# Electronic Voting by Means of Digital Terrestrial Television: the Infrastructure, Security Issues and a Real Test-bed.

Roberto Caldelli, Rudy Becarelli, Francesco Filippini, Francesco Picchioni, and Riccardo Giorgetti

MICC, University of Florence, Florence, Italy
{roberto.caldelli,rudy.becarelli,francesco.filippini,
francesco.picchioni}@unifi.it,giorgetti@lci.det.unifi.it
http://lci.micc.unif.it/index.php

**Abstract.** Electronic voting has been largely studied in different forms and applications. Typical objectives of electronic voting are to enhance security and to grant easy accessibility. Security can be pursued by means of several strategies oriented to secrete the vote, to check the voter identity, to decouple the voter from his choice and to allow the ballot to be audited. On the other hand, accessibility too can be greatly improved by providing the opportunity to vote remotely or by using voting machines, located at polling stations, equipped with appropriate interfaces for disabled people.

In this paper a Digital Terrestrial Television (DTT) based voting system is presented. This kind of electronic voting technology allows disabled users (especially people with mobility problems), but not only, to cast their vote from home and, above all, by using common well-known devices. In fact, the needed basic equipment are a TV set, a Set Top Box (STB) with its remote control and a telephone line. The complete infrastructure consists of an MHP (Multimedia Home Platform) application that acts as a client application; a server application that acts as a network/counting server for e-voting; and a security protocol based on asymmetric key encryption to ensure authentication and secrecy of the vote. The MHP application is broadcasted by a certified (e.g. national) TV channel that grants its originality. The user needs a smart card issued by a national authority (e.g. the electronic identity card) to authenticate himself to the application and to sign the encrypted ballot to send to the server application. The voter can simply browse the application, displayed on his TV screen, by acting on the STB remote control. The server application is in charge to verify user identity, to gather and store user's encrypted ballots and finally to count votes. The communication between the client application and the server takes place by means of a secured channel (using HTTPS), established over the common telephone line, while the voting operations are secured with the help of asymmetric keys encryption. The whole infrastructure has been proven in laboratory tests and also in a public demonstration for USA Presidential Election on 2008 November $4^{th}$.

**Key words:** DTT, MHP, e-voting, e-democracy, digital divide, disabled people accessibility

## 1 Introduction

Electronic, mechanical, or electromechanical voting are nowadays form of voting commonly accepted in various countries worldwide. Despite of this diffusion, these voting techniques have been always criticized for many reasons. Typical critics are related to the possibility, for the voter, to audit his vote or have some kind of control on the underlying mechanism during the polling phase that is when the vote is cast. This kind of frights arise naturally from the intrinsic complexity and/or from the opacity of the mechanism itself and can be amplified by a justified sense of caution. These critics highlight that one of the most important open issues is security and in particular how to achieve or, eventually, increase it with respect to a traditional voting scenario. Electronic voting can, besides, enhance the accessibility to vote even for people living outside the country of origin or for disabled persons. Electronic voting, that is widely exploited, offers mainly two different approaches to solve both security and accessibility issues. The first approach aims to substitute traditional voting form in the polling stations with electronic machines trying to match the requirements for accessibility and security. The second tries to solve the accessibility issue by making people vote through web based or broadcasted applications, not disregarding the security of the communication channel that must be used in this case. Electronic voting machines in polling stations, named DRE (Direct Recording Electronic) voting systems [1][2], have been widely used especially after US presidential elections in 2000 when mechanical punching machines led to a large number of invalid ballots. Actually, despite of the confidence given by citizens to such a solution, DRE machines are very sensitive to various kind of attacks, as detailed in [11][6]. In order to improve the security of DREs in terms of capability of performing an audit by the user, secrecy of vote, and relative independence from technical flaws the *receipt* approach has been proposed. As explained in [3][4][7][9], the central idea is to give the user an encrypted receipt which can be used to audit the vote as an evidence that the vote has been cast and that can be seen like the ballot itself, since the user's choice is encrypted. Typically these systems, implemented as electronic or manual, give as a result of the voting operation two distinct ballots. After the voting phase (this is part of the security mechanism) the user is asked to destroy one of these ballot, chosen by himself, and scan the other one. The scanned ballots are sent to a server that acts like a ballot repository. Since both the ballots are encrypted and only the combination of the two can give some chance of recovering the vote, at the end of the operation the voter owns an *encrypted receipt*. The actual ballot is readable only with the help of some codes owned by the trusted authority that controls the voting operation (e.g. the Ministry of Internal Affairs). To allow the user to audit his vote, every encrypted ballot is identified by a readable unique number. The number, that is

decoupled from the user's identity, can be used to audit the ballot via web with the help of a specific web application.

Recently electronic voting has proposed a new approach based on web applications allowing user to vote from worldwide. One of the first experiment in this direction has been SERVE (Secure Electronic Registration and Voting Experiment) [12], a web based application developed for military personnel deployed overseas. Its security is mostly based upon asymmetric key encryption and HTTPS connection. An analysis of possible security flaws can be read in [5]. Other similar systems have been developed starting from the SERVE experience, as for the Estonian e-voting system used during political elections in 2005 [10]. The SERVE security architecture and the Estonian experiment have been used as a reference for the implementation described in the proposed work. This paper is not aimed at presenting a new technology for voting security, but a new architecture whose purpose is to provide an usable voting system in order to allow, in particular, people with mobility problems to vote.

The architecture presented is based on the DTT (Digital Terrestrial Television) infrastructure and on the use of Java interactive applications running on a common television by means of a simple decoder (Set-Top-Box, STB). DTT is worldwide spread as a family of different standards for digital TV (DVB in Europe, Australia and Asia [13]; ATSC in North America [15]; ISDB in Japan and Brazil [16]). Specifically the proposed architecture is built upon DVB-T/MHP (Multimedia Home Platform) [14] technology but it could be extended to the other standards. MHP applications, named Xlets, are similar to Applets for the web, both in structure and life-cycle; they are broadcasted by multiplexing them with the digital TV transport stream and can be accessed on the television screen. Such application, through the STB and a return channel (e.g. the telephone line), can allow a *one-to-one* interaction between the user and the server side.

The proposed system is based on two applications: an MHP application running on the client side and a server application running on the server side. The first, broadcasted by the authorized TV channel (trusted authority), when downloaded on the STB, permits to authenticate the user, and to send the digital ballot to the server application. On the other side the server application is in charge to check the user ID and eventually allows the user to vote. Then the server application manages the secret ballot and decouple the vote from the voter.

The whole system has been tested in laboratory emulating a real world situation by broadcasting the TV signal and checking the functionalities provided by various kinds of STBs present on the market.

During the phase of development an usability study has been performed to make on interface easy to use and understand even for users subject to "digital divide". Also the entire architecture, in an ad-hoc setup, has been presented and tested, as a demonstration, on $4^{th}$ of November 2008 during the latest USA Presidential Election. A convention, organized by the TAA (Tuscan American

Association [20]), was held in Florence (Italy) to wait for the actual result of the elections.

The paper is organized as follows: an exhaustive description of the system is given in section 2. Security issues are debated in section 3, while usability aspects are discussed in section 4. The experience faced during the "Election Night Event" is presented in section 5 and section 6 concludes the paper.

## 2 System Description

This architecture for electronic voting was basically designed as an aid to people with mobility problems and not only. The idea behind this infrastructure is to provide the chance to vote remotely by using very *well-known* devices, like a television and a remote control, in a domestic environment. In addition to that, the simplicity of the required equipment (i.e. a STB for DTT, a telephone line connection and a TV screen) allows to easy access to vote not only persons with computer science skills but also people without a personal computer, living in needy areas of the country.

The infrastructure is based on the DTT (Digital Terrestrial Television) platform and on the use of Java interactive applications called Xlets. Such applications run on common television through the use of a simple decoder called Set-Top-Box (STB). The standard that in Europe rules this architecture is based on DVB-T/MHP (Multimedia Home Platform) technology. DVB-T defines the caracteristics of the signal transmission, while MHP covers the part related to the interactive applications. MHP applications in DTT are similar to Java Applet for the web; they are broadcasted by multiplexing them with the digital TV transport stream and loaded in the STB where runs a Java Virtual Machine with MHP stack.

The whole platform is reported in Figure 1 . The MHP application certified by the authority which is responsible for voting operations (e.g. the Ministry of Internal Affairs) is broadcasted by the TV channel/channels authorized to distribute the service (e.g. the national TV broadcaster like RAI in Italy, BBC in UK, CNN in USA, etc.). The user by selecting the channel can download the application and watch it on the television screen. Being equipped with the appropriate smart card (e.g.Electronic Identity Card - EIC), he can proceed to the authentication phase and then to the vote by using the return channel on the PSTN connection. It is easy to understand that the structure for this kind of electronic vote is basically composed by two main parts: the "Client" (MHP application) and the "Server" (Server-side application) which communicate each other through an HTTPS connection over the Return Channel. Decoders on the market have to types of network interface: modem 56k or Ethernet.

The Client application is a Xlet (Java Application) broadcasted together with TV transport stream that a voting user can launch from the remote control of STB. STBs have a slot for electronic cards, typically used for pay TV or other interactive services, where user can insert his EIC to be authenticated by the system. Each smartcard has a secret numerical PIN that the user must enter by

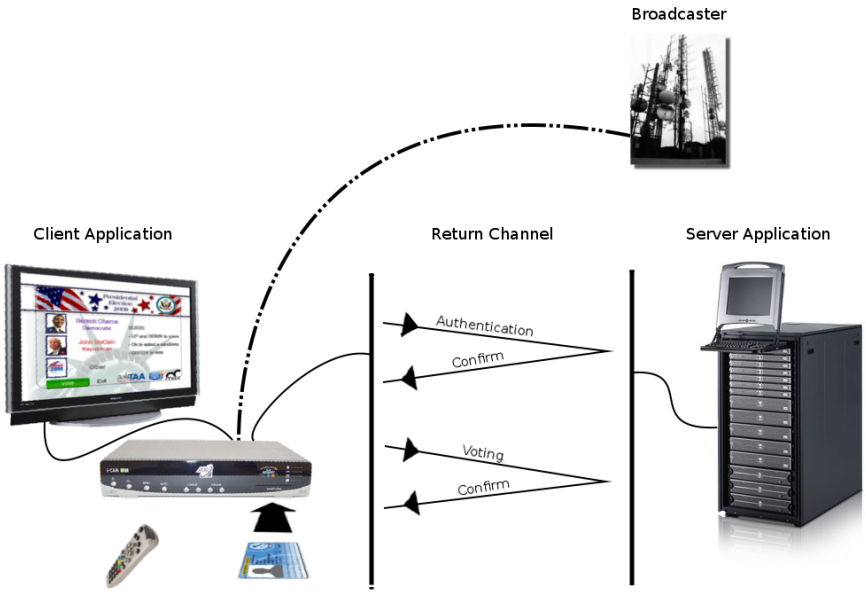


**Fig. 1.** System infrastructure

using the Remote Control. Through the remote control it is possible to easily navigate the application; instructions are provided on the screen too. When the user sends his vote, the application activates the Return Channel and establishes a secure connection with the server via HTTPS protocol.

The Server application has two principal roles: user authentication and vote registration. Authentication is done by receiving the identifying data contained in the Smart-Card by the return channel, validating the public certificate (X.509) and verifying if the user has already voted. When the user has been authenticated, he can select his favorite candidate or party and his vote is sent to the server that records user's preference.

The following paragraphs illustrate in detail each part of the proposed system.

### 2.1 Set-Top-Box, Smart Card and Client Application

The client side is essentially composed by four parts: STB (Set-Top-Box) and its remote control, Smart Card (like Electronic Identity Card), Return Channel and MHP application (Java Xlet). Let's see in detail these components. STB is a decoder that transcodes the broadcasted digital television signal into a classical analog one by following the standard DVB-T (Digital Video Broadcasting - Terrestrial). It is essential that these devices implement the MHP stack with

the appropriate Java virtual machine, without that Java applications can not be executed (common STBs on the market satisfy that). The user can launch the application by the remote control, generally clicking on the *APP* button (it depends on models). Another prerequisite is that the STBs are equipped with the SATSA (Security And Trust Services API) [17] libraries to interact with this type of smart card (again common STBs on the market satisfy that). Once launched, the application needs to authenticate the user and at this stage the smart card plays an essential role. There are many types of cards used for pay-TV such as Irdeto® [18] and Nagravision® [19]. This application supports all SmartCards that refers to ISO 7816 standard [21] like EIC (Electronic Identity Card) and NCS (National Services Card). Application is able to authenticate the user after he has introduced his card in the slot of the decoder. For security two checks are performed: one directly with the request of the PIN (Personal Identification Number), the other, once typed the code, using a secure cable connection with the server side exchanging the certificate (X.509) of the card using a system of asymmetric key encryption. Once the user has chosen his candidate, the digital ballot is encrypted and sent to the server following a protocol described in section 3. Consumer electronic STBs can access the Internet or a local network. Usually they have two types of interface: 56k modem or Ethernet. In the first case the connection is made using an ISP (Internet Service Provider) which the modem connects to. The parameters for the connection (i.e. ISP telephone number, user name, password) can be passed in two ways: either directly by setting the STB through its communication interface, or embedded inside the MHP application itself. After opening the return channel, the application can make requests to the normal internet network through protocols like http, ftp, https, pop etc. For save of conciseness, the MHP application features are not described here but in Section 4.

### 2.2 Server Application

On the server side, a server application (Servlet) exists which is in charge for the authentication phase and for vote storaging. It receives the public certificate of the smart card, extracts the personal data (i.e. an unique Revenue Service Code) and verifies that the user has not already voted. If the user has already voted the server returns a message stating that the session is going to be closed. On the contrary if voting is admitted it gets and records the user's encrypted ballot decoupling it from the voter. As previously mentioned, all transactions between client and server are made using the return channel within a HTTPS connection (Hypertext Transfer Protocol over Secure Socket Layer).

## 3 Security Issues

In this section an implementation of the security protocol used to hide sensible data and authenticate the user is drawn. The architecture presented in the section 2 consists of two applications (a client-side and a server-side one) that

exchange data along a secure channel using HTTPS and asymmetrical key encryption. In this scenario two main security issues arise: application security and communication security.

Application security is easily achievable by trusting two main authorities: the broadcaster of the MHP application and the server side application warrantor. The broadcaster must ensure that the application, acting like a client, is the original one. The server application instead must ensure that votes coming from the client are sent by a certain person and that, above all, the vote is decoupled from the voter. The adopted model of communication security, instead, makes use of an encrypted channel and two pairs of asymmetrical keys, one for the server and one for the client. The first necessary step is to build a secure channel between the client and the server. In order to do that an encrypted HTTPS connection is established adopting SSL (Secure Socket Layer) and TLS (Transport Layer Security) technologies. Typical attacks to the secure channel (e.g. Man-In-The-Middle attacks operated by poisoning the ARP - Address Resolution Protocol etc.) can be neutralized with an appropriate *double check* of the SSL certificate. Once the secure connection is established, the protocol makes use of the client and the server keys to encrypt the user's vote. The *Server* pair of keys are $PK_S$ (public) and $SK_S$ (secret). This pair is used to encrypt and make anonymous the vote of the user. The *Client* instead has a pair of keys addressed as $PK_C$ and $SK_C$. These keys are used to authenticate the user and sign the data exchanged along the secure channel. The exchange of public keys is done as follows: $PK_S$ is broadcasted directly with the MHP application, $PK_C$ is sent from the client through the SmartCard certificate.

The encryption mechanism used in a voting session is pictured in Figure 2 and detailed hereafter. The user logs into the client application by inserting his smartcard into the slot and by entering the secure PIN code to unlock the card. After this client side authentication, the user digital certificate is sent via HTTPS to the server that checks whether the user is allowed to cast the vote or not by extracting from the digital certificate the unique public data for identifying the voter (e.g. the Revenue Service Code). If the user is allowed to vote, the $PK_C$ key is extracted from the digital certificate and stored in a session object ready to be used for further operations. A message, containing the positive or negative acknowledgement, is sent back to the client through the established HTTPS channel. The message is signed with $PK_S$ in order to ensure the user that the message sender is the server. If a positive acknowledgement is sent, the user is invited by the client application to cast his vote. Once his choice has been expressed, the client application generates a random number $r$, encrypts the message with the server public key ($PK_S$) and signs it with the private user key ($SK_C$) as follows: $SIGN_{SK_C}(ENC_{PK_S}(v, r))$, where $v$ is a unique identifier for the user's choice and $r$ is the random number. Random number $r$ is concatenated with the vote in order to generate each time a different encrypted ballot to improve security, otherwise the types of encrypted ballots would be as much as the number of eligible candidates/parties. $ENC_{PK_S}(v, r)$, the encrypted ballot, is also stored as a *receipt* for checking back the server

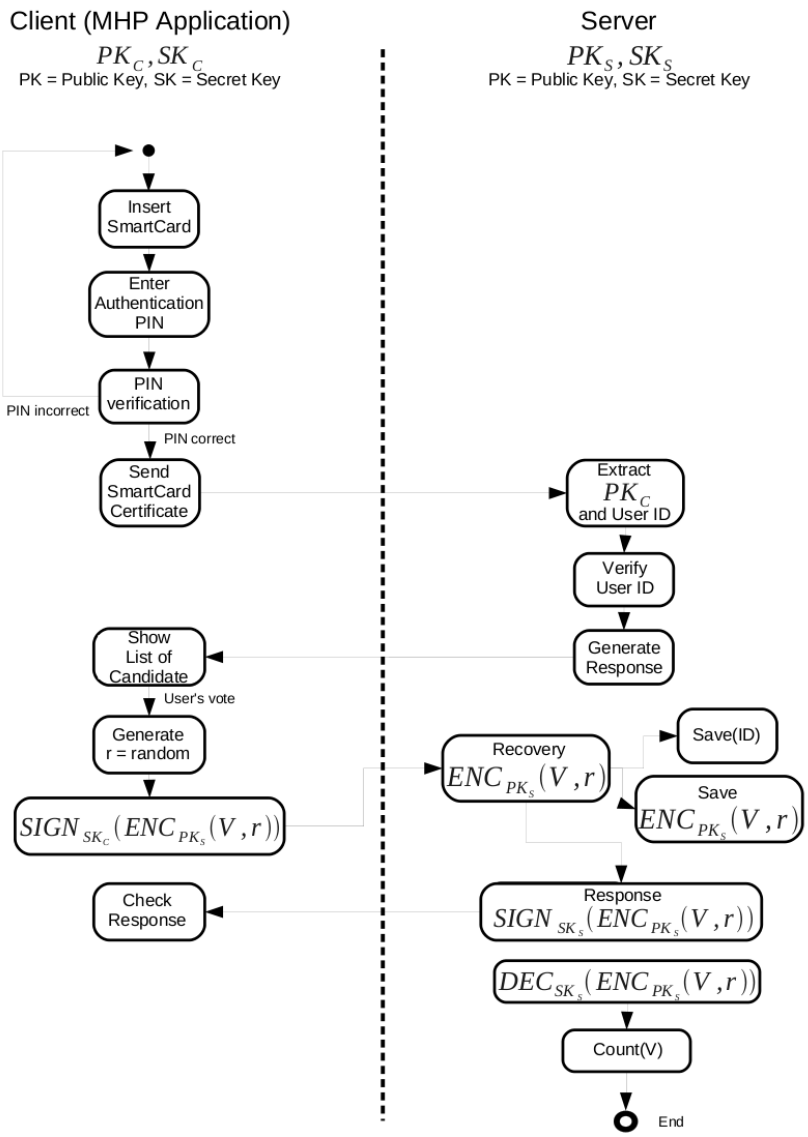


**Fig. 2.** Security mechanism

answer while the entire message is sent to the server. The client application stands by for a positive acknowledgement that will close the transaction. When the server gets the message it is confident that the message comes from the user (that is granted by the signature) and that none but its own is able to read the content of the inner envelope; not even the user who does not hold the necessary server private key $SK_S$. The server, verified the sender identity, obtains an encrypted ballot $ENC_{PK_S}(v, r)$ containing the user's choice. The encrypted ballot is then stored in a database, ready to be balloted. In this phase the voter's identity is decoupled from the ballot as it happens in mail voting systems. The security model implemented is then, at least, the one granted with mail voting. If all these phases terminate with success, the server sends back to the client a signed answer containing the encrypted ballot. The answer is signed with the server secret key $SK_S$. Once the client receives this message is able to recover the anonymous envelope and compares it with the previously stored copy. If the comparison is positive the transaction closes. Once the whole voting operations are closed, the server side authority, who is responsible to manage the votes, can disclose every single encrypted ballot using the private server key and taking into account the $r$ random number, since $r$ is separated from vote by a sequence of known characters. Then votes are computed and the final results are published.

## 4 System Characteristics: an analysis

In this section an usability analysis and an explanation of some of the advantages and drawbacks of the presented system is carried out.

### 4.1 Usability

During the phase of client application development an usability study has been performed. It is important to underline that this system uses a structure and devices well-known to everybody. The application is accessed through a common TV and the remote control of STB, devices already used by millions of people.

To simplify as much as possible the operations which the user must perform, the use of buttons on the remote control was limited. The application can be used only by resorting at the arrow keys (*UP/DOWN* and *LEFT/RIGHT*) and by pressing *OK* button. Usually MHP applications exploit the four *coloured* buttons (red, green, yellow and blue), making more complicated the use of the interfaces. The interface consists of a sequence of screens thought to be easy to use and understand even for users subject to "digital divide". The application however fully comes in help of people by using a written guide (see Figure 3(a)) to give the user some basic hint for interacting with the interface in a correct way (audio-guide messages have also been implemented). The most difficult step for the user can be the typing of the *PIN*, but because it is simply numerical the operation is reduced to just press the code on the numeric keypad on the remote control, typically used to change TV channels, and then confirm with


(a)


(b)

**Fig. 3.** MHP application screenshots

the *OK* button. It is important to highlight that at each step, in case of an error by the user, it is possible to go back. Both for the *PIN* and for any selection of the candidate, the pressure of the *OK* button displays a *confirmation dialog* (see Figure 3(b)). Without this check is not possible to proceed to the next step. It was also used a very specific criterion in the choice of using the arrow keys. Vertical arrows (*UP* and *DOWN* buttons) are used only for the selection of candidates/parties (see Figure 3(a)). These have the role to move the focus on the respective boxes of the candidates/parties. While the horizontal arrows (*LEFT* and *RIGHT* button) are used only in the *confirmation dialog* to move the focus on *YES* (confirm) or *NO* (cancel) buttons (see Figure 3(b)).

### 4.2 Advantages and drawbacks

In this subsection advantages and drawbacks of the system are outlined and comparisons with classical voting systems and other electronic voting forms are carried out. Basically the proposed system allows the user to vote from home like the classical mail voting. Additionally the system encrypts the digital ballot and the communication towards the server with asymmetric key encryption. Besides the use of a smart card and a Personal Identification Number simplifies the authentication phase increasing at the same time the security level. A possible critic relies on the fact that such a system is not able to recognize whether a voter has actually cast his vote, because it's possible that the smart card and the PIN are used by someone else but the owner. It can be noted that the same critic can be arisen to a well known and accepted system like the mail voting. Another advantage of this system is the fact that the MHP application is broadcasted (i.e. with a *push* technology) to everyone making the Man In The Middle attack almost impossible. This kind of intrinsic security is given by the nature of the on-the-air broadcasting that can be modelled as a *one-to-many* scheme. Other possible drawbacks can come out from the structure of the system based on the client-server paradigm. The most attackable part of the system is the return channel that is exposed to MITM attacks. Similar attacks can be minimized by using HTTPS and performing a double security check from the client as requested by the protocol itself. Another clear advantage is the fact that the digital ballot can not be repudiated and the secrecy of the ballot itself ensured by the use of asymmetric cryptography properly implemented within the system. Besides, the use of the application does not affect the user in any way by demanding some particular cost. No extra hardware is needed to make the application run. The only things needed is a commercial STB and at least a PSTN connection.

## 5 A real test bed (*Election Night*)

The entire architecture, in an ad-hoc setup, has been presented and tested, as a demonstration, on $4^{th}$ of November 2008 during the latest USA Presidential Elections. A convention named "Election Night", organized by the TAA (Tuscan

American Association, http://www.toscanausa.org/Election_EN.asp) and sponsored by Tuscany Region (http://www.regione.toscana.it), was held in Florence (Italy) to wait for the actual result of the elections.

The architecture previously presented has been modified in order to simplify the voting phase during the *Election Night*. These changes have been necessary since a large number of people (about one thousand persons) were expected to cast their vote in few polling booths. The polling booths were equipped with a STB receiving the digital signal directly from a PC running a DVB-T/MHP object carousel and acting as a broadcaster. This PC used an appropriate hardware interface to modulate the digital signal to RF (Radio-Frequency) and sent it, via coaxial cable, to all STBs. These configuration has been adopted since no real broadcaster were available that night. Besides, since the STBs in use were fitted with the necessary connectors and interfaces, another PC were connected via Ethernet to each STBs to act like the server side machine as described above.

People who wanted to try the system, had to ask a hostess a smartcard with an associated dummy identity used during the tests. The voter, then, had to enter the polling booth and insert the card into the STB slot. The authentication phase, in this scenario, has been skipped since the smart card available were few and in order to speed up all the operations. In particular, the voter was not asked to unlock the card by entering the PIN code. The only security operation performed by the application was to check the presence of the card inside the slot to authorize the voter to go on. The following phases of voting took place as described in the previous chapters. During the night some exit polls has been simulated by projecting partial votes to a wall screen. People who took part in the convention expressed their appreciation for the initiative. The large number of voters clearly demonstrates that such a system matches the minimal usability requirements and that is easily comprehensible. A photograph of some tests can be seen at http://www.toscanausa.org/gallery/images/dnvye__C0B1305.jpg.

## 6 Conclusions

The system presented in this paper is mainly aimed to reduce the "digital divide" in e-voting by exploiting DTT technology. People having mobility problems or leaving in weedy areas can vote by means of a STB and a PSTN connection even from their home. The security is ensured using HTTPS connection and an appropriate encryption of sensible data; in this case the user's vote. The encryption is operated by means of two pairs of asymmetrical keys owned respectively by the server and by the user's smartcard. The security protocol is similar to the ones designed for web based e-voting applications like SERVE [12] and the estonian e-voting system [10]. The architecture is composed by a client application running on a STB and broadcasted by a trusted TV channel that is responsible to preserve its integrity. This application is able to communicate, via PSTN or digital connection, with a server that authenticates the user and eventually authorizes him to vote. The server stores his cyphered ballot, decoupling it from

the user identity, until the scrutiny phase. The scrutiny takes place using a secret key used to decypher the encrypted ballots. This system is easily extending to other platform describe in the section 1 and not only. For the characteristics of the MHP standard, it is possible to reproduce this architecture in other system like: DVB-S (Satellite), DVB-C (Cable) or DVB-H (Handheld).

## References


1. Federal Election Commission, Voting system Standards, Performance Standards, Introduction; http://www.fec.gov/agenda/agendas2001/mtgdoc01-62/v1/v1s1.htm
2. Federal Election Commission, Voting System Standard; http://www.fec.gov/agenda/agendas2001/mtgdoc01-62/overview.htm
3. Chaum, D., Essex, A., Carback, R., Clark, J., Popoveniuc, S., Sherman, A., Vora, P., Scantegrity: End-to-End Voter-Verifiable Optical- Scan Voting. Security & Privacy, IEEE Volume 6, Issue 3, May-June 2008 Page(s):40 - 46.
4. Chaum, D., Secret-ballot receipts: True voter-verifiable elections. Security & Privacy, IEEE Volume 2, Issue 1, Jan.-Feb. 2004 Page(s):38 - 47
5. D. Jefferson, A.D. Rubin, B. Simons, D. Wagner. A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE). 2004. http://www.servesecurityreport.org/. 21.01.2007
6. T. Kohno, A. Stubblefield, A. D. Rubin, D. S. Wallach. Analysis of an Electronic Voting System. 2004. http://avirubin.com/vote.pdf. 21.01.2007.
7. Essex et al., The Punchscan Voting System: Vo-Comp Competition Submission, Proc. 1st Univ. Voting Systems Competition (VoComp), 2007; http://punchscan.org/vocomp/PunchscanVocompSubmission.pdf.
8. S. Garera and A.D. Rubin, An Independent Audit Framework for Software Dependent Voting Systems. Proc. 14th ACM Conf. Computer and Comm. Security (CCS 07), ACM Press, 2007, pp. 256-265.
9. D. Chaum, R. Carback, J. Clark, A. Essex, S. Popoveniuc, R. Rivest, P. Ryan, E. Shen, A. Sherman, Scantegrity II: End-to-End Verifiability for Optical Scan Election Systems Using Invisible Ink Confirmation Codes, IEEE Security & Privacy, May/June 2008.
10. Triinu Mägi, Practical Security Analysis of E-voting Systems. Master Thesis Tallinn University of Technology, Faculty of Information Technology, Department of Informatics, Chair of Information Security, Tallinn, 2007.
11. E. A. Fisher, K. J. Coleman, The Direct Recording Electronic Voting Machine (DRE) Controversy: FAQs and Misperceptions, Congressional Research Service, The Library of Congress, 2005.
12. D. Jefferson, A. D. Rubin, B. Simons, D. Wagner, A Security Analysis of the Secure Electronic Registration and Voting Experiment (SERVE), http://www.servesecurityreport.org/.
13. DVB Project, http://www.dvb.org.
14. MHP-DVB Project, http://www.mhp.org.
15. The Advanced Television Systems Committee, Inc, http://www.atsc.org.
16. Integrated Services Digital Broadcasting, Japanese standard, http://www.dibeg.org.
17. Security and Trust Services API for J2ME (SATSA), http://java.sun.com/products/satsa.

18. Irdeto Access B.V, http://www.irdeto.com.
19. Nagravision SA, http://www.nagravision.com.
20. TAA, Tuscan American Association, http://www.toscanausa.org/Election_EN.asp.
21. ISO/IEC 7816, http://www.iso.org/iso/search.htm?qt=7816&searchSubmit=Search&sort=rel&type=simple&published=on.

## 7 Acknowledgements

The authors would like to thank Tuscany Region for supporting the project and RTV38, Italian TV broadcaster, for the technical effort during testing phases and for contributing to the "Election Night"event.