On-line Voting System with Illegal Ballot Filtering Using Homomorphic Encryption

Mun-Kyu Lee, and Jong-Hyuk Im
Department of Computer and Information Engineering, Inha University
Incheon 22212, Korea
mklee@inha.ac.kr, imjhyuk@gmail.com

Abstract. A simple on-line voting protocol using homomorphic encryption is proposed. In addition to the basic properties required of a voting system, e. g., a voter's privacy, the system has additional functionalities such as automatic filtering of illegal ballot. Moreover, it is also possible that a voter who inadvertently cast a spoilt vote may vote again without revealing its secret ballot.

1 Introduction

Electronic voting, or on-line voting is a voting method where a computer or automated voting equipment is used [1] instead of a paper ballot. It aims at providing faster, more cost-effective, and more accurate voting. There have been a number of voting systems and protocols in the literature [2-9], each of which defines and satisfies various security goals. In this paper, we propose a new protocol for on-line voting with the property that an illegal or faulty vote can be filtered in real time without revealing its content and the voter of this vote can have the second opportunity to cast a correct vote. The proposed protocol uses homomorphic encryption schemes where degree-2 equations can be evaluated on the ciphertext domain.

2 Preliminaries

2.1 On-line Voting

A voting protocol is composed of voters, candidates, and the authority [1]. Voters are the participants that cast votes, candidates are the choices that voters can select, and the authority is an entity responsible for conducting the voting. There could be some adversary who attempts to manipulate the voting. An adversary may try to breach the privacy of voters or modify the result of voting. An adversary could be an internal one, or it could be an external one who cooperates with a voter.

There can be many different types of voting. For example, in a 1-out-of-2 (yes/no) voting, a voter chooses his/her answer from yes and no. In a 1-out-of-t voting, a voter chooses one of t candidates, while in a k-out-of-t voting, k candidates are chosen. In 1-out-of-t and k-out-of-t voting types, the candidate(s) chosen by the voter obtains 1 point, while other candidates obtain 0 point. The accumulated value over voters will be the final score for a specific candidate. Our proposal covers all the above types, but do not cover an ordered voting (multiple choices with preference) and a write-in voting where no candidates are given.

There has been an extensive research on on-line voting [2-9], and various security requirements for voting systems have also been proposed in previous works. The details for these requirements may be found in e.g., [1]. However, here we summarize the common requirements as follows:

- Completeness: the voting system should record and collect the votes correctly.
- Soundness: no voter can vote more than what is allowed. That is, s/he can only give 1 or 0 to candidates. S/he cannot give a value > 1 to his/her favorite candidate, nor a value < 0 to competitors.
- Privacy: no one, including even the authority, should be able to know any voter's choice.

The other important requirements contain authentication, integrity, non-reusability, auditability (accountability), etc. But we do not explicitly consider these properties in this paper because they can be achieved using legacy cryptographic techniques.

2.2 Homomorphic Encryption

Let $Enc(m)$ and $Dec(c)$ denote encryption and decryption operations over a plaintext message m and a ciphertext c, respectively. That is, $Dec\big(Enc(m)\big) = m$. A homomorphic encryption scheme is an encryption scheme where arithmetic operations can be properly done on its encrypted domain. For example, if an encryption scheme satisfies $Dec\big(Enc(m_1) + Enc(m_2)\big) = m_1 + m_2$ for any pair of plaintexts (m_1, m_2), then we say that this scheme is additively homomorphic. Similarly, if $Dec\big(Enc(m_1) \times Enc(m_2)\big) = m_1 \times m_2$, this scheme is multiplicatively homomorphic. If an encryption scheme is both additively and multiplicatively homomorphic for an arbitrary number of operations, it is called a fully homomorphic encryption scheme [10]. On the other hand, if it allows only a limited number of operations, we say it is somewhat homomorphic.

The idea of homomorphic encryption was first proposed by Rivest et al. [11] in 1978. Paillier proposed an additive homomorphic encryption scheme which provides homomorphic addition operations on an encrypted domain [12], and Boneh et al. proposed a homomorphic encryption which provides an arbitrary number of additions and depth-1 multiplications on ciphertexts, enabling evaluation of degree-2 equations [13]. In 2009, Gentry first proposed a fully homomorphic encryption scheme which provides an arbitrary number of additions and multiplications on an encrypted domain [10]. Since the fully homomorphic encryption scheme of Gentry, many works on fully homomorphic encryption schemes have appeared. While Gentry's scheme is based on the sparse subset sum problem over lattices, homomorphic encryption schemes based on the approximate GCD (Greatest Common Divisor) problem over the integers [14-16] and the (ring) learning with errors problem [17, 18] have been proposed.

Recently, Catalano and Fiore proposed a technique to transform an arbitrary additive homomorphic encryption scheme, e.g., the Paillier system, into a homomorphic encryption scheme which can evaluate degree-2 equations on ciphertexts [19].

3 Proposed Voting Protocol

The proposed voting protocol is composed of n voters ($n \geq 1$), t candidates ($t \geq 2$), and the authority as the traditional voting systems. In addition, it requires m intermediate collector(s) ($m \geq 1$). The protocol will be described for the 1-out-of-t voting, but it can be easily modified to the yes/no case, by regarding 'yes' and 'no' as the first and second candidates, respectively. It is also applicable to the k-out-of-t case, although we do not explain the detail in this paper. The voting protocol is composed of three stages; initialization, collection, and decision stages.

3.1 Initialization Stage

1. Let R be the authority. R generates a key pair composed of a private key S_R and a public key P_R for an underlying homomorphic encryption scheme, and sends them to voters as well as the information on t candidates. The homomorphic encryption we use is a somewhat homomorphic encryption with multiplication depth 1. That is, it should be able to deal with equations up to multiplicative degree 2 on its encrypted domain, e.g., $Dec(Enc(a) \times Enc(b) + Enc(c) \times (Enc(d) + Enc(e))) = ab + c(d + e)$. However, it is not guaranteed that $Dec\big(Enc(a) \times Enc(b) \times Enc(c)\big) = abc$, because the degree of this equation is 3. For this purpose, we may use the BGN scheme [13]. We may also use the somewhat homomorphic version of recently developed fully homomorphic encryption schemes such as integer-based schemes [14-16], LWE-based schemes [17, 18], and lattice-based schemes [10]. However, the most efficient one is to combine an additive homomorphic scheme such as the Paillier system with the recent general transformation technique which transforms an additive homomorphic scheme to a somewhat homomorphic scheme with multiplication depth 1 [19].
2. R decides the mapping between voters and collectors, i.e., which voter to send the ballots to which collector, and informs the corresponding voter and collector of this information.

3.2 Collection Stage

1. Each voter i $(1 \leq i \leq n)$ assigns 1 for his/her favorite candidate and 0 for all the other candidates. That is, if we denote voter i's vote on candidate j as v_{ij} and voter i's choice is candidate x, $v_{ix} = 1$ and $v_{ij} = 0$ for all $j \neq x$. If the voter does not want to choose anyone of the candidates, s/he may set $v_{ij} = 0$ for all j $(1 \leq j \leq t)$. The voter then encrypts each v_{ij} using the authority's public key P_R. As a result, s/he obtains

$$V_{ij} = Enc(P_R, v_{ij})\ (1 \le j \le t), \tag{1}$$

and sends them to the corresponding intermediate collector.

2. The intermediate collector $k \in \{1, \ldots, m\}$ assigned for voter i computes the verification vector for voter i as

$$E_i = ((V_{i1} + \cdots + V_{it}) \times (V_{i1} + \cdots + V_{it} - 1), \\ V_{i1} \times (V_{i1} - 1), V_{i2} \times (V_{i2} - 1), \ldots, V_{it} \times (V_{it} - 1)) \tag{2}$$

and sends it to the authority.

3. The authority collects all E_i for $1 \le i \le n$ and decrypts each element in $E_i = (E_{i0}, E_{i1}, E_{i2}, \ldots, E_{it})$ using its private key, producing

$$c_{ij} = Dec(S_R, E_{ij}) \tag{3}$$

for $0 \le j \le t$. If there is any nonzero c_{ij}, this means that voter i cast an illegal vote. This could be a failed trial for breach, or it could be a just inadvertent one, which are indistinguishable. Anyway, the authority notifies the intermediate collector corresponding to voter i of this fact.

4. The corresponding collector may inform voter i to vote again to prevent a spoilt vote, if it is the policy of the authority.
5. Each intermediate collector $k\ (1 \le k \le m)$ aggregates the encrypted votes (except the spoilt ones), and computes

$$U_1^{(k)} = \textstyle\sum_i V_{i1},\ U_2^{(k)} = \sum_i V_{i2},\ \ldots,\ U_t^{(k)} = \sum_i V_{it}. \tag{4}$$

The collector then sends theses values to the authority.

3.3 Decision Stage

1. Now the authority has m sets of encrypted aggregation,

$$(U_1^{(1)}, U_2^{(1)}, \ldots, U_t^{(1)}),, \ldots, (U_1^{(m)}, U_2^{(m)}, \ldots, U_t^{(m)}).$$

2. It computes

$$W_i = \textstyle\sum_{k=1}^{m} U_i^{(k)} \tag{5}$$

for $1 \le i \le t$.

3. It then decrypts each W_i using its private key S_R and obtains

$$w_i = Dec(S_R, W_i). \tag{6}$$

4. Finally, it finds the index z such that w_z is the maximum among w_i. Candidate z is elected.

3.4 Example

Let us assume that there are $t = 3$ candidates, $n = 100$ voters, and $m = 4$ intermediate collectors. Let's say, voter 5 is assigned to collector 2, and he wants to vote for candidate 3. Then, he may set $(v_{51}, v_{52}, v_{53}) = (0,0,1)$ and send its component-wise encryption to collector 2. (This choice will pass the filtering procedure in step 3 of the collection stage, because $c_{50} = c_{51} = c_{52} = c_{53} = 0$. For more details, see section 4.2.) Let's assume that collector 2 has 20 voters assigned to it, including voter 5. Then it will receive 20 tuples of (V_{i1}, V_{i2}, V_{i3}). If no illegal vote was found in steps 2 to 4 of the collection stage, it will compute $U_1^{(2)}$, $U_2^{(2)}$, and

$U_3^{(2)}$ by accumulating those 20 tuples. In step 2 of the decision stage, the authority will compute $W_1 = \sum_{k=1}^{4} U_1^{(k)}$ using the values from four collectors. It also computes W_2 and W_3 in a similar manner. Finally, it decides the winner by finding the W_i decrypted to the maximum.

4 Security Analysis

In this section, we show that the proposed voting protocol satisfies the security requirements mentioned in section 2.1.

4.1 Completeness

According to the additive homomorphic property of the underlying encryption scheme, combining (1), (4), (5) and (6), we obtain

$$w_j = Dec(S_R, W_j) = Dec\left(S_R, \sum_{k=1}^{m} U_j^{(k)}\right) = Dec\left(S_R, \sum_{all\ i} V_{ij}\right)$$

$$= \textstyle\sum_{all\ i} Dec(S_R, V_{ij}) = \sum_{all\ i} v_{ij}$$

for $1 \le j \le t$. Because $v_{ij} = 0$ (for preference) or 1 (for non-preference), w_j represents the exact number of voters who voted for candidate j. It is easy to see that $\sum_{j=1}^{t} w_j \le n$, where the inequality is for the case where (1) either some of the voters did not choose any candidate, i.e., gave 0 to all candidates, or (2) some of the votes were not counted because it was filtered in step 3 of the collection stage. The latter only happens if the authority's policy does not give another chance for voting to a voter who cast a spoilt vote.

4.2 Soundness

It is sufficient to show that a voter cannot give a value > 1 to his/her favorite candidate, nor a value < 0 to competitors. We achieve this goal by establishing the following lemma.

Lemma 1. If all $c_{ij} = 0$ for $0 \le j \le t$, one of the following two cases holds:

1) all v_{ij} *are 0 for* $1 \le j \le t$, *or*

2) $v_{iJ} = 1$ *for some* $J \in \{1, \dots, t\}$, *and* $v_{ij} = 0$ *for all* $j \ne J$.

Proof. Because we assumed that the underlying encryption scheme provides a depth-1 multiplication, (2) and (3) implies $c_{i0} = Dec(S_R, E_{i0}) = (v_{i1} + \cdots + v_{it}) \times (v_{i1} + \cdots + v_{it} - 1)$. Thus, $c_{i0} = 0$ implies that $v_{i1} + \cdots + v_{it}$ is either 0 or 1. Similarly, for $1 \le j \le t$, $c_{ij} = 0$ implies that v_{ij} is either 0 or 1. Because v_{ij} is either 0 or 1 for all $1 \le j \le t$, we can enumerate all 2^t combinations of $(v_{i1}, \dots, v_{it}) \in \{0,1\}^t$.

Among these, any combination with more than two nonzero v_{ij}, i.e., more than two 1's, produces $v_{i1} + \cdots + v_{it} > 1$, which contradicts the condition that $c_{i0} = 0$. Then, the only remaining possibilities are the two cases mentioned in the lemma. Finally, it is easy to see that these two cases make $c_{ij} = 0$ for $0 \leq j \leq t$. To be precise, the first case makes $v_{i1} + \cdots + v_{it} = 0$ and the second makes $v_{i1} + \cdots + v_{it} = 1$. This proves the lemma.

The first case in lemma 1 is for the case that voter i did not choose any candidate, which is reasonable if s/he cannot make a decision. The second case stands for the normal situation where the voter selected only one candidate. If it is not guaranteed that a voter can only give 1 or 0 to candidates, the election may be corrupted. For example, s/he might try to give a huge score to let her favorite candidate elected irrespective of other voters' choices, or s/he might give a negative score to let her enemy fail to be elected. In addition, it can also be detected if a voter tries to choose multiple candidates. However, by slightly modifying the equation for E_{i0}, we can also support the k-out-of-t voting where a voter can select up to k candidates. Finally, we remark that the detection of illegal votes based on the values of c_{ij} only reveals that there is something wrong, but it does not discriminate which is the case among the above three possibilities; a huge score, a negative score, or multiple choices.

4.3 Privacy

The privacy of voters is guaranteed by aggregation. In step 5 of the collection stage, each intermediate collector sends the authority only the aggregated value of the votes that it collected. Even when the authority tries to separately decrypt $\left(U_1^{(j)}, U_2^{(j)}, \ldots, U_t^{(j)}\right)$ for $1 \leq j \leq m$ in step 3 of the decision stage instead of their aggregated values $W_1, W_2, \ldots, W_t$, the authority cannot know the choice of a specific voter. What the authority learns is the sum of votes for each candidate from all voters connected to a specific collector. For example, by decrypting $U_1^{(k)}$, the authority will recover $\sum_{(for\ all\ i\ connected\ to\ collector\ k)} v_{i1}$. That is, the collector finds out how many people in the community managed by collector k voted for candidate 1, but not who did. This situation is exactly the same as the way the secret ballot principle is observed in a local polling station for an off-line voting. We remark that, however, the privacy may not be protected if a collector and the authority colludes.

4.4 Support for Second Voting

If the authority adopts the policy that it allows for failed voters to vote again, it may raise the ratio of valid votes without harming the completeness and soundness. Moreover, the failed voter's privacy is guaranteed even in the spoilt vote and the second vote.

4 Security Analysis

We presented a new protocol for on-line voting with the property that an illegal or faulty vote can be filtered in real time without revealing its content. Then, the voter of this vote may have the second opportunity to cast a correct vote. Our protocol uses somewhat homomorphic encryption schemes where degree-2 equations may be evaluated on the ciphertext domain. The proposed protocol guarantees the voters' privacy, and at the same time, it prevents voters from giving illegal scores to candidates.

Acknowledgments. This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2016-H8501-16-1008) supervised by the IITP (Institute for Information & communications Technology Promotion).

References

1. Mursi, M., Aggassa, G., Abdelhafez, A., Sarma, K.: On the Development of Electronic Voting: A Survey. In: International Journal of Computer Applications, 61(16), 1-11 (2013)
2. Chaum, D.: Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms. In: Communications of the ACM, 24(2), 84-88 (1981)
3. Sako, K., Killian, J.: Receipt-Free Mix-Type Voting Scheme: A Practical Solution to The Implementation of A Voting Booth. In: EUROCRYPT '95, 393–403 (1995)
4. Benaloh, J.: Verifiable Secret-Ballot Elections. In: Ph.D. thesis, Yale University (1987)
5. Cramer, R., Gennaro, R., Schoenmakers, B.: A Secure and Optimally Efficient Multi-Authority Election Scheme. In: EUROCRYPT '97, 103-118 (1997)
6. Schoenmakers, B.: A Simple Publicly Verifiable Secret Sharing Scheme and Its Applications to Electronic Voting. In: CRYPTO '99, 148-164 (1999)
7. Fujioka, A., Okamoto, T., Ohta, K.: A Practical Secret Voting Scheme for Large Scale Elections. In: AUSCRYPT '92, 248–259 (1992)
8. Park, C., Itoh, K., Kurosawa, K: Efficient Anonymous Channel and all/nothing Election Scheme. In: EUROCRYT '93, 248–259 (1993)
9. Juels, A., Catalano, D., Jakobsson, M.: Coercion-Resistant Electronic Elections. In: WEPS '05, 61-70 (2005)
10. Gentry, C.: Fully Homomorphic Encryption Using Ideal Lattices. In: STOC ' 09, 169-178 (2010)
11. Rivest, R., Adleman, L., Dertouzos, M.: On Data Bank and Privacy Homomorphisms. In: Proceedings of the 19th Annual Symposium on Foundations of Secure Computation-FSC 1978, Academic Press, 169-180 (1978)
12. Paillier, P.: Public-Key Cryptosystems based on Composite Degree Residuosity Classes. In: EUROCRYPT '99, 223–238 (1999)
13. Boneh, D., Goh, E., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Theory of cryptography, 3378, 325-341, (2005)
14. Dijk, M., Gentry, C., Halevi, S., Vaikuntanathan, V.: Fully Homomorphic Encryption over the Integers. In: EUROCRYPT '10, 24-42 (2010)
15. Coron, J., Naccache, D., Tibouchi, M.: Public Key Compression and Modulus Switching for Fully Homomorphic Encryption over the Integers. In: EUROCRYPT '12, 446-464 (2012)
16. Cheon, J., Coron, J., Kim, J., Lee, M., Lepoint, T., Tibouchi, M., Yun, A.: Batch Fully Homomorphic Encryption over the Integers. In: EUROCRYPT '13, 315-335 (2013)

17.Brakerski, Z., Gentry, C., Vaikuntanathan, V.: (Leveled) Fully Homomorphic Encryption without Bootstrapping. In: ITCS ’12, 309-325 (2012)
18.Clear, M., McGoldrick, C.: Multi-Identity and Multi-Key Leveled FHE from Learning with Errors. In: CRYPTO ’15, 630-656 (2015)
19.Catalano, D., Fiore, D.: Using Linearly-Homomorphic Encryption to Evaluate Degree-2 Functions on Encrypted Data. In: ACM-CCS ’15, 1518-1529 (2015)