations cannot be tampered with by an attacker.

4.3.4 Attacking the link between feature extraction unit and comparison unit. The same attacks as in section 4.3.2 can be applied against this link. It is therefore essential to implement the same protection means.

4.3.5 Attacking the comparison unit. The comparison unit must of course be tamper-resistant: it is of no use to carefully design an identification protocol if an attacker can simply break into the system and modify the program so that it always grants access, independently of the input. This is true for every protocol.

Another, more subtle way of attacking the comparison unit seems specific to biometrics. We have seen that biometric data measured at different times will not be exactly equal, so that comparison between measure device's output and reference database must be based on "loose equality", which means, in short, that two inputs will be considered equal if they are "close enough".

The way loose equality is defined must be carefully studied, at least in two directions: first, of course, to prevent different persons having "close" characteristics to be considered as equal (this is clearly the scope of biometrics itself). Second, its impacts on cryptographic models must not be neglected: the signature forgery problem (see section 4.3.2) is no more "forge a signature for message m", but rather "forge a signature for a message $\widehat{m}$ that is *close to* m, in the sense of loose equality". This problem is significantly different from the classical signature forgery problem and must be taken into account when choosing the signature scheme. Note that this consideration is also valid for the link between the measure device and the feature extraction unit.

A simple solution could be to hash the message before signing it. As section 3.1 showed that hashing destroys loose equality, the message would have to be transmitted separately (i.e. no message recovery). This scheme could work, but the absence of message recovery would imply an increase in the bandwidth, which can be problematic, e.g. if a smart card is involved.

Another case in which loose equality can have impact is when encryption must be performed – we said it was not necessary in this model, but as we will see (section 5.1), this is no more true if we adopt more general hypotheses.

As far as we know, this problem has not been extensively studied in the signature case. However, some attacks have been found in the encryption case (encrypting quite similar messages). Examples can be found in (Misarsky, 1999; Coppersmith, 1996; Coppersmith, 1997; Jutla, 1998; Patarin, 1995). We will not extend any further on this important issue, as it is closely related to the specificities of the chosen biometric scheme and therefore out of the scope of this paper.

4.3.6 Attacking the database. Another obvious possible target for the attacker is the reference database: if Carol is able to edit Alice's entry in the database and replace the biometric data by her own, then she will easily gain access to the system. The adequate countermeasure is simply to protect the database against writing.

In the same way that it was not necessary to encrypt communications between measure device and comparison unit, there is no real need to read-protect the database either: knowledge of Alice's characteristics will not help Carol.

4.3.7 Attacking the link between database and comparison unit. Once again, it is of no use to protect the database if we do not ensure that the data submitted to the comparison unit actually comes from the database. It is therefore necessary for the database to sign data sent to the comparison unit.

Another possible threat is the possibility for some valid user to impersonate someone else (presumably with more power). This could be done by a replay attack conducted as follows:

1 Carol presents herself to the measure device, pretending to be Alice (e.g. with Alice's smart card); her biometric data are measured and sent to the comparison unit;

2 the comparison unit asks the database for Alice's reference data;

3 Carol intercepts the database's answer and sends a new answer with her own database entry, that she eavesdropped from the database in a previous genuine authentication; as this data actually comes from the database, it is correctly signed;

4 the comparison unit compares the two data, and grants Carol Alice's permissions.

To prevent this type of impersonation, it is needed to link the biometric data stored in the database to some information about the person's identity[5]. When the database is questioned, it answers not only the biometric data, but also the information related to the corresponding person.

This is immediate to implement in a non-anonymous system: it suffices to use directly the identity; in the anonymous case, it is necessary to define groups, or privilege levels, reflecting the various permissions. Access is granted only if the comparison matches *and* the person's privilege correspond to the requested access.

4.4. SUMMARY

In the next few figures, a summary of our proposed model is presented. The signature algorithm must be chosen carefully especially if a message recovery algorithm is going to be considered.

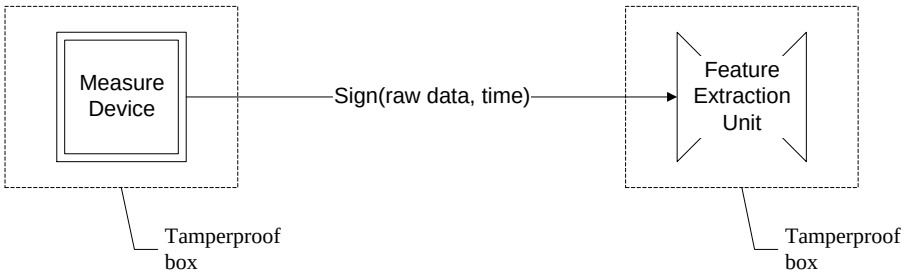

Figure 3 Secure link between the measure device and the feature extraction unit

5. A MORE PARANOID MODEL

Above we have assumed that biometric data could not be reproduced: even with knowledge of the precise characteristics of somebody's iris, it is still impossible to reproduce it in another person's eye. This is probably true at present, but what about the future?

Such fraud does not necessarily involve organ transplant, as the verifier we are trying to cheat is only a machine, checking very specific

[5]This remark may appear obvious at first sight: one may indeed think it is for example unavoidable, just to index the database. We would thus like to insist on the fact that this is in no way mandatory in a general authentication scheme. Anonymous authentication can for example be useful to protect a smart card against theft, by allowing it to check its user's legitimacy before accepting to work.

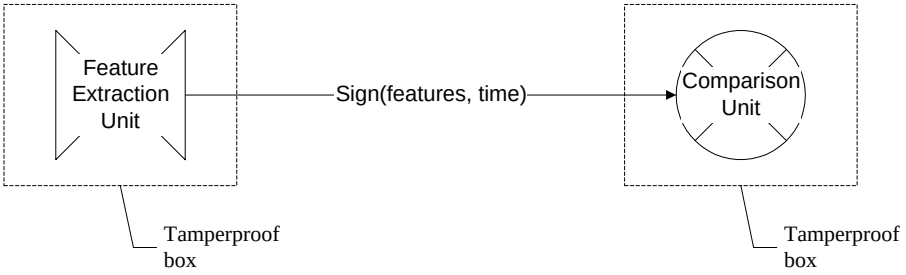

Figure 4 Secure link between the feature extraction unit and the comparison unit

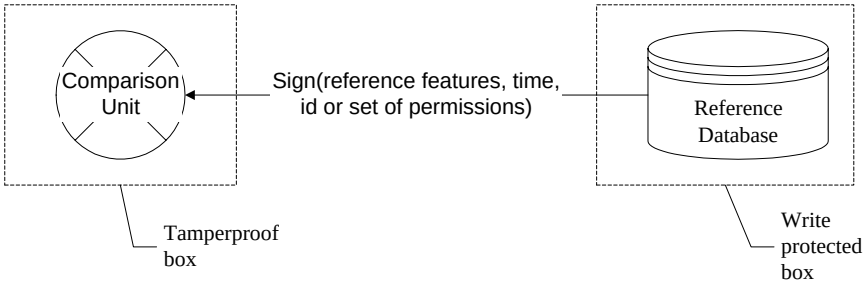

Figure 5 Secure link between the comparison unit and the reference database

features and completely ignoring the rest of them. In other words, the question is: "with a good knowledge of what the measure device is looking at, is it really impossible to build a 'fake organ' that will behave like a genuine one to the measure device and present the desired biometric signature ?"

If we assume not, then the model becomes significantly more complicated, and may even lead to complete rejection of biometrics.

5.1. STRENGTHENING THE PROTOCOL

Basically, what we have to do is keep biometric data secret, just like passwords. However, this is much more difficult to enforce with biometrics.

First of all, the reference database must be read-protected. In the same way, communications between measure device, comparison unit and database must now be encrypted. These conditions do not seem too difficult to enforce.

Last but not least, it is no more sufficient to ensure that no fake or corrupted measure device can communicate with the comparison unit: we must also take care of fake measure devices *not connected to the system*. The attack we have in mind is very similar to the false ATM attack: Carol builds a fake measure device and hopes Alice will try to use it. When she does, the device simply records her biometric data before outputting a message explaining a system malfunction. This is a very simple way of obtaining Alice's biometric profile.

The only way to prevent this type of attack is to have the measure device authenticate itself to Alice before she presents her biometric data. This is of course possible, but implies Alice must dispose of some computing device, powerful enough to perform cryptographic functions and that she completely trusts. A simple example of this is a smart card.

This yields an interesting and, to our knowledge, quite new model of joint use of biometrics and smart card to implement security, in which the smart card role is twofold: not only does it perform cryptographic transactions after having checked its user's validity (this is the commonly accepted joint use of smart card and biometrics), but also – and first – does it make the user confident about the protocol. Basically, this could work as follows:

1 the smart card identifies itself to the terminal, proving it is a valid card, but with no guarantee about its possessor;

2 the terminal identifies itself to the smart card, proving it is valid and has not been tampered with;

3 once the two devices have performed self authentication, Alice provides the terminal with her biometric data, proving she is the valid user of the smart card;

4 access is granted.

Such protocol implies that Alice must dispose, not only of a smart card, but also of a trusted user interface, as she of course cannot trust the terminal in reporting it has been correctly identified by the smart card. What we have in mind is a small device that Alice carries with her (so that she can trust it as much as the smart card) and that disposes of some graphical interface (which, in our case, can be as simple as a LED). Rather than inserting her card in a terminal, Alice puts it in her "user interface" and connects this device to the terminal. In this way, the smart card has some mean to communicate directly with Alice, to inform her that "things are going well".

This is extra investment, but it begins to be considered very seriously as an unavoidable component of smart card-based security. It will be

used in the implementation of SET in France (Megglé, 2000) and is the purpose of the European sponsored project FINREAD (FINREAD, 1999).

An interesting property of this protocol is that the loss or stealing of Alice's smart card would no more be damageable. On the other hand, corruption or replacement would really become an issue, as it could cause Alice to erroneously trust a pirated device and provide it with her biometric data. The pirate's work would however be more difficult, since he would have to simultaneously attack both Alice's smart card *and* the next terminal she is going to use, for otherwise the intrusion would be detected.

5.2. OUTSIDE THREATS

The above measures ensure that the identification application cannot be corrupted to lead to data disclosure. However, this is not sufficient yet, as we have no guarantee these data cannot be recovered *externally* to the application. If another society with a weaker security policy uses the same identification method, then their protocol could be corrupted and allow to break into the first one. This is similar to passwords too: in the same way you should not use the same password for two different purposes, you should also avoid using the same biometric identification type twice.

But even then, problems are not solved: how do I know my optician is not making a copy of my iris while observing my eyes? Besides that, as I am typing this report, I am leaving many information about my fingerprints on the keyboard. If we really want to be paranoid, these problems seem to become unsolvable.

This is intrinsically related to another specificity of biometric data: as opposed to passwords, the user does not even always know when he transmits / reveals them[6].

The smart card is considered as a secure device in this paper. However, if it is not the case, this can also be considered as an outside threat.

[6]This is not true for every biometrics: for example, biometrics related to features (writing speed, pencil pressure, ...) of a handwritten signature can probably not be observed without the person being aware that he is performing a signature! However, some other "active" biometrics, such as voice with pass phrase, are less obvious: think for example about the movie "Sneakers", in which a long conversation is recorded, during which the attacker tries to have her interlocutor say every word of the pass phrase; these words are then cut-pasted in right order. We will not discuss the realism of such attack here: we just want to illustrate that, as always, great care has to be taken to examine the validity of such assumption.

6. OTHER POSSIBLE SOLUTIONS

Recently Yair Frankel proposed (Frankel, 2000) a method to circumvent the loose equality problem of biometrics. He proposes to store an error correcting code in a database. The system will take as input the biometrics features, use the error correcting code and produce, almost each time, exactly the same error-corrected biometric data. Then, it can be used as a password or a key. We will not describe in details this method; see (Frankel, 2000) for those details.

This solution seems promising at a first stage but some questions remains open mainly when the biometrics data are used as a key:

- The biometric features are not randomly distributed in the key space. As some subparts of the biometrics characteristics are similar for some users, the subparts of the keys will be also quite similar. Is that weakening the cryptographic scheme?
- In his presentation, the author stressed that there is no information leakage in the error correcting code. We are not as confident as him in this statement because, by essence, error-correcting codes have been designed with the opposite goal (recover information rather than hide it). This question at least deserves an extended examination.

Several authors (Alexandre, 1997; Jain et al., 1998) propose some biometrics allowing the use of standard PC devices (such as scanner, digital camera or typing frequency on the keyboard). It can be tempting to use them as biometrics devices for many purposes: cost reduction, widespread existence of the devices, easy to use, ... As we have shown, this approach is wrong. For a biometrics-based protocol to be secure, it is essential to use specialised hardware, with strong self-identifying properties[7].

7. CONCLUDING REMARKS

As we have shown here, biometrics can be useful to add features to security protocols but they must be used carefully. Because they have specific properties, they must be treated as a specific part of the protocol and not as replacement of another part such as keys or passwords.

Taking into account these particularities, we have built a skeleton of a secure protocol based on biometrics with the help of smart cards.

[7]Note that this does not mean it must be dedicated hardware: a secure, tamper-proof and self-identifying scanner aimed at checking hand geometry could probably also be used to scan regular documents, without interfering with security.

8. ACKNOWLEDGEMENTS

The authors wish to thank the anonymous referees for their useful comments, especially Mr. X for pointing out a minor flaw in a first version of the protocol.

They are also grateful to Jean-Marc Boucqueau for the fruitful discussions that gave birth to this paper.

References

Alexandre, T. J. (1997). Biometrics on smart cards: An approach to keyboard behavioral signature. *Future Generations Computer Systems*, 13(1):19–26.

Calabrese, C. (1999). The trouble with biometrics. *;login:*, 24(4):56–61.

Coppersmith, D. (1996). Finding a small root of a univariate modular equation. In Maurer, U., editor, *Advances in Cryptology - EUROCRYPT'96*, volume 1070 of *LNCS*, pages 155–165. Springer-Verlag.

Coppersmith, D. (1997). Small solutions to polynomial equations, and low exponent RSA vulnerabilities. *Journal of Cryptology*, 10(4):233–260.

FINREAD (1999). Financial transactional IC card reader project. http://www.ispo.cec.be/isis/98finrea.htm.

Frankel, Y. (2000). Biometric identification and authentication with privacy preservation. In *RSA Conference 2000*.

Infineon (2000). FingerTIP. http://www.infineon.com/products/chipcds/portfol/biometr/faq.htm.

Jain, A. K., Ross, A., and Prabhakar, S. (1998). Biometrics-based web access. Technical Report TR98-33, Michigan State University.

Jutla, C. S. (1998). On finding small solutions of modular multivariate polynomial equations. In Nyberg, K., editor, *Advances in Cryptology - EUROCRYPT'98*, volume 1403 of *LNCS*, pages 158–170. Springer-Verlag.

Lamport, L. (1981). Password authentication with insecure communication. *Communications of the ACM*, 24(11):770–772.

Megglé, C. (2000). Personal communication.

Misarsky, J.-F. (1999). *Cryptanalyse et spécification de schémas de signature RSA avec redondance*. PhD thesis, Université de Caen.

Patarin, J. (1995). Some serious protocol failures for RSA with exponent e of less than 32 bits. In *Luminy Workshop on Cryptography*. http://www.cp8.bull.net/sct/uk/partners/bull/page/c_publication.html.

Schneier, B. (1998). Biometrics: Truths and fictions. Crypto-Gram, August 15. http://www.counterpane.com.

STMicroelectronics (1999). Biometric subsystem in smartcard environment. Exhibitor Workshop in CARTES'99.

Vandenwauver, M. (1998). *Practical Network Security Aspects*. PhD thesis, Katholieke Universiteit Leuven.