Open Access
Baghdad Science Journal **Vol.15(1)2018**

DOI: http://dx.doi.org/10.21123/bsj.2018.15.1.0110

CVOTING: An Anonymous Ballot E-Voting System

Ali Fawzi Najm Al-Shammari

Received 22/3/2017
Accepted 20/11/2017

Abstract:

One of the concerns of adopting an e-voting systems in the pooling place of any critical elections is the possibility of compromising the voting machine by a malicious piece of code, which could change the votes cast systematically. To address this issue, different techniques have been proposed such as the use of vote verification techniques and the anonymous ballot techniques, e.g., Code Voting. Verifiability may help to detect such attack, while the Code Voting assists to reduce the possibility of attack occurrence. In this paper, a new code voting technique is proposed, implemented and tested, with the aid of an open source voting. The anonymous ballot improved accordingly the paper audit trail used in this machine. The developed system, which we called CVOTING, further demonstrated the efficacy of the Code Voting technique against systematic vote change attacks and provides some features to make it easily configurable for different elections and elections in countries with right-to-left and up-to-down languages.

Keywords: e-voting, open source, code voting, QR code.

Introduction:

Elections like any other field affected accordingly by technology. This effect is illustrated by different e-voting systems like the Direct Recording Electronic (DRE), the scanner based, and the online (Internet) voting systems. However, dealing with voting electronically is not so easy task because of the fact that the votes have to be anonymous all the time. In this case, the challenge is how to be sure that a vote does not change by any compromised component in the system at any phase. For this purpose, different techniques have been used such as the verifiability techniques, that aim to allow a voter or any interested participant to verify that the vote has been processed as intended by the system. Beside vote verification, there are some protection techniques, which tries to add barriers against compromising the system, like for vote coding.

Concerning to the protection techniques, there are different approaches proposed to reduce the probability of systematic vote change attack, by ensuring that the voting machines themselves have an anonymous view of the ballots they are dealing with.

David R. et al. (1) proposed an approach in which the e-voting terminal configured by a ballot image and a ballot layout XML file. The e-voting terminal shows the image to the voter and reads the locations of voters' ticks. The tick locations for the selected options stored in XML file inside the machine to be processed later by another device in order to retrieve the clear text of the selected choice. However, scanner based system has been suggested as a typical solution for this implementation. More specifically, the voter casts the vote using a voting terminal supported with a touch screen and a printer. The voting terminal shows the vote and the voter have to select the candidate by touching the screen. After the voter cast the vote, the terminal produces two outputs: the first is an XML file contains information about voters' ticks' coordinates and the second is a paper vote printed by the attached printer. The voter has to take the paper vote to a scanner device located nearby the voting terminal in order to scan. The scanner scans only the locations of voters' ticks in order to produce another XML file similar to the one produced by the voting terminal. After the election day, counting devices interpret the XML files generated by the scanner to calculate the election outcome. Furthermore, the other XML files generated by the voting terminals and the physical paper votes are used for auditing purposes.

Similarly, there is the Code Voting approach which is used by some internet voting systems (e.g., the VeryVote (2)). In this method, the voter

Department of Prosthetics and Orthotics Engineering, University of Kerbala. Iraq.
E-mail: alifawzi@uokerbala.edu.iq

indicates the selected candidate in term of entering codes apart from selecting a clear text. More specifically, the election authorities send by e-mail a code sheet to the voter. This sheet maps the available candidates with random codes differs from voter to voter. The e-voting client asks the voter about the code reflecting the preferred candidate in order to upload to the tallying machine. The client application will have an anonymous view of the cast code. In this case, a compromised e-voting client will not be able to change the vote systematically.

Another related approach proposed by Rolf O. et al. (3) named as CAPTCHA voting, which is another online e-voting system. In this approach, the e-voting server sends a ballot image and a list of codes to the e-voting client, in real time, during the vote cast session. The ballot image formed as a set of candidate names shown as CAPTCHA areas (displaying text characters in a way that it is readable by human but not by machine, more information about CAPTCHA can be found in (4)). Each code in the received list is associated with a single CAPTCHA area. In this case, when the voter selects the CAPTCHA of the preferred candidate the machine maps the selection to an anonymous code. Then, the code is uploaded to the voting server to be processed for election result calculation.

All of the mentioned approaches have in common the motivation of making voting ballot anonymous for the e-voting machine. This in order to mitigate the attack of changing the vote systematically, just in case if the client machine is compromised.

After analyzing the current state of the art, we have observed some issues in these approaches, among them we mention: The David R. et al. solution relies on a unique pattern for the code mapping which may allow the attacker to manipulate the e-voting machine on the election day if a copy of the ballot image leaked outside the poll. Regarding to the online Code Voting, a corrupted voter may reveal the code sheet to a compromised client for different reasons like vote selling. Furthermore, both of the online code voting and CAPTCHA system limits to work in an online environment.

Based on the results of our analysis, a new anonymous ballot structure introduced and applied in an improved voting machine, we called CVOTING, which is an open source DRE machine. The aim of this system is to mitigating some of the mentioned issues. Even though the CVOTING system is a polling place voting system, the proposed anonymous ballot technique could be applied to online voting systems too. Furthermore, as an extra result, an open source solution is provided to the e-voting community, that is more immune against the systematic vote change attack, supports verifiability, and could be configured easily for different elections without need to touch the code.

This paper is structured as follows, Section 2 describes the CVOTING system components and how to configure the CVOTING machine to run a specific election. Furthermore, it illustrates the system procedures in different phases and highlights the main required assumption by the system to achieve the integrity of the election result. Section 3 delineates how the vote verification being supported. Section 4 draws the main results of applying the CVOTING machine. Finally, the future work and general conclusions are discussed in Sections 5 and 6.

System Description

The CVOTING can be classified as a DRE based system to be used in a controlled environment election (e.g., polling place election). In this system, the Code Voting technique is used apart from using a clear text ballot. More specifically, the machine provides the voter with human readable ballot image on the machine screen. The voter can select the preferred option by touching the screen. Then, the machine maps voter's touch into an anonymous code. These codes stored in the machine as the cast votes. Later, the votes are collected from the different DREs and transferred to the tallying machine for decoding and tallying.

The following subsections introduce the different components of the system and their responsibilities, then describe the technical configuration of the CVOTING machine. The last subsection outlines the system procedures in the different phases of the election and finally summarize the required assumptions to fulfill the integrity of the election result.

System components

The CVOTING system includes a number of components, namely: Election Management System (EMS), e-voting client software (CVOTING machine), printing device and tallying machine. A brief detail, about how different components works, discussed in the following.

Ballot anonymity is the main feature of the CVOTING. The anonymity achieved by using an image to represent ballot instead of text. The image mapped into multiple areas that represent multiple selection. Each area location is mapped into an anonymous code. Mapping codes are organized in tables, that is called "image-code mapping table".

In the election, different voting machine will be configured by ballot images with different mapping table. The EMS is the responsible for generating the ballot images and image-codes mapping tables.

More specifically, each voting machine will be loaded with different ballot image and its relevant image-codes mapping table. The ballot image design represents the user interface that the voter will see in the machine screen. When the voter selects one of the available options shown in the screen, then the machine reads the coordinates of voter's finger touch to be mapped into a relevant anonymous code. At the end, the machine stores the selected anonymous codes to produce the cast vote. The CVOTING machine is equipped with a printing device, to be used as Voter Verifiable Paper Audit Trail (VVPAT) device (5). More specifically, during the vote cast session, the printer prints in real time a paper trail provides vote feedback in order to be verified by voter. If the voter verifies that the cast vote and the paper trail are not contradicted, then the audit trail will be stored in the poll, to be used later for auditing purposes.

The cast votes are collected from the different DREs and transferred to the tallying machine. The responsibilities of the tallying machine used to decode the anonymous codes found in the cast votes to the clear text using the codes-text mapping tables, and then count the election result.

CVOTING configuration

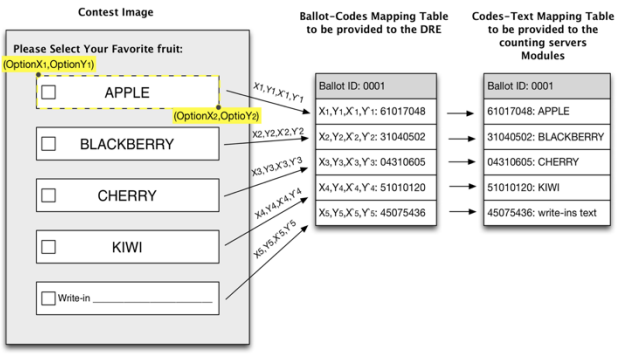

Figure 1. The mapping of the ballot image.

The CVOTING machine has been designed to be configured for different elections without the needs of customizing its code apart from customizing configuration files. The configurations files are: set of images and single XML file.

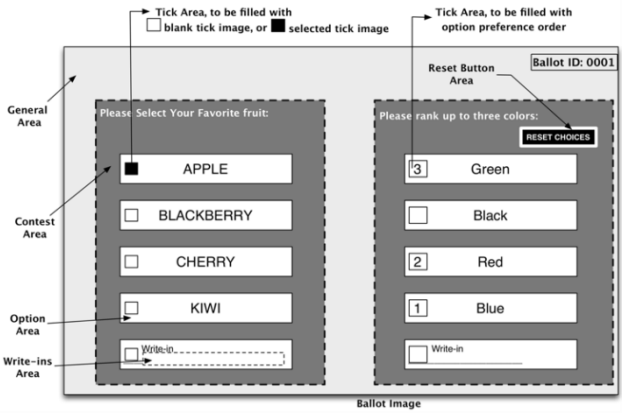

Figure 2. Sample of a ballot image contains two contests, one is ordered and the other is non-ordered.

There are three images to be configured in configuration phase, that are: ballot image, blank tick image and selected tick image. The **ballot image** can include one or more contest, inside each one of them there are a set of options. An option area is formed as a rectangular that is defined by two coordinate points forming the radius of the area. Each option area is mapped into a unique code which is different from contest to contest and from machine to machine, Figure (1) shows how an option area in the ballot image is mapped into a code. If the contest allows write-ins, then the last option area must be dedicated for this purpose. The write-ins area must be defined to be inside the option area and it must be large enough to include the write-in text, it is important to consider the maximum number of characters versus the font size. Furthermore, the CVOTING machine supports two types of contest which are the ordered and non-ordered. The ordered contest allows the voter to select options based on the order of preferences. However, if the contest is ordered then it is required to have the reset button area inside the its area which can be used to reset the selected options in case if the voter aims to re-edit the selection. Finally, in what related to the ballot, the areas outside the contests areas can be used for general purposes (e.g., may contain the Ballot ID, election title, ...etc).

Concerning to the blank and selected tick images, they are small images in which one of them must be located inside each option area to indicate if the option is selected or not. Figure (2) shows the different areas of a ballot comes with two contests, one is ordered and the other is not.

The image files must be supported by a configuration data in order to allow voting machine to identify the different option areas. This data like: the coordinates identifying the option areas, the tick images location's coordinates, the contest specifications (e.g., maximum votes, minimum

votes, maximum write-ins, write-ins font, the coordinates identifying the reset button, ...etc). All of the required configurations by voting machine are organized as an XML file. The configuration data, inside the XML, support three different levels, namely: election level, contest level and option level.

The ***election level*** provides general election configuration data (e.g., election identifier, location, and date) and some private data (e.g., machine ID and the name of the images files to be used by the machine).

The ***contest level*** provides configuration data for each contest in ballot, like: the minimum and maximum votes, whether the contest is ordered or not, the reset button coordinates (set only in the ordered contest), and the write-ins configuration, like: max characters allowed, the coordinates of the write-ins area, and font size (set only if write-ins allowed).

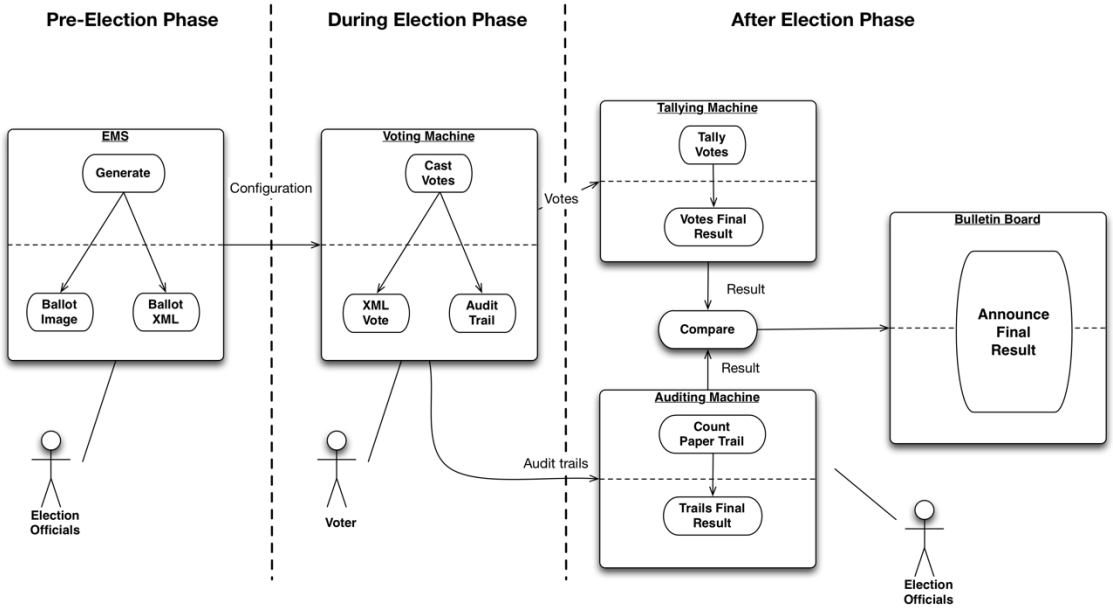

Figure 3. General System Components

The ***option level*** provides configuration data of each option in ballot contests, like: the tick image coordinates, where to put the select tick image, the option area coordinates, and the anonymous option code. The option codes are random unique code allocated for each option in the different ballot contests.

The configuration files illustrate the input for the CVOTING machine, while the output of the machine after each successful vote cast session is an XML file that contains a list of codes reflect the selected options by voter. This means that the e-voting machine have an anonymous view of the cast votes and only the tallying machine is able to count the election result, since it has the ability to map the anonymous codes into clear text.

System procedures in election phases

It is possible to describe the steps to configure and use the CVOTING system during different election phases, namely: pre-election, during the election and after the election.

In the pre-election phase, the election authorities use the EMS to prepare the CVOTING machine configuration files and the tallying machine code-text mapping tables. More specifically, the authorities produce the ballot images for the CVOTING machines, in which the order of the contests and options inside may differ from machine to another machine. Also produce the XML configuration file which include the ballot codes mapping information. Furthermore, generate the tallying machine XML files which include the codes-text mapping information for different machines in the election. All of these files are loaded to the CVOTING machines and tallying machine to be ready for conducting the processes of the next election phase, i.e., the during election phase.

In the during election phase, the polling officer opens the poll by activating the CVOTING machine. The machine processes the information provided by the configuration XML file in order to start the election day. When an eligible voter arrives, the polling officer activates the CVOTING machine for a vote cast session. The machine loads the ballot image on the screen and asks the voter to start vote casting process. When the voter selects an option area by touching the screen, the machine maps touched location to a relevant code. Meanwhile, the CVOTING machine draws the tick image in a location that indicates the selected

option. Before casting the vote by voter, the CVOTING produces a two votes feedback, a soft copy feedback shown on the screen and a hard copy feedback printed as paper trail. The voter has to verify that both feedbacks are the same as intended to be and then cast the vote. After vote confirmation, the CVOTING machine generates the cast vote as XML file containing the selected option codes. Furthermore, it encodes the list of the selected codes into a QR code (6, 7, 8) to print in the last inch of the paper trail. The paper trail stored inside the poll and can be used for auditing purposes after the election.

In the after election phase, the cast votes are collected from the different DREs and transferred to the tallying machine. The counting modules decodes the cast votes to the clear text and counts the final result. Furthermore, the auditing process performed by authorized auditors to support the universal verifiability.

Integrity Assumptions

Any critical system, like e-voting system, have to consider a number of security and procedural assumptions in order to achieve maximum integrity of the election results. For example, the DRE systems assumes that the voter must be attentive enough to verify the vote summary shown in the screen before casting the vote. This is to verify that the vote has been cast as intended to be. However, the required assumption of the CVOTING system summarized in the following:

- Mapping tables (i.e., images-codes and codes-texts) must be kept safe during election phases, i.e., nobody except authorized entities can access it.
- It is not possible to access the configuration files in the machine by any attacker.
- Any voter cannot access the paper trails that stored in the poll.
- Sufficient number of the voters must check and verify the paper trail during the vote cast session.
- Election officials must audit a sufficient amount of paper trails in the after election phase.

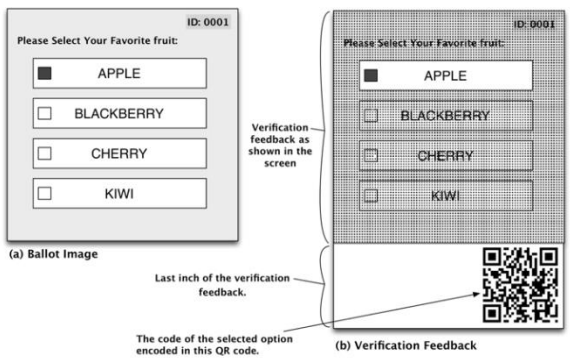

Figure 4. Verification Feedback

The achievement level of the mentioned assumptions may vary according to different regions and environments, and the fail in achieving them would make it easy to compromise the system. So, it is recommended to consider these assumptions before adopting the CVOTING in any election.

Verifiability in CVOTING

The level of verifiability supported by the CVOTING is not so different than the level supported by other existing DREs, e.g. the ES&S (9), the ProVote (10). However, there are two technical issues that we need to discuss here, the first one is how the machine can provide the verification feedback to the voter while it has an anonymous view of the ballot. The second issue is how the election authorities can verify the system against any machine failure (or even any random vote change attack caused by a compromised machine).

Concerning the first issue, the machine provides the verification feedback by highlighting the selected option areas on the screen. This is done by overlying a transparent layer covering the ballot image except the areas of the selected options, (option area coordinates are defined by the configuration XML), as shown in Figure (4).

The printer prints out in the paper trails the same feedback image as shown in the screen. The voter responsibility is to verify that both of the screen and the printer feedbacks are the same as what intended to be, before casting the vote. After casting, the printer prints out a QR code in the last inch of the paper trail which contains the codes of the selected options. This process would help the voter to verify that the vote has been captured and cast as intended (which is a part of the individual verifiability).

The second issue covered by the auditing process for the paper trails in the after election phase. More specifically, the auditors can select randomly a set of audit trails, then audit each trail by checking the codes decoded from the QR code if they are correctly reflecting the screen images that is printed in the trail. This process assists to verify the system against any technical failure or any random votes change attack. However, this auditing process supports the universal verifiability.

Results:

Even though the CVOTING system is generally similar to the existing DRE based systems, and the principles of Code Voting is not new, but still we believe that the outcome of this work includes some contribution. We classified the contribution

into technical and theoretical parts as described by the following.

Technically, we produced the CVOTING open source machine software which could be found in the online Github repository. In fact, it is an improved version of the EVM2003 machine (11), which is an open source DRE developed in Python by the Open Voting Consortium (OVC). The significant technical improvement in the CVOTING machine illustrated by:

- Among the available open source DREs, this machine is the first supporting the offline Code Voting.
- The machine is fully configurable for different elections without need to touch the Python code. Such easily configurable machine would provide an interesting tool for the e-voting researchers to re-use for different research purposes.
- The machine can be configured to support any human language (e.g., Arabic, Chinese, ... etc.) as it does not require to render or deal with any text apart from dealing with images mapped into codes. Furthermore, most of the interfaces of the machine (e.g., screen states, buttons, messages) are image based.

Theoretically, the CVOTING system design ameliorate the issues mentioned in Section 1. For example, considering the Rolf O. et al (3). approach, the CVOTING reduces the possibility of being manipulated by leaking the ballot image as it has a different ballot image that comes with random order of the contest and the options inside. So, leaking a ballot image from one machine may cause to compromise that single machine instead of compromising the whole system. About the issue of the online Vote Coding, the CVOTING does not provide the voter with the codes list, instead, each CVOTING machine has its own private list. This would make the codes kept inside the controlled environment apart from given to the voter.

Future Work:

The improvements of the CVOTING machine targets some goals out of improving the verifiability, which lets us to think about ameliorate the verification technique used by the system. Otherwise, the CVOTING will have the same issues as the ones reported for the VVPAT (e.g., the issues reported in EVEREST report (12).

For this purpose, we are investigating to make the CVOTING machine compatible with an open specification protocol (13), which would allow the election authorities to ask different firms to develop Verification Modules (VM) to run in parallel with e-voting machine. More specifically, the machine needs to be developed to broadcast a verification message that is compatible with the specifications of the OVVM during the vote cast session. The VM reads these messages and provides the feedback to the voter to verify during the session. More specifically, the VM will be able to decode the message broadcast from the DRE to the clear text of the selected option as it has the codes-text mapping table. When the voter confirms both of CVOTING and DRE's the feedbacks, both of them store a copy of the cast ballot. After the election day, the election officials can verify the records comes from both devices in order to verify the behavior of the voting terminal (i.e. the DRE machine). This would reduce attack possibility as the attacker needs to compromise two different devices to perform undetectable attack, improve the verifiability, and help to distribute the trust between the machine software and other VM developed by other firms.

Conclusions:

On the one hand, the anonymous view of the ballot could be a strong barrier against compromising the e-voting client to change the votes. From the other hand, the ability to verify the code of an open source machine would make its process more transparent. In this paper, we introduced the CVOTING machine, which is a code voting based open source machine to be more immune and transparent. Furthermore, the easiness in machine configuration makes it an interesting tool to be used for researches or real election purposes. This also would make the e-voting technology available for any research group or country to experiment and use it freely.

References:

1. Webber DR, Borras J, Johnson, RC. Transparent Open Secure e-Voting Exploring the Paradox. EVT07 proceeding, 2007.
2. Joaquim R, Ribeiro C, Paulo F. Very vote: A voter verifiable code voting system. Proceedings of the 2nd International Conference on E-Voting and Identity VOTE-ID09, 2009.
3. Oppliger R, Schwenk J, Christoph L. Captcha-based code voting. Proceeding of 3rd international conference on electronic voting, Austria , 2008.
4. Luis A, Blum M, Langford J. Telling humans and computers apart automatically. Communication of the ACM, 47(2): 56–60, 2004.
5. Mercuri RT. Electronic Vote Tabulation Checks and Balances. PhD thesis, University of Pennsylvania, 2001.

6. Soon TJ. QR Code: The 2D Code of 21st Century and Its Applications. ITSC Synthesis journal, pp:7–14, 2002.
7. Cao X, Feng L, Cao P, Jianhua, H. Secure QR Code Scheme Based on Visual Cryptography. 2nd International Conference on Artificial Intelligence and Industrial Engineering, 2016.
8. Falkner S, Kieseberg P, Simos DE, Traxler C, Weipp E. E-voting Authentication with QR-codes. Proceeding of International Conference on Human Aspects of Information Security, Privacy, and Trust, 2014.
9. Cooper K. The iVotronic Voting System Operator's Manual. Election Systems and Software Inc., 2005.
10. Villafiorita A, Fasanelli, G. Transitioning to evoting: the provote project and trentinos experience. Proceedings of EGOV-06, 2006.
11. Open Voting Consortium. EVM 2003: The Electronic Voting Machine Project. Available from:http://evm2003.sourceforge.net/index.html , 2003.
12. McDaniel P, Butler K, Enck W, Hursti H, McLaughlin S, Traynor P, Blaze M, Aviv A, Cerny P, Clark S, Vigna S, Kemmerer R, Balzarotti D, Banks G, Cova M, Felmetsger V, Robertson W, Valeur F, Hall JL, Quilter L. EVEREST: Evaluation and Validation of Election-Related Equipment. Standards and Testing, Ohio Secretary of State's EVEREST Project Report, 2007.
13. Al-Shammari AFN, Weldemariam K, Villafiorita A. Towards an open standard vote verification framework in electronic voting systems. Proceeding of 7th International Conference on Availability, Reliability and Security (ARES), 2012.

التصويت الالكتروني باستخدام ورقة الاقتراع المجهولة

علي فوزي نجم الشمري

قسم هندسة الأطراف والمساند، جامعة كربلاء. العراق.

الخلاصة:

من الامور المقلقة في انظمة التصويت الالكتروني هو امكانية اختراقها بواسطة برامج خبيثة تهدف لتغيير اصوات الناخبين بشكل ذاتي. عده طرق تم اقتراحها للتقليل من وطء هذه المشكلة، ومنها استخدام تقنيات التحقق، وكذلك بالاعتماد على ورقة الاقتراع المجهولة .حيث ان تقنيات التحقق تعمل على التحري عن انتهاكات النظام، اما ورقة الاقتراع المجهولة تعمل على التقليل من امكانية اختراق النظام .في هذا البحث، سوف نستعرض نظام تصويت الكتروني مفتوح المصدر قمنا بتطويره يعتمد على تقنية ورقة الاقتراع المجهولة. النظام المقترح يدعم وصف التصويت بواسطة الشفرة ويستعرض امكانية تقليله لمشكلة محاولة اختراق ماكينة التصويت المذكورة أعلاه. بالإضافة لذلك، فان النظام المقترح يدعم فكرة دعم مجموعة من الانتخابات المختلفة دون الحاجة الى تغيير شفرة البرنامج، وكذلك دعم لغات مختلفة (مثلا العربية والصينية بجانب الانكليزية).

الكلمات مفتاحية: التصويت الالكتروني، مفتوح المصدر، رمز الاستجابة السريعة، التصويت بواسطة الشفرة.