Electronic Elections Based on Group Signatures

Lukas Malina, Jan Smrz, Jan Hajny

***Abstract*—This work deals with electronic elections and voting systems. The paper presents a secure electronic voting solution for small and medium groups of voters. The proposed solution is based on modern cryptographic schemes such as ElGamal encryption and a group signature scheme that keeps user privacy, ballot authenticity and confidentiality. The solution offers a user revocation that can be accomplished only by the cooperation of two system entities. The solution is experimentally implemented and tested and the performance results are measured. The results demonstrate that the solution is practical and can be run on various devices such as PCs, laptops, smartphones, etc.**

***Keywords*—Cryptography, electronic election, group signature, privacy, security, voting.**

I. Introduction

Electronic election and voting systems that run via Internet become more and more popular in many nations, e.g., the United States, the UK, Switzerland, Estonia etc. [1]. Many companies and organizations start to use Internet voting to privately elect board members and officers. Internet voting systems which are based on information and communication technologies, e.g. web access via Internet, can significantly speed up the counting of electronic ballots and can provide the remote access for voters in abroad, persons with disabilities etc. Electronic voting systems have to usually provide many phases, e.g. setup, distributing, voting, collecting and ballots counting. These systems have to be secured as well as possible to minimize the possibility of cyber attacks, frauds or privacy leaks. In addition, there are online voting systems which offer the digital interaction between government and citizens as a part of the electronic government (E-government). For example in Estonia [2], the citizens can vote in public elections via Internet. On the other hand, these systems have to keep the privacy for civilians and must be trustworthy for public.

Nowadays e-voting systems have to guarantee the privacy of the votes, preserve the correctness of the results and have to be organized in a trustworthy way. In this paper, we deal with secure and privacy-preserving electronic voting systems which use strong cryptographic primitives. We focus solely on the cryptographic concept that allows voters and election observers to verify that ballots are correctly recorded, tallied and declared. We propose a novel solution that is based on the combination of ElGamal encryption and group signatures. The solution keeps voters in anonymity due to their election pseudonyms and group signatures. This work extends our previous work [3] by the experimental implementation of the proposed solution and by measuring the performance results. The experimental implementation helps to verify the practical use of the solution on current devices and PCs.

This paper is organized as follows: Firstly, Section II presents the related work. Then, we introduce our proposal of a secure digital voting solution which is suitable for the Internet environment in Section III. In Section IV, we present the security analysis of the solution and we evaluate the performance of our solution. In Section V, the implementation of the solution and experimental results are described. The last section (VI) concludes our work.

Authors are with Department od Telecommunications, Brno University of Technology, Brno, Czech Republic, e-mail: malina@feec.vutbr.cz and xsmrzj00@stud.feec.vutbr.cz and hajny@feec.vutbr.cz

Manuscript received November 19, 2015, revised January 14, 2016. Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

II. State of the Art

The first electronic voting scheme was proposed by David Chaum [4] in 1981. Since then many e-voting schemes have been proposed, e.g. e-voting schemes with publicly verifiable secret sharing [5], [6], [7], e-voting based on mix of cryptosystems (ElGamal encryption, threshold public-key cryptosystem, proofs of knowledge) [8], e-voting based on homomorphism encryption [9] or Pailliar's crypto system [10] and e-voting schemes based on the discrete logarithm problem with secret sharing [11] or on secret sharing techniques with a secure multi-party computation [12]. Nevertheless, only several schemes have been tested in practice. The paper [2] describes e-voting experience from Estonian local elections in 2005. The e-voting system uses special ID cards which replace ordinary identity card. These cards are equipped with electronic microchips, which hold personal information about cardholders, two digital certificates and private keys protected by a PIN number. One certificate is used for authentication and the second one is used for digital signature. These cards can be used also by different organizations for different services. A digital double-envelope approach is used to ensure the security and anonymity of voters. The paper [13] describes e-voting experience from Switzerland. In 2004 and 2005, five e-voting pilot trials were studied in order to grapple all security risks, technical risks and possible attacks.

Chaum *et al.* [14] present a practical voter-verifiable election scheme which allows voters to verify the success of their votes. This verification ensures that their digital ballots are included in the poll. Digital ballots are encrypted twice into onion envelopes. The verification of digital ballots is provided by a public bulletin board where voters check that their encrypted ballot receipts appear correctly. Nevertheless, this system has a few disadvantages. For example, if the authority (knowing the associations of all onions) is compromised then it could jeopardize the secrecy of the votes. Another issue is the possibility of double voting, and the new unwished ballot can replace the old ballot. In 2008, the paper [15] presents an e-voting scheme that uses code voting and linkable group signatures. The proposal

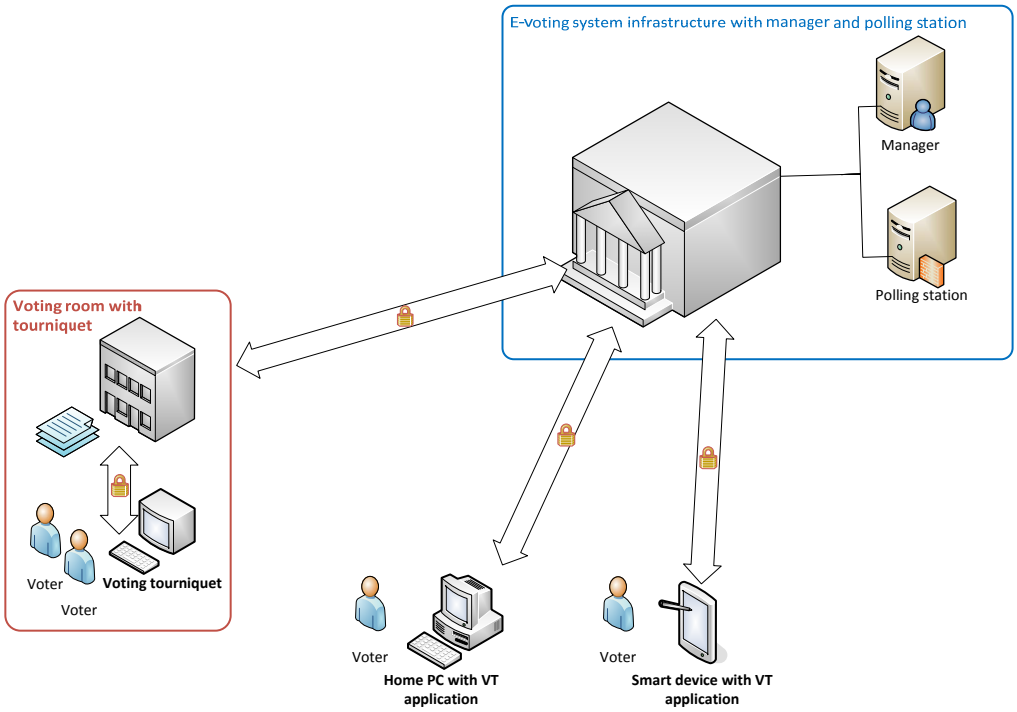

Fig. 1. System model of our e-voting solution.

uses a group signature scheme with two authorities: issuer and opener. Nevertheless, the proposed scheme uses the group signature scheme which requires a secure storage to store a private key on voter's PC. Kremer *et al.* [16] present an election verifiability property in electronic voting protocols. The election verifiability is based on boolean conditions and allows to identify which parts of an e-voting system need to be trusted. Their approach can be applied to systems using blind signatures, homomorphic encryption and mixnets. The paper [17] presents a commitment consistent encryption (CCE) that enables to build the universally verifiable voting schemes with a perfectly private audit trail and practical complexity. CCE is a public key encryption scheme that derives a commitment on the encrypted message and the private key is used to open the commitment. This approach enables to keep the privacy of votes perfectly. The paper [11] presents a practical and secure anonymous Internet voting protocol. The solution is based on a modified ElGamal blind signature scheme and a secret sharing cryptosystem.

In this paper, we propose a solution which is based on the combination of group signatures and probabilistic ElGamal encryption to keep voter in anonymity during the voting and tallying phases. Only the cooperation of a cryptographic manager and an election authority can reveal the user identity and revoke this user from the e-voting system. Further, the proposed solution is implemented to get the performance results.

III. Our Solution of Privacy-preserving E-Voting

In this section, we present our solution of a secure and privacy-preserving electronic voting system based on modern cryptographic schemes. Firstly, we describe a system model which is used in our solution. Then, we introduce cryptographic primitives used in our solution and present the phases of our e-voting solution.

A. System Model

The system model of our solution is depicted in Fig. 1. The proposed solution consists of four system entities (manager, polling station, voting tourniquet and voters) that can be described as follows:

- **Voter (V)** - a person who is able to vote after joining the e-voting system.
- **Manager (M)** - is an entity which manages, sets and generates all system keys and system cryptography parameters. The manager adds/removes all voters into the e-voting system. Before voting, the manager adds all applicants (voters), who are entitled to vote, into a database. The manager generates cryptographic parameters and keys. The manager as the main authority in the system securely keeps all personal information of applicants (full name, address and identification number). The manager can disclose the personal data of revoked voters only in certain election events which require to prevent misusing the system.
- **Polling Station (PS)** - is an entity which organizes elections, generates ballots with the names of the candidates and starts the electoral period. The polling station sends ballots with candidates to the voters after their requests. PS receives and stores the encrypted ballots into a ballot box (database). After the end of the voting period, PS calculates voting results. PS may request the revocation of malicious ballots by sending these data to the manager.

- **Voting Tourniquet (VT)** - is an application which enables voters to interact with the manager and the polling room in the e-voting. VT can be run on users' PC, smartphones or terminals in the voting rooms.

B. Cryptography Used

In our solution the cryptography is based on a group signature scheme, the ElGamal encryption, a secure hash function and the TLS (Transport Layer Security) protocol. The TLS protocol establishes the secure channels between entities in our system model. The ElGamal encryption scheme is used for hiding the content of ballots and for keeping the user privacy. We choose this asymmetric key encryption scheme because it is probabilistic (a single message can be encrypted to many possible ciphertexts). This property is needed for maintaining the user privacy and unlinkability. Further, the non-repudiation, integrity and authentication of ballots have to be provided. Classic signature schemes, e.g., RSA, ECDSA, DSA, provide these properties but do not offer privacy and unlinkability. Hence, we employ Group Signature (GS) schemes because these schemes provide the non-repudiation, integrity and authenticity of signed ballots and keep user privacy and unlinkability. In our scheme, we use the group signature scheme described in [18] which is suitable also for smart phone devices. Nevertheless, we can use other group signature schemes, e.g., [19], [20].

C. Phases of Our Solution

Our solution consists of seven phases: System Setup, User registration, Election Setup, Voter Join, Voting, Tallying and Voter Revocation.

1) System Setup: In this phase, all cryptographic parameters are generated. Between Polling Station (PS) and Manager (M) is established a secure connection via TLS. PS generates a public ElGamal encryption key ($PSpubkey$) and a private ElGamal decryption key ($PSprivkey$) for secure communication.

2) User Registration: Every new user (voter) has to make a registration. The user has to fill in his personal information and provide these data to manager (M). M verifies the voter's personal data, for example, by scanned ID card. Then, M securely stores the personal data into his database and generates authentication credentials V_{ac} (e.g. a login, an authentication code, a password, a certificate) for every member of the system (voter). After the successful registration, the user of the voting system, voter (V), securely obtains his/her authentication credentials V_{ac} and can download a voting tourniquet application. The voter registration phase is depicted in Figure 2.

3) Election Setup: In the Election Setup phase, Polling Station (PS) which organizes elections creates election data (ballots, election start and stop dates, the list of candidates, the number and area of voters, ...). After the creation of election data, PS sends to M the message Start of Election, which contains election data designed for M. M generates group signature scheme parameters such as one public group key ($pubgkey$), n group member private keys for voters

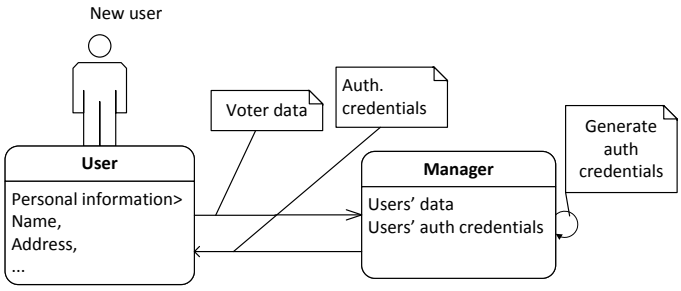

Fig. 2. User Registration phase.

($V_{privgkey}$) where n denotes the number of voters. Further, M generates revocation key ($revgkey$) for this election. M generates voters' election pseudonyms Voter IDs. M sends the list of Voter IDs to PS. PS generates an election encryption public key ($EEpubkey$), an election encryption private key ($EEprivkey$). To secure election encryption it is needed some probabilistic encryption scheme, such as ElGamal. Finally, PS randomly chooses a secret election key ($elsec$). The election setup phase is shown in Figure 3.

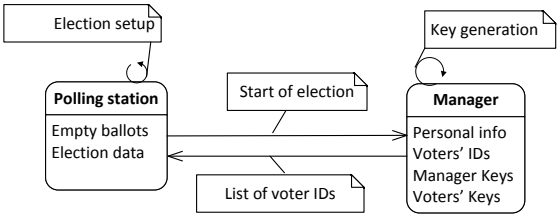

Fig. 3. Election Setup phase.

4) Voter Join: Vouting Tourniquets (VT), which are involved into the election, establish secure connections with manager (M) via TLS. Firstly, voters have to login into the VT terminal/application and connect with M by using his/her authentication credentials V_{ac}. Voters are authenticated by the manager only if they provide valid authentication credentials V_{ac}. The authentication process should permit only few attempts for one voter's login and after the unsuccessful authentication the algorithm holds and the voter's login is blacklisted. After the successful authentication to the manager, V obtains the Voter ID which is needed for the election via different communication channel (secure email communication, encrypted SMS). V also securely gets the group signature private key ($V_{privgkey}$) and group signature parameters (e.g. $pubgkey$), for the voting phase. These data are securely sent by an encrypted connection via TLS that uses authentication credentials V_{ac} to establish the encryption key.

Later, voters use their voters' ID to download ballots from PS. VT sends the encrypted Voter ID to PS by using the public ElGamal encryption key of PS ($PSpubkey$). PS can decrypt this message by the private ElGamal decryption key ($PSprivkey$). The voter who provides the valid voter ID obtains only one empty ballot (bal) that contains the list of candidates. V also obtains an election token (tok) which is unique and is derived from the Voter ID and the secret election key of PS ($elsec$) by a secure hash function h. Finally, V gets

the election encryption public key ($EEpubkey$) from PS. If V does not provide a valid voter ID to PS then PS stops this phase and does not send any ballot, token and public key to the voter. Figure 4 depicts the Voter Join phase with ballots withdrawing.

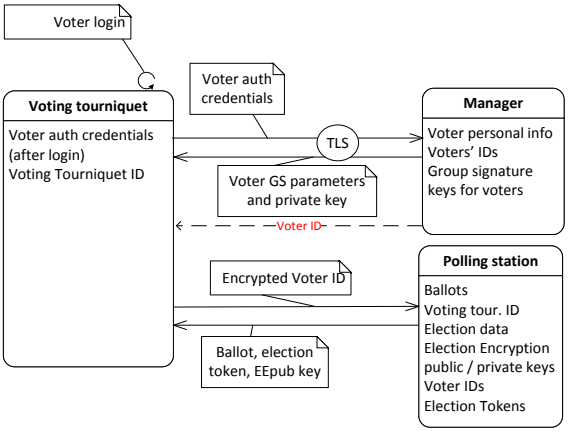

Fig. 4. Voter Join phase

5) Voting: During the voting phase, voters access into Polling Station (PS) which simulates a polling room via the Vouting Tourniquet (VT) application. Voters use their group signature private keys $V_{privgkey}$ to sign the filled ballots ($fbal$) and election tokens (tok). Every voter computes the group signature of the ballot and the election token and encrypts this signature, the ballot and the election token by the election encryption public key ($EEpubkey$) This encrypted message $enc_{EEpubkey}(sig_{V_{privgkey}}(fbal||tok)||fbal||tok)$ is sent to PS. The voting phase is shown in Figure 5.

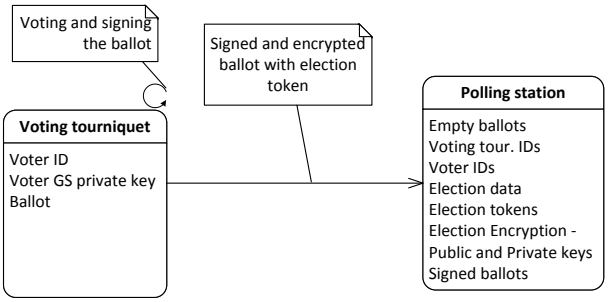

Fig. 5. Voting phase.

6) Tallying: In the tallying phase, the voting phase is stopped. Firstly, PS sends the election stop message to M. After that, M stops providing the group signature private keys to voters who miss the election event. Firstly, PS decrypts the message $enc_{EEpubkey}(sig_{V_{privgkey}}(fbal||tok)||fbal||tok)$ by the private key $EEprivkey$. PS checks via election tokens tok if there are no duplicates (e.g. by checking their receiving time stamp). All newer ballots connected with the election tokens which have been already used are discarded. Then, PS uses the group public key $pubgkey$ to check the signatures $sig_{V_{privgkey}}(fbal + tok)$. Some group signatures enable to employ a batch verification which provides the verification of n signatures in one period. Only a voter with a valid group private key $V_{privgkey}$ is able to sing a message correctly. All ballots with a wrong signature are dropped and not counted by PS. PS counts valid ballots in the final result. The tallying phase is shown in Figure 6.

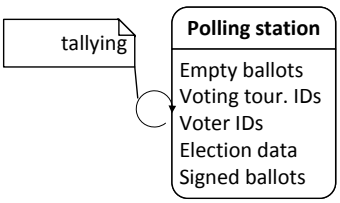

Fig. 6. Tallying phase.

7) Voter Revocation: If malicious voters break the rules of the election than the e-voting system is able to revoke this voter. The voter can be revoked by adding his/her election token to a black list. To disclose a voter identity, PS can send Voter ID to M, which is able to find the user real ID in a database. The manager with PS can revoke the voter for the next election events. Moreover, M is able to determine the user ID from the group signature by the manager revocation key ($revgkey$). Nevertheless, PS must send to M a suspicious signed ballot (e.g. a corrupt ballot). M uses it for identifying the malicious Voter. The voter revocation phase is depicted in Figure 7.

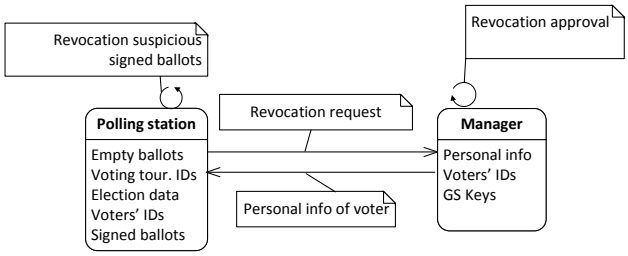

Fig. 7. Voter Revocation phase.

IV. Security Analysis and Evaluation of Our Solution

In this section, we outline the security analysis and the performance evaluation of our solution.

A. Security Analysis

We focus on these main security requirements which are typical for e-voting systems:

- **Ballot correctness and integrity** - Only ballots connected with an election token which are correctly signed are valid. An attacker (A) who wants to modify the ballot has to recompute the group signature of the ballot and election token. The attacker must use a valid group member private key and an election token or a Voter ID. Nevertheless, the voter keeps these parameters in secret. All invalid signed ballots are discarded.
- **Election non-repudiation** - A voter who signs a filled ballot and election token by his/her valid group member private key is not able to deny this action later.

- **Duplicates elimination** - The valid signature of ballots and election tokens are stored in PS voting into a ballot-box database which contains ballots and election tokens. If a signature with the election token which has been already used is sent to PS than PS discards this newer signed ballot and the election token.
- **Ballot non-multiplicity** - The designed e-voting system generates same number ballots as the number of potential voters. The manager of the e-voting system generates and releases only one Voter ID per one voter. The Voter ID is sent to the voter by different communication channel, e.g. via encrypted SMS. The voter needs this Voter ID to obtain an election token from PS and also needs the valid group member private key. Only one valid signed ballot with the one valid election token is counted in the final result.
- **Election privacy** - Only the manager knows voter's identities. In the voting phase, voters send the group signatures that are signed behalf of the group of voters. The group signature contains election token which is connected to the Voter ID. Only PS and the voter is able to connect the election token and Voter ID. The filled ballot and the election token are encrypted by the encryption ElGamal public key. Only PS can decrypt the ballot and the election token. PS is able to detect the Voter ID from the election token, but PS is not able to determine voter identity. On the other hand, the manager is able to detect who signed the filled ballot by the revocation key and determine the voter's group signature private key which leads to the voter's identity. Nevertheless, M is not able to decrypt the encrypted and signed ballots with election tokens that are sent during the voting phase. M and other voters without the valid ElGamal decryption private key of the Polling Station are not able to decrypt signed ballots and election tokens. Then, M is not able to revoke the signed ballot without cooperation with PS. Only the cooperation of PS and M can reveal the real identity of the voter.

B. Performance Evaluation

Table I shows the cryptographic primitives used in our solution in the main phases such as Voter Join, Voting and Tallying. Voting takes one ElGamal encryption (EncEG) and one group signature signing (SigGS). Tallying takes n ElGamal decryptions (DecEG) and n verifications of the group signature (VerGS), where n is a number of signed messages. Generally, ElGamal encryption requires two exponentiations and EG decryption requires one exponentiation. Nevertheless, the number of exponentiations depends on the total length of the message m where m has to be split and converted into elements $m_i \in G$ of order q. The length of the message m depends on the size of the group signature, e.g. 2636 bits if the GS scheme [18] is used (ca. 1500 bits with the BBS04 GS scheme [19]), the size of the election token, e.g. 256 bits if SHA-256 hash is used, and the size of the ballot, e.g. 256 bits if the election has 256 candidates.

The performance of group signatures depends on the scheme which is used. For example, the signing in the GS scheme [18] takes 0 pairing operation (3 pairings can be precomputed) and 9 exponentiations and the verification takes 5 pairings and 10 exponentiations. Nevertheless, this GS scheme [18] supports a batch verification which reduces the pairing operations from $5 * n$ to 2.

TABLE I
CRYPTOGRAPHIC PRIMITIVES USED IN OUR E-VOTING SOLUTION

Phase	VT	M	PS
Voter Join with M	TLS	TLS	-
Voter Join with PS	EncEG	-	DecEG + hash
Voting	EncEG + SigGS	-	-
Tallying	-	-	n (DecEG + VerGS)

The batch verification verifies many signatures in one process. The batch verification reduces the number of bilinear pairing operations e from $n * k$ to l, where n is the number of signatures, k is the number of bilinear pairing operations during an individual message verification, and l is the number of bilinear pairing operations during the batch verification. Equation (1) describes pairing operations in the batch, where $f_i \in G_1, h_i \in G_2, c_i \in Z_q^*$ are parameters (e.g. elements mapped in groups and the finale field of integers' modulo q) for each i signature from the total number of signatures n, and, A is a constant value which is known by a verifier and does not depend on the concrete parameters of signatures.

$$e(\prod_{i=1}^{n} {f_i}^{c_i}, h_i) = A \tag{1}$$

If all signatures are valid, then the batch verification is valid. If one signature is invalid, then the batch verification is invalid and the computational complexity of the batch verification, which is linear in case of presence n valid signatures, degrades to logarithmic. The batch must be split to two batches that are verified again separately. This procedure is performed until all invalid signatures are detected.

V. IMPLEMENTATION AND EXPERIMENTAL RESULTS

This section describes the implementation of the proposed solution and presents the performance results of main solution phases.

A. Implementation of Our E-voting Solution

The proposed solution is implemented by the JAVA programing language (JDK 1.8). The solution runs in Java Runtime Environment 8 and consists of three projects that represent system entities: Manager, Voting tourniquet and Polling station. The graphical user interfaces for Manager, Voting tourniquet, Polling station and Ballot are depicted in Fig. 8. The Manager application can add and revoke voters. The Polling Station application sets, starts and ends election events. The Voting tourniquet application enables to choose election candidates and create ballots. The Ballot form contains the ballot ID, token, time of election, and if the ballot is signed and verified correctly.

The standard cryptographic schemes and key builders are implemented by methods specified in javax.crypto.*, java.security.*. The advanced cryptographic schemes (e.g.

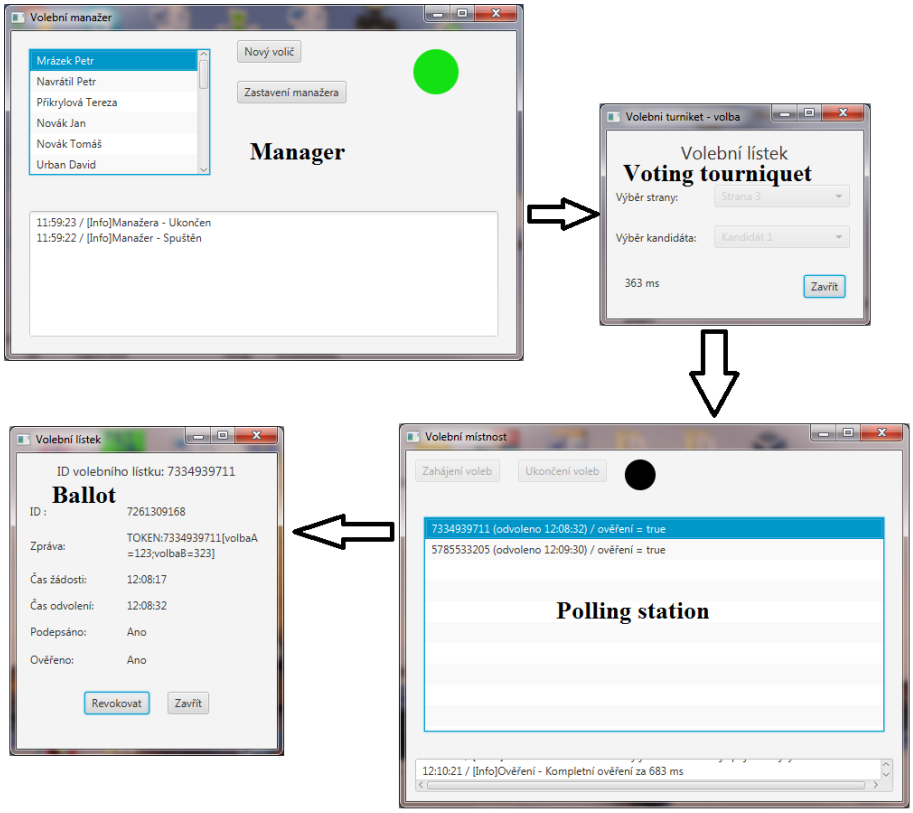

Fig. 8. Graphical user interfaces of entities during election event.

the BBS group signature scheme) are implemented by using external libraries, i.e. the bouncy castle library (available on www.bouncycastle.org) and the Java Pairing Based Cryptography (jPBC) Library (available on http://gas.dia.unisa.it/projects/jpbc/index.html). The implementation of the BBS scheme uses the MNT curves type D with the embedding degree $k = 6$, the 175-bit order of curves and the pre-generated parameters d840347-175-161.param.

B. Performance Results

We focus on main phases that are Election setup, Voting and Tallying because these parts employ the group signature scheme (BBS). The BBS scheme represents the most time and memory expensive cryptographic method that is used in our solution. The voting phase takes about 380 ms per one ballot on PC (Intel(R) Core(TM) i5-4210U CPU @ 1.7 GHz 2,4 GHz, 8 GB RAM, Windows 8.1 Pro) where the most time consuming part is the BBS group signature signing phase. The results of the BBS scheme, which is implemented and measured on smartphones in the paper [21], show that current smartphones need several seconds to generate the BBS signature. In an election event, the voter can use a smartphone to sign, encrypt and send the ballot but these devices takes several seconds only for computing one group signature.

The tallying phase is performed on the Polling Station that is simulated by PC (Intel(R) Core(TM) i5-4210U CPU @ 1.7 GHz 2,4 GHz, 8 GB RAM, Windows 8.1 Pro). The most time consuming part of this phase is the verification phase of the BBS group signatures. The BBS verification phase takes about 430 ms without any optimization tricks.

Figure 9 shows the performance of group member key generation for n voters and the tallying phase for n ballots. The results demonstrate that the proposed solution is practical for small (up to hundreds members) and medium (up to several thousands members) groups of voters because the tallying of 10 ballots takes about 2.6 s and the tallying of 1000 ballots takes about 4 minutes which is reasonable time in practice during an election event. We assume that servers for counting and verifying ballots can be more powerful than used PC in order to improve the performance of the tallying phase. Moreover, the verification process of group signatures can be optimized by several tricks, e.g. precomputation, batch verification, that are studied in [21], [22] and [23].

During the Election Setup phase, group member keys used by voters have to be generated in a practical time. The results show that the key generation takes about 1.5 s for 10 voters and about 38.4 s for 1000 voters.

VI. Conclusion

In this paper, we present the electronic voting solution which provides secure elections and keeps the user privacy. Our solution is based on group signature schemes that ensures non-repudiation, integrity and authenticity and ElGamal encryption ensures the confidentiality of the ballots. The performance of voting and tallying phases depends mainly on the group signature scheme. The implementation of the proposed

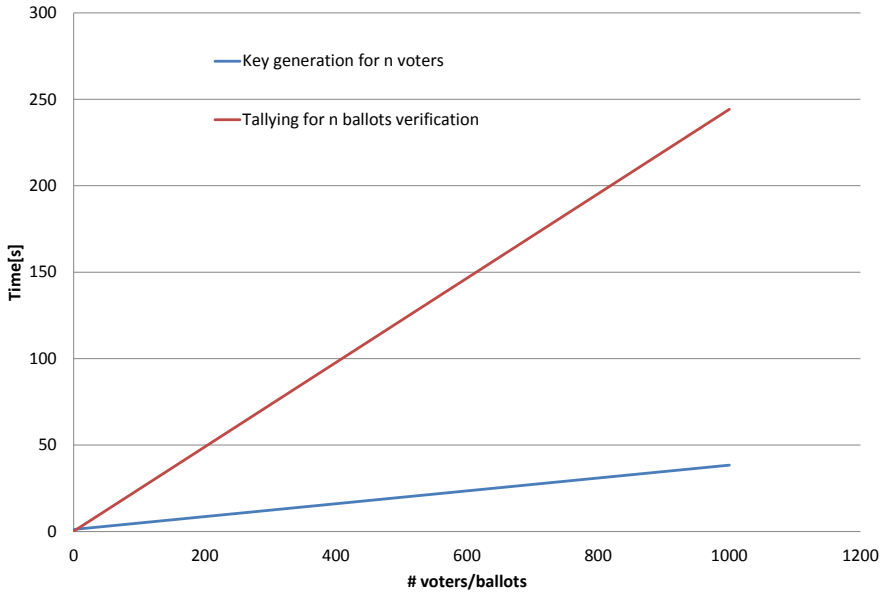

Fig. 9. Performance of group member key generation and the tallying phase.

solution with the BBS group signature scheme demonstrates that the performance results are practical on current PCs if the number of voters and ballots are from hundreds to thousands. Nevertheless, the proposed e-voting solution can use a different group signature scheme that can be more efficient than the BBS scheme. Verifying the signed ballots can be optimized by employing the group signature scheme that supports the batch verification. These optimization tricks can improve the performance of the tallying phase that runs in the polling station.

REFERENCES

[1] J. Budurushi and M. Volkamer, "Feasibility analysis of various electronic voting systems for complex elections," in *Conference for E-Democracy and Open Governement*, 2014, p. 141.

[2] Ü. Madise and T. Martens, "E-voting in estonia 2005. the first practice of country-wide binding internet voting in the world," *Electronic Voting*, vol. 86, 2006.

[3] L. Malina, J. Smrz, J. Hajny, and K. Vrba, "Secure electronic voting based on group signatures," in *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*. IEEE, 2015, pp. 6–10.

[4] D. L. Chaum, "Untraceable electronic mail, return addresses, and digital pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, 1981.

[5] B. Schoenmakers, "A simple publicly verifiable secret sharing scheme and its application to electronic voting," in *Advances in Cryptology-CRYPTO99*. Springer, 1999, pp. 148–164.

[6] B. L. K. Kim, "Receipt-free electronic voting through collaboration of voter and honest verifier," 2000.

[7] C. A. Neff, "A verifiable secret shuffle and its application to e-voting," in *Proceedings of the 8th ACM conference on Computer and Communications Security*. ACM, 2001, pp. 116–125.

[8] A. Juels, D. Catalano, and M. Jakobsson, "Coercion-resistant electronic elections," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, ser. WPES '05. New York, NY, USA: ACM, 2005, pp. 61–70. [Online]. Available: http://doi.acm.org/10.1145/1102199.1102213

[9] M. Hirt and K. Sako, "Efficient receipt-free voting based on homomorphic encryption," in *Advances in CryptologyEUROCRYPT 2000*. Springer, 2000, pp. 539–556.

[10] I. Damgård, M. Jurik, and J. B. Nielsen, "A generalization of pailliers public-key system with applications to electronic voting," *International Journal of Information Security*, vol. 9, no. 6, pp. 371–385, 2010.

[11] C.-L. Chen, Y.-Y. Chen, J.-K. Jan, and C.-C. Chen, "A secure anonymous e-voting system based on discrete logarithm problem," *Appl. Math*, vol. 8, no. 5, pp. 2571–2578, 2014.

[12] D. G. Nair, V. P. Binu, and G. Santhosh Kumar, "An improved e-voting scheme using secret sharing based secure multi-party computation," 2014.

[13] N. Braun and D. Brändli, "Swiss e-voting pilot projects: Evaluation, situation analysis and how to proceed." *Electronic Voting*, vol. 86, pp. 27–36, 2006.

[14] D. Chaum, P. Y. Ryan, and S. Schneider, *A practical voter-verifiable election scheme*. Springer, 2005.

[15] J. Helbach, J. Schwenk, and S. Schäge, "Code voting with linkable group signatures," in *Electronic Voting*, 2008, pp. 209–208.

[16] S. Kremer, M. Ryan, and B. Smyth, *Election verifiability in electronic voting protocols*. Springer, 2010.

[17] E. Cuvelier, O. Pereira, and T. Peters, "Election verifiability or ballot privacy: Do we need to choose?" in *Computer Security–ESORICS 2013*. Springer, 2013, pp. 481–498.

[18] L. Malina, A. Vives-Guasch, J. Castellà-Roca, A. Viejo, and J. Hajny, "Efficient group signatures for privacy-preserving vehicular networks," *Telecommunication Systems*, pp. 1–19, 2014.

[19] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Advances in Cryptology–CRYPTO 2004*. Springer, 2004, pp. 41–55.

[20] L. Malina and J. Hajny, "Efficient security solution for privacy-preserving cloud services," in *Telecommunications and Signal Processing (TSP), 2013 36th International Conference on*. IEEE, 2013, pp. 23–27.

[21] L. Malina, J. Hajny, and V. Zeman, "Usability of pairing-based cryptography on smartphones," in *Telecommunications and Signal Processing (TSP), 2015 38th International Conference on*. IEEE, 2015, pp. 617–621.

[22] A. L. Ferrara, M. Green, S. Hohenberger, and M. Ø. Pedersen, "Practical short signature batch verification," in *Topics in Cryptology–CT-RSA 2009*. Springer, 2009, pp. 309–324.

[23] L. Malina, J. Hajny, and V. Zeman, "Trade-off between signature aggregation and batch verification," in *Telecommunications and Signal Processing (TSP), 2013 36th International Conference on*. IEEE, 2013, pp. 57–61.