

On The Use of Smart Cards and Secure Terminals for Implementing a TCB for REVS Client Applications

André Zúquete, Carlos Costa

IEETA / Universidade de Aveiro, Portugal
{avz@det.ua.pt, ccosta@det.ua.pt}

Pedro Martins, Jorge Pontes

Universidade de Aveiro, Portugal
{a20352@alunos.det.ua.pt, a20325@alunos.det.ua.pt}

1. Introduction

Trusted voting clients for Internet voting systems are difficult to implement in most hosts running general-purpose operating systems, such as Windows or Linux. The complexity of these systems and the degree of freedom in their configuration makes it nearly impossible to assure the correct behavior of a local voting client application. Therefore, critical parts of client voting applications should be deployed in restricted computing environments, capable of providing a proper Trusted Computing Base (TCB).

In this document we describe the upgrade of REVS [1] voting client for using a TCB composed by a FINREAD terminal and a smart card. The FINREAD terminal provides the proper protection of user input – authentication input, voter's choices – against disclosure and a proper display of the voter's choices. The smart card provides the proper protection of the voter's authentication credentials – asymmetric key pair and voter's signatures.

2. Weaknesses in the REVS Voter Module

The REVS Voter Module is an ordinary Java application conceived to work on a trusted host. By trusted host we mean a host not interfering with the privacy of voters or with the accuracy of the election. Examples of actions affecting either privacy or accuracy are:

- Stealing voters' credentials – identification and passwords (impersonation issue).
- Using otherwise the identification of the voters and their votes (anonymity issue).
- Sending false votes instead of the ones expressed by voters (accuracy issue).
- All these actions can be engaged by a tampered Voter Module or by other applications running on the same host (e.g., key loggers).
- Thus, to improve the privacy and accuracy of the Voter Module is required to:
- Protect the input and usage of voters' credentials from stealing.
- Protect the input of voter's choices for ballots.
- Protect all sensitive information related with the vote (e.g., the bit commitment).

To enforce such protection we need to execute all these actions in a TCB. This requires moving part of the Voter Module to a TCB, in order to protect all critical input/output operations and all permanent or temporary, personal data involved in the voting process.

3. New REVS TCB Client architecture

The main goal was to migrate part of the Voter Module into a TCB, providing the required protection of private data and preventing tampering of votes. For implementing the TCB we chose a trustable smart card/smart-terminal environment, namely a FINREAD device and a smart card. These two components are described in more detail in the two following sections.

We considered that this set of components, a FINREAD terminal and a smart card, was suitable for assuring the required level of security for the new REVS Voter Module: the asymmetric key pair for voter authentication is protected by the smart card, the PIN to use it is handled by a Finlet and using the FINREAD pin-pad, voting options are presented in the FINREAD display and introduced using the FINREAD pin-pad, and critical data associated with votes is stored inside the FINREAD protected memory.

3.1. Smart cards

One good way to store sensitive information, such as personal details or cryptographic keys, is through the use of smart cards [2]. There are various ways to use this technology [3], but when correctly combined with the use of other security technologies like PKC (Public Key Cryptography) and biometrics, it strongly enforces effective access control through personal identification and/or authentication [4].

There are various types of smart cards, but the most interesting in terms of security for implementing our TCB are those that have an embedded microprocessor capable of executing strong cryptographic algorithms on the card itself, thus not requiring protected information to be moved from the card. The use of those storage and processing devices, with native cryptographic capabilities and protected by user-provided secrets, can improve the level of security to the whole system. Smart cards are an ideal solution for PKC authentication: the private key lies in a secure, tamper resistance storage, a “second factor” authentication must be introduced to unlock it (the PIN), and a crypto accelerator provides cryptographic hardware operations, such as key pair generation and digital signature generation/verification. We have been working with two distinct types of smart cards, the Schlumberger (Axalto) Cryptoflex (16kbytes) [5] and the Javacard Cyberflex e-gate (32 bytes) [6].

PKC-based authentication systems, using asymmetric key pairs, are more secure than password or PIN based systems because there is no shared knowledge of the secret between transaction intervenient. The private key needs only to be known in one side of transaction. If this key is placed in a crypto smart card, a PIN protects its access and it never leaves the smart card at any circumstances. The operations involving its use are performed directly inside the card, which disables Trojan horses to spy the secret key on computers.

By providing a voter with a PKC digital credential, supported by a smart card, we can enforce strong voter authentication and protected, authentication signatures performed inside the smart card. Nevertheless, smart cards cannot control the data provided for signing. Therefore, they cannot prevent the adulteration of votes in client PC environment (c.f. Figure 1). This issue is handled by using a trusted card reader.

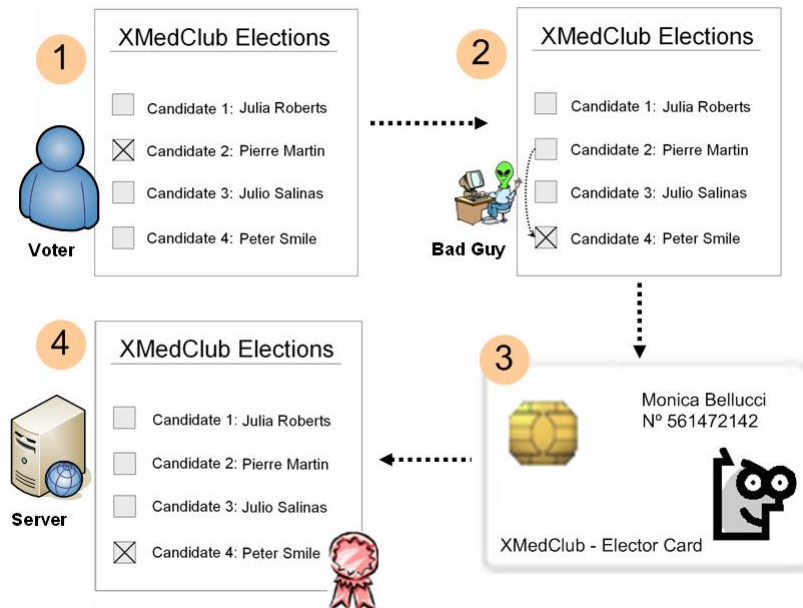


Figure 1 – Vote Adulteration in the Client PC Environment

3.2. OMINEY CardMan Trust FINREAD

As previously explained, the usage of a smart card does not prevent the possibility of vote adulteration in the client PC environment. For instance, the REVS java-based client module could be cracked or “maliciously cloned” without being detected by the voter.

Steps 1 and 2 in Figure 1 are actually executed in the PC environment, representing a potential risk to the voting process. After analyzing some possibilities to enforce the voting environment, we decide to separate this component from the Voter Module running on the PC. The evaluation of several tamperproof devices conducted us to the FINREAD smart terminal as a promising option.

Therefore, our TCB was built around a Java-based smart card reader, namely the Omnikey CardMan Trust FINREAD [7]. This is an ISO 7816 and EMV 3.1.1 compliant smart card reader, which also adopts the latest FINREAD specifications [8].

The FINREAD platform adopts and extends the Java applet technology too. In a FINREAD environment applets are called “Finlets”. Consequently, FINREAD terminals are provided with internal memory (1MB) capable of hosting different software modules and a JVM (Java Virtual Machine) to execute them. The microprocessor used by the device is a 32 bit ARM 7.

Only Java applications signed by the appropriate entity may be downloaded into the FINREAD terminal. The FINREAD specification defines secure distribution of Finlets by public key based code-signing mechanisms. When a Finlet is activated, the terminal operates in secure mode, i.e. the access to the smart card is always mediated by a terminal Finlet [9].

Our FINREAD device has a small LCD display, containing 4 lines of 20 characters, as well as a pin-pad with 16 keys for user input interaction. Finlets running on the device can control both sensitive transactions with the smart card and interactions with the user (voter) using the device’s display and pin-pad.

3.3. REVS FINREAD Implementation

As we have already highlighted, the FINREAD capacities not only allowed us to prevent the PIN capture outside the reader, but also to display the ballot and retrieve the user vote.

Because the REVS platform is fully implemented in Java, it was relatively simple to isolate the software parts that should be implemented inside the terminal environment as a Finlet.

Like all blind signatures based voting [10], REVS is ballot independent. The system flexibility is largely due to the use of XML for describing ballots. This way, the ballot can be easily transferred to the terminal.

Due to the FINREAD display limitation, it was decided display simultaneously the ballot in both the PC monitor and the LCD terminal (c.f. Figure 2). Of course, the terminal ballot presentation is graphically less rich than in the PC display. However, the idea is to validate the PC display. If some inconsistency appears between both displays, then the software of the Voter Module may be compromised.

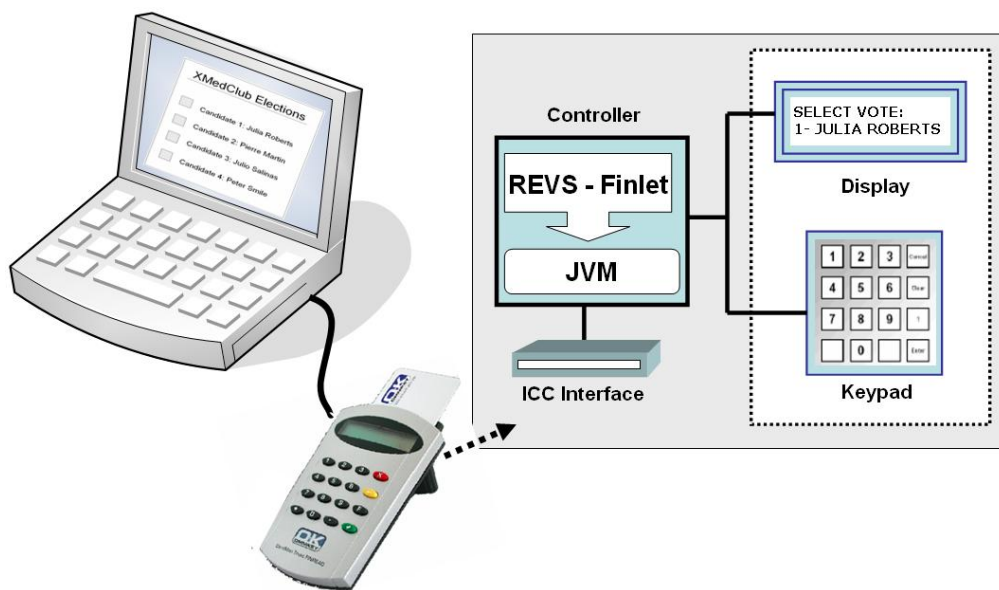


Figure 2 – REVS Voter Module with FINREAD Implementation

After the voter performed his action, i.e. introduced voting choices on the keypad terminal, the vote is stored inside the terminal and used in the rest of the voting protocol. First, the terminal generates a bit commitment, hashes it with the vote, generates a blinding factor, blinds the hash with it and digitally signs the result with the voter's private key. The PIN that grants the access the signing process with the private key, which will run inside a smart card, is equally introduced in the terminal keypad when local signing operations are required.

The signed, blinded hash of the vote is then sent, together with the voter identification, to the Administrators for get their authorization signature. This is performed by the Voter Module part that runs in the PC, which is responsible all for the network interaction with REVS electoral servers. The results returned by Administrators, which should include a signature of blinded hash of the vote, are sent to the terminal for validation. After getting the required number of valid Administrators' signatures, the terminal produces the vote submission package and sends it to the Voter Module part that runs in the PC for sending it to all Anonymizers/Counters.

Along all this process, the Voter Module that runs in the PC has no direct access to the voter's authentication credentials – smart card PIN – neither to the vote – the vote submission package can only be open at the end of the election. Furthermore, the identification of voters sent to Administrators, when getting their authorization signature, is previously encrypted by the terminal with the Administrators' public keys.

4. Implementation details

The first step in the development of the new architecture for the REVS Voter Module was to introduce in REVS a new way of authenticating voters: with asymmetric key pairs. This first step was accomplished using IAIK-JCE (IAIK Java Cryptography Extension) [11] and ordinary smart card readers for interacting with smart cards.

The second step was to separate the Voter Module in two different parts: a PC part, responsible for interacting with the Electoral Servers and for displaying the rich image of ballots, and a Finlet part responsible for all the critical aspects of voting: (i) input of authentication credentials; (ii) input and storage of votes and other related information (e.g., the bit commitment); and (iii) presentation of voter's choices.

Regardless the technologic difficulties associated to the Finlet development, we faced some other development problems while handling the communication between all Voter Module entities: the PC application, the Finlet and the smart card. First, all communication functions between the Finlet and the smart card was implemented using hexadecimal ISO7618 APDU commands [5, 12]. Second, the communication between the PC application and the Finlet had to be supported by a C/C++ API. So, we add to develop also a Java interface, on top of the C/C++ one, to allow the pretended Java-to-Java communication between the PC application and the Finlet.

5. Conclusions

In this document we described a new architecture for the REVS Voter Module using a TCB formed by a smart card and an OMNIKEY FINREAD reader. To adopt the TCB we had also to modify the authentication process of voters in REVS – from username/password pair into asymmetric key pairs. The final system provides a robust PKC-based authentication of voters and protects all critical actions and data of voters, during the vote process, with tamper-proof devices. We believe that the proposed TCB for the Voter Module could be integrated in other e-voting platforms.

From a financial perspective, namely the cost associated to the terminal acquisition, the system is perfectly affordable, since the FINREAD terminals are not very expensive (~140€). Moreover, the device can be used on other smart card based payment and e-commerce transactions on open networks.

6. References

- [1] Joaquim, R., A. Zúquete, and P. Ferreira, *REVS – A Robust Electronic Voting System*. IADIS Int. Journal of WWW/Internet, December 2003. 1(2).
- [2] Marvie, R., Pellegrini, M. et al. *Value-added Services: How to Benefit from Smart Cards*. in *GDC2000*. 2000. Montpellier, France.
- [3] Gobioff, H., S. Smith, et al. *Smart Cards In Hostile Environments*. in *Proceedings of The Second USENIX Workshop on Electronic Commerce*. 1996. Oakland, U.S.A.
- [4] Hachez, G., F. Koeune, and J. Quisquater, *Biometrics, Access Control, Smart Cards: A Not So Simple Combination*, in *Security Focus Magazine*. 2001 October.
- [5] Schlumberger, ed. *Cryptoflex Programmer's Guide*. rev1 ed. Cyberflex Access Software Development Kit 4.3. 2002, Schlumberger.
- [6] SchSDK, ed. *Cyberflex Access Cards Programmer's Guide*. rev1 ed. Cyberflex Access Software Development Kit 4.3. 2002, Schlumberger.
- [7] Onikey, A., *OMNIKEY FINREAD SDK Manual, v1.22.3*. 2005.

- [8] FINREAD, C., *FINREAD Technical Specifications Part1-8*, C. WS/FINREAD, Editor. 2003.
http://www.finread.com/pages/about/specifications/01_specifications.html.
- [9] Kurt, S. and Z. Harald, *FINREAD Whitepaper, rev 1.0*. OMNIKEY AG. 2003.
- [10] Fujioka, A., T. Okamoto, and K. Ohta, *A Practical Secret Voting Scheme for Large Scale Elections*, in *Advances in Cryptology -- AUSCRYPT '92 Proc. (LNCS 718)*. 1992: Queensland, Australia.
- [11] IAIK-JCE, *IAIK-JCE 3.14 API Documentation*. Stiftung SIC. 2005.
http://javadoc.iaik.tugraz.at/iaik_jce/current/index.html.
- [12] ISO 7816, P., *ISO 7816 Identification Cards - Integrated circuit(s) cards and terminals. Part 4: Inter-industry Commands for Interchange*. 1997.