Filing in the gap of Electronic Elections, the JCElection framework

Nikodimos Eustathiou*, Sotirios Kontogiannis*, Stavros Valsamidis* and Alexandros Karakos*
* Department of Electrical and Computer Engineering, Democritus University of Thrace
12, Vas. Sofias, Xanthi, Greece
ne9732@ee.duth.gr, svalsam@ee.duth.gr, skontog@ee.duth.gr, karakos@ee.duth.gr

***Abstract*—The growth of electronic services, on the web, lead to the development of premier web application that try to cover several aspects of electronic democracy. Such aspects are: *electronic elections, electronic debates, electronic pre-election concentrations, public speeches, electronic parliament, electronic government* and other. Electronic elections is nowadays one of the most popular issues of e-democracy and the increasing need for electronic elections. The latter lead to the development of some applications and several security mechanisms to address this necessity. The problem that adheres is that such applications are created either on demand for a specific election process, or experimentally for scientific purposes.**

In order to further assist the development of such applications we developed an open source, generic purpose election application using Joomla CMS framework that can sustain unlimited number of elections. This application was designed to be modular under a well known by developers platform. It's modularity derives from the fact that can serve the non-uniform election processes and requirements of organisations, political parties, governments, groups or even teams. Also different security modules or policies may be applicable accordingly. This is in fact an effort to promote the idea of electronic elections and attract developers, giving them the motive of an open framework, for further development.

***Index Terms*—Electronic democracy, Electronic elections, web based Electronic Elections, Joomla CMS**

I. Introduction

A trusted for delegation electronic voting system has to meet a satisfactory level for a set of standards. These standards are: *Accuracy, Availability, Democracy, Authentication, Privacy* and *Verifiability*. As *accuracy* we mean that it is not possible for a validated vote to be altered, eliminated, or even an invalid vote to be counted in the final tally. The term *Availability* means that the voting system is operational from the beginning till the end of the voting process and must satisfy two opposite requirements of the participating voters: (a) The requirement of proper system functionality during the voter's voting process and (b) the ability to allow the voter to resume an interrupted voting process. The term *democracy* covers the fundamental democratic principles of all voting processes electronic or not that are conducted in democratic societies [5], [12]. That is, only eligible voters can vote and vote only once. Furthermore no electoral entity (administrators, committee), or group of entities, running the election can work in a non democratic way to introduce votes or to prevent voters from voting. This also includes non electoral entities but the ability of a non electoral entity to introduce votes or to prevent voters of an electronic election framework is negligible since it does not control structural parts of the voting process. The term *authentication* covers all those mechanisms that provide non user repudiation, while the term *privacy* assures that neither authorities nor anyone else including the voter himself, is able to prove that the voter voted in a particular way. Finally *verifiability* term covers the ability of anyone to independently verify that all votes have been counted correctly.

There are several studies that pinpoint the problems of electronic elections that use either an electronic embedded machine-software or generic purpose machines and the HTTP protocol over the Internet. Those threats include: network vulnerabilities, delays, correctness, robustness and security of the voting terminal system and electronic elections application-protocol flaws in terms of an application that does not satisfy all of the aforementioned standards [9]. Similar problems also exist on other types of "sensitive" electronic applications such as: e-commerce, e-banking and so forth. Many reports of illegal activities that derive from the use of these application were reported. Nevertheless the development and the increasing need for such services especially over the web continues to rise.

II. Related Work

There are three major types of electronic systems that support electronic elections:

- Manuscript ballot voting systems with PIN or smart-card voter authentication or punch card voting systems. Such systems maintain some characteristics of the classical voting process like the paper voting ballots or the existence of specific areas where the voter may cast his vote.
- Direct recording electronic voting systems (DRE). This category systems are consisted of embedded electronic devices called *voter terminals*, where the voter may authenticate and cast an electronic ballot, and the central server where the ballots are sent. DRE systems completely eliminate paper ballots from the voting process. They also use their own protocol specification to connect to the server, and commonly this specification is hidden inside the embedded *voter terminal* [10]. Nevertheless DRE systems portability, are kept on specific areas where the voter may cast his vote. An example of a DRE voting system is Accuvote-TS DRE system that was written in C++ and designed to run on Windows CE devices, which act as voting terminals [1].

- Web electronic voting systems. These systems use the *WWW*, the HTTP protocol and web browsers as a *voter terminals*. Such systems have the tendency not to keep any of the characteristics of the classical voting process. These systems are also easily established, but due to the wide spread of protocols used and the Internet connectivity among the voter and the voting server, are more susceptible to network delays, attacks and security threats:

The SENSUS web voting system, implemented in Perl and C was the first attempt to develop a modular election application over the Internet. As far as *authentication* and *privacy* is concerned, SENSUS platform, uses RSA keys for encrypting the ballots while they are sent to the voters, and blind signatures when the ballots are sent to the tallier [4]. REVS is a latter implementation in Java of a fault-tolerant voting system designed for voting through the Internet. It uses replication as a basic mechanism to tolerate system failures and RSA keys provided by the administrative group to the voters so as the voters to blindly sign their votes [8], just like SENSUS application. In this paper we deal with Web electronic election voting systems.

III. The JCElection Application

We developed an application called JCElection, based on Joomla CMS framework [11], for the process of conducting elections on the web. More specifically this application is a component consisted of modules-bots that can be installed directly to Joomla CMS from its installation manager. This application can be used to conduct electronic elections that use single, sequential and rated voting methods but not ranked, proportional or semi-proportional ones. These elections may take place on different dates or concurrently. The JCElection application also manages election entities per election, organise election entities to groups and provide a secure voting mechanism for voters. The administration panel of the application is depicted in Fig. 1.

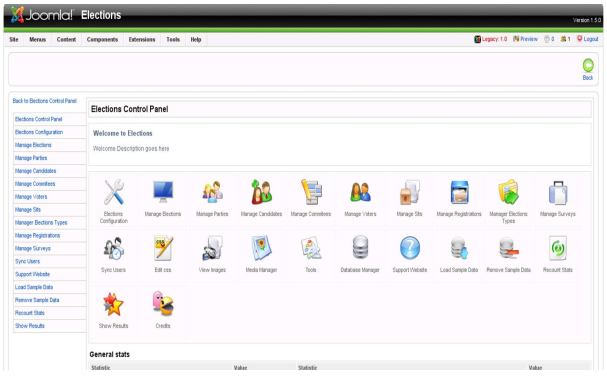

Fig. 1. The JCElection application administration panel

The JCElection application is consisted of the following modules:

The election core module:

This is the core module attached to the JCEletion application and allows the administrator to insert election instances define the attributes per instance and the election method used.

The roles management module:

This module is used by authorities to arrange voters, candidates and committee attributes per election. It can also assign candidates to parties and voters to election groups.

The security and authentication module:

This module is responsible for the user authentication per election process, voter privacy and election process verifiability. That is accomplished with the use of a secret passphrase that the user obtains from the election system in order to enter the application and a set of RSA public-private keys that are given by each election committee to every user in order to participate to a specific election.

The voting front-page module:

This is the front election panel, where the registered voters per election may vote for a specific election.

The committee display module:

This module is responsible for counting the votes of each election process and graphically display the results according to the tallying method. The algorithm for determining the outcome can be modularly selected. This outcome may be a single winner or may involve multiple winners such as in the election of a legislative body. The algorithm may also specify how voting power is distributed among the voters and how the voters are divided into groups whose votes have different weights and counted independently. This module also includes the mechanism that the committee entity members or everyone else may check whether the number of signed votes is equal to the number of participated voters per election.

A. JCElection Application modules

The **election core module.** controls the way each election is instantiated. There are three prebuilt election types and each type controls some other application modules. The main election process attributes defined are: the type, a unique election id, a short description, the voting method, the election date-time limits and the allowed for voting voters entity groups. Each election must be associated with only one election type and this is the important part of the process. According to this association the rest of the process is strictly defined by the constructor in order to create the desired election structure. The three prebuilt election types are:

1) Simple election. This type of election is the simplest prebuilt type and it contains only candidates. Parties and sits are completely disabled by the core module. In this case a candidate can participate with only one election, and cannot participate with any parties or sits.
2) Party-Candidate election. This type contains candidates and parties, but no sits. The functions that associate a candidate to an election and a candidate to a sit are disabled. Only the function that associates a candidate to a party is enabled so that a candidate may only participate in parties. In this case a candidate cannot participate directly to an election or to a sit, but only parties can. When a candidate receives a vote, the candidate's party also receives a vote, so at the end we may have multiple results; per candidate and per party.
3) Sit election. This type of election contains only candidates and sits. Parties are completely disabled. Each

candidate can participate on the election for a specific sit. At the end of this election, for each sit there is a candidate who is proclaimed the winner.

The **roles management module.** JCElection platform is consisted of entities that may participate or not to an election process. The number of entities participating depends strictly on the election type. These entities are: The Joomla users, the voters, the candidates, the administrator, the committee and the election managers. So, the rules management module is a set of independent modules that each one administers each one of the existing platform entities.

The main module is controlled by the application's core module according to the election controls generic attributes-functions. For each election some functions are disabled and some others are enabled accordingly. This module is hidden from all the election entities apart from the manager election entity. As far as management of the platform entities is concerned, the following management modules exist inside the roles management module:

Voters Manager: The Joomla CMS has a prebuilt SQL table for the application's users, and a built-in manager for the users called Users Manager. There is also a prebuilt tool named Sync_Users and this may be used by the election managers for synchronising the user's attributes with the voter's attributes. So each application user may also be a voter. Another available tool is the voters monitor tool called JCmon, which monitors user attributes from the user's SQL table and automatically synchronises all the attributes to the JCElection voter's SQL table.

The Voters Manager module also controls some extra voter attributes such as: voter id, country, group and if the voter is a member to any committee the committee id. The voters personal info is retrieved from the users SQL table. This module also manages the operation that associates a voter with a committee and the function to assign a voter to a group. The default state for this function is enabled, and it can change state to disabled if the election manager wishes to prevent the committee from voting. In this case, if a voter is assigned to a committee, is automatically blocked from the voting procedure.

Party Manager: This module controls the partie's attributes. A party can only be associated to an election set to type that supports parties, and this is mandatory, so the party manager can only be activated if there are at least one election set up to type 2. Some extra attributes controlled by this manager is the name of the party, the cv of the party, and a link to the party's official site.

Candidates Manager: This module manages candidate attributes. These attributes are: candidate code, email, address, telephone, fax, photo, cv and a personal web page url. The rest of the candidate attributes are controlled by the main module. There is also the election function which enables or disables elections set up with type 1 or 2 and the sit function which is only enabled if there are any sits present in the application. The sit function can be combined with the election function, but only if the selected election is set up to type 3, else the manager will deny this with an error message notifying the administrator that this operation is not allowed. Finally the party function is enabled only if there are any parties present in the application. This function cannot be used in combination with the other two functions, and the manager is responsible to deny operation in such case.

Committee manager: The committee manager is not controlled by the election module at all. In JCElection application there can be unlimited election committees but only one committee is allowed to associate with each election. Attributes of this manager are: the committee name, description and a committee code used for the association with elections and voters.

Sits Manager: This manager module is not controlled by the core module. There can be unlimited sits, and unlimited candidates associated with each sit, but each sit name must be unique, else the manager will deny the operation with an error message. The attributes available in this manager are: the sit name, the sit code, and a description about the sit. Tools available with this manager is jcmail, and this tool operates as a security tool for fatal error or suspicious activities, but also it may be used by the voters or the candidates for direct contact with the committee.

Registration manager: This manager module operates as an extended tool in the front-page of the application. For each election there can be only one registration, if this is permitted by the manager in any case. Attributes available here are: registration name, associated election, a description, the time limits, and as non mandatory attributes such as the maximum number of users that may register for an election or the election participating voters groups.

The **voting front-page module.** is responsible for the election output in the front page. What the voters first see upon authentication is a list of available elections, previously set up by the administrator. Depending on the type of each election, there are only three ways a voter can navigate. In each way the voter follows the generated structure and can see only the associated items of each election.

The voting process is divided into three steps. The first step is the authentication system and the exchange of the ballot RSA encryption keys. The second step starts after the voter has successfully logged in. In this step the voter may choose whom to vote according to the election type (sit, candidate, party), or just vote nobody. In this step the administrator can permit the voters to log in multiple time, or force the voters to vote the first time the log in and do not allow any further login from voters who have already voted. After the vote submission the ballot is encrypted with the public key associated to this election process. The third step involves user second authentication process in order for the election authority to blindly sign his encrypted ballot. Note that a voter may enter the system and vote but if the ballot signature is not valid this means that the vote is not taken into account by the election committee. The second type of authentication is described in subsection B.

The **committee display module.** generates one result report for each election and each report is reviewed by the committee members associated with the election. The committee members can make several functionality checks per each election

they are assigned to, before and during the election process. After termination of an election process, the committee may also verify that the votes accounted originate from authenticated voters in total and furthermore that all of those votes are properly signed by the election management entity. This module may even automatically check if there are any non signed votes and compare the total received votes with the total voters.

The voting method used, depends on the election type. For elections type 1 and 3, the single voting method is used. The sequential voting method is also supported, but with the form of two different election instances. The number of votes per ballot can also be defined by the commitee. For election type 2 there are two schemas. In the first schema the party vote is considered valid while in the second invalid.

B. JCElection voters privacy and verifiability

The **security and authentication module** delivers all security options the application can support and these settings are global for all elections. Each security option is implemented by classes that leave inside the security module. There are several proposed secure electronic voting methods that may achieve privacy and verifiability of the voting process. The most commonly used are blind signatures [7], mix nets [2] and homomorphic encryption [3].

With *blind signatures* the user interacts with an authority and after authentication the authority issues a blind signature on the ballot. With *mix nets* a private digital signing key is assigned to each voter. To cast the vote, the voter encrypts his ballot, then signs it and sends it to the tally. When all votes are collected, and the signatures have been verified then the ballot decryption process occurs. The decryption process that follows is successful only when all members of the election committee contribute their part of a secret key that is constructed by some specific permutation. With *homomorphic encryption* the voter encrypts his vote using a public-key cryptosystem as a number and the cryptosystem comes out with a method to calculate the sum of all voted numbers by combining the encrypted messages of the voters.

The security and authentication module is mainly responsible for the user authentication per election process. That is accomplished with the use of a secret passphrase. As a voter privacy mechanism we implemented RSA private-public key exchange between the election committee and the authenticated user in order to encrypt his vote. As a voter verification mechanism we use blind signatures issued from the election committee to the encrypted ballot upon successful voter authentication. This second type of authentication is achieved with the upload from the voters side of an MD5 hash of information involving voter attributes and election attributes. This MD5 hash is generated and sent by e-mail to all election participating voters, by the election committee before the election process. If the voter manages to upload a successful MD5 fingerprint then the election committee issues a blind signature for the voter on his encrypted ballot.

The voter authentication class also controls the available attributes used by the application in order to generate the available RSA public-private keys, but these options are available only to the election committee. The implementation of other cryptographic and signing methods is considered a future work.

IV. Conclusions and future work

The need to perform electronic elections as described in the *FLAK* [6] association statute, lead us to the search of web based, non commercial, open source electronic election applications. Since no such application with the aforementioned characteristics existed, we focused on the development of such application. The result of our effort is the JCElection application framework, a component of Jooma! CMS GPL platform. This component has the advantage to be modular enough due to our effort and since it is developed in a well known by developers framework.

As a future work we plan to extent the JCElection potentials, improving its scalability with the use of a clustered database schema, in order for an election process to have locality aware entities of committee and voter groups, with every group of a particular election to be assigned to a specific cluster election server. Furthermore, in order to cover other forms of voter privacy and verification techniques, *mix nets* procedures are set to be implemented.

References

[1] AVTSCE source tree. (2003) Diebold Election System. [Online]. Available: http://users.actrix.co.nz/dolly/Vol2/cvs.tar

[2] D. Chaum, “Untraceable electronic mail, return addresses and digital pseudonyms,” *Commun. ACM*, vol. 2, no. 24, pp. 84–88, 1981.

[3] R. Crammer, R. Gennaro, and B. Schoenmakers, “A Secure and Optimally Efficient Multi-Authority Election Scheme,” in *Proceedings of Eurocrypt*. Springer Verlag LNCS, 1997, pp. 103–118.

[4] L. F. Cranor and R. K. Cytron, “Sensus: A security-conscious electronic polling system for the internet,” in *Proceedings of the Hawaii Internationa Conference on System Sciences*, 1997, pp. 85 – 93.

[5] W. H. Dutton, A. Elberse, and M. Hale, “A case study of a Netizen’s guide to elections,” *Commun. ACM*, vol. 42, no. 12, pp. 48–54, 1999.

[6] Electronic Democracy Team. (2006) Greek and Cyprus Friends of Open Source Association (FLAK). [Online]. Available: http://www.flak.gr

[7] A. Fujioka, T. Okamoto, and K. Otha, “A practical secret voting scheme for large scale elections,” *Adv. in Cryptology*, pp. 244–251, 1992.

[8] R. Joaquim, A. Zuquete, and P. Ferreira, “Revs - a robust electronic voting system,” in *Proceedings of IADIS International Conference e-Society 2003*, 2003, pp. 95 – 103.

[9] A. Juels, D. Catalano, and M. Jakobsson, “Coercion-resistant electronic elections,” in *WPES ’05: Proceedings of the 2005 ACM workshop on Privacy in the electronic society*. ACM, 2005, pp. 61–70.

[10] T. Kohmo, A. Stubbledield, A. D. Rubin, and D. S. Wallach, “Analysis of an electronic voting system,” in *Proceedings of the 2004 IEEE Symposium on Security and Privacy*, 2004, pp. 76 – 90.

[11] E. Moglen and the Joomla Core Team. (2005, Aug.) The Joomla free open source CMS. [Online]. Available: http://www.joomla.org

[12] R. T. Watson and B. Mundy, “A strategic perspective of electronic democracy,” *Commun. ACM*, vol. 44, no. 1, pp. 27–30, 2001.