A PROTECTION SCHEME FOR MOC-ENABLED SMART CARDS

Claude Barral[1,2] *and Serge Vaudenay*[2]

[1]GEMALTO,
Technology & Innovation
La Ciotat, France
claude.barral@gemalto.com

[2]EPFL Lausanne, Switzerland (www.epfl.ch)
{claude.barral,serge.vaudenay}@epfl.ch

ABSTRACT

The concept of *Match-on-Card* (MoC) consists of a smart card which receives an applicant's candidate template T to be compared with the stored reference template T_{ref} by processing the complete matching algorithm during a biometric authentication request. The smart card will then output whether this comparison is positive or not. The main argument against MoC-enabled smart cards is that it opens the way for *YesCard* (i.e. an attack path previously seen in Banking, a card always returning "yes"). The threat regarding Biometrics is not only YesCard, but also *NoCard* as we will see in this paper. We will propose a protocol to easily thwart these attacks by using simple cryptographic primitives such as symmetric encryption. This protocol will however only protect the system from malicious smart cards, but will not protect the smart card against malicious systems. Finally we will enhance this protocol to protect the smart card against its use as a so-called *oracle* to guess the stored reference biometric template.

1. INTRODUCTION

The need for user authentication and user identification in Information Technology (IT) world seems to date back to the late sixties [1], and the idea to use fingerprints was already there [2, 3]. More than thirty years later, the use of a personal trait to be closely linked with our identification documents (e.g. passports, visas, national ID cards) will invade our everyday life [4, 5].

To prove our identity, we can use three ways [1]:

1. Something we have (e.g. a Smart Card)
2. Something we know (e.g. a PIN code, a Password)
3. Something we are (Biometrics, e.g. Fingerprint, Face, Iris)

In everyday life, we usually give our trust to a combination of *something-we-have* and *something-we-know* (e.g. banking cards, SIM card in mobile phones) but a password can be communicated or guessed and a personal device can be lost or borrowed. Building a three-factor authentication with the addition of one or several biometric techniques brings high confidence in our authenticated interlocutor and provides non-repudiation.

2. AUTHENTICATION FACTORS

2.1. Smart Card

A conventional smart card is a silicon electronic chip embedded in a plastic rectangle printed with information concerning the application or the issuer, as well as readable information about the card holder (for instance, a validity date or a photograph). This support can also carry a magnetic stripe or a bar-code.

While for the time being smart card microprocessor cores are mainly 8 or 16-bit (the most common cores are Motorola's 68HC05 and Intel's 80C51), new 32-bit devices have recently become available. From a functional standpoint a smart card is a microcontroller, or let's say a miniature computer. A small on-board RAM serves as a temporary storage of calculation results and the card's microprocessor executes a program etched into the card's ROM at the mask-producing stage. This program cannot be modified or read-back in any way. For storing user-specific data individual to each card, cards contain EEPROM (Electrically Erasable and Programmable ROM) or flash memory, which can be written and erased hundreds of thousands of times. Java cards even allow the loading of executable programs (applets) into their nonvolatile memory according to the card holder's needs.

The smart card chip contains a communication port for exchanging data and control information with the external world. This communication port may be a contact interface or

a contactless interface. The smart card chip is the ideal container for cryptographic secrets such as symmetric secret keys and asymmetric private keys. The use of contactless smart card chip is now mandatory in numbers of travel documents [6] and national ID programs. Here, the role of the electronic chip is to authenticate the document (*something-we-have*) using cryptographic tools [7].

2.2. Password

A password is certainly the oldest and best known solution to provide user authentication. However this sounds simple to use, we have to take care about how the password is communicated: a secure channel between the authenticator (the system or person controlling the authentication) and the applicant (the candidate user) must be available, notably at the primary exchange to set up the shared password. If these minimal precautions aren't taken, very simple man-in-the-middle attacks such as eavesdropping are possible. One of the most used password-based authentication is the PIN (Personal Identification Number) code authorizing the use of a banking card. In this case, precautions must be taken when entering the PIN code since this is very easy to spy over the shoulder of the user (attack known as "shoulder-surfing").

To ensure security in IT systems, the password is never stored nor in clear text, nor only encrypted (reversible function) but only a cryptographic hash (short signature built with a one-way function) of the password is kept is the system's memory. During the authentication, the candidate password is hashed and the hash value is compared with the hash value stored in the system (i.e. the hash value of the reference password). We will see later that this approach of passwords security is not usable with Biometrics, that need clear data for the comparison process, hence decreasing the security level.

2.3. Biometrics

The biometric authentication [8] has the advantage of checking the user's personal characteristics. These characteristics can be physical ones such as fingerprints, face, iris or behavioral ones such as voice, handwritten signature, keyboard tapping.

This brings to a possible split in the usually called something-we-are:

1. Something we are (physical Biometrics)
2. Something we know how to do (behavioral Biometrics)

Behavioral characteristics are much less stable than physical characteristics because of their poor resistance to user's stress or health troubles. The authentication process is a comparison between a pre-registered reference image, or template (representative data extracted from the raw image, built during an *enrolment* step) and a newly captured candidate image, or template. Depending on the correlation between these two samples, the algorithm will determine if the applicant is accepted or rejected. This statistical process leads to a False Acceptance Rate (FAR, i.e. the probability to accept a non-authorized user) and a False Rejection Rate (FRR, i.e. the probability to reject an authorized user).

3. SECURITY ISSUES

3.1. Biometrics and Smart Card

The use of Biometrics without any personal device to store the reference template leads to privacy concerns: the centralized database which stores all biometric information from every user could be hacked. The use of a smart card here allows building up a distributed database where every user is the carrier of his own biometric reference, hence downsizing the previous privacy concern. Depending on the application, the smart card could handle differently the biometric data:

1. Storage-on-Card (SoC): the reference biometric template is stored on the smart card and is read by the system at any authentication request. This only uses non-volatile memory, hence allowing cost-effective smart card. However the reference template is exposed to different attacks when communicated out of the smart card.

2. Match-on-Card (MoC): the reference biometric template will never be communicated out of the smart card once written during the enrolment step. The candidate template is sent to the smart card and the comparison is processed internally. This protects the reference biometric template but needs a more powerful smart card in terms of processor and memory resources. This is particularly interesting when the result of the authentication has only to be used locally: applet activation, access to private key for digital signature. A malicious terminal capturing a candidate template will never have the information in return if it matches or not.

3. Partial Match-on-Card (PMoC): this solution has the advantages of both previous solutions, permitting cost-effective smart card and protecting the biometric information. The biometric information is split in two on the smart card: a public part to be read by the terminal and a private part which will be locked on the smart card [9]. The process is like a biometric challenge-response: the terminal reads the public part of the biometric information, process complex computation and send a candidate to the smart card to be compared with the private part of the biometric information using very light computation on the smart card's chip. Processing and decision entities are clearly separated.

The combination of Biometrics and smart card is an old topic [10] but the idea of using Biometrics to replace the PIN code for security reasons is too often cited. Biometrics capabilities are always overestimated. First of all, any biometric data is not a *secret*: a face can be seen on any picture or video recording even without the owner's authorization, fingerprints are left everywhere, voice can be recorded. Let's say Biometrics are *public* data, hence a biometric data can be captured and replayed[11, 12].

Different attacks and countermeasures are possible depending on the context of use of Biometrics and smart card. We define here three contexts of use:

1. Attended Terminal : the applicant is in front of the authority (e.g. face to face with a policeman)

2. Trusted Third-Party: under video surveillance (e.g. ATM, banks, shops)

3. Uncontrolled Area: user at home with the smart card and biometric device (e.g. e-voting, e-commerce)

For instance, only the last context of use would permit a manipulation of the biometric reader to bypass the captured image and replay a matching candidate; idem for using a large man-in-the-middle device.
Only the first context of use will prevent from the discrete usage of a fingerprint copy or bad-looking smart card copy; idem for using a discrete man-in-the-middle device. Classical attack paths are:

1. Man-in-the-middle (capture and replay)

2. Finger substitution (gummy fingers)

3. Smart card substitution (forged cards, yes-cards)

4. Fingerprint and smart card readers manipulation (probing)

Most of these attacks, finger substitution apart, can be stopped by using cryptographic tools (e.g. mutual authentication, session key). The countermeasures against finger substitution are liveness detection systems built in the biometric reader itself (e.g. pulse detection, skin conductivity).

A miscellaneous of threats and countermeasures can be found in the following tables:

Context of use	Threats
Attended terminal	False cards, YesCards
Trusted third-party	same as above + false finger
Uncontrolled Area	same as above + Reader manipulation

Context of use	Countermeasures
Attended terminal	Secure printing, signed data
Trusted third-party	signed data, liveness detection
Uncontrolled Area	signed data, liveness detection, tamper resistance

3.2. Biometrics and Password

First of all, the security of a password-based authentication tool such as ones in Unix or Windows systems are based on the local storage of only cryptographic hashes of passwords, no passwords themselves. This is possible because of the *deterministic* nature of password authentication: if the entered candidate password is the right one then its hash value equals the stored hash value and the authentication succeeds; if the entered candidate password is a wrong one then its hash value is different and the authentication fails [13].

This previous approach of security is impossible with biometric data. Any new capture of a biometric candidate results in slightly different data which leads to the *statistical* nature of Biometrics-based authentication (distance evaluation between two samples) [8]. The hash value of a reference biometric template will be totaly different from the hash value of any matching candidate, this means that biometric references have to be store locally, in clear text or maybe encrypted but encryption is a reversible function unlike a hash function which is a one-way function. For more information on encryption functions and hash functions see [14].

A deep characteristics analysis of both passwords and Biometrics shows a clear opposition. This opposition confirms the good complementarity of passwords and Biometrics. The replacement of one with the other should be carefully studied depending on the targeted application.

Previous things being said, we now need to counterbalance with situations where Biometrics are in any case more secure than passwords: weak passwords, bad-managed passwords, password-based authentication deactivated by the user. Many Information System administrators complain about users writing their password on a Post-It® note stuck under their keyboard or even on their computer's screen. Many mobile phone users leave the default PIN code (e.g. 0000, 1234) to unlock the phone or even deactivate this security feature considered as counter user convenient. Too many passwords, to be memorized, are short and explicit hence could be easily guessed with a simple dictionary attack [15] or more sophisticated attacks [16].

Thus, in most cases involving non security-aware users in an environment requesting a minimum of security, the use of Biometrics will anyway provide a "weak, but easy" security tool.

3.3. Three-Factor Authentication

Any combination of two among three authentication factors will miss at least one of the different security criteria. *Something-we-know* with *something-we-are* will miss privacy since no personal device implies the use of a database to centralized all biometric data. *Something-we-have* with *something-we-are* will miss a secret in the architecture since Biometrics are public data. *Something-we-have* with *something-we-know* will miss real user authentication since there is no proof of link between the user and his card/PIN code.

Three-factor authentication provides the highest security level in IT. Without being paranoid, some applications need to duplicate one factor in the authentication scheme: sometimes we need to show both ID card and Passport, we need to present both face and fingerprints, we need to enter the password to log in a system and then enter another password for the application we intend to use. For instance, the use of smart card, PIN code, fingerprints and facial recognition remains a three-factor authentication and not a four-factor authentication as we can sometimes read in press releases and marketing messages.

In today's digital world, most of communication channels are insecure since the first goal was to provide user convenience. When delivering a password or a biometric data, a particular attention must be paid to this communication channel to avoid very simple way to bypass authentication in the system. The use of cryptographic tools is mandatory to ensure the security of any three-factor authentication, the ultimate solution being to combine three-factor authentication with a Public Key Infrastructure (PKI). Nevertheless PKI being hard and costly to set up, manage and maintain, more simple solutions to provide secure communications over insecure channels [17] and to provide confidentiality and integrity of data [14] must be considered.

4. THE YESCARD / NOCARD ISSUE

The YesCard is a smart card which has been maliciously modified to always answer with a positive authentication, whatever is the biometric data it receives. This helps an attacker to enter in the system by presenting his own fingerprint and the biased smart card. This attack was popular few years ago in the banking area, exploiting a security flaw in ATM during off-line transaction.

Conversely, the NoCard is a smart card which has been maliciously modified to always answer with a negative authentication, whatever is the biometric data it receives. This provides denial of service for an authorized person to whom an attacker has replaced the card and then get some benefits from this situation (afterward, the attacker could impersonate the authorized user with a YesCard to enter in the system).

The Match-On-Card feature has the unique advantage of protecting the reference template of the user against capture and replay attacks by storing this reference template in a "safe". Once written at the enrolment, the smart card will never output this reference, only the candidate will be sent to the smart card to be internally compared with the reference. However, since the smart card takes the decision, the MoC feature opens the path for YesCard and NoCard. This widely used argument against Match-on-Card can be easily thwarted by the protocol described hereafter.

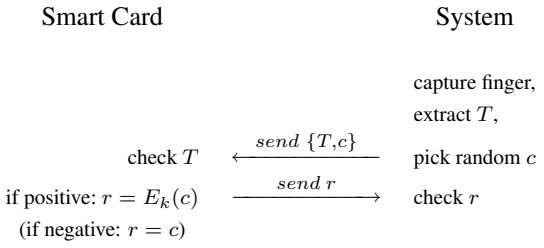

Fig. 1. Protocol #1

Firstly, we assume the use of a secure block cipher E (e.g. AES, 3DES) and a cryptographic key k shared between genuine smart cards and the system. The first idea was to use a challenge-response protocol to output the decision of the smart card: a/ if positive verification of candidate template T, the smart card send $r = E_k(c)$ (the response) where c (the challenge) is a random value given by the system together with the candidate template T b/ if negative, the smart card send any value different from r (c for instance). See Figure 1.

However this only protects from the Yescard. Then we replace c by $c\|b$, denoting the concatenation of c with a bit b where $b = 0$ if negative authentication or $b = 1$ if positive authentication. The smart card will then send $r = E_k(c\|b)$. See Figure 2.

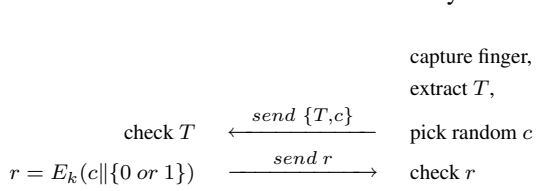

Fig. 2. Protocol #2

This protocol obviously protects from a non-authorized smart card to be a Yes-or-No Card.

5. THE ORACLE ISSUE

An oracle is a device or an algorithm to which we can submit questions and get answers, the oracle model is a powerful tool to evaluate the security of a system by estimating the average number of necessary queries to guess the content of the ora-

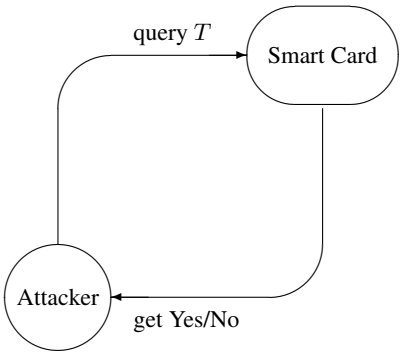

Fig. 3. Smart Card as an oracle

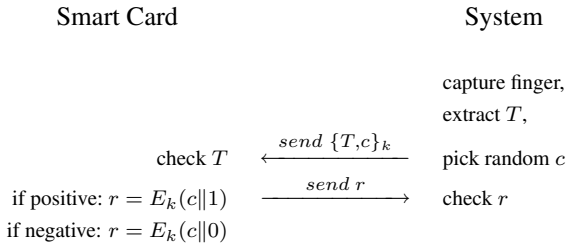

Fig. 4. Protocol #3

cle. Of course, a smart card could be used as an oracle by a malicious system to guess a matching candidate T (see Figure 3).

We can thus enhance our protocol to resist it (even if a practical countermeasure could be the use of a try-counter of non-matching candidate T to turn off the card). The Protocol #2 does not protect against a malicious system which will send different T with always the same challenge c and analyze differences in answers (a practical countermeasure could be the comparison of the challenge received with a log table of previously used c to turn off the card). Moreover, classical side-channel attacks against smart cards could be used to find the value of the concatenated bit (which represents the decision) by carefully looking at the microprocessor operations (e.g. power consumption or processing time will differ between computation with $\|0$ or $\|1$), all other bits being known since the challenge c is transmitted in clear.

A simple way to protect the smart card against unauthorized system is to encrypt the couple $\{T, c\}$ under the shared key k. This also protects against side-channel attacks to retrieve the concatenated bit since all the other bits of the challenge c are no longer known. See Figure 4.

6. CONCLUSION

In this paper we introduce the notion of *NoCard*, being as problematic as YesCard in the Biometrics domain and propose a protocol (Figure 4) that thwarts both attacks in a unique simple way. Moreover this protocol prevents the smart card from being used as an oracle by an unauthorized system to guess its biometric content.

7. REFERENCES

[1] International Business Machines Corp., "The consideration of data security in a computer environment," *IBM, Data Processing Division*, 1968.

[2] J.H. Wegstein, "A computer oriented single fingerprint identification system," *National Bureau of Standards, NBS Technical note 443*, Jan. 1968.

[3] J.H. Wegstein, "Matching fingerprints by computers," *National Bureau of Standards, NBS Technical note 466*, Jul. 1968.

[4] Joint Research Centre, "Biometrics at the frontiers: assessing the impact on society," Tech. Rep. EUR 21585 EN, European Commission, 2005.

[5] ICAO, "Biometrics deployment for Machine Readable Travel Documents," Tech. Rep., May 2004, Available at http://www.icao.int/mrtd/download/documents.

[6] ICAO, "Annex 1 - Use of Contactless Integrated Circuits," Tech. Rep., May 2004, Available at http://www.icao.int/mrtd/download/documents/Annexs.pdf.

[7] ICAO, "PKI for Machine Readable Travel Documents offering ICC read-only access," Tech. Rep., Oct. 2004, Available at http://www.icao.int/mrtd/download/documents/TR-PKIy

[8] Anil Jain, Ruud Bolle, and Sharath Pankanti, *Biometrics - Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999.

[9] Claude Barral, Jean-Sébastien Coron, and David Naccache, "Externalized fingerprint matching.," in *ICBA*, David Zhang and Anil K. Jain, Eds. 2004, vol. 3072 of *Lecture Notes in Computer Science*, pp. 309–315, Springer.

[10] Gaël Hachez, François Koeune, and Jean-Jacques Quisquater, "Biometrics, Access Control, Smart Cards: A not so simple combination.," in *CARDIS*, Josep Domingo-Ferrer, David Chan, and Anthony Watson, Eds. 2000, vol. 180 of *IFIP Conference Proceedings*, pp. 273–288, Kluwer.

[11] Tsutomu Matsumoto, Hiroyuki Matsumoto, Koji Yamada, and Satoshi Hoshino, "Impact of artificial gummy fingers on fingerprint systems.," *Datenschutz und Datensicherheit*, vol. 26, no. 8, 2002.

[12] Tsutomu Matsumoto, "Gummy and conductive silicone rubber fingers.," in *ASIACRYPT*, Yuliang Zheng, Ed. 2002, vol. 2501 of *Lecture Notes in Computer Science*, pp. 574–576, Springer.

[13] David C. Feldmeier and Philip R. Karn, "Unix password security - ten years later.," in *CRYPTO*, Gilles Brassard, Ed. 1989, vol. 435 of *Lecture Notes in Computer Science*, pp. 44–63, Springer.

[14] Serge Vaudenay, *A Classical Introduction to Cryptography: Applications for Communications Security*, Springer, 2005.

[15] Stéphanie Delaune and Florent Jacquemard, "A theory of dictionary attacks and its complexity.," in *CSFW*. 2004, pp. 2–15, IEEE Computer Society.

[16] Philippe Oechslin, "Making a faster cryptanalytic time-memory trade-off.," in *CRYPTO*, Dan Boneh, Ed. 2003, vol. 2729 of *Lecture Notes in Computer Science*, pp. 617–630, Springer.

[17] Serge Vaudenay, "Secure communications over insecure channels based on short authenticated strings," in *CRYPTO*, Victor Shoup, Ed. 2005, vol. 3621 of *Lecture Notes in Computer Science*, pp. 309–326, Springer.