Sáchez-Reíllo, R. (2015) Tamper-Proof Operating System. In: Li S.Z., Jain A.K. (eds.) *Encyclopedia of Biometrics*. Springer, Boston, MA

DOI: https://doi.org/10.1007/978-1-4899-7488-4_291

Metadata of the chapter that will be visualized online

Chapter Title	Tamper-Proof Operating System	
Copyright Year	2014	
Copyright Holder	Springer-Verlag London	
Corresponding Author	Family Name	**Sanchez-Reillo**
	Particle	
	Given Name	**Raul**
	Suffix	
	Division	GUTI (University Group for Identification Technologies)
	Organization	Carlos III University of Madrid
	Address	28911, Avda. Universidad, 30, Leganes, Madrid, Spain
	Email	rsreillo@ing.uc3m.es
	Email	raul.sanchezreillo@gmail.com

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

Tamper-Proof Operating System

Q1
Q2

Raul Sanchez-Reillo*
GUTI (University Group for Identif cation Technologies), Carlos III University of Madrid, Leganes, Madrid, Spain

Synonyms

Malicious-code-free operating system; Secure biometric token operating system

Definitio

Operating system with a robust design, as not to allow the execution of malicious code. Access to internal data and procedures are never allowed without the proper authorization. In its more strict implementations, this operating system will have attack detection mechanisms. If the attack is of a certain level, the operating system may even delete all its code and/or data.

Introduction

The handling of sensible data in Information Systems is currently very usual. Which data is to be considered sensible is up to the application, but at least we can consider those such as personal data, f nancial data, as well as access control data. Actors dealing with such Information System (clients/citizens, service providers, integrators, etc.) have to be aware of the security level achieved within the system.

Although this is a very important issue in any system, when biometric information is handled, it becomes a critical point. Reason for this is that biometric information is permanently valid, as it is expected to be kept the same during the whole life of a person. While a private key can be changed as desired and even cancelled, a user cannot change his f ngerprint (unless changing f nger) or even cancel it. If cancelling biometric raw data, the user will be limited, in case of f ngerprints, to 10 successful attacks during his/her whole life. These kinds of considerations have already been published even back to 1998, as it can be read in [1].

Therefore, biometric systems have to be kept as secure as possible. There are several potential vulnerable points (PVPs) in any biometric system, as it can be seen in Fig. 1. All those 9 PVPs have to be considered when designing a biometric solution. A good introduction to threats in a biometric system can be found in [2, 3] and in BEM [4]:

- PVP 1 has to deal with user attitudes, as well as capture device front end. Regarding user attitude, an authorized user can provide his own biometric sample to an impostor unknowingly, unwillingly, or even willingly. From the capture device front-end point of view, such device may not be able to:

*E-mail: rsreillo@ing.uc3m.es
raul.sanchezreillo@gmail.com

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

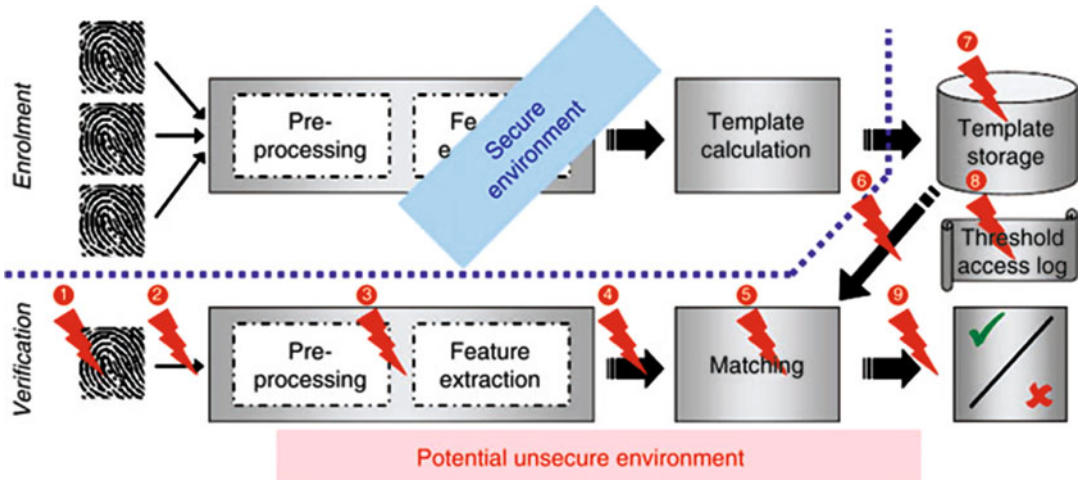

Fig. 1 Potential vulnerable points in a biometric system where enrolment is considered secured

- Detect a nonlive sample
- Detect the quality of the input sample, being able to discard those under a determined threshold
- Protect the quality threshold against manipulation
- Detect degradation of its own degradation
- Resist environmental factors
- Eliminate residual information from previous captures
- Detect and discard sample injection
- Deny successive and fast sample presentation

- PVP 2 is directly related to the threat group 3 of BEM. It is basically focused on the capture device back end as well as the front end of the Biometric Algorithm. Captured sample could be intercepted and/or reinjected, to provide a replay attack. Major problem relies on the potential loss of the user's biometric identity. Also, another threat is a hill-climbing attack by injecting successive biometric samples.
- PVPs 3, 4, 6, 7, and 8 could be treated as in any other IT system (Trojans, viruses, communications interception, data injection, hill-climbing attacks, etc.). So the same kind of study shall be done. It is in this kind of PVPs where a tamper-proof operating system can be of help. It is important to note that sensibility related to biometric-related information covers not only the sample data, feature vectors, and templates but also thresholds, access logs, and algorithms.
- PVP 5, being also a typical point of study in any IT system, has here more importance depending on the information that could be given by the system after the matching. If matching result is not given just by an OK/ERROR message, but also carries information about the level of matching acquired, this could be used by an attacker to build an artif cial sample, by hill-climbing techniques. For this PVP also, the tamper-proof O.S. can play an important role.

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

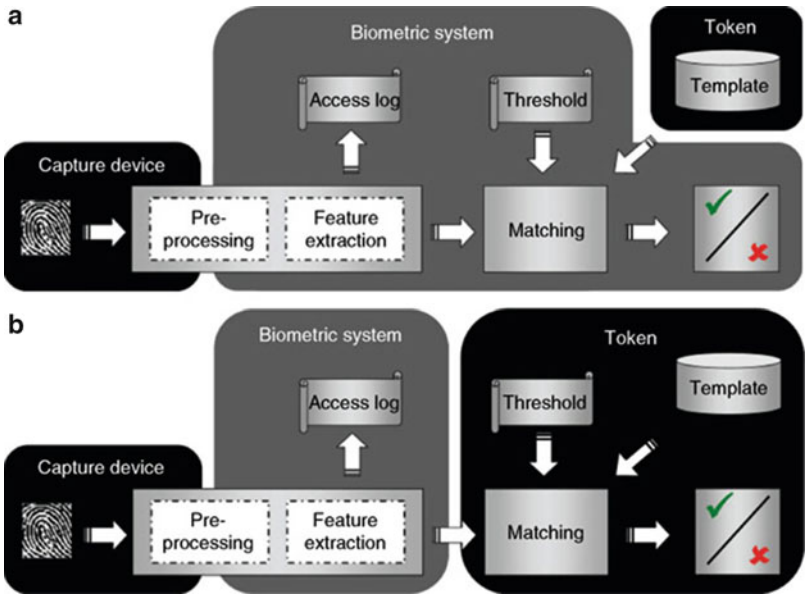

Fig. 2 Some architectures of biometric authentication systems, splitting tasks in several devices

Biometric Devices

Regarding biometrics, a tamper-proof operating system is intended to be running in some (or any) elements which are part of the biometric system. The idea of this kind of operating systems is not new, as they are already implemented in other areas, such as smart cards for f nancial services. This kind of electronic devices are designed under a basic security rule: "Not only the device has to work under its constrained conditions of user, but also has to stop working outside those conditions." In few words, this means that, for example, if the smart card is expected to work with a supply voltage from 4.5 to 5.5 volts, it does not have to work outside the range (e.g., if supply voltage is 4.4 or 5.6, not even a response has to be obtained from the card). Related to the operating system inside the card, this covers things like not allowing the execution of any undef ned/undocumented command or not being able to install new functions that can behave as Trojan horses or viruses.

With this example, the reader can think that this kind of products does not really exist, because several papers have been published related to security problems with smart cards (e.g., [5] and some general audience press). It has to be stated that not all times an integrated circuit identif cation card is referred; it is really a smart card (i.e., a microprocessor-based identif cation card with a tamper-proof O.S.). Also, some real smart cards have not been properly issued, leaving some critical data f les unprotected, or not using the security mechanisms provided. Rules to be followed to properly use a smart card can be found in [6].

This same kind of rules can be applied to all kinds of biometric devices. Obviously, depending on the system architecture, biometric devices can be of very different kinds. Figure 2 shows two possible architectures of a biometric authentication system, which are usually known as (a) match-off-card (also known as match-off-token) and (b) match-on-card (or match-on-token). Apart from these two, many other schemes can be designed. The term "match" should be changed for "comparison." So instead of "match-off-card," "off-card biometric comparison" should be used. But within text "match-off-card" and "match-on-card" are used due to being terms widely used among the industry.

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

In a match-off-card system (e.g., [7]), we can consider a simplif cation of the system as composed of three devices: the capture device, the token or card where the user's template is stored, and the rest of the system, which will be named as "biometric system." Major difference with the match-on-card system (e.g., [8]) is that here the token only stores the user's template, while in the match-on-card version, it also performs some biometric-related computations.

In any case, any of those devices should be designed following the rules given below regarding a tamper-proof O.S. Being this viable, if those devices are developed as embedded systems, major problems can arise when one of those devices (typically the biometric system) is running on a general-purpose computer, where little or no control is available for installed applications and data exchange.

Requirements for a Tamper-Proof O.S.

Q3

Once focused on the environment where a tamper-proof O.S. has to be found in biometric devices, it is time to start its design. A good starting point will be following all previous works dealing with smart cards. The reason for that is to transfer the know-how of near 30 years of secure identif cation tokens given by the smart card industry [9]. This same ideas can be extrapolated to other biometric devices, not only personal tokens.

First thing to consider when designing a tamper-proof O.S. is the different life phases that the biometric device will have. All devices, specially those related to personal authentication, should go through different life stages, from manufacturing to its use by the end users. As information handled by them is really sensible, extra protection should be taken to avoid robbery, emulation, or fraudulent access to the device or its information. Therefore, security mechanisms will be forced in each life stage. Those mechanisms are mainly based on Transport Keys, which protect access to using the device in each change of its life phase. Life phases define are:

- Manufacturing: where the device is assembled. The microcontroller within the device should be protected by a Transport Key, before being delivered to the next stage. The way to compute that Transport Key for each microcontroller will be sent to the company responsible of the next phase by a separate and secured way.
- Personalization: In this phase, each device is differentiated from all others by storing some unique data related to the fina application, user, and access conditions. Sometimes this phase is split in several subphases, specially when the device has to be personalized for the application (prepersonalization) and then for the f nal user (personalization), as it may happen with identif cation tokens. In this phase, also data structure regarding the applications may be created, as well as the full security architecture.
- Usage: The end user is ready to use the device.
- Discontinuation: Due to aging, limited time use, accidents, or attack detection, the device may be out of use. This can be temporary (e.g., when keys are blocked) or permanent (no reactivation is allowed). It has to be guaranteed that once discontinued, such device shall not be able to be used.

Entering in details regarding the requirements for a tamper-proof operating system, we can state the following general rules:

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

- Mutual authentication mechanisms have to be used before exchanging any kind of biometric information. In any communication, both parts have to be sure that the other party is a reliable one.
- To avoid replay attacks, some time-stamping-like mechanisms have to be used (e.g., generation of session keys to sign/cipher each message exchanged).
- Only the manufactured designed commands can be executed. No possibility of downloading new commands has to be allowed. Therefore, flash reprogramming and device updating are strongly discouraged.
- Before executing anything in the biometric device, full integrity check (both cryptographic and semantic) of the command and its data has to be performed. Some attacks would try to exploit undefined cases in the semantics of a command exchanged.
- All sensible data (sample data, feature vectors, templates, and thresholds) has to be transmitted and ciphered.
- If there is a command related to changing parameters, it has to be sent with all security mechanisms allowed, as the system can be even more vulnerable to attacks related to changing those parameters (e.g., quality or verification thresholds).
- Feedback information from the device to the external world has to be as short as possible to avoid hill-climbing attacks. For example, a device performing comparisons in an authentication system has to provide only a YES/NO answer, but not giving information on the matching score obtained.
- Attack detection mechanisms have to be considered. If an attack is detected, then the device has to stop working, and a reinitialization has to be made. If the detected attack is considered extremely serious, the device may consider deleting not only all temporal data but also its permanent data or even its programming code.
- Successive failed attempts to satisfy any security condition has to be considered as an attack, and therefore, the device has to be blocked, as it happens with a PIN code in a smart card.
- No direct access to hardware resources (e.g., memory addresses, communication ports, etc.) can be allowed. Most virus and Trojan horses benefit for not following this rule.
- As soon as data is no longer needed by the operating system, it has to be erased as to prevent latent data to be acquired in a successful attack.

Most of these requirements can be satisfied by defining a security architecture based on cryptographic algorithms. Several implementations can be followed. If the developer is not familiar with these mechanisms, it is suggested to follow the secret codes/secret keys architecture of a smart card and the Secure Messaging mechanism [6, 9]. These can be directly applied to personal tokens and upgraded to another kind of biometric devices.

Example of an O.S. Instruction Set

When implementing a tamper-proof O.S., several design decisions have to be made: frame formats, time-outs, number of retries, etc. All these issues depend on the communication strategy followed by the whole biometric system. Therefore, no general rule can be given to the designed.

Regarding the instruction set, a minimal list of functions can be considered, depending on the device where the O.S. is to be included. This is also dependent on the platform chosen. As an example, the instruction set for a limited-resources platforms is given. This instruction set has been

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

proposed to ISO/IEC JTC1/SC37 to be considered as a lighter version of BioAPI, the standardized Application Program Interface for biometric applications. This lighter version is called BioAPI Lite and is being standardized as ISO/IEC 29164.

Commands needed by a limited biometric device depend on the functionality of such device. Q4 Obviously a capture device is not the same as a personal token. But in general terms these commands can be classif ed in four major groups: module management, template management, biometric enrolment, and biometric process.

Q5 Management commands relate to managing the overall module behavior. Four commands can be considered in this group:

- Initialize: Tells the module to initialize itself, opening the offered services, and initialize all security for ciphered data exchange. This command is to be called any time a session is started (power on, session change, etc.). Without being called, the rest of the commands shall not work.
- Close: Tells the module to shut down.
- Get properties: Provides information on capabilities, conf guration, and state.
- Update parameters: Updates parameters in module. One of such parameters can be the comparison threshold. For that reason, this function is recommended to be used with all security mechanisms available.

Template management commands refer to those functions needed to store and retrieve templates from the module. These functions will be supported by those modules that are able to store users' templates. These set of commands are expected to be used by personal tokens or small databases. The functions def ned are:

- Store template: stores the input template in the internal biometric module database
- Retrieve template: obtains the referenced template from the biometric module

The next group is the biometric enrolment commands. This group of functions will be considered in systems where enrolment is to be made internally. Due to the different processes of enrolment, even for a single biometric modality (e.g., different numbers of samples needed), in limited devices, a multistep procedure is suggested. First, user will call related functions to obtain samples for the enrolment, and then a call to the enrolled function will have to be done. Commands def ned are:

- Capture for Enrol: Performs a biometric capture (using on-board sensor), keeping the information in module for later enrolment process. The number this function is called depends on the number of samples the module needs to perform enrolment. As this operation involves user interaction, biometric module manufacturer shall consider time-out values to cancel operation, reporting that situation in the status code returned.
- Acquire for Enrol: Receives a biometric sample to keep the information in module for later enrolment process. The number this function is called depends on the number of samples the module needs to perform enrolment. Depending on module capabilities, input data can be a raw sample, a preprocessed one, or its corresponding feature vector.
- Enrol: Performs an enrolment to create a template and stores the template in module. To execute this function, either Capture for Enrol or Acquire for Enrol functions has to be called in advance. Enrol with process with the samples temporally stored in the module. The return value is the number of template internally assigned.
- Erase Enrolments: Erases all enrolment templates or the indicated (by number) template.

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

Finally, the fourth group is dedicated to all those commands that are dealing with biometric functions. It covers the capture process, feature extraction, and comparison. Even with comparison, it handles comparisons with internal templates or templates coming from the external world:

- Capture: Performs a biometric capture (using on-board sensor), returning biometric sample.
- Process: Processes biometric sample to create comparable recognition data (feature vector). Depending on module capabilities, the input sample can be a raw sample or a preprocessed one.
- Capture and process: Performs a biometric capture (using on-board sensor), returning its feature vector.
- Compare external: Compares a feature vector with the template sent by the external world.
- Process and compare external: Processes a biometric sample and compares it with the template sent by the external world.
- Capture and compare external: Performs a biometric capture (using on-board sensor), processes the biometric sample, and compares it with the template sent by the external world.
- Compare internal: Compares a feature vector with templates stored in the module. If the input parameter is 0xFF, comparison will be done with all templates stored. In another case, comparison is done only with the template whose internal number is given at the input parameter.
- Process and compare internal: Processes a biometric sample and compares it with templates stored in the module. If the input parameter is 0xFF, comparison will be done with all templates stored. In another case, comparison is done only with the template whose internal number is given at the input parameter.
- Capture and compare internal: Performs a biometric capture (using on-board sensor), processes the biometric sample, and compares it with templates stored in the module. If the input parameter is 0xFF, comparison will be done with all templates stored. In another case, comparison is done only with the template whose internal number is given at the input parameter.

Some of these instructions involve user interaction. Therefore, manufacturer shall consider time-out values to cancel operation if it is exceeded, reporting that situation within the protocol used.

Applicability of Tamper-Proof O.S.

As mentioned above, this kind of operating system is desirable to be included in all devices related to biometric identif cation, but unfortunately this is not always possible. As in many applications, a general-purpose computer is used, and general-purpose operating systems are used (such as Windows, Linux, etc.). Developing those O.S. in a tamper-proof way without restricting usability and generality is nearly impossible. Therefore, tamper-proof operating systems are meant for those embedded systems, sensors, and personal tokens dealing with personal identif cation.

Using this kind of tamper-proof O.S. in these devices, restrict the number of security holes to the minimum within the device and to be concentrated only in those general-purpose systems used. As some tasks will be performed in such secured devices, security leaks will be avoided. For example, if a biometric system uses personal tamper-proof tokens with match-on-card capability, the user's template will never be exposed, and possibility of hill-climbing or replay attacks will be cancelled. Thus, all comparison and decision blocks will be secured, restricting the potential security problems to the relevant previous modules.

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

Summary

Due to the sensibility of biometric data, security in biometric devices has to be considered. One of the ways to protect privacy is to include a tamper-proof operating system. This O.S. would not allow direct access to hardware resources of the device, neither to temporary nor permanent data. This O.S. has also to control the different life stages of the device. A set of requirements have been def ned that have to be considered when developing such tamper-proof O.S. Finally an example of the commands to be covered by some devices have been given. Including this kind of O.S. in all biometric devices will improve the security of the whole system. Unfortunately, when some parts of the biometric system has to be implemented in a general-purpose computer with an open operating system, applying these rules is not easy.

Related Entries

Q6

- Biometric Security Threat
- Biometric Token
- Biometric Vulnerabilities
- Match-off-Card
- Match-on-Card
- Template Security

References

1. M. Rejman-Greene, Security considerations in the use of biometric devices. Inf. Secur. Tech. Rep. **3**, 77–80 (1998)
2. N.K. Ratha, J.H. Connell, R.M. Bolle, Enhancing security and privacy in biometrics-based authentication systems. IBM Syst. J. **40**(3), 614–634 (2001)
3. C. Roberts, Biometric attack vectors and defences. Comput. Secur. **26**(1), 14–25 (2007)
4. C. Criteria, Biometric evaluation methodology supplement (BEM). Common methodology for information technology security evaluation (2002), http://www.cesg.gov.uk/site/ast/biometrics/media/BEM_10.pdf
5. A. Matthews, Side-channel attacks on smartcards. Netw. Secur. **2006**(12), 18–20 (2006)
6. R. Sanchez-Reillo, Achieving security in integrated circuit card applications: reality or desire? IEEE Aerosp. Electron. Syst. Mag. **17**, 4–8 (2002)
7. R. Sanchez-Reillo, A. Gonzalez-Marcos, Access control system with hand geometry ver- if cation and smart cards. IEEE Aerosp. Electron. Syst. Mag. **15**(2), 45–48 (2000). doi:10.1109/62.825671
8. R. Sanchez-Reillo, Smart card information and operations using biometrics. IEEE Aerosp. Electron. Syst. Mag. **16**(4), 3–6 (2001). doi:10.1109/62.918014
9. ISO/IEC_JTC1/SC17, ISO/IEC 7816 Parts 3, 4, 8, 9 & 11 (1987–2005)

Encyclopedia of Biometrics
DOI 10.1007/978-3-642-27733-7_291-3

Author Queries

Query Refs.	Details Required
Q1	The chapter title "Tamper-Proof Operating System" is not listed in the ToC. Please check.
Q2	Please check if author affi iation is okay.
Q3	Please check if edit to sentence starting "Once focused on the environment. . ." is okay.
Q4	Please check if edit to sentence starting "Obviously a capture. . ." is okay.
Q5	Please check if edit to sentence starting "Management commands relate. . ." is okay.
Q6	Titles "Biometric Security Threat," "Biometric Token," "Match-off-Card," and "Match-on-Card" are not listed in the ToC. Please check.