Verisoft – Secure Biometric Identification System

The Verisoft project is a long-term research project, funded by the German Federal Ministry of Education and Research. It aims at verifying the correctness of concrete application tasks, one from academic and up to four from industrial backgrounds. This paper gives an introduction of one of the industrial applications, which is subproject 4 "Chipcard based Biometric Identification System (CBI-System)". Firstly, biometric systems in general are discussed in order to define security requirements of a more secure system. Then the security functions and the overall design of the CBI-System as well as verification task is given. Finally a more detailed view of the implemented function is provided.

Dr. Gunter Lassmann works as a Chief Scientific Adviser of the DC Security at T-Systems SL SI and has been working in Cryptography and IT-Security for Deutsche Telekom since 1987. He is one of the founders of the committees DIN NI 27 "IT Security" and TeleTrusT WG "Biometric Identification" and editor of the "Catalogue of Criteria for Evaluating the Comparability of Biometric Methods" [6].

Dipl. Inf. Mathias Schwan studied computer science at Humboldt-University Berlin with a focus on cryptography. He works as a scientific assistance at the University as well as at T-Systems and has been involved in various joint research and development projects in the field of PKI solutions, cryptographic protocols, biometrics, evaluation criteria and formal methods.

Dipl. Inf. Lassaad Cheikhrouhou studied computer science at the University of Kaiserslautern. He works as a research assistant at the German Research Center for Artificial Intelligence (DFKI) in the field of formal methods and security analysis. He has been involved in the formal analysis of several practical cryptographic protocols.

Dr. Georg Rock studied computer science at the University of Saarland and was awarded a doctorate in 2004. Since 1997 he has been working as a research assistant at the German Research Center for Artificial Intelligence (DFKI). His main focus is on projects regarding the formal verification of software and formal security analysis.

1 Introduction

1.1 The Project Verisoft

The main goal of the Verisoft project is the pervasive formal verification of computer systems, which means from the processor design up to the application layer. This way, human errors are excluded, full coverage is achieved and the results are based

on a well-known small set of assumptions. Hence, the verified systems are of very high quality as required in many industrial sectors, such as chip design, automotive engineering and security systems. The Verisoft project consists of six subprojects (SP) and is realized by a consortium of eleven partners and is headed by the Universität des Saarlandes. Further partners are DFKI, MPII and OFFIS, Universities of Darmstadt, Munich (TU), Koblenz and Karlsruhe. The industrial partners are Infineon, T-Systems, AbsInt and BMW. The defined subprojects cover different industrial sectors as mentioned above. In addition in SP2 an “Academic System” is put into account in order to be free of publication restrictions. It is managed by University of Koblenz and develops a system for signed e-mails. In SP3 “Correct Industrial Hardware System” (managed by Infineon), the hardware of a 32bit-Microcontroller is to be verified. In SP 6 “Automotive” (TU Munich/BMW) the automatic emergency call (e-call) from the automotive sphere is about to be persistently verified. T-Systems manages SP4 “Biometric Identification System”

1.2 The Subproject 4: Biometric Identification System

The *Chipcard based Biometric Identification System (CBI-System)* realizes a secure access control system. A host system compares biometric data from any biometric sensor with a reference template stored on a smartcard and grants or denies access depending on the degree of similarity between both sets of data. The access software, the cryptographic primitives and their combination as well as the security of the underlying cryptographic protocols will be formally verified. Protecting the individual’s reference template from misuse by malicious attackers on the host system is of high relevance in this subproject. The subproject 4 is divided in four work packages (WP). Content of WP4.1 is in a first step the formalization of the standardized communication protocol between the chipcard, the chipcard terminal and the host (T=1, ISO/IEC 7816-3). In a second step the deadlock characteristic of the communication protocol will be verified (common work with Saarland University (Group Prof. Finkbeiner)). The aim of WP4.2 is the proof that the security functions meet the security requirements using various techniques as cryptographic protocol verification and information flow analysis. The specification of the system is done in Unified Modeling Language (UML). The formalization and verification will be done in UMLsec by TU Munich and VSE by DFKI. The formalization and verification of the cryptographic primitives is done by TU Darmstadt. The WP4.3 will answer the question: Does the source code implement the same security functions as in the top level design? The CBI-System will be implemented in C-Code (later C0 Code) in order to verify its correctness against the formal specification. It's a cooperative work with DFKI and TU-Munich. A first demonstrator works since 12/04. The Integration of the CBI-System in the Academic System is done in WP4.4. The example application will implement a secure login procedure or electronic signature with biometrics within the Academic System. This makes the realization of extra system requirements and further system design necessary as well as the specification and realization of a secure and safe protocol between the CBI-System and the calling application.

2 The properties and problems of biometric methods

There are three main methods of verifying an identity presented to the system:

- something you know (PIN, Password, Passphrase);
- something you have (smartcards, RF-ID, other token);
- something you are (physiological or behavioral characteristics = biometrics).

A biometric system uses physiological and typical behavioral traits for authenticating the user. Biometric traits have the advantage that they can not be stolen and are difficult to copy. The outstanding characteristic of biometrics is its ability to verify the trait to be identified as well as its lawful possession. Examples of the most common biometric methods are fingerprint and face recognition, iris scan, signature dynamics and voice recognition. In general, biometric systems can operate in two main modes. In the mode *verification* the actual captured biometric data are compared only against the reference data of the particular person, he or she claims to be. It is a one-to-one comparison. In the mode *identification* the actual captured data are compared against a set of reference data of many people. The access is granted, if the biometric feature is sufficient similar to one reference data of the set. It is a one-to-many comparsion. The quality of an individual biometric system is expressed in terms of *False Accept Rate (FAR)* and *False Reject Rate (FRR)*, whereas the FAR is more security relevant. Please note, that the FAR will never be zero and is mostly lower in verification mode than in identification mode, see also [6].

In January 1999 T-Systems run a series of projects for the examination of biometric systems. Basic tests, examinations, attacks and field tests with approximately 800 people have been done for a selection of at least 46 biometric systems, which covered the features fingerprint, iris, face, voice, signature, handgeometry as well as multifeature systems. The tests gave an evidence that the *total error rate* of a biometric system consists of three additional parts:

- *The Device error rate*: The idealized error rate of the biometric device, tested under good conditions in the laboratory and tested on people with good biometric features.
- *The Quality of the biometric database*: If the enrolment is not made in the correct way, the quality of the biometric database will decrease.
- *The Quality of the fresh biometric data*: If a biometric method is used under changing or bad operational conditions the error rate will rise significantly.

From the results of field trials we have learned that there are some unavoidable problems in all biometric systems:

a. A fault in the biometrics authentication, i.e. the admittance of unauthorised or the rejection of authorised people, is part of the normal course of operations in a biometric system.
b. For each biometric system there are always people with no sufficient biometric features, each biometric method needs an alternative backup system.
c. Because most live checks in biometric systems do not work efficiently, these biometric systems have to be supervised.

d. Because this is a lifelong fixation of a user to his biometric data and according to German Federal Data Protection Act we need a complete solution for the privacy-problems before we can put the biometric system into operation.

3 The basic idea of subproject 4

Biometric methods fail with a not negligible probability. In combination with well-established security components like smartcards, it remains the basic security of the smartcard. The results of biometric methods in the operation mode *verification* are much better than in the operation mode *identification*. For that the system needs another element to indicate the claimed identity, the smartcard is the ideal solution. Password and smart card systems do not check whether the current user is the legal owner. This security defect can be eliminated with biometric methods. According to the German law the central storage of biometric data is strong restricted. With a smartcard we can store biometric templates only on the smartcards and not in a central register. A majority of users of biometric system is concerned with the problem of privacy and data abuse. A smartcard with the evaluated TCOS operating system is the ideal secure habitation for the biometric data.

The basic idea of the CBI-System is the smart combination of the established smartcard based methods and the novel biometric based methods in order to avoid the problems mentioned above. We formulate the following security requirements that have to be fulfilled by the CBI-System:

1. the host only accepts valid smartcards
2. the smartcard only accepts valid hosts
3. the biometric data and the biometric reference data must be handled confidentially
4. the host only accepts valid biometric reference data
5. after the matching process the fresh biometric data and the reference data have to be deleted
6. the biometric authentication is successful if the biometric data and the biometric reference data are sufficiently congruent
7. failed biometric matchings are tolerated, but their number is limited, therefore they have to be counted

4 Security functions and design of the CBI-System

In order to fulfill the requirements mentioned in the previous section the following security functions have to be implemented by the CBI-System:

- smartcard and host do mutual authentication and communicate in a secure way;
- an error counter of failed authentications between host and smartcard is introduced;
- the reference data is digitally signed;

- all biometric data is overwritten after the matching;
- an error counter of failed biometric verifications is introduced.

Now we can formulate the verification tasks of the subproject 4:

a. Do the security functions meet the security requirements?
b. Does the source code implement the same security functions as in the top level design?

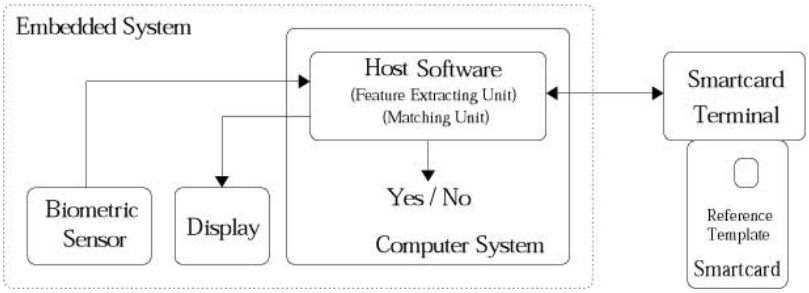

Figure 1: the overall design of the CBI-System

In figure 1 the overall design of the CBI-System is given. It consists of the host computer, the biometric sensor and the smartcard that communicates with the host via a smartcard terminal. A typical behavior of the system can be described as follows:

1. The smartcard and the host perform a mutual authentication using preshared symmetric keys given in the initial step. The count of failed authentication trials must not be reached. After a successful mutual authentication a symmetric session key is known to both parties and the error counter is set to the default value. Otherwise the error counter is decremented and the session terminated.
2. The host reads the second error counter from the smartcard that indicates the count of failed biometric verifications. If the error counter is not equal to 0 the host writes the decremented error counter back on the smartcard.
3. The host reads the data set consisting of the reference template, the identification number of the smartcard and the electronic signature generated by the administrator. The host verifies the electronic signature
4. The host reads the fresh biometric data from the biometric sensor.
5. The host compares the fresh biometric template and the reference template. If the verification is successful the host deletes the fresh biometric data and the reference data, the host writes the default error counter back on the smartcard and gives the result back to the user via the display.
   In the negative case the host deletes the fresh biometric data and continues with step 4. It does the loop at a maximum of three times.
6. The output of the CBI-System is the card ID and the ``biometrically authenticated'' information. In the negative case the session is closed and the output of the CBI-System is the card ID and the ``biometric authentication failed'' information.

In the following section we present the formal representation of the cryptographic protocol used in the communication between the different communication partners indicated in Figure 1.

5 Formal specification and verification of the CBI-cryptographic protocol

The specification and verification of the CBI protocol that is illustrated in Figure 2 is performed with the help of the Verification Support Environment System (VSE) [5]. VSE contains a library of predefined data types that is used to specify cryptographic protocols. The formalism we are using in VSE is based on the work of Paulson [2]. It is a trace-based approach where every trace represents one possible run of the protocol. The methodology allows for

- the representation of arbitrary many interleaved runs of a protocol and
- the representation of arbitrary many communication partners belonging to different runs.

The attacker model is based on that of Dolev and Yao [1]. The attacker has the possibility to

- observe all messages ever sent,
- interrupt message transfer,
- analyze messages,
- create new messages from its current knowledge and
- sent these new messages to arbitrary communication partners.

The verification task in the cryptographic protocol analysis is mainly concerned with the proof of desired properties, such as data integrity, authenticity or secrecy. The verification task in VSE is supported by many heuristics that lead in some cases to a nearly automatic verification of the desired properties. Especially in the field of protocol verification where proofs can get very big, it is essential to have a strong proof support.

After this short overview of the methodology VSE is based on with respect to the cryptographic protocol verification we will present the CBI protocol, its formalization and the specification and verification of some selected properties.

5.1 CBI - Protocol specification

Figure 2 shows the steps of the CBI protocol which corresponds to the identification scenario in the CBI system as described in the previous section. Here, the words in typewriter are constants and the other words are protocol variables, which can be substituted by different instances in different protocol sessions.

In the following we go through the steps of the protocol and explain their meaning:

- In step one the host (with identifier) ***Host*** sends the message consisting of a command (**askRandom**) and its identifier to the chipcard with identifier ***CK***.

This message allows for the chipcard to determine the identifier of the other protocol participant.

1. $Host \longrightarrow CK : \texttt{askRandom}, Host$
2. $CK \longrightarrow Host : Rsc$
3. $Host \longrightarrow CK : \{Rhost, Rsc, CK, Host\}_{K_{auth}(CK,Host)}$
4. $CK \longrightarrow Host : \{Rhost, Rsc2, Host\}_{K_{auth}(CK,Host)}$
5. $Host \longrightarrow CK : \{\texttt{getSessKey}, Rsc\}_{K_{enc}(CK,Host)}, MAC(K_{mac}(CK, Host), \{\texttt{getSessKey}, Rsc\}_{K_{enc}(CK,Host)})$
6. $CK \longrightarrow Host : \{K_{CH}, Rsc\}_{K_{enc}(CK,Host)}, MAC(K_{mac}(CK, Host), \{K_{CH}, Rsc\}_{K_{enc}(CK,Host)})$
7. $Host \longrightarrow CK : \texttt{askFBZ2}, MAC(K_{CH}, \texttt{askFBZ2})$
8. $CK \longrightarrow Host : \texttt{sendFBZ2}, FBZ2, MAC(K_{CH}, \{\texttt{sendFBZ2}, FBZ2\})$
9. $Host \longrightarrow CK : \texttt{writeFBZ2}, (FBZ2 - 1), MAC(K_{CH}, \{\texttt{writeFBZ2}, (FBZ2 - 1)\})$, if $FBZ2 \neq 0$
10. $CK \longrightarrow Host : \texttt{sendFBZ2}*, (FBZ2 - 1), MAC(K_{CH}, \{\texttt{sendFBZ2}*, (FBZ2 - 1)\})$
11. $Host \longrightarrow CK : \{\texttt{askRefData}\}_{K_{CH}}$
12. $CK \longrightarrow Host : \{Data, \{sha(\{Data, CK\})\}_{sk(Admin)}\}_{K_{CH}}$
13. $Host \longrightarrow Interface : \texttt{askData}$
14. $Interface \longrightarrow Host : Data$
15. $Host \longrightarrow Interface : \texttt{Ok}$

Figure 2: CBI - protocol

- In step two the chipcard sends a random ***Rsc*** (a fresh nonce), which is generated by the chipcard.
- The message in step three is encrypted with the shared key $K_{auth(CK,Host)}$ of the card ***CK*** and the host ***Host***. It contains among others a challenge (a new nonce) ***Rhost***, which is generated by the host.
- Message 4 contains a new nonce ***Rsc2***, which is generated by the chipcard.
- In the messages 5 and 6 the chipcard is asked by the host to generate a new session key (**getSessionKey**). The new session key K_{CH} is used for secure messaging in the subsequent steps. The confidentiality of the messages is guaranteed by encryption using a second shared key $K_{enc(CK,Host)}$ between the chipcard ***CK*** and the host ***Host*** and the integrity is reached by the MAC (Message Authentication Code) generation using a third shared key $K_{mac(C,Host)}$ between the protocol parties ***CK*** and ***Host***.
- Messages 7 to 10 correspond to the read- and write-steps of the error counter FBZ2.
    - In message 7 the host asks for the error counter and the chipcard sends it to the host in message 8.
    - In message 9 the error counter is reduced by one (if it is not already zero) before the biometric data check is performed.

  - Message 10 simulates the access to the decremented value of the error counter ***FBZ2*** on the chipcard.

The required integrity of messages 7 to 10 is provided by the Message Authentication Codes (MACs) using the session key K_{CH}.

- In the messages 11 the host asks for the reference data that are sent in step 12. The sent reference data are additionally digitally signed. The signature is created in the enrolment phase by a third party ***Admin*** which corresponds to a trusted administrator of the CBI system. The signature of the reference data $\{sha(\{Data, CK\})\}_{sk(Admin)}$ belongs to the secrets that are stored on the card and it is sent after being encrypted with the session key K_{CH}. It allows for the host to verify that the reference data ***Data*** belong indeed to the chipcard ***CK***.
- The messages 13 and 14 allow for the host to obtain the biometric data ***Data*** of the card holder. The host generates then a template from the biometric data and matches this with the received reference data from the chipcard.
- Message 15 occurs when the biometric data (***Data*** in message 14) match the reference data (***Data*** in message 12). Here it is assumed that the stored data and the new read data are identical. Therefore, ***OK*** is sent in the final step of the protocol.

The specification that we have presented in this chapter is translated by the VSE system into a formal representation of the protocol that is based on the before mentioned cryptographic protocol library. Starting from this specification we need to specify and verify the desired properties of the protocol.

5.2 CBI-Protocol properties

The desired protocol properties are formulated informally in chapter 4 of this paper. In the following sections we outline the formal specification of the mutual authentication between the host and the chipcard and the secrecy of the session key that is generated in step 6 of the protocol.

5.2.1 Mutual authentication

One of the most important properties of the CBI system is the mutual authentication of the chipcard and of the application host. This is reached in steps 2 to 4 of the CBI protocol (see Figure 2). The chipcard ***CK*** authenticates the host ***Host*** with the message in step 3, which contains the challenge sent in step 2 and which is encrypted by the shared key $K_{auth}(CK,Host)$ (for authentication) between the card and the host. This is formulated from the point of view of the chipcard in the following formula:

$$\begin{aligned}&\forall tr, CK, Host, Rsc, Rhost :\\ &\quad (tr \in CBI \ \wedge\ \mathtt{says}(CK, Host, Rsc) \in tr \ \wedge\\ &\quad \mathtt{gets}(CK, \{Rhost, Rsc, CK, Host\}_{K_{auth}(CK,Host)}) \in tr\\ &\quad \wedge\ CK \notin bad \ \wedge\ Host \notin bad)\\ &\quad \Rightarrow \mathtt{says}(Host, CK, \{Rhost, Rsc, CK, Host\}_{K_{auth}(CK,Host)}) \in tr\end{aligned}$$

Figure 3: Authentication of the host

The formula expresses that if we have a valid CBI trace ***tr***, i.e. a sequence of messages or protocol events that fit to the protocol description given in Figure 2, and this trace ***tr*** contains a message where the chipcard sends a random ***Rsc*** to the host and the chipcard has received a message that is encrypted with the key $K_{auth}(CK,Host)$ and that consists of four items according to the formula above, then we there has been the corresponding message in the trace tr that was sent by the host ***Host*** to the chipcard ***CK***. In proving this formula with respect to the traces tr belong to the CBI-protocol we can guarantee that the chipcard ***CK*** communicates with the host ***Host***.

Similarly, the application host ***Host*** authenticates the chipcard ***CK*** with the message in step 4, which contains the challenge ***Rhost*** sent in step 3 and which is encrypted by the shared key $K_{auth}(CK,Host)$. This is formulated from the point of view of the host in the following formula:

$$\begin{array}{l}\forall tr, Host, CK, Rhost, Rsc, Rsc2: \\ \quad (tr \in CBI \ \wedge \ \mathtt{says}(Host, CK, \{Rhost, Rsc, CK, Host\}_{K_{auth}(CK,Host)}) \in tr \\ \quad \wedge \mathtt{gets}(Host, \{Rhost, Rsc2, Host\}_{K_{auth}(CK,Host)}) \in tr \\ \quad \wedge CK \notin bad \ \wedge \ Host \notin bad) \\ \quad \Rightarrow \mathtt{says}(CK, Host, \{Rhost, Rsc2, Host\}_{K_{auth}(CK,Host)}) \in tr\end{array}$$

Figure 4: Authentication of the chipcard

In both cases we have to assume that the protocol participants are not compromised, i.e. not ***bad***.

5.2.2 Secrecy of the session key

The following formula expresses that the attacker (spy) is not able to obtain a session key K_{CH} which is sent by a chipcard ***CK*** to a host ***Host*** in an arbitrary session of the CBI protocol, which is represented as (part of) an arbitrary trace ***tr*** from the CBI set.

$$\begin{array}{l}\forall tr, CK, Host, K_{CH}, Rsc: \\ \quad (tr \in CBI \ \wedge \\ \quad \mathtt{says}(CK, Host, \{\{K_{CH}, Rsc\}_{K_{enc}(CK,Host)}, \\ \qquad\qquad MAC(K_{mac}(CK, Host), \{K_{CH}, Rsc\}_{K_{enc}(CK,Host)})\}) \in tr \\ \quad \wedge CK \notin bad \ \wedge \ Host \notin bad \ \wedge \ \mathtt{notes}(spy, \{Rsc, K_{CH}\}) \notin tr) \\ \quad \Rightarrow K_{CH} \notin analz(spies(tr))\end{array}$$

Figure 5: Secrecy of the session key

This holds under the assumptions that the protocol participants ***CK*** and ***Host*** are not compromised and that the session key K_{CH} is not revealed accidentally to the attacker. In addition to the confidentiality of the session key, the integrity of the message containing this key is required in chapter 4. This is guaranteed by the authenticity of message 6. Noteworthy, the corresponding theorem comprehends that the session key K_{CH} is generated in the same session, after the creation of the random ***Rsc***. This excludes that the message containing the session key belongs to an earlier session.

Further properties that have been formally verified are concerned with the integrity of the misuse counter and the authenticity of certain messages.

5.2.3 CBI-protocol verification

The properties we have presented in the previous sections are all verified with the help of the Verification Support Environment (VSE) system. Besides some small lemmas all the proofs are performed by structural induction on the protocol trace. Therefore, the proofs itself can only be performed in an interactive style. The proof search is supported in VSE by intelligent proof heuristics that lead in some cases to a nearly complete automatic verification. Generally an automation grade of approximately eighty percent is achieved using these heuristics.

5.3 Conclusion

The mission of the VERISOFT project is the pervasive formal verification of computer systems. Especially in the field of cryptographic protocol verification we have achieved a substantial step towards the overall aim of the VERISOFT project. The future work will be concerned with the implementation of the protocol in a C-like language and the invention and realisation of a concept to connect the implementation and the protocol verification layers. Furthermore, our experience shows that it is even possible to further automate the proof search in this area. The aim of complete automatic protocol verification even if we consider confidentiality and authentication properties seems to be achievable.

6 Bibliography

[1] D. Dolev and A. C. Yao. On security of public key protocols. In *Transactions of Information Theory*, volume 29 of *IEEE*, pages 198 – 208, 1983

[2] L. C. Paulson. The inductive approach to verifying cryptographic protocols. *Journal of Computer Security*, 6:85 – 128, 1998.

[3] Giampaolo Bella and Lawrence C. Paulson. Mechanical Proofs about a Non-Repudiation protocol. In Richard J. Boulton and Paul B. Jackson, editors, *Proc. Of TPHOL 2001, 14th international conference on Theorem Proving in Higher Order Logics*, volume 2152 of lncs, pages 91 - 104, 2001.

[4] Gunter Lassmann, Bernd Kernbaum, Matthias Schwan. Übersichtsspezifikation: Verisoft – Chipkartenbasiertes Biometrisches Identifikationssystem. Technical Report, T-Systems Nova, 2004

[5] D. Hutter, B. Langenstein, C.Sengler, J.H. Siekmann, W. Stephan, and A. Wolpers. Verification Support Environment (VSE). *High Integrity Systems*, 1(6):523 – 530, 1996

[6] G. Lassmann (Ed.), Bewertungskriterien zur Vergleichbarkeit biometischer Verfahren, TeleTrusT Working Group 6, 2002 &1998, new in 2006, http://www.teletrust.de