An Approach for Detecting ID Frauds in a Traditional Voting System Using a Smartphone Stand

Paul Strimbeanu
University "Lucian Blaga" of Sibiu
Faculty of Engineering,
CS3E Student
Victoriei Boulevard 10,
Sibiu 55002, Romania
paulstrimbeanu@gmail.com

Alexandru Butean
University Politehnica of Bucharest,
Faculty of Automatic Control and Computer Science
Splaiul Independentei 313,
Bucharest 060042, Romania
alexandru@butean.com

Diana Butean
The KPI Institute
European Division
Marketing and Online Platforms Department
Harbour Esplanade, Suite 606,
Melbourne 198, Australia
diana@butean.com

Florica Moldoveanu
University Politehnica of Bucharest,
Faculty of Automatic Control and Computer Science
Splaiul Independentei 313,
Bucharest 060042, Romania
florica.moldoveanu@cs.pub.ro

ABSTRACT
All over the world, the voting systems are very important tools that contribute to the basic principles of democracy and equality. In every developing country, the problem of online voting becomes a necessity, therefore solutions must be found. Nowadays mobile smart devices are equipped with high performance lens and cameras, therefore they can capture high quality pictures in almost no time. Optical character recognition methods (OCR) are getting 99% accuracy on standard font documents with little or no noise at all. Having this in mind we propose a system that verifies in real-time the ID of an elector, right before a traditional voting system procedure. This solution is based on a stand containing a mobile device that takes a quality picture of that ID and sends it to a server to extract the data. The use of such a system does not affect the actual voting process and does not interfere or relate with the voter's choice at all.

Author Keywords
Mobile devices, E-government; voting system;

ACM Classification Keywords
H.5.m. Information interfaces and presentation - Human-Computer Interaction, Miscellaneous. ; J.1 – Computer Applications – Administrative Data Processing - Government; I.7.5 Document and text processing– Document Capture – Optical Character recognition

General Terms
Human Factors; Design; Security;

INTRODUCTION
Electronic election systems are used since 1960 when punched cards were introduced. The first country to use them on a large scale was United States of America, who implemented the system during the 1964 presidential elections. As a result, a new automated system to verify the authenticity of the voters was needed.
Online voting systems discard any physical evidence of the process, relying solely on the electronic infrastructure. During our research, we have developed a transition model between the traditional and online voting systems. In this paper we propose an architecture where mobile technologies and devices can be used to check in real-time a voter's IDs before the actual vote.

ONLINE VOTING SYSTEMS
Internet voting systems proposed in research literature [1, 2] use cryptographic techniques to get to a property called end-to-end (E2E) verifiability [3]. This feature ensures that the ballots have been counted accurately without trusting the computers or officials to behave honestly. The Estonian voting system is the largest Internet voting system in the world. It was introduced in 2005, being used by 30% of participating voters to cast their vote at the most recent elections [10] and because of this it can be used as an example. The system is based on the Estonian national ID infrastructure. These smartcards have the ability to perform cryptographic functions, which in combination with card readers and client software allow Estonians to log in to websites and make legally binding signatures on documents [5]. Based on this, they are used to authenticate and sign the ballots. As an extra security measure, the smartcards are

associated with a PIN code to authorize each operation. The source code from the server is published to a GitHub repository 2-3 weeks before the elections and the infrastructure is configured one week before the election in a public ceremony. It consists of four machines: Vote forwarding server (VFS), Vote storage server, Log server and Vote counting server. Before each election, a set of voting applications is published by the election authority.

In order to vote, a person has to launch a client application, insert his ID and enter the PIN code. By doing this, a secure connection with the VFS is established. After the server verifies the voter`s eligibility, a list of candidates for the voters district is returned [4]. After the voter selects his choice, he enters his PIN code again to sign the vote. The client pads the choice using RSA-OAEP and randomness, encrypts it with a 2048-bit encryption public key and signs the encrypted vote with the private key of the voter. The result is sent to the server. Voters are allowed to vote multiple times but only the last vote counts. Earlier votes are revoked and while the system indicates if the user voted previously, it does not show the number of times. The vote can also be overridden by voting in person on the Election Day. If the voter wants to confirm that his vote was recorded correctly he can use a smartphone app provided by the election authority [6]. The verification can be done three times per vote and up to 30 minutes after casting.

The storage server processes the encrypted votes to verify the signatures and removes any invalid or revoked votes after the online voting has ended. Officials export the set of valid votes in a public counting session, making sure only the anonymous encrypted votes remain after the signature is stripped away. These votes are burned to a DVD and transferred to a counting server. The counting server decrypts each vote and the results are combined with the totals from the in-person polling stations and published as overall results for the election.

The problem with the system is that it is vulnerable to denial-of-service attacks against the voting process. By sending many specially created requests containing fields with long names, an attacker can exhaust the server`s log storage, thus blocking it from accepting new votes. Another problem is a shell-injection vulnerability in the user interface of the server. It would allow operators to execute arbitrary shell commands on the election servers with root privileges. This can be very dangerous and proves the fact that open source doesn`t guarantee the absence of vulnerabilities [7].

TRADITIONAL VOTING SYSTEMS

In the United States, conventional voting systems are formed by joining many state-wide elections conducted independently by local election jurisdictions. States can use different voting systems, the decision being made by each county. The systems vary [11] from paper ballots and punch cards to mechanical lever machines, optical scan and direct recording electronic devices. In addition, a variety of voting processes are employed throughout the nation. Traditionally, people cast their vote on the Election Day. However, some alternative methods do exist:

- Absentee ballots, which allow people to vote-by-mail before the election and are available to voters who prove that are unable to get to the polling place;
- Vote-by-mail, available to everyone who registers as a voter, the person only has to fill-in the ballots and return it by mail, thus removing the need of polling places;
- Satellite voting, allows early voting from sites around the county for a period of time (several weeks to a few days) prior to elections.

While all these methods allow people to cast their votes in any condition, increasing turnout and convenience, a big problem appears because the process is very hard to manage and the counting of the votes is slow. These systems require voters to register before voting, making the process even more complicated. People need to make sure they don`t make a mistake when they cast their vote because they are not allowed to use multiple paper ballots, which leads to many votes being canceled.

However, there are advantages to paper ballot voting [8]. They give a reliable audit trail and perhaps more importantly, they guarantee the protection of electoral neutrality.

PROPOSED ARCHITECTURE

The idea behind this concept is that simplicity is the ultimate sophistication. Trying to solve a complex problem using a complicated solution only makes the matter worse. Because of that, the architecture of the system is as follows: using a smartphone’s camera a photo of an ID is taken; the photo is then transferred to database storage; the processing platform uses a secure connection (see Figure 1). After the information contained in the photo is processed, the data is stored online in a database so it is available anytime to anyone who has access to the system.

Data security is a big concern because we are dealing with personal information. As a result, a secure connection to communicate between the phone stand, the cloud processing system and the government database is used. All the data transferred is secured with a 128-bit encryption keys. This guarantees that the transfer is safe and no information is disclosed without approval.

In order to verify the IDs and determine if they are indeed genuine the system tracks several fields. It uses a combination of 3 unique identifiers from the ID to create a criterion on which each entry will be verified. Each time a new person tries to vote, the document is first verified against the existing items in the database so that no duplicates exist. This gives us the assurance that each entry is indeed unique and no ID can be used to vote multiple

times. In order to get the information from those specific fields, this project uses an online OCR processor [9]. The OCR processor takes the captured photo as an input and converts the information into text with a high success rate, 99.8% [9]. The information provided by the OCR engine contains personal data (name, surname, place of birth and address) and identification data (serial number, ID expiration date and PNC) that are stored in a database.

To eliminate the possibility of someone using a fake ID that has a unique combination of fields, the system can compare all the data gathered after the voting process has ended to a government database. This database contains all the relevant information regarding each person that has the right to vote. Because of this, every time an invalid ID appears it is flagged as illegal voting and subsequent actions should be taken to verify the validity of it and the people who voted using such IDs.

A 2-step verification is implemented because of this. The first step and the one that gives instant notifications ensure no one votes twice. After each entry is verified so there are no duplicates in the database, a notification is sent back to the phone if the confirmation went through or if there is a duplicate and the validation failed. This first notification provides real time verification. The second notification is implemented as a collective response from all the IDs used in the process after they have been compared with the government database and proved to be fake. Using these two steps ensures that no person votes using a fake ID.

ADVANTAGES OF THE PROPOSED ARCHITECTURE

The proposed architecture grants a simple and secure way to verify each person who wants to vote. By eliminating pre-registration, privacy and neutrality are guaranteed. Also the use of a smartphone reduces the cost and grants portability. Client or server fraud attempts are eliminated as no crucial vote related information is stored and cloud processing ensures the process moves fast and without any delays. Using a secure internet connection coupled with data encryption guarantees no personal information is accessible to anyone. By keeping paper ballots in combination with online ID verification the architecture combines the advantages from both online and traditional voting systems. It doesn`t require major modifications to polling places. Because of that and the accessible price of the components, it can be seen as a first step in a migration process from a traditional system to an online system.

PROOF OF CONCEPT

In order to make a proof of concept for the proposed model we have implemented an alpha version of this system. Testing was made using standard plastic ID's (released after 2002) and two Android devices for multi-point data acquisition. Each devices has the applications installed and connected to the secure server which was established on Google AppEngine. The entire scenario made for one voter lasted between 34 – 56 seconds depending on the availability and speed of the cloud system. Since cloud scalability is not a problem nowadays, we assume that the number of simultaneous voters is not a problem.

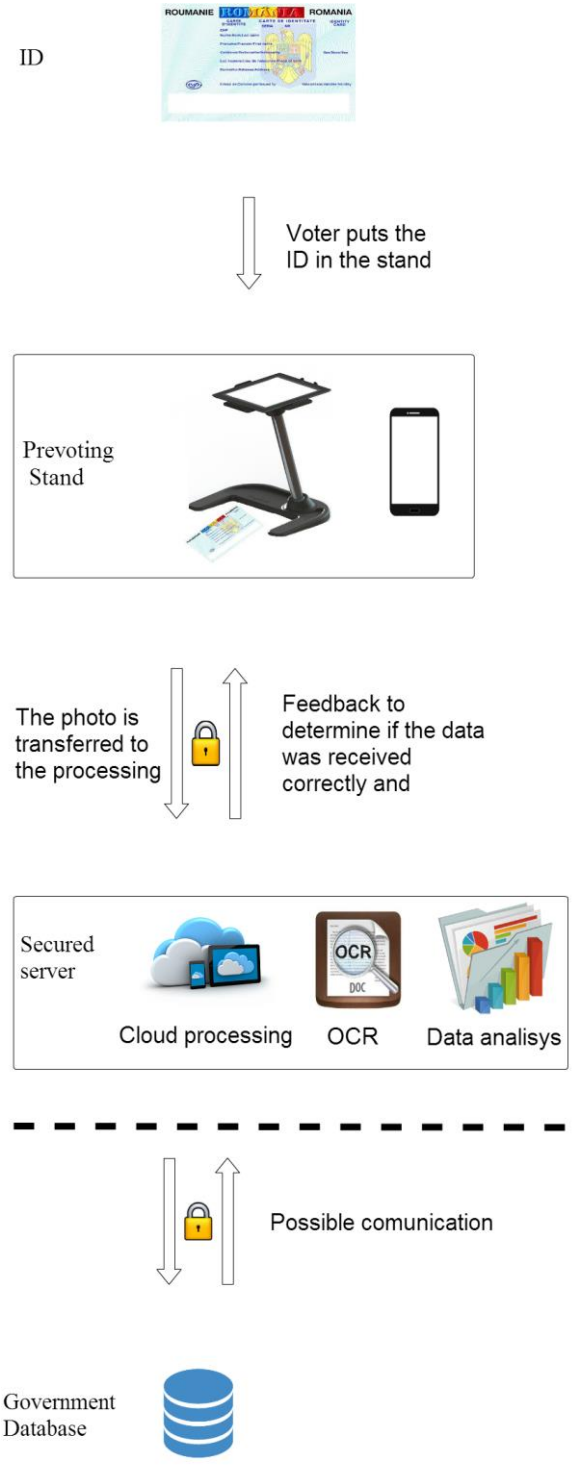

Figure 1: System Architecture

CONCLUSION

In this paper we have presented a solution that offers the possibility of checking an ID before a traditional voting procedure. Notifications are triggered in real-time concerning the legitimacy, validity and uniqueness of the vote. The procedure requires positioning the ID in a special stand that contains a smart device with a good camera that can capture an image and send it to the cloud server. The server processes the image and extracts the data using OCR. Every information is stored encrypted in the database for later use or for immediate notifications.

After the development of an alpha version just for proof of concept purposes we have identified a series of problems: blurring of the lens needs to be detected and a notification has to be triggered; the distance between the stand and the ID needs to be calibrated for every device type; there are at least 4 slightly different types of plastic card ID's and the OCR match must be made individually for every card type; internal flash of the camera cannot be used because on different types of plastic outputs different mirror-light effects.

FUTURE DEVELOPMENTS

Making the system available in the entire voting area is a very important goal. Taking that into consideration, it would be a good idea to implement local temporary databases in polling places that don`t have internet connection. This allows the architecture to work in the most remote places. The processing and checking of the information would be done after the voting process has ended and fake IDs and duplicate votes are flagged. We believe this would reduce the necessity of a real-time synchronization between an existing database and the government database with valid ID's.

Another improvement that can be implemented to allow operating in locations without access to internet is local phone OCR processing. This means that the information authentication should be done on the device, requiring more powerful smartphones so that the time needed for each ID verification is reasonable.

These features will raise the costs of the system but still offer the possibility of real time ID validity notifications, local vote duplication detection or later on synchronizing and statistics. The most important aspect is that they would allow system usage in extreme conditions.

ACKNOWLEDGMENTS

This work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreement POSDRU/159/1.5/S/134398.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 643636 "Sound of Vision".

REFERENCES

1. Adida B., Helios: Web-based open-audit voting. In *Proceedings of the 17th USENIX Security Symposium*, Aug. 2008.
2. Ansper, A., Buldas, A., Oruaas, M., Priisalu, J., Veldre, A., Willemson, J., Virunurm K.m E-voting concept security: analysis and measures. Technical Report EH-02-01, Estonian.
3. Chaum, D. Secret-ballot receipts: True voter-verifiable elections. *IEEE Security & Privacy*, 2(1):38–47, 2004.
4. Cybernetica AS. Internet voting solution, 2013. http://goo.gl/62qv2h
5. Estonian Certification Authority. Mis on ID-tarkvara? In Estonian. https://goo.gl/lXKNF0
6. Estonian National Electoral Committee. Valimised: Android Apps on Google Play, Oct. 2013. In Estonian. Accessed: May 13, 2014, https://goo.gl/GcXkxe.
7. Kitcat, J., Source availability and e-voting: An advocate recants. *Commun. ACM*, 47(10):65–67, Oct. 2004.
8. Mote C.D. Jr., Report of the National Workshop on Internet Voting: Issues and Research agenda. Proceedings of the 2000 annual national conference on Digital government research, Pages 1-59, 2000
9. OCR (ABBYY) Processor: http://goo.gl/ZS8YJc.
10. Springall, D., Finkenauer, T., Durumeric, Z., Kitcat, J., Hursti, H., MacAlpine, M., Halderman, J.A., Security Analysis of the Estonian Internet Voting System. Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security, Pages 703-715. 2014
11. Voting system: https://goo.gl/sEB9WG.