Security protocols for biometrics-based cardholder authentication in smartcards

Luciano Rila and Chris J. Mitchell

Information Security Group
Royal Holloway, University of London
Surrey, TW20 0EX, UK
luciano.rila@rhul.ac.uk and c.mitchell@rhul.ac.uk

Abstract. The use of biometrics, and fingerprint recognition in particular, for cardholder authentication in smartcard systems is growing in popularity, and such systems are the focus of this paper. In such a biometrics-based cardholder authentication system, sensitive data will typically need to be transferred between the smartcard and the card reader. We propose strategies to ensure integrity of the sensitive data exchanged between the smartcard and the card reader during authentication of the cardholder to the card, and also to provide mutual authentication between card and reader. We examine two possible types of attacks: replay attacks and active attacks in which an attacker is able to calculate hashes and modify messages accordingly.

Keywords. smartcards, biometrics, cardholder authentication.

1 Introduction

Traditional methods for automated personal identification mainly employ the possession of a token (magnetic card, USB-token) and/or the knowledge of a secret (password, PIN) to establish the identity of an individual. However a token can be lost, stolen, misplaced, or willingly given to an unauthorised person, and a secret can be forgotten, guessed, or unwillingly — or willingly — disclosed to an unauthorised person. Biometrics has emerged as a powerful tool for automated identification systems. Since it is based on physiological and behavioural characteristics of the individual, biometrics does not suffer from the disadvantages of the traditional methods.

In parallel, smartcards have steadily grown in popularity, as have their storage capacity and processing capabilities. The computing power of modern smartcards allows a wide variety of applications, from support for PKI to decentralised applications requiring off-line transactions [4, 14, 15]. Smartcards also offer the possibility of executing multiple applications on a single card. To implement controlled access to the functionalities of the card, smartcard systems typically require a method for cardholder authentication. Not only does cardholder authentication address the issue of card theft or misappropriation but it also allows the system to grant different access rights to different users of the same card.

An example of the latter can be drawn from health care applications where the patient and the physician access the same card belonging to the patient.

Biometrics and smartcards have the potential to be a very useful combination of technologies. On the one hand the security and convenience of biometrics allow for the implementation of high-security applications on smartcards. On the other hand smartcards represent a secure and portable way of storing biometric templates, which would otherwise need to be stored in a central database. Among the various biometrics technologies in use today, fingerprint recognition seems to be particularly suitable for smartcard systems.

A smartcard system is composed of two main physical units: the smartcard itself and the card reader. Depending on how the logical modules of the biometric system are distributed between the card and the card reader, biometrics-based cardholder authentication may require the transmission of sensitive data, such as a biometric live sample or a biometric template, between the two units. It is therefore fundamental to ensure the integrity of transmitted data during cardholder authentication. It is also important to provide mutual authentication between the two units, so as to prevent use of fraudulent cards or card readers.

In [3], weaknesses of the biometric system model are identified and some countermeasures are suggested, although no particular security protocol is proposed. Moreover, no assumptions as to the actual architecture of the system is made and the analysis is rather general in nature. In this paper, we analyse a specific system architecture that reflects the current state of the art for smartcard systems. Our focus is on security issues associated with the communications link between the smartcard and the card reader during fingerprint-based cardholder authentication. We propose strategies to ensure integrity of the data exchanged between the smartcard and the card reader, and also to provide mutual authentication between the two components. We examine two possible attacks: replay attacks, and active attacks in which an attacker is able to make minor modifications to the messages.

For the purposes of this analysis we do not make any assumptions about encryption or other cryptographic protection of the card/card reader communications link. Note that, in many cases, the requirements for high speed data transfer across this link, combined with the limited computational capabilities of the card, may severely limit the possibilities for such protection. We also assume throughout that fingerprint recognition is used as a method of cardholder authentication to the smartcard.

Given our focus on card/reader communications link, we make other simplifying assumptions. We assume that the smartcard is a tamper-proof device and any transmission between biometric system modules taking place within the card is therefore secure. We do not discuss the impact of using fake biometrics, such as plastic fingers, to fool the system, although it was shown in [17] that this is a possible attack with the current technology. This is an issue relating to the fingerprint-based biometric technology itself, and as such is outside the scope of this paper.

This paper is organised as follows. In Section 2 we define the system architecture analysed in this paper. In Section 3 we identify the possible threats to the communications link and propose two security protocols to prevent replay attacks. In Section 4, we discuss active attacks and propose security protocols to prevent them. We also propose a security protocol for mutual authentication of the smartcard and the card reader and analyse some methods for sharing of a secret key between the card and the reader. Finally we present our conclusions in Section 5.

2 Biometric System Architecture

2.1 General model for biometric authentication

According to [5], a general biometric system is composed of the following logical modules:

1. Data collection subsystem;
2. Signal processing subsystem;
3. Matching subsystem;
4. Storage subsystem;
5. Decision subsystem;
6. Transmission subsystem.

A block diagram for the general authentication model is given in Figure 1. The authentication process involves the raw biometric data of the claimant being captured in the data collection subsystem by an input device or sensor, and then transferred to the signal processing subsystem where the feature extraction takes place. The matching subsystem receives the extracted features from the signal processing subsystem and retrieves the biometric reference template associated with the claimed identity from the storage subsystem. The matching subsystem then compares the submitted biometric sample with the reference template yielding a score, which is a numeric value indicating the degree of similarity between the submitted sample and the reference template. The decision subsystem receives the score and, according to a confidence value based on security risks and risk policy, decides whether to accept the claimant or not. The authentication decision is finally passed on to the application.

Note that these are logical modules, and therefore some systems may integrate several of these components into one physical unit.

2.2 Biometric system architecture in a smartcard system

The architecture of the biometric system within a smartcard system, i.e. how the logical modules of the biometric system are distributed between the smartcard and the card reader, determines the nature of the data to be transferred between the card and the card reader during biometric-based cardholder authentication. Hence different architectures are open to different threats. If all modules of the

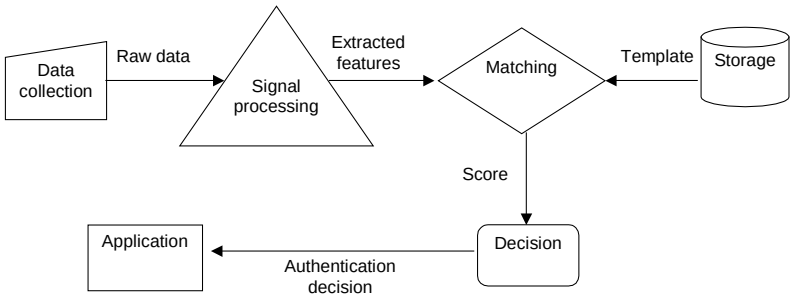

Fig. 1. General model for biometric authentication.

biometric system reside in the card, the whole cardholder authentication process takes place on the card and therefore no data transfer is needed. Provided that the card is a secure physical entity, this is the most secure architecture possible. However it is also the most costly and, with current technology, it is still a challenge to manufacture an ISO-compliant card containing all the modules of the biometric system.

Current biometrics-based smartcard systems must therefore distribute the modules of the biometric scheme between the card and the card reader in some way. In general, the signal processing module is very likely to be located in the card reader, since it requires both significant computational power and, most significantly, large amounts of RAM, and these requirements are likely to be beyond the capabilities of current smartcards. The location of the other modules, however, varies according to the system design and the biometric technology being used.

For fingerprint recognition, a recent prototype implemented in the Finger_Card project [18, 13] has incorporated all but the signal processing module of the biometric system into the smartcard. Within such an architecture, the biometrics-based cardholder authentication is carried out as follows. The fingerprint image is collected by a sensor on the smartcard and transferred from card to reader. The reader performs feature extraction only, and transfers the extracted features back to the card. The card then performs the matching process and reaches the

authentication decision. This architecture is represented in Figure 2 and is the basis of our analysis throughout the remainder of this paper.

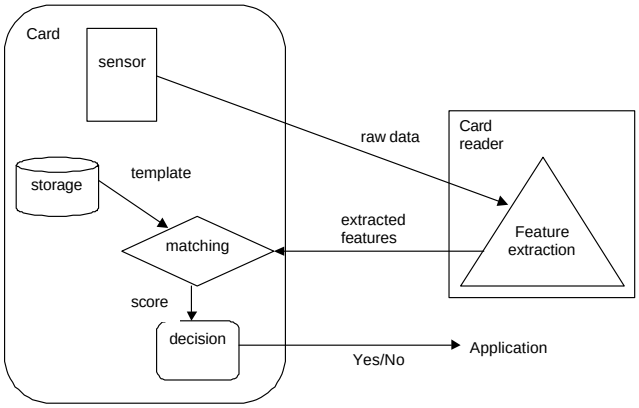

Fig. 2. The architecture of the biometrics-based authentication system within a smartcard system.

3 Replay Attacks

In this paper we focus on possible attacks to the biometrics-based cardholder authentication process. For the purpose of this analysis, we assume that the smartcard is a tamper-proof device, and hence trust the security of any transmissions between those modules of the biometric system within the card. We therefore assume that the only communications link open to attacks is that between the smartcard and the card reader. The main threats to this link can be divided into threats to the 'up-link' (i.e. smartcard to reader) and 'down-link' (i.e. reader to smartcard).

In this section we assume that a possible attacker can only listen to the communications channel, record messages, and/or subvert the protocol by replaying old messages, but the attacker is unable to modify messages or create new messages. This is known as a replay attack [14]. Challenge-response protocols [9–11] and zero-knowledge based protocols [12] may be used to counteract replay attacks.

3.1 Possible threats

Suppose that an attacker is able to insert a bug in the communications link so that he can record the messages being transferred, and he can also intercept

messages and replace them with old recorded messages. According to [16], the main threats to the communications link are as follows:

1. Up-link threats: an attacker can record the fingerprint image sent from card to reader while the legitimate cardholder is authenticating himself to the card. Once the attacker is in possession of a legitimate fingerprint image, he can steal the card, insert the card into the card reader, intercept whatever fingerprint image (probably his own) is being transferred to the card reader, and replace it with the legitimate fingerprint image. The card reader would then extract the features from a legitimate fingerprint image, and send them to the card which would most probably authenticate the user (an attacker) to the card.
2. Down-link threats: another possibility would be to record the (genuine) extracted features sent via the down-link, when the legitimate cardholder is authenticating himself to the card. As before, the attacker can steal the card once in possession of a set of extracted features for the legitimate cardholder. When required to authenticate himself to the card, he can use his own fingerprint to enable the protocol to proceed. The card reader will extract the features from his illegitimate fingerprint image, which the attacker then intercepts and replaces with the legitimate set of features when they are sent across the down-link so that the attacker is authenticated to the card.

Note again that we are assuming here that the attacker can neither modify the messages nor create them, only replay them. Note also that both these attacks require the attacker to monitor card/reader communications when the card is in use by its legitimate holder, prior to stealing or otherwise misappropriating the card.

In the next two sections we propose two different security protocols designed to address the threats we have just described.

3.2 Security protocol using a random string generator on the card

In this protocol, the freshness of the data is ensured by the use of random strings and hash functions. The following protocol prevents both replay attacks.

1. The reader generates a random string R_1 and sends it to the card.
2. The card (in fact the fingerprint sensor on the card) captures the fingerprint image ($BioData$), generates a random string R_2, and sends to the reader: $R_2 \mid\mid BioData \mid\mid h(R_2 \mid\mid R_1 \mid\mid BioData)$. Note that h denotes a cryptographic hash-function (see, for example, [14, 6]), and $\mid\mid$ denotes concatenation of data items.
3. The reader computes $h(R_2 \mid\mid R_1 \mid\mid BioData)$ and verifies that it is identical to the value sent by the card.
4. The reader extracts the features (EF) from the fingerprint image and sends to the card: $EF \mid\mid h(R_1 \mid\mid R_2 \mid\mid EF)$.
5. The card computes $h(R_1 \mid\mid R_2 \mid\mid EF)$ and verifies that it is identical to the value sent by the reader.

Since the reader generates a new random string R_1 for each transaction, if the value $h(R_2 \mid\mid R_1 \mid\mid BioData)$ received by the reader is correct, then the message from the card is not a replay of an old message. Therefore steps (1), (2), and (3) prevent the up-link threat.

The down-link threat is prevented in steps (4) and (5). Since the card generates a new random string R_2 for each transaction, if the value $h(R_1 \mid\mid R_2 \mid\mid EF)$ received by the card verifies correctly, then the message from the reader to the card is not a replay of an old message.

Note that this protocol also partially ensures the integrity of the data being transferred in both directions even though integrity is not an issue when replay attacks are considered. In the up-link, any change in $BioData$ during transmission would result in an incorrect hash-code $h(R_2 \mid\mid R_1 \mid\mid BioData)$ in step 3. In the down-link, any change in EF during transmission would result in an incorrect hash-code $h(R_1 \mid\mid R_2 \mid\mid EF)$ in step 5. Of course, a sophisticated attacker could change the data and also recompute the hash-value; however, such an attack requires a level of sophistication beyond the scope of the countermeasures considered in this section.

3.3 Security protocol using biometric data as a random string

The nature of biometric data is such that two different measurements of the same biometric feature from the same person are very likely to be different, although the difference may be small. For example, when fingerprint recognition is used, the fingerprint image captured by the sensor may vary, e.g., due to skin conditions, dirt, grease, or the position in which the finger is placed on the sensor. The biometric data can therefore be regarded as a unique random string for each transaction, and can as such be incorporated by the card into the security protocol. Although this approach may restrict the possible values for the random string, it may have practical advantages if generating a random string is a demanding task for the smartcard.

The protocol presented in the previous section would then be modified as follows:

1. The reader generates a random string R and sends it to the card.
2. The card captures the fingerprint image ($BioData$) and sends to the reader: $BioData \mid\mid h(BioData \mid\mid R)$.
3. The reader computes $h(BioData \mid\mid R)$ and verifies that it is identical to the value sent by the card.
4. The reader extracts the features (EF) from the fingerprint image and sends to the card: $EF \mid\mid h(EF \mid\mid h(BioData \mid\mid R))$.
5. The card computes $h(EF \mid\mid h(BioData \mid\mid R))$ and verifies that it is identical to the value sent by the reader.

As in the previous section, a new random string R is generated by the reader for each transaction, and the use of the hash-code $h(BioData \mid\mid R)$ prevents replay attacks in the up-link — steps (1), (2), and (3).

The down-link threat is prevented in steps (4) and (5). Under the assumption that the biometric data is very likely to be different at every collection, $BioData$ plays the role of a random string generated by the card. When the reader returns to the card the value $h(EF \mid\mid h(BioData \mid\mid R))$ in step (4), the verification of the hash-code in step (5) ensures that this is not a replay of an old message.

Note again that this protocol also partially ensures the integrity of the data being transferred in both directions.

4 Active Attacks

Suppose now that the attacker can modify messages and also knows the details of the protocol being used (including the hash-function). Since we have assumed that all data (including the random strings) are transferred in plaintext, the protocols of Section 3 can be defeated, since the attacker could intercept either the fingerprint image or the extracted features and simply generate the hashes as the protocol goes along. The integrity of the data being transferred between the card and the card reader becomes an issue in this case. In order to prevent such active attacks, we describe a slightly different protocol.

4.1 Message Authentication Codes (MACs) and Mutual Authentication

Suppose the card and the card reader share a secret key. This allows both parties to use a Message Authentication Code (MAC) in the place of the hash-code in the protocols of Section 3. A MAC function, as specified in [14, 7, 8], is a key-dependent cryptographic function with the property that it can map arbitrarily long messages to fixed length strings. Moreover, only someone who knows the key can produce a valid MAC for a data string. MACs are validated by re-computing them using the shared secret key.

Hence, given that the secret key shared by card and card reader is not known to the attacker, only a legitimate card and card reader can generate and verify valid MACs. The protocols described in Section 3 remain the same except that MACs are used instead of one-way hash-functions, and the protocols now provide mutual authentication between the card and the card reader (see, for example, [14, 11]).

4.2 Mutual authentication protocol prior to transaction

The sharing of a secret key would also make it possible for the card and the card reader to run a mutual authentication protocol before any sensitive data is transferred. In this way the card would be assured that it has not been inserted into a hostile card reader. Conversely, the card reader can verify that the card inserted is a legitimate one. A possible protocol for this scenario is as follows. This protocol actually conforms to the relevant ISO/IEC standard for authentication protocols, [9], and is closely related to the SKID3 protocol, [1]. Note that in

this protocol m denotes a MAC function; specifically, $m_K(X)$ denotes the MAC computed on data string X using key K.

1. The reader generates a random string R_1 and sends it to the card.
2. The card generates a random string R_2 and sends to the reader the two values: $R_2 \,||\, m_K(R_2 \,||\, R_1)$.
3. The reader computes $m_K(R_2 \,||\, R_1)$ and verifies that it is identical to the value sent by the card.
4. The reader sends to the card: $m_K(R_1 \,||\, R_2)$.
5. The card computes $m_K(R_1 \,||\, R_2)$ and verifies that it is identical to the value sent by the reader.

One possible reason for taking this approach — rather than the one in Section 4.1 — might be that computing a MAC on the fingerprint data is not feasible. This is because, whereas PINs are very short, fingerprint samples are rather large, and the limited computational and storage capabilities of the card may severely limit the possibilities of a MAC computation on the fingerprint data. Hence conducting the authentication protocol before the exchange of sensitive data would be a reasonable alternative. Note however that this approach does not prevent active attacks such as those described at the beginning of Section 4.

4.3 Sharing a secret key

Methods by which the card and the card reader can share a secret key are now considered. One possible way to accomplish this is to have a single key shared by all cards and card readers in the application. The fact that a universal key is shared by all parties in the application domain is a potential source of weakness, but the use of a secret key nevertheless makes things more difficult for an attacker. Measures could also be adopted to improve the overall security of such a system.

One such measure is to have expiry dates for keys. Keys would expire periodically and new keys would be generated — together with a new card possibly. Card readers would have to be equipped with copies of all the current keys. If a key is compromised, this measure would at least limit the period of time in which attacks could be carried out. A second approach would have all parties in the application sharing a list of keys (K_1, K_2, ..., K_n say) rather than a single key. At the beginning of the protocol, in step 2, the card chooses at random a key K_i ($1 < i < n$) from the list, and sends to the card reader:

$$R_2 \,||\, \mathit{Biodata} \,||\, i \,||\, m_{K_i}(R_2 \,||\, R_1 \,||\, \mathit{Biodata}).$$

The card reader uses the key index i to determine which key is being used. The protocol then proceeds with all MACs generated using K_i. In this case, even if one key is compromised (e.g. by cryptanalytic means), the attacker would have to make many attempts until the compromised key is used again. However, after a few failed attempts to use the card, the card would be blocked, making it unlikely that this attack would work. In order to further improve security, cards could be given different subsets of the same set of keys, although all readers would have all possible keys.

A third approach avoids the weakness of using a universal key by giving a unique key to each card. Each card reader would then need to have all such keys or, at least, on-line access to a centralised database of keys. The need for a key database could be avoided by deriving all card keys from a single secret master key. Specifically it would be possible to derive a card secret key from a combination of a secret master key with a card serial number (e.g. using a one-way function such as a cryptographic hash-function exhibiting pseudorandomness properties). Such a scheme will work as long as the card reader has access to the master key. Card key derivation from a card issuer secret master key is a solution described in Annex A1.4 of EMV Book 2 [2], an industry standard governing interactions between a card and terminal, and used for smartcard-based debit and credit cards.

5 Conclusions

All the threats examined in this paper arise from an assumed lack of integrity for the communications link between card and card reader. If it is assumed that the card reader is a trusted device and has not been interfered with or replaced, then guaranteeing the integrity of the link between the card and the card reader would effectively prevent all the threats, even in the absence of any confidentiality for data transferred.

If the integrity of the communications link is untrusted, then security protocols are able to prevent some attacks. Replay attacks can be prevented using random numbers and hash-functions to ensure the freshness of the data. This countermeasure requires the card to be capable of computing hash-functions and/or generating random numbers. Alternatively, by its very randomic nature, the biometric data may play the role of the random string generated on the smartcard. Active attacks can be prevented using similar protocols, using MACs instead of hash-functions. MACs can also be used to provide mutual authentication between the card and the reader. If the cryptographic capability of the card is limited, then a mutual authentication protocol can be conducted before any sensitive data is exchanged, instead of protecting the actual data transfer. In any case, the use of MACs requires that the card and the reader share a secret key, and possible means by which such keys could be managed have also been discussed.

6 Acknowledgments

The work described in this paper has been supported by the European Commission through the IST Programme under Contract IST-2000-25168 (Finger_Card). The information in this document is provided as is, and no guarantee or warranty is given or implied that the information is fit for any particular purpose. The user thereof uses the information at its sole risk and liability.

The authors would like to thank their colleagues in the Finger_Card Project for their encouragement and advice.

References

1. A. Bosselaers and B. Preneel (editors), *Integrity primitives for secure information systems: final report of RACE integrity primitives evaluation RIPE-RACE 1040*, LNCS 1007, Springer-Verlag, 1995.
2. EMV 2000, Integrated circuit card specification for payment systems, Book 2 — Security and key management, version 4.0, 2000.
3. G. Hachez, F. Koeune, and J.-J. Quisquater, "Biometrics, access control, smart cards: a not so simple combination", in *Proc. 4th Smart Card Research and Advanced Applications Conference (CARDIS 2000)*, J. Domingo-Ferrer, D. Chan, and A. Watson (editors), pp. 273-288, Kluwer Academic Publishers, Bristol, UK, Sep. 2000.
4. M. Hendry, *Smart Card Security and Applications*. Artech House, 1997.
5. ISO/DIS 21352: 2001, Biometric information management and security, ISO/IEC JTC 1/SC 27 N2949.
6. ISO/IEC 10118-3: 1998, Information technology — Security techniques — Hash-functions — Part 3: Dedicated hash-functions.
7. ISO/IEC 9797-1: 1999, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 1: Mechanisms using a block cipher.
8. ISO/IEC 9797-2: 2002, Information technology — Security techniques — Message Authentication Codes (MACs) — Part 2: Mechanisms using a dedicated hash-function.
9. ISO/IEC 9798-2: 1999, Information technology — Security techniques — Entity authentication — Part 2: Mechanisms using symmetric encipherment algorithms.
10. ISO/IEC 9798-3: 1998, Information technology — Security techniques — Entity authentication — Part 3: Mechanisms using digital signature techniques.
11. ISO/IEC 9798-4: 1999, Information technology — Security techniques — Entity authentication — Part 4: Mechanisms using a cryptographic check function.
12. ISO/IEC 9798-5: 1999, Information technology — Security techniques — Entity authentication — Part 5: Mechanisms using zero knowledge techniques.
13. M. Janke, "Bio-System-On-Card", in *SecureCard 2001*, Jun. 2001, Hamburg, Germany.
14. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. CRC Press, 1997.
15. W. Rankl and W. Effing, *Smart Card Handbook*. John Wiley & Sons, 2001.
16. L. Rila and C. J. Mitchell, "Security analysis of smartcard to card reader communications for biometric cardholder authentication", in *Proc. 5th Smart Card Research and Advanced Application Conference (CARDIS '02)*, pp. 19-28, USENIX Association, San Jose, California, Nov. 2002.
17. T. van der Putte and J. Keuning, "Biometrical fingerprint recognition: don't get your fingers burned", in *Proc. 4th Smart Card Research and Advanced Applications Conference (CARDIS 2000)*, J. Domingo-Ferrer, D. Chan, and A. Watson (editors), pp. 273-288, Kluwer Academic Publishers, Bristol, UK, Sep. 2000.
18. B. Wirtz, "Biometric System On Card", in *Information Security Solutions Europe 2001*, Sept. 2001, London, UK.