An Optical Scan E-Voting System based on N-Version Programming

The authors present a multi-agent prototype for an e-voting system based on optical character recognition technology. This case study demonstrates how using N-version programming and improving an e-voting authentication system and the resulting data transmission could further enhance the security of the electoral process.

IÑAKI GOIRIZELAIA, MAIDER HUARTE, AND JUANJO UNZILLA
University of the Basque Country

TED SELKER
Massachusetts Institute of Technology

Researchers are working hard to ensure that the e-voting systems used today—and the ones that will be used in the near future[1,2]—are secure, democratic, and well-designed, especially given that there has been concern about possible dangers of e-voting systems.[3]

For example, when the Irish Government looked into e-voting for the 2004 European elections, it decided not to use e-voting systems.[4] The Commission on Electronic Voting of Ireland concluded that, because of public concerns about transparency, it couldn't recommend using the proposed system at the local and European elections. Its reports state that testability and the ability to audit the ballots would help maximize trust in voting systems. But if the audit trail is paper, and requires conventional counting, it wouldn't be able to achieve the accuracy level of electronic counting.[5] Counting paper ballots by hand is a very difficult task when you have lot of different questions and different people counting

Here we discuss how we could enhance e-voting machines so they could reliably produce separate auditable records of a voter's selections. Electronic ballot copies and other improvements to the auditing capacity of the machines could make voting more secure without having to rely on paper. New technologies, or even better, new ways of implementing existing technologies, might be necessary to fully accomplish this goal.[6]

We propose improvements to Demotek, an e-voting system whose main goal is to count votes by reading them automatically.[7] Demotek has been tested in several local elections in the Basque Country and in Barcelona, but so far it hasn't been checked in large-scale state elections. Our analysis in this article will propose new capabilities for this voting system, specifically, improvements to the data transmission and authentication system, and will demonstrate that existing technology is good enough to provide reliable e-voting systems.

Demotek experiment

As an experiment in modernization by the Basque Government, researchers at the University of the Basque Country and authors of this article, working together with other research institutions and private companies, designed Demotek, an electronic system that uses optical character recognition (OCR) to scan paper ballots and cast votes, in order to automate the voting process. We designed this system to account for the electoral requirements of the Spanish and Basque electoral laws, such as the use of preprinted party ballots and the need for IDs to identify registered voters.[7] Our electoral law states that each paper ballot represents the choices of one political party.

Automatic paper ballot reading

With Demotek, an OCR scanner replaces the traditional transparent ballot box lid with one with an OCR scanner and two slots: one for reading and validation and one for counting (Figure 1).

The ballot resembles traditional ballot papers with one electoral possibility printed on it but with the addition of a special strip where non-visible text, which is only visible using ultraviolet light, is written.

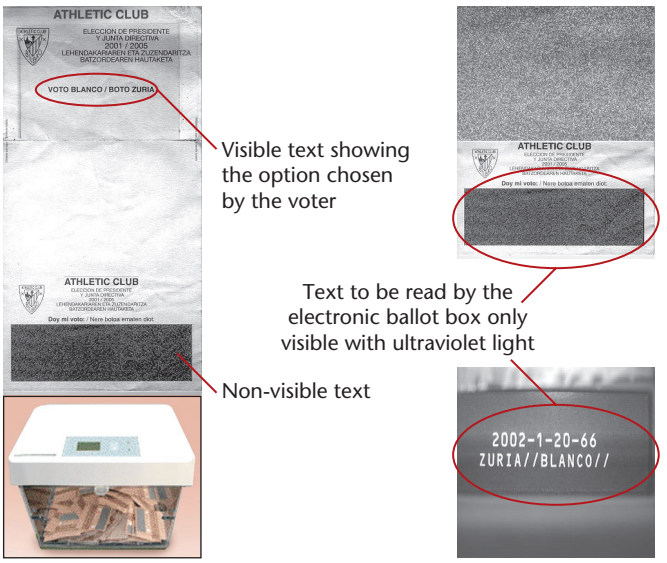

Figure 1. An example of a ballot for the e-voting system and a folded ballot ready to be introduced into the ballot box.

Electoral booths have UV light systems so that voters can illuminate their paper ballots to check the correspondence between the visible and non-visible text. Once the voter has checked it, he or she then folds the ballot, covering only the visible text (Figure 1). The Demotek prototype secures privacy by scanning UV-readable labels that voters have checked. Voters give their paper ballot to the polling place president, who then deposits the ballot into the box through the second slot. Paper ballots remain inside the transparent ballot box until the end of Election Day. If someone wants to audit the results given by the e-voting system, the paper ballots are available for manual counting.

Voter authentication

This process is done manually. Voters present their IDs to the polling place president, who checks whether or not they're registered and have the right to vote at that particular electronic ballot box. (New digital IDs will soon be available for all voters and might be used to improve voter authentication.)

Act of voting

As noted earlier, the act of physically voting means that the voter chooses the desired paper ballot and gives it to the polling place president—another process that could be improved. The most significant criticism to our proposal is that the process doesn't require voters to examine their ballots for authenticity, much less to do this in private. If voters didn't closely examine the ballot's UV-readable portion, they might mistake a fake ballot for a real one. Of course, visual ballots must also be interpreted for sightless people.

Vote deposit and increment register

The polling place president presents the paper ballot to the automatic reading system. If the system can read the ballot without any problem, it opens the ballot box and the president inserts the ballot correctly. Once this is complete, the internal register (register memory associated with each political party that's used to count the number of votes of each party) associated with the political party in question is incremented, and the system updates and displays the total number of votes. Representatives of each political party check their lists and keep track of who votes, as well as the total amount of votes.

There are several checkpoints during this process. First, when the president presents the paper ballot to our electronic ballot box, Demotek automatically reads it and ensures that the paper ballot is valid. If valid, the appropriate ballot box slot opens electronically. If the president tries to insert more than one vote at a time, the sensors will detect it, and the ballot box will close. Sensors also detect when the paper ballot is fully inside the box, so the ballot box closes and is ready for another vote.

Closing the electronic ballot box

Once all votes are recorded and secured in the ballot box, the polling place president must close the electronic ballot box, using an administrator card designed for that purpose. When he or she presents this card to the OCR system, it displays the final results and transmits them to the Central Electoral Office (where all the results of all the electronic ballot boxes are collected and counted) via GSM short messages. The president writes the results on the official certificate; this is the legally valid result for that ballot box. Political party representatives should also sign this certificate. After the votes have been recorded, Demotek erases all the information stored in memory. As noted earlier, to verify that the GSM transmission was correct, all the votes inside a ballot box can be saved as an auditable record.

Demotek vs. other e-voting systems

Table 1 shows the results of the evaluation of the Demotek proposal for an e-voting system vs. other voting systems, taking into account how well our system responds to the expectations that any voting machine must fulfill with regard to electoral tradition in our country as well as others.

Improving Demotek with N-version programming

Our analysis of Demotek shows that it's a first step in the design of new voting technology. Several important functions could help automate the electoral process, but they have either not been implemented

Table 1. Evaluating Demotek.*

	PREPRINTED BALLOT	DEMOTEK PREPRINTED	DIRECT RECORD ELECTRONIC WITHOUT VOTER VERIFIED PAPER AUDIT TRAIL	OPTICAL MARK RECOGNITION	REMOTE INTERNET VOTING SYSTEM
Privacy	Yes	Yes	Yes	Yes	Yes if no external attack
Accuracy	Even multiple people made two to three errors per thousand counted	Possible problems due to system failure or attacks	Possible problems due to system failure or attacks	Possible problems due to system failure or attacks	Possible problems due to system failure or external attacks
Auditability	Yes	Yes	No	Yes	No
—Physical	No	Yes	Yes		Yes
— Separate record of transaction	No	Yes	No	Yes	No
— Separate record of selection					
Convenience	Yes	Yes	Yes	Yes	Yes
Ballot creation	Not flexible	Not flexible	Very flexible	Usually not very flexible	Very flexible
Availability	Yes	Sensitive to system failure	Sensitive to system failure	Sensitive to system failure	Sensitive to system failure and external attacks
Handicap accessible	No	No	Yes	No	Yes
Easiness*	Very easy	Very easy	Easy	Easy	Complex
Automation level	Manual	Partial	Total	Almost total	Total
Ability to vote anytime, anywhere	No	No	No	No	Yes
Coercion and vote sale	Traditional	Traditional	Safe	Only with physical security	Not safe
Compatible with electoral tradition in Basque Country	Yes	Yes	No	No	No
Compatible with electoral tradition in different countries	Various	No	Yes	Yes	Yes

*We evaluated ease of use by surveying participants in a recent university election where the technology was used. In general, the participants found the new system as easy to use as the traditional voting system.

yet, or could be improved significantly. That's the case with the voter authentication system and with data transmission.

Two important goals—compatibility with the previous version of Demotek and reliability of existing technology—must be taken into account when improving Demotek's reliability. As far as software development technology is concerned, N-version programming techniques would make undetected attacks far more difficult to carry out.[8, 9]

An N-version system consists of several software modules developed in controlled isolation. We implemented such modules for each function in Demotek. A distributed decision algorithm determines that each stage of the process has consensus from its modules.[10,11] There has been significant research during the past few years in the field of N-version programming;[12–14] its main advantage is that there's no way to attack the whole system without separately attacking an important number of software modules, making hacking the system more difficult.

Diversity in computer systems refers to systems that run on different hardware, operating systems, or algorithms, and its a fundamental requirement of N-version programming.[14] With enforced diversity, the probability of identical software failures occurring in

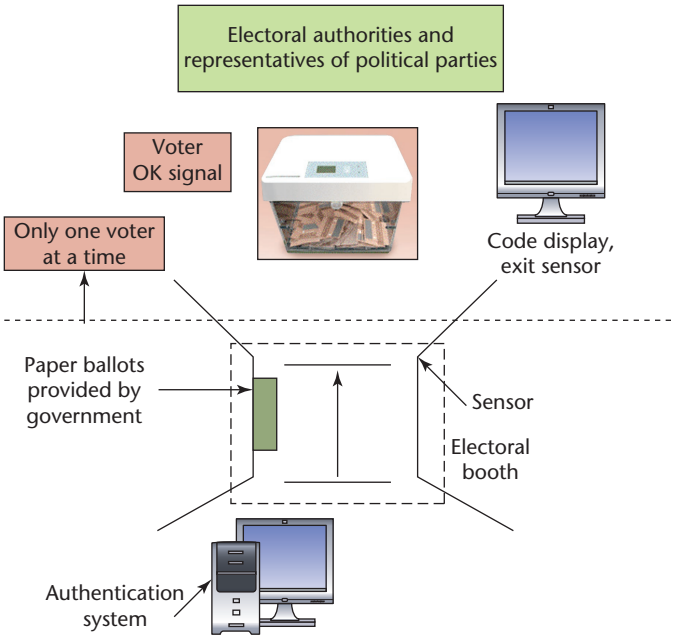

Figure 2. Demotek's authentication system, electoral booth, and e-voting ballot box. This voting procedure adds an authentication capability to an e-voting system.

multiple modules at the same time is greatly reduced. That's why we propose forced diversity at each stage.

The N-Version Execution Environment (NVEX) is another fundamental component of N-version programming. It's the software or hardware component that manages the *N* individual software modules, constructing N-version software units (NVS) from the inputs, outputs, and behaviour of individual modules. The system uses the decision algorithm to decide if a message agrees with enough other messages calculated by other versions of the software units to be able to corroborate its accuracy. This decision will be made based on the number of exact matches among input XML messages. We improve Demotek by using N-version programming technology and the philosophy behind this concept. Every software module of the system is written again with this in mind. We want to use this philosophy to make the system more difficult to be attacked by hackers.

The use of N-version programming has been criticized, especially when different people recoded the same algorithm. Nancy Leveson and John Knight[15] have shown that caution would be appropriate when implementing software systems based on N-version methodology. This becomes important in those software applications "where loss of life is possible, such as commercial aircraft" though this isn't the case when used for voting purposes.

Following those recommendations, we redesigned the new version of Demotek as follows:

- We made the concept of diversity fundamental to the redesign, not only during software development, but also for when we establish new voting procedures.
- Experienced programmers belonging to different organizations developed our software.
- We used different algorithms. For instance, we automatically read paper ballots using two different algorithms that run on different operating systems.
- We used different data transmission technologies (adding redundancy to our system).

We strove to protect our voting system from external attacks especially during voter authentication, data transmission, and the display of results.

Authentication system improvements

Let's look at the authentication system more closely. Adding an authentication facility to Demotek requires some thought about the actual voting process. As noted earlier, every voter in the Basque Country must show an ID. Figure 2 shows how such a system could be used when a person is ready to cast his or her vote.

Voters now present their ID cards to the authentication system. Once it's proven to belong to a registered voter, the authorities and party representatives receive the signal "voter ok," and the voter may enter the booth. From that point on, only authenticated voters are allowed in the booth, having chosen a paper ballot without anyone present and therefore without the threat of coercion. The voter then casts his or her vote, and the e-voting system automatically reads it. Simultaneously, the voter receives a code that he or she can check against the list of codes published at the end of the day to ensure that the vote was counted. This code isn't associated with the voter's option or ID. The exit sensor detects when the voter leaves the Central Electoral Office, and the system is then ready to authenticate a new voter.

This process has no way to link a voter's ID and the vote cast given that our electronic ballot box doesn't keep any ordered list of votes in memory.

Casting votes in this manner has another important advantage over the way Demotek originally was designed. The combination of persons and computers can prevent mistakes and many kinds of collusion. For instance, the procedure in Figure 2 eliminates opportunities for a person to vote twice. For a double vote to occur, the poll worker and the system would both need to accept that person twice. In this sense, the concept of diversity that underlies N-version programming is also apparent in this new procedure.

Figure 3 shows the new design we propose to add authentication functionality to any e-voting system that doesn't support it. In this case, it's particularized to improve the Demotek e-voting system.

1. Voters insert their digital IDs into the authentication machine, inserting it in different readers to improve robustness. The digital ID contains, among other things, the voter's PIN and private and public keys. The identity number is signed and blinded by multiple identity number modules. The modules use their distributed algorithm to demonstrate that they agree about the identity number to accept it.
2. The digitally signed and blinded identity number is then transmitted to the ID Check stage by using different transmission resources.
3. All the ID Reception and Checking N version programs ensure that the information (after being decrypted using the public key) has been transmitted without any problems and that it's consistent (that is, the decision algorithm evaluates all the inputs and says that reception was OK).
4. The system, using the algorithm, checks each elector's database versions to determine whether the voter is authorized. If accepted, a "voter ok" signal is sent to the electoral office president and political representatives. This signal also allows the voter to enter the booth to verify the UV-readable party name on a ballot of choice.
5. The voter casts his or her vote by giving it to the polling place president of that electoral office. We might add that the usual custom of having representatives of the parties present is itself an N-version approach. At this point, the polling place president could also ask for the ID to double check that this person is a valid voter. If laws would change—and we hope they do—voters could place it in the voting box themselves.
6. Once the vote is inserted in the ballot box, the "vote in ballot box" signal is on, and the code generator receives it.
7. The code generator provides a code for each voter that's displayed on a monitor. Voters can write it down and use it later to make sure their vote was included in the actual count by checking the list of codes published at the end of the electoral day. By listing these codes, we try to increase voter's confidence in our system. No official paper receipt containing the voter's option (which could be sold later) is given to the voter. This validates voting by making it clear that all votes have been deposited in the ballot box. Finally, the exit sensor detects that the voter leaves the electoral office, and the system is ready for another voter.

Forced diversity in each stage makes any collusion among many programmers incredibly difficult and discoverable. The authentication system uses digital IDs to identify voters. The encrypted ID information is transmitted by at least three different transmission channels. At the following stage, the system receives and checks the ID information using at least three diverse software versions. At the next stage, three electors' databases are used to check whether that ID corresponds to a valid voter.

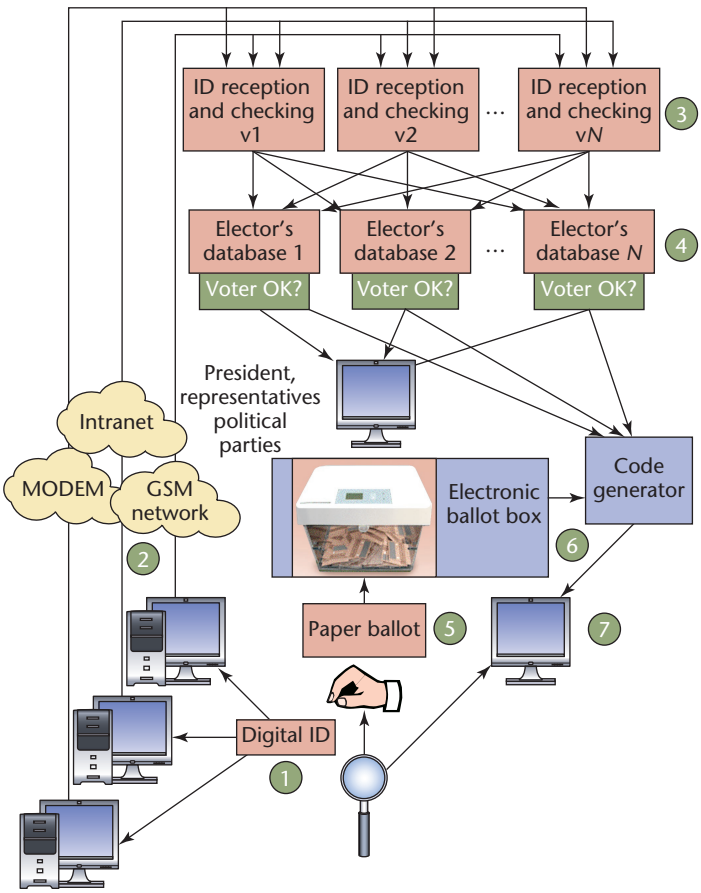

Figure 3. Voter authentication system based on N-version programming techniques showing the authentication machine, different ways of ID transmission, and elector's database.

Data transmission improvements

Data transmission between each polling place and the Central Electoral Office could be accomplished using SMS messages and the GSM network, but this method is particularly vulnerable to attack. One of the best ways to improve data transmission mechanisms in voting technology is to apply existing technology on top of cryptography techniques, but in a different way—in this case, we use the N-version programming to reduce external attacks to our system. Although today's networks are robust, if one transmission is intercepted, it won't reach its destination. If, on the other hand, one transmission occurred on land lines, another via the Internet, and a third by cell phone, the agreement of two of out three would be enough to boost security.

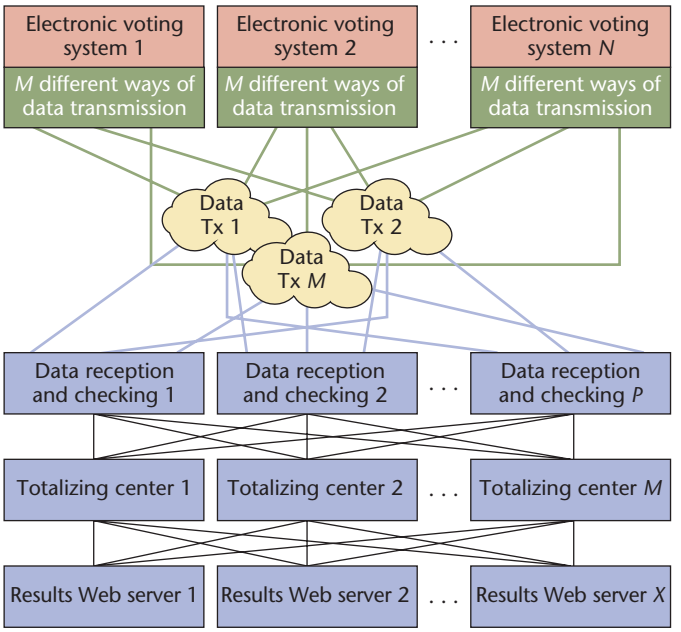

Figure 4. Data transmission based on *N*-version technology showing different ways of transmission, Totalizing Centers, and Web servers showing the election results.

As shown in this design for the authentication system, N-version programming methodology could be a first-rate approach to solving a very difficult problem. That means that each e-voting system must have *M* different ways to transmit partial vote counts to the totalizing centre (see Figure 4). Therefore, to cover a large-scale state election, *N* different e-voting systems are needed, each one having *M* different ways of data transmission.

At the Central Electoral Office, there would be *P* different modules, the goal of which would be to receive and check data transmitted using each different transmission method. Each module would generate an XML message that's sent to all the modules in the next stage with any parameters these modules need. At this point, the N-version execution environment would execute the different decision algorithms (one per each module) and the "most likely correct" input would be elected as an input for this module. There would be *M* Totalizing Centers and all of them would be in charge of calculating final results. Each module would also generate an XML message containing final results, which would be sent to the *X* results Web servers where, after approval by authorities, they would be made publicly available. (Based on the concept of N-version technology, the system should be implemented using several Web servers, *X* being an undefined number. Each Web server takes as input data coming from all the Totalizing Centers and uses the decision algorithm to check data; if OK, it publishes the results.) At this stage, the NVEX would also run the decision algorithm to make sure that final results match.

We understand that e-government is the future, and our e-voting system means a first step to support it. Electronic counting of votes implies a better accuracy level than hand counting. Our system also helps voters avoid errors when depositing their vote, and it costs less money than traditional ways of voting (less human resources needed).

Possibilities for the future would be to create an audio output (upon scanning) to earphones on the Demotek, thereby letting all voters confirm that the system interpreted the vote as they intended. If this audio output were recorded, it would complement the paper and electronic record with a record the voters could hear as they deposited their ballot. Yet another possibility is to redesign Demotek and make it possible for voters to vote using any device connected to the Internet or intranet. □

Acknowledgments

We thank the Carnegie Corporation, the Knight Foundation, and the General Directorate for Electoral Processes of the Basque Government for their support and funding of this work.

References

1. Caltech/MIT Voting Technology Project, "What is What could be," http://web.mit.edu/voting/, July 2001; http://web.mit.edu/newsoffice/nr/2001/VTP_report_all.pdf.
2. California Internet Voting Task Force, "A Report on the Feasibility of Internet Voting," Jan. 2000, www.ss.ca.gov/executive/ivote/final_report.htm.
3. T. Kohno et al., "Analysis of an Electronic Voting System," *Proc. IEEE Symp. Security and Privacy*, IEEE CS Press, 2004; http://avirubin.com/vote.pdf.
4. Commission on Electronic Voting, "Secrecy, Accuracy and Testing of the Chosen Electronic Voting System," Apr. 2004; www.cev.ie/htm/report/V02.pdf.
5. T. Selker, "Security Vulnerabilities and Problems with VVPT," Caltech/MIT Voting Technology Project, April 2004; www.vote.caltech.edu/media/documents/vtp_wp13.pdf.
6. Caltech/MIT Voting Technology Project, "Statement on Verifying the Vote and Auditing Elections," Feb. 2004; www.vote.caltech.edu/media/documents/Caltech_MIT_Audit.pdf.
7. I. Goirizelaia et al., "An Electronic Secure Voting System Based on Automatic Paper Ballot Reading," LNCS 3287, Springer, 2004, pp. 470–477.
8. S.D. Liburd "An N-Version Electronic Voting System," Caltech/MIT Voting Technology Project, May 2004,

www.vote.caltech.edu/media/documents/wps/vtp_wp17.pdf.
9. T. Selker and J. Goler, "The SAVE System: Secure Architecture for Voting Electronically," *BT Technology J.*, vol. 2, no. 4, 2004, Springer, pp. 89–95.
10. A.A. Avizienis, "The Methodology of N-Version Programming," *Software Fault Tolerance*, M. Lyu, ed., John Wiley & Sons, 1995, pp. 24–46.
11. A. Avizienis and L. Chen, "On the Implementation of N-Version Programming for Software Fault-Tolerance during Program Execution," *Proc. Int'l Computer Software and Applications*, 1977, IEEE Press, pp. 145–155.
12. P. Popov and L. Strigini, "The Reliability of Diverse Systems: A Contribution using Modelling of the Fault Creation Process," *Proc. Int'l Conf. Dependable Systems and Networks*, IEEE CS Press, Jul. 2001, pp. 5–14.
13. M.R. Lyu and A. Avizienis, "Assuring Design Diversity in N-Version Software: A Design Paradigm for N-version Programming," *Proc. Dependable Computing for Critical Applications 91*, Springer-Verlag, 1991, pp. 197–218.
14. B. Littlewood, P. Popov, and L. Strigini, "Design Diversity: An Update from Research on Reliability Modelling," *Proc. 65-Critical Systems Symposium*, Springer-Verlag, 2001; www.csr.city.ac.uk/diversity/Papers/SSS2001/SSS2001.pdf.
15. J. Knight and N. Leveson, "An Experimental Evaluation of the Assumption of Independence in Multi-Version Programming," *IEEE Trans. Software Eng.*, vol. 1, SE-12, no. 1, 1986, pp. 96–109.

Iñaki Goirizelaia *is a full professor at the University of the Basque Country's School of Engineering. His research interests include e-voting technology, Web-based virtual learning environments, and security schemes based on digital image watermarking. Goirizelaia has a PhD in electrical engineering from the University of the Basque Country. Contact him at inaki.goirizelaia@ehu.es.*

Ted Selker *is an associate professor at the MIT Media Laboratory, the Director of the Context Aware Computing Lab, and codirector of the Caltech/MIT Voting Technology Project. His research interests include product design of the future, voting technology, and human-computer interaction. Selker has a PhD in computer science, information sciences, and applied mathematics from City University, New York. Contact him at selker@media.mit.edu.*

Maider Huarte *is a lecturer at the University of the Basque Country's School of Engineering. Her research interests include e-voting technology, e-learning, and secure programming techniques. Huarte has a master's in telecommunication engineering from the University of the Basque Country. Contact her at maider.huarte@ehu.es.*

Juanjo Unzilla *is an associate professor at the University of the Basque Country's School of Engineering. His research interests include e-voting technology, Web-based virtual learning environments, security schemes based on digital image watermarking, and wireless network security. Unzilla has a PhD degree in telecommunication engineering from the University of the Basque Country. Contact him at juanjo.unzilla@ehu.es.*