A Practical Solution against Corrupted Parties and Coercers in Electronic Voting Protocol over the Network

Thi Thao Nguyen, Tran Khanh Dang

A Practical Solution Against Corrupted Parties and Coercers in Electronic Voting Protocol over the Network

Ai Thao Nguyen Thi and Tran Khanh Dang

Faculty of Computer Science and Engineering, HCMC University of Technology
268 Ly Thuong Kiet Street, District 10, Ho Chi Minh City, Vietnam

{thaonguyen, khanh}@cse.hcmut.edu.vn

Abstract. In this paper, we introduce a novel electronic voting protocol which is resistant to more powerful corrupted parties and coercers than any previous works. They can be the voting authorities inside the system who can steal voters' information and the content of their votes, or the adversaries outside who try to buy the votes, and force voters to follow their wishes. The worst case is that the adversaries outside collude with all voting authorities inside to destroy the whole system. In previous works, authors suggested many complicated cryptographic techniques for fulfilling all security requirements of electronic voting protocol. However, they cannot avoid the sophisticated inside and outside collusion. Our proposal prevents these threats from happening by the combination of blind signature, dynamic ballots and other techniques. Moreover, the improvement of blind signature scheme together with the elimination of physical assumptions makes the newly proposed protocol faster and more efficient. These enhancements make some progress towards practical security solution for electronic voting system.

Keywords: Electronic voting, blind signature, dynamic ballot, uncoercibility, receipt-freeness.

1 Introduction

Along with the rapid growth of modern technologies, most of the traditional services have been transformed into remote services through internet. Voting service is among them. Remote electronic voting (also called e-Voting) system makes voting more efficient, more convenient, and more attractive. Therefore, many researchers have studied this field and tried to put it into practice as soon as possible. However, that has never been an easy work. It is true that e-voting brings many benefits for not only voters but also voting authorities. Nevertheless, benefits always come along with challenges. The biggest challenge of e-voting relates to security aspects.

In previous works, authors proposed several electronic voting protocols trying to satisfy as many security requirements as possible such as eligibility, uniqueness, privacy, accuracy, fairness, receipt-freeness, uncoercibility, individual verifiability, universal verifiability. However, security leaks cannot be rejected thoroughly in recent

electronic voting protocols when voting authorities collude with each other. In the protocol of Cetinkaya et al [5], for example, though the authors announced that their protocol fulfilled the requirement of uncoercibility, once the adversaries corrupted the voter and colluded with the voting authorities taking responsibilities of holing ballots and voter's cast, they could easily found out whether that voter followed their instruction or not. In addition, in voting protocol of Spycher et al [1] and JCJ protocol [6], if the coercer can communicate with the registrars, no longer can voter lie about their credentials. Therefore, the uncoercibility cannot be satisfied. Moreover, in order to satisfy the receipt-freeness, some protocols employed the physical assumptions such as untappable channels that are not suitable for the services through internet.

Most of the previous electronic voting protocols applied three main cryptographic techniques to solve the security problems. Thus, we classify these protocols into three types as: protocols using mix-nets, blind signatures, and homomorphic encryption. The concept of mix-nets was firstly introduced by Chaum in [11]. Since then, there have been some proposed voting protocols such as [7]. However, these protocols met with the big difficulties because of the huge costs of calculations and communications which the mix-net required. Moreover, the final result of voting process is dependent on each linked server in mix-net. If any linked server is corrupted or broken, the final result will be incorrect. So far, no election system based on mix-net has been implemented [13]. Besides mix-net, homomorphic encryption is another way to preserve privacy in electronic voting system. Though homomorphic encryption protocols like [9][14] are more popular than mix-net, they are still inefficient for large scale elections because computational and communicational costs for the proof and verification of vote's validity are quite large. In addition, homomorphic encryption protocols cannot be employed on multi-choices voting forms. As for the blind signature protocols, they also provided anonymity without requiring any complex computational operators or high communicational cost. Until now, there have been many protocols based on blind signature such as [5][8][10][15]. Some of them employed blind signature on concealing the content of votes, others concealed the identifications of voters. Protocol [10], for example, conceals the content of votes; then at the end of voting process, voters had to send the decryption key to the voting authority. This action might break the security if the adversaries conspired with these voting authorities. Therefore, our proposal applies the blind signature technique which is used to hide the real identification of a voter. Besides that, in order to protect the content of votes we apply dynamic ballots along with a recasting mechanism without sacrificing uniqueness to enhance security in the electronic voting protocol and make good the previous protocol 's shortcomings as well.

In this paper, we propose an inside and outside collusion-free electronic voting protocol which guarantees all security requirements. The remarkable contribution is that our proposal is able to defeat the more powerful adversaries which can collude with most of the voting authorities. Another improvement is the enhancement of blind signature scheme that makes our protocol faster and more efficient.

The structure of this paper is organized as follows. In Section 2, we summarize the background knowledge of electronic voting. We describe the details of our proposal

protocol in Section 3. Then, in Section 4 security of the protocol is discussed. Finally, the conclusion and future work are presented in Section 5.

2 Background

2.1 Security Requirements

According to [8], the security requirements of electronic voting system are introduced as follows: (1) privacy: no one can know the link between the vote and the voter who casted it; (2) eligibility: only eligible and authorized voters can carry out their voting process; (3) uniqueness: each voter has only one valid vote; (4) accuracy: the content of vote cannot be modified or deleted; (5) fairness: no one, including voting authorities, can get the intermediate result of the voting process before the final result is publicized; (6) receipt-freeness: the voting system should not give voter a receipt which he uses to prove what candidate he voted; (7) uncoercibility: the adversary cannot force any voters to vote for his own intention or to reveal their votes; (8) individual verifiability: every voter is able to check whether their vote is counted correctly or not; (9) universal verifiability: every voter who is interested in tally result can verify it is correctly computed from all the ballots casted by eligible voters or not.

2.2 Cryptography Building Block

Bulletin Boards. In [2], Bulletin board is a communication model which can publish information posted on its body, thus everybody can verify these information. Electronic voting system applies this model to fulfill the requirement of verifiability. In the protocol using bulletin board, voters and voting authorities can post information on the board. Nevertheless, no one can delete or alter these things.

Blind Signature. The concept of blind signature was first introduced by Chaum in 1982. It stemmed from the need of verifying the valid of a document without revealing anything about its content. A simple method to implement the blind signature scheme is to apply the asymmetric cryptosystem RSA. We have some notations: (1) m: the document needs to be signed; (2) d: the private key of authority (signer); (3) (e, N): the public key of authority; (4) s: the signature of m.

The RSA blind signature scheme is implemented as follows:

The owner generates a random number r which satisfies $gcd(r, N) = 1$. He blinds m by the blind factor $r^e \pmod N$. After that, he sends the blinded document $m' = m.\ r^e \pmod N$ to the authority. Upon receiving m', the authority computes a blinded signature s', as illustrated in Eq. (1), then sends it back to the owner.

$$s' \equiv (m')^d \bmod N \equiv (m.\ r^e)^d \bmod N \equiv (m^d.\ r^{ed}) \bmod N \equiv (m^d.\ r) \bmod N \quad (1)$$

According to Eq. (1), the owner easily obtains the signature s, as Eq. (2).

$$s \equiv s' r^{-1} \bmod N \equiv m^d \bmod N \quad (2)$$

Dynamic Ballot. The concept of dynamic ballot was introduced in [5]. This is a mechanism that helps voting protocol fulfill the requirement of fairness. In most of e-voting protocols, authors have used usual ballots in which the order of candidates is

pre-determined. Therefore, when someone gets a voter's casting, they instantly know the actual vote of that voter. Alternatively, the candidate orders in dynamic ballot change randomly for each ballot. Hence, adversaries need the voter's casting as well as the corresponding dynamic ballot in order to obtain the real choice of a voter.

In voting process, each voter can randomly take one of these ballots. He chooses his favorite candidate. Then he casts the order of this candidate in his ballot (not the name of this candidate) to a voting authority and his ballot to another voting authority.

Plaintext Equality Test (PET). The notion of PET was proposed by Jakobsson and Juels [4]. The purpose of PET protocol is to compare two ciphertexts without decrypting. It based on the ElGamal cryptosystem [3].

Let $(r_1, s_1) = (a^{y1}, m_1.a^{x.y1})$ and $(r_2, s_2) = (a^{y2}, m_2.a^{x.y2})$ be ElGamal ciphertexts of two plaintexts m_1 and m_2 respectively. The input I of PET protocol is a quotient of ciphertexts (r_1, s_1) and (r_2, s_2), and output R is a single bit such that $R = 1$ means $m_1 = m_2$, otherwise $R = 0$.

$$I = \left(\frac{r_1}{r_2}, \frac{s_1}{s_2}\right) = \left(a^{y_1 - y_2}, \frac{m_1}{m_2}.a^{x(y_1 - y_2)}\right)$$

According to ElGamal cryptosystem, I is the ciphertext of the plaintext (m_1/m_2). Therefore, if someone who owns the decryption key x, they can obtain the quotient of m_1 and m_2 without gaining any information about the two plaintexts m_1 and m_2.

3 The Proposed Electronic Voting Protocol

3.1 Threats in Electronic Voting Protocol

Vote Buying and Coercion. In a traditional voting system, to ensure a voter not to be coerced or try to sell his ballot to another, voting authorities built some election precincts or kiosk in order to separate voters from coercers and vote buyers. Therefore, they could vote based on their own intentions. When electronic voting system is brought into reality, there are no election precincts or voting kiosks, but voters and their devices which can connect to the internet. Hence, the threats from coercers and vote buyers quickly become the center of attention of the voting system.

Corrupted Registration. Registration is always the first phase of a voting process where voting authorities check voters' eligibilities and give voters the certificates to step into the casting phase. However, in case a voter abstains from voting after registration, the corrupted registrars can take advantages of those certificates to legalize the false votes by casting the extra vote on behalf of the abstaining voters. Sometimes, corrupted registrars can issue false certificates to deceive other voting authorities.

Corrupted Ballot Center. Some protocols have a ballot center as providing voters with ballots. Others, in [5], utilize it for holding the choices that voters made until the casting phase completes. If the ballot center becomes a corrupted party, it can modify the content of the votes or sell them to vote-buyers and coercers who want to check whether the coerced voters cast the candidate they expect. Hence, a feasible electronic voting protocol has to possess the mechanism to protect the system against this threat.

Corrupted Tallier. Tallier takes the responsibility for counting up all the votes to get the final result of voting process. If tallier becomes a corrupted party, it will be able to

do that job though the voting process does not come to the end. In this case, it will release the intermediate voting result which has the influence on the psychology of the voters who have not casted the ballots yet. This threat makes the fairness fail.

3.2 The Proposed Electronic Voting Protocol

Before explaining each step in the protocol, we introduce some notations: (1) (e_x, d_x): a public-private key pair of user X; (2) $E_x(m)$: an encryption of *m* with the public key e_x; (3) $D_x(m)$: a decryption/sign of *m* with the private key d_x; (4) H(m): an one way hash function with an input *m*; (5) $E_{PET}(m)$: an encryption of *m* using ElGamal cryptosystem; (6) PET(x, y): a PET function applying PET protocol with two inputs *x, y*.

Registration Phase. In this phase, the blind signature technique is applied to conceal the real identity of a voter through creating an anonymous identity for communicating with other voting authorities. The following paragraphs will show how voters get their anonymous identification from *Privacy of Voter server* (hereafter called *PVer*).

Firstly, the voter sends his real ID to *Registration server* (hereafter called *RS*) to start registration process. Based on the real ID, *RS* checks whether that user is registered or not. If he did this job before, *RS* will terminate his session; otherwise, *RS* will ask *CA* to check the current rules of the voting process in order to find out whether this person can become an eligible voter or not. Then, *RS* creates a certificate and sends it to the voter. This certificate includes: a serial number, a digital stamp, a session key, a signature of *RS*.

Upon receiving the certificate, voter generates his unique identification number:

$$uid = Hash(D_V(\text{Digital stamp}))$$

To get the signature of a voting committee on uid, a voter applies the blind signature technique as introduced in Section2.2. He uses a random blind factor to blind uid, and then sends it together with the certificate to *PVer*, which takes the responsibility for preserving privacy of voters. *PVer* saves the serial number in certificate in order to ensure that each certificate asks for the blind signature just one time. After checking the validity of certificate, *PVer* blindly signs the uid, then send the result s' to the voter. He, then, unblinds s' to get the signature s of the voting committee on his uid. Since then, the voter sends uid and corresponding s to other voting authorities for authentication. The detail steps are illustrated in Fig.1.

To avoid man-in-the-middle-attacks, the asymmetric cryptosystem is used at the 1^{st}, 6^{th}, and 8^{th} steps. However, at the 10^{th} step, asymmetric key pairs are not a good choice because they are used only one time for encrypting message, not authenticating. Therefore, the symmetric-key cryptosystem with Tripple DES algorithm is proposed in this blind signature scheme because it has some significant benefits: (1) it does not consume too much computing power so we can shorten encryption time and simplify the period of encryption certificate as well; (2) although symmetric encryption is not as safe as an asymmetric encryption, high level of security still be guaranteed for some reasons that: Triple DES has high complexity, the session key generated randomly by system is long enough to against Brute Force and Dictionary Attack, and the period of using session key is limited in one step with a short time.

Another improvement of this blind signature scheme is that a voter generates list of anonymous identifications including uid, uid_1, and uid_2 instead of just one. The purpose of uid is to communicate with other voting servers; and uid_1 and uid_2 are to ensure the dynamic ballot of voter is not modified by any adversaries.

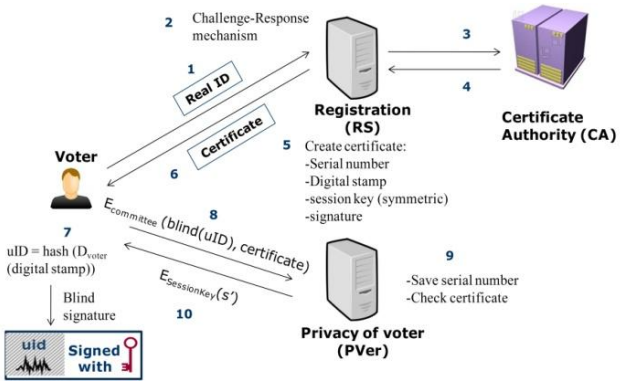

Fig.1. Registration scheme.

Authentication and Casting Phase. To protect privacy of votes from coercers, voting buyers, or sometimes adversaries who stay inside the system, we propose the scheme as shown inFig.2, which applies dynamic ballots, plaintext equivalent test, bulletin boards as introduced in Section 2.2.

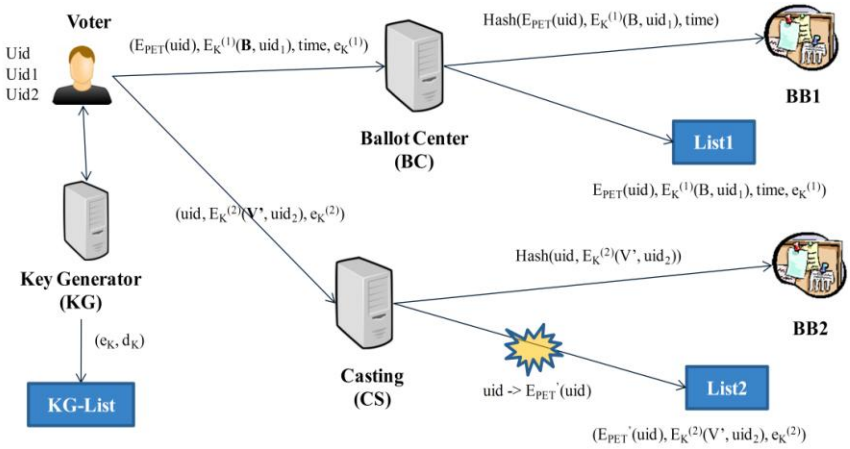

Fig.2. Casting scheme

In previous works, to avoid coercion as well as vote buying, authors apply a fake identification mechanism to deceive coercers (in JCJ protocol [6]); and others utilize a recasting mechanism without sacrificing uniqueness to conceal voters' final decision [5]. The fake identification requires the condition that at least one voting authority knows the real identification of a voter in order to determine what the real votes are.

Therefore, if this voting authority becomes adversary, the requirements of uncoercibility and vote-buying can be violated. As a result, our proposal uses recasting mechanism to achieve higher level of security.

Firstly, an eligible voter receives the list of candidates from *Ballot Center* (called *BC*). He then, mixes the order of candidates randomly, sends this dynamic ballot B to *BC* and casts his cloaked vote V' by picking the order of the candidate in B he is favor, and sending it to the *Casting server* (called *CS*). To ensure B and V' cannot be modified by others, the voter encrypts them with uid_1 and uid_2 by the couple of public keys $e_k^{(1)}$ and $e_k^{(2)}$ generated by *Key Generator server* (called *KG*). *KG* also saves the private key d_k along with corresponding e_k in KG-List for decrypting in the next phase. After that, voter sends ($E_{PET}(uid)$, $E_k^{(1)}(B, uid_1)$, time, $e_k^{(1)}$) to *BC*, and (uid, $E_k^{(2)}(V', uid_2)$, $e_k^{(2)}$) to *CS* as illustrated in Fig.2. Each message receiving from voter, *CS* checks uid of this voter. If uid is invalid, *CS* discards it; otherwise, it hashes the whole message, and publishes the result on the Bulletin Board BB2 for individual verifiability. It also stores the message into List2 for matching with B in tallying phase. As for *BC*, it does the same things with every message it receives, except authenticating the eligibility of voters.

Voters are allowed to recast. Because the actual vote V of a voter consists of B and V', voters just need to change one of two components to modify the value of V. In this protocol, voters are able to change the orders of candidates in their dynamic ballot B. In order to recast, a voter sends another message ($E_{PET}^*(uid)$, $E_k^{(1)}(B^*, uid_1)$, time*, $e_k^{(1)}$) to *BC* in which $E_{PET}^*(uid)$, B* and time* are respectively new ElGamal encryption of uid, new dynamic ballot and the time when he sends the message.

Tallying Phase. At the end of casting phase, *PET* server applies PET protocol in Section 2.2 to each $E_{PET}(uid_i)$ in List1. The purpose is to find which pair of encryptions of uid_i and uid_j is equivalent without decryption. After that, *PET* server removes the record holding the earlier time parameter. Concretely, we consider two records R_i and R_j of List1:

$$R_i = (E_{PET}(uid_i), E_{Ki}^{(1)}(B_i, uid_{1i}), time_i, e_{Ki}^{(1)})$$
$$R_j = (E_{PET}(uid_j), E_{Kj}^{(1)}(B_j, uid_{1j}), time_j, e_{Kj}^{(1)})$$

If $PET(E_{PET}(uid_i), E_{PET}(uid_j)) = 1$, and $time_i > time_j$; then the system removes R_j from the system. The purpose of this process is to remove duplicated votes and gain the latest choices of all voters. After that, *PET* server continues to compare each $E_{PET}(uid_i)$ in List2 to each $E_{PET}(uid_j)$ in List1 to find out which B in List1 is corresponding to V' in List2. If there exists a record in List1 which does not match with any record in List2, this record must have come from an invalid voter, so it is discarded at once. The purpose of this process is to remove invalid dynamic ballots B in List1.

After determining pairs of records, KG-List publishes the list of session keys (e_K, d_K) for List1 and List2 to find d_K related to each e_K which is attached to every record in List1 and List2. With the corresponding d_k, $E_K^{(1)}(B, uid_1)$ and $E_K^{(2)}(V', uid_2)$ are decrypted. *Tallying server* (called *TS*) checks the valid of uid_1 and uid_2 to ensure B and V' not to be modified by any parties, then combines the valid values of B and V' to find out the actual vote V of a voter. Finally, *TS* counts up all the actual votes and publishes the result of voting process.

4 Security Analysis

In this section, we provide the security analysis of our proposal and draw the comparisons with the previous typical electronic voting protocols.

Table 1. Comparing the earlier typical protocols with our proposal.

Security flaw/requirement	Hasan [12]	Cetinkaya [5]	JCJ [6]	Our protocol
No Vote buying/ Coercion	-	-	√	√
No corrupted *RS*	√	√	-	
No corrupted *BC*	-	-	√	
No corrupted *Tallier*	-	-	√	
No physical assumption	√	√	-	
Privacy	√	√	√	
Eligibility	√	√	-	
Uniqueness	√	√	√	
Uncoercibility	-	-	√	
Receipt-freeness	-	-	√	
Accuracy	-	√	√	
Fairness	-	√	√	
Individual verifiability	-	√	√	
Universal verifiability	-	√	√	

In our protocol, a voter employs the blind signature technique to get the voting authority's signature on his created identity. Therefore, the *RS* and *PVer* do not know anything about the anonymous identity that voters use to authenticate themselves. Hence, if these voting authorities become corrupted, they cannot take advantages of abstention to attack the system. So do the protocols of Hasan [12] and Cetinkaya [5]. In JCJ protocol [6], the *RS* establishes the credentials and passes it to voters through an untappable channel. In the worst case, if the *RS* is a corrupted party, it can give voters fake credentials, and use the valid ones to vote for other candidates. Thus, the corrupted *RS* becomes a security flaw of JCJ protocol. Using physical assumption, i.e. an untappable channel, is another weak point of JCJ protocol in comparison with the previously proposed protocols.

In the voting protocols of Hasan [12] and Cetinkaya [5], though eliminating the abstention attack from corrupted *RS*, these protocols are not stronger enough to defeat sophisticated attacks. The voting protocol of Hasan is quite simple; it has no mechanism to protect the content of votes against being modified. Thus, if *CS* or *TS* collude with attackers, the system will collapse. As a result, the accuracy and fairness properties cannot be guaranteed. In ideal case which every server is trusted, the protocol cannot avoid vote-buying and coercion if voters reveal their anonymous identities to vote-buyer or coercer. As for the protocol of Cetinkaya, it guarantees some security requirements (as illustrated in Table 1). However, the weakness point of this protocol is that the voters are still coerced if the servers holding ballots connive with coercer. In the worst case, voters also able to sell their ballots by providing buyers with their anonymous identities, and then if buyers collude with Ballot Generator, Counter, and Key Generator, they can find out whether these anonymous identities are attached to

the candidate they expect or not. In other words, corrupted *BC* is a security flaw that Cetinkaya has not fixed yet. Our protocol makes good Cetinkaya's protocol shortcomings by encrypting uid using ElGamal cryptosystem before sending it to *BC*. Therefore, when a voter recasts, *BC* itself cannot recognize his uid. Only *Casting* server has responsibility to authenticate the eligibility of uid. However, the recasting process does not take place in *CS*, coercers cannot collect any information from this server.

If *TS* becomes corrupted, our protocol cannot be broken even though *TS* colludes with others voting authorities in protocol. In previous protocol using dynamic ballot, corrupted *TS* just needs to bribe *BC* and *CS* for getting intermediate result. However, in our protocol, B and V' are encrypted with the session key generated by *KG*, so *BC* and *CS* cannot provide the value of B and V' for *TS* without the decrypt key. Even if *KG* is also corrupted, the intermediate result of our protocol is still safe because the uid of voters are encrypted using ElGamal cryptosystem. Attackers have no way to combine B and V' or to remove invalid and duplicated votes. Therefore, corrupted Tallier is no longer a threat for our protocol. However, regarding sophisticated attacks which many voting authorities conspire together, Hasan [12] and Cetinkaya [5] are not strong enough to defeat these kinds of attacks.

According to the blind signature technique, no voting authorities know the link between voter's real ID and his uid and, no one can find out the link between a vote and a voter who casted it. It means that the privacy requirement is guaranteed.

This protocol has multiple layers of protection. For instance, *RS* checks the validity of requesters by CRAM; then, *PVer* check the eligibility of voters by their certificates. Another interesting point of our protocol is that there is a voter's signature d_V in the uid of a voter so the *RS* cannot create a fake uid to cheat other voting authorities without detecting. In brief, our protocol achieves eligibility.

Recasting is allowed in our protocol. If an adversary coerces voters to cast for his intention, the voters can send another vote to replace the previous one. According to the analysis above, this process cannot be discovered by coercers though they connive with many voting authorities. Therefore, the uncoercibility requirement is guaranteed.

Receipt-freeness is also fulfilled when the voters cannot prove their latest casting to vote-buyer. In case that an adversary penetrates into List1 and gets voters' uid through bribing, if the uid is not encrypted, the adversary can easily find out a certain uid does recasting process or not. Consequently, he can threat the voter or discover what the latest casting of voter is. Nevertheless, this assumption has never occurred in our protocol, according to the analysis at the beginning of this section.

The requirement of individual verifiability is guaranteed by applying bulletin boards. *BC* publishes Hash(E_{PET}(uid), $E_K^{(1)}$ (B, uid_1), time) in BB1 and Hash(uid, $E_K^{(2)}$ (V', uid_2)) is published in BB2. Thus, voters just have to hash the necessary information which they have already known, and compare their results to all records in bulletin boards to check whether the system counted his vote correctly.

At the end of election, all voting authorities publish their lists. Any participant or passive observer can check the soundness of final result based on the information on these lists and the bulletin boards as well. Hence, universal verifiability is fulfilled.

5 Conclusion

In this paper, we have proposed an unsusceptible electronic voting protocol to most of sophisticated attacks. The proposed protocol protects the privacy of voters and the content of votes from both inside and outside authorities even though more and more adversaries collude together. Furthermore, the fact no physical assumptions and no complex cryptographic techniques need to be used makes our proposal more practical. In the future, we intend to formalize an electronic voting protocol using process calculi such as pi-calculus for describing concurrent processes and their interactions.

References

1. Spycher, O., Koenig, R., Haenni, R., Schlapfer, M.: A new approach towards coercion-resistant remote e-voting in linear time. In: Financial Cryptography, pp. 182-189 (2012).
2. Araujo, R., Rajeb, N.B., Robbana, R., Traore, J., Youssfi, S.: Towards practical and secure coercion-resistant electronic election. In: 9th International Conference on Cryptology and Network Security, pp. 278-297 (2010).
3. Kohel, R.D.: Public key cryptography. In: Cryptography, Book, pp. 67-74 (2010).
4. Meng, B.: A critical review of receipt-freeness and coercion-resistance. Journal Information Technology, 8 (7), 934-964 (2009).
5. Cetinkaya, O., Doganaksoy, A.: A practical verifiable e-voting protocol for large scale elections over a network. In: 2nd International Conference on Availability, Reliability and Security, pp. 432-442 (2007).
6. Juels, A., Catalano, D., Jakobsson, M.: Coercion-resistant electronic elections. In: ACM workshop on Privacy in the electronic society, pp. 61-70 (2005).
7. Camenisch, J., Lysyanskaya, A.: A formal treatment of onion routing. In: 25th Annual International Cryptology Conference, pp. 169-187 (2005).
8. Liaw, H.T.: A secure electronic voting protocol for general elections. In: Journal Computers and Security, 23, 107-119 (2004).
9. Baudron, O., Fouque, P.A., Pointcheval, D., Poupard, G., Stern, J.: Practical Multi-Candidate Election System. In: 20th ACM Symposium on Principles of Distributed Computing, pp. 274-283 (2001).
10. Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. In: Advances in Cryptology Auscrypt'92, pp. 244-251(1993).
11. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: Journal Communications of the ACM, 24(2), 84-88 (1981).
12. Hasan, M.S.: E-Voting Scheme over Internet. In: Conference on Business and Information (2008).
13. Brumester, M., Magkos, E.: Towards secure and practical e-Elections in the new era. In Secure Electronic Voting, 7, 63-76 (2003).
14. Acquisti, A.: Receipt-free homomorphic elections and write-in voter verified ballots. ISRI Technical Report CMU-ISRI-04-116, Carnegie Mellon University, PA, USA, http://eprint.iacr.org/2004/105.pdf (2004).
15. Juang, W.S., Lei, C.L., Liaw, H.T.: A verifiable multi-authority secret election allowing abstention from voting. In: The Computer Journal, 45, 672-682 (2002).