A MULTIPLE BALLOTS ELECTION SCHEME USING ANONYMOUS DISTRIBUTION

Manabu Okamoto[1]

[1] Kanagawa Institute of Technology
1030 Shimo-Ogino, Atsugi, Kanagawa 243-0292, Japan
manabu@nw.kanagawa-it.ac.jp

ABSTRACT

Electronic voting is an important application for security protocols. Most existing voting schemes are designed for elections in which each voter has only one ballot. However, some elections permit voters to cast multiple ballots. In this paper, we present a new voting scheme in which each voter can have multiple ballots, and can vote for multiple candidates. The proposed scheme allows the voter to simply pick their candidates and post a single encrypted message. Anonymous distribution of secret information is used so that no one knows which information is being passed to whom.

KEYWORDS

Elecronic voting, Anonymity

1. INTRODUCTION

Electronic voting is an important application for security protocols. Everybody hopes that electronic voting is as secure and efficient as traditional voting systems. It needs to ensure privacy, universal verifiability, fairness, and eligibility.

Many electronic voting schemes have previously been proposed, though most of these voting schemes are intended for elections in which each voter has only one ballot. However, there are instances, in which each voter is permitted to cast multiple ballots [1], [2]. For example, a multiple ballot election is often used at a general meeting of stockholders.

When applying the existing methods to a multiple ballot election, each voter would need to perform multiple actions. For example when I want to cast 3 ballots for Alice, I need to do 3 actions like sending 3 e-mails to the Vote-system server. These scheme are non-efficiency.

In this paper, we present a new voting scheme in which each voter can have multiple ballots, and can vote for multiple candidates. The proposed scheme allows the voter to simply pick their candidates and post a single encrypted message. It is a very simple scheme and can reduce the cost of a voting system as well as the amount of effort required by the voter. The system can be divided into layers, so that voting in organizations is easily facilitated. An anonymous distribution scheme is used, employing secret information so that no one knows what’s actually being transmitted, or where it’s being sent.

2. RELATED WORK

Many electronic voting schemes have previously been proposed. These are classified below.

A. Shuffle

DOI : 10.5121/ijnsa.2011.3502

This scheme uses an anonymous channel, such as Mix-net [3], [4]. Mix-net is a server for the shuffles. It shuffles ballots and provides anonymity and privacy. Anonymity is retained unless all of the servers conspire. Mix-net is a fundamental technique used for transmitting ballots, and is often incorporated into other technologies [5]-[9].

B. Blind Signature

Anonymity is facilitated using Mix-net; however, an illegitimate vote cannot be checked. A blind-signature technique could be used to overcome this [10]-[12], in which an eligible voter is only allowed to vote once. All voters request a blind-signature from a central authority, which then permits them to vote. Voters can then get their ballot signed by this authority without them being permitted to see the contents of the vote. This scheme requires an anonymous channel to work properly, such as Mix-net.

C. Homomorphism based schemes

Homomorphism based schemes [13]-[19] use homomorphic encryption functions and require ZKIP to prove that each vote is fairly cast. These schemes have an efficiency bottleneck to check vote validity.

3. OUR GOAL

In this paper, we propose a new multiple ballot election schemes that uses an anonymous distribution. Most of the voting schemes previously mentioned apply to elections in which each voter can cast only one ballot. However, there are instances in which each voter can have multiple ballots to cast. When applying the methods described above to a multiple ballot election, each voter would need to perform multiple actions; one for each vote. This is inefficient.

Our proposed scheme is very simple. Each voter requires only one action to cast multiple ballots, and the principles of privacy, universal verifiability, etc. are honored.

Fairness is an important issue for a multiple-ballot election, meaning that nothing must affect the voting. That is, no participant is allowed to have any knowledge of the tally before the tally is complete. In a multiple ballot election, if a voter were to learn how many ballots had already been cast, then he might say to someone: ‘Would you please cast all of your votes for Alice.’ In a single ballot election, that person would only have two choices: voting or not voting. However, in a multiple ballot election, the individual could decide HOW MANY ballots will be cast. It is therefore very important that knowledge of the voting progress remains concealed. By knowing the voting percentage, he might also decide how many ballots will be cast for each candidate. For example if he knows that the voting percentage is high, the he might decide: ‘I will cast all my 10 ballots to Alice’. If the voting percentage is low, then he might decide: ‘I will divide my ballots into 2 candidates: 5 ballots for Alice; and 5 ballots for Bob.’ This is a difficult issue for multiple ballot elections.

Our proposed scheme achieves this fairness principle. Other schemes require that ballots be divided into pieces, or that the counting is divided into groups, which incurs a cost penalty. However, in our scheme, there is no need to divide the ballots into pieces. It also enables the ballots to be rapidly counted by anyone.

4. ANONYMOUS DISTRIBUTION

Our technique uses an anonymous distribution, which uses secret information so that no one knows about the information being transmitted, or where it’s being sent.

An anonymous distribution has previously been applied to electronic voting schemes [20], and is easily realized. Different secret values are written one by one onto a series of CD-ROMs. We do not attach labels to the CDs, and we mix them up randomly. We therefore cannot know which pieces of information are written onto any given CD-ROM.

We can perform an anonymous distribution using email [21]. Such a scheme achieves an anonymous distribution method via one or more trusted third parties. A receiving person does not need to participate in the distribution itself, and can receive data very easily. Since this high risk activity is separable, the trusted third party can be divided into any arbitrary number of facilities. Reliability is therefore achieved, unless all of the third parties conspire together.

5. How To Vote

In our proposed scheme, the voting system consists of various entities as described below.

- Voters

We use V_i to denote voters who are permitted to vote. $\{V_1,V_2,\cdots,V_N\}$ denotes the set of all voters. Each voter has m_i ballots that can be cast across M candidates.

- Candidates

In this scheme, we can have multiple candidates as well. We use $\{ C_1 {}_, C_2, \cdots, C_M \}$ to denote these candidates.

- Election Administration Committee

In this scheme, the Election Administration Committee (EAC) bears an important role. The EAC creates all the voting cards and keeps them secret until the end of the election. For tallying votes, the EAC make the voting cards public on the bulletin board system (BBS), and then counts the ballots. Essentially, the EAC is a trusted third party, but we have assumed that the EAC may in fact be corrupt.

- Anonymous Distribution System

The Anonymous Distribution System (ADS) distributes the voting cards that the EAC have made. For security, the ADS must be two or more separate organizations. The ADS distributes the voting cards such that no one can identify the recipients. In this paper, we will not discuss this distribution system in any detail.

- Mix-net

We use Mix-net to provide an anonymous channel. Voters send their voting cards through Mix-net, and this provides them with the required anonymity. Mix-net must consist of two or more servers, and we assume that all voters are given public keys to connect to it. In this paper, we will not discuss the details of Mix-net.

- Tally up Facility

The Tally up Facility (TF) collects ballots from Mix-net and keeps them a secret until the end of the election. When the vote count commences, the TF makes the data available on the BBS. Essentially, the TF is a trusted third party, but we once again assume that the TF could be corrupt. However, we assume that the EAC and the TF are unable to conspire together.

The TF needs a homomorphic public encryption function E, and the EAC obtains a public key and uses it to encrypt voting cards. The public key is effectively secret information for the voters, and for any other authorities.

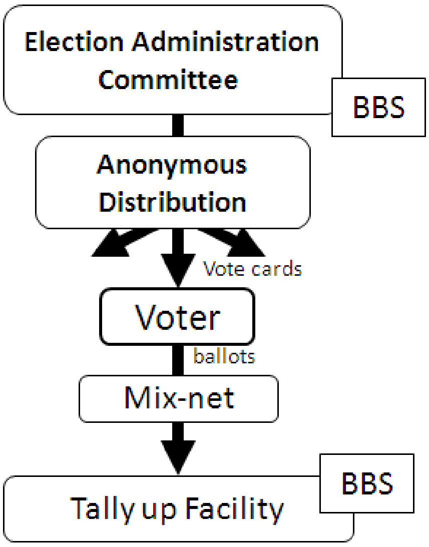
The homomorphic property that we use here is a product of two values, such as:

$$E(a)\times E(b) = E(a\times b). \qquad (1)$$

RSA and ElGamal cryptosystems have this property, and we therefore assume the use of either of these.

- Bulletin Board System (BBS)

The BBS is used for public communication by any party or individual, but each legitimate party can only write messages at designated times. Nobody can erase messages from the BBS. Figure 1 shows the details of this system.



Fugire 1. The voting system.

We will describe the actual voting method below, to ensure clarity. We will also describe a simple example in the subsequent chapter.

(1) EAC creates the voting cards.

First of all, the EAC creates the voting cards. Voting cards consist of prime numbers that correspond to each candidate. With candidates $\{C_1,C_2,\cdots,C_M\}$, voting cards $\{VC_i\}$ are created (with i=1$\cdots$$N$, where N is the number of voters) = $\{B_{i1},B_{i2},\cdots,B_{iM}\}$ = $\{E(P_{i1}),E(P_{i2}),\cdots,E(P_{iM})\}$. For any x and y, P_{xy} is a unique prime number and must be different from $P_{x'y'}$ for any other x' and y'.

B_{xy} is an encrypted value of P_{xy} using the TF's public key. In addition, we recommend that an electronic signature by the EAC is performed on these cards. The EAC must calculate the hash value of all voting cards that are public on the BBS, and they cannot be untruthful.

Figure 2 shows the voting card VC_i. A voter that receives this card will understand that B_{ij} is a ballot for C_j but does not know the value of P_{ij}.

B_{i1}	B_{i2}	$\vdots$	B_{iM}
C_1	C_2	$\vdots$	C_M

Figure 2. Voting card VC_i.

(2) Anonymous distribution

The EAC distributes the voting cards to the voters using an anonymous distribution system. No one can know to whom the voting cards are being sent. If any given voter has the right to vote *m* ballots, then he receives *m* voting cards.

(3) Voters' action

Voters receive their voting cards, and some voters may receive multiple voting cards.

Now, we assume here that voter V_i receives only one voting card $VC_i=\{B_{i1},B_{i2},\cdots,B_{iM}\}$. That is, he has a right to vote only one vote. He obtains a prime number B_{ij} that corresponds to the candidate C_j that he elects to vote for.

He sends $B_{ij}=E(P_{ij})$ to Mix-net. Each voter can connect to Mix-net only once, and all voters need to be authenticated. In this paper, we will not describe this authentication, nor how the mixing occurs.

If a voter V_k receives multiple voting cards, then he can obtain multiple prime numbers $\{B_{kl},B_{k'l'},\cdots\}$ with his voting card. If he wants to cast multiple ballots for the same candidate, then the each prime value is used on EACH voting card.

He creates products of the prime values obtained and sends the following to Mix-net:

$$\begin{aligned} & B_{kl}\times B_{k'l'}\times\cdots \\ & = E(P_{kl})\times E(P_{k'l'})\times\cdots \\ & \qquad = E(P_{kl}\times P_{k'l'}\times\cdots) \end{aligned} \tag{2}$$

(4) TF's action

The TF receives the voting cards from Mix-net and keeps their contents secret until the end of the election.

The TF can consist of two or more facilities, or the encryption function *E* can be comprised of two or more functions. Any one TF cannot therefore decrypt these until the voting deadline has passed.

Assume however that the TF could indeed decrypt the products of ballots, and could also factor them into various prime numbers. It is important to realize that the TF still cannot understand what each prime number represents, and which candidate each prime number corresponds to. For this to occur, the TF would need to conspire with the EAC.

(5) Tally up ballots

Once the voting period has expired, the TF counts the votes. The TF decrypts all the values and factors them into prime numbers, then makes them publicly available on the BBS. The EAC makes all voting cards public on the BBS with their prime values P_{xy}, which are a plain text of

B_{xy}. The EAC cannot be untruthful because the hash values of cards have previously been made public.

A vote P_{ij} that corresponds to a given candidate can be counted with reference to the voting cards that the EAC have made available on the BBS.

Anyone can perform this calculation, since the data are openly available on the BBS. This process enables universal verifiability

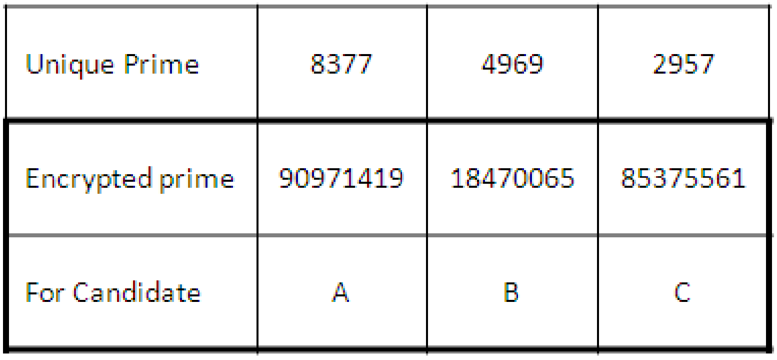
6. EXAMPLE

We describe this process with a simple example, in which the number of voters and the number of rights for each voter are manageably small.

We have 3 candidates (Alice, Bob, Carol) and 4 voters in this example. Each voter can cast 2 ballots.

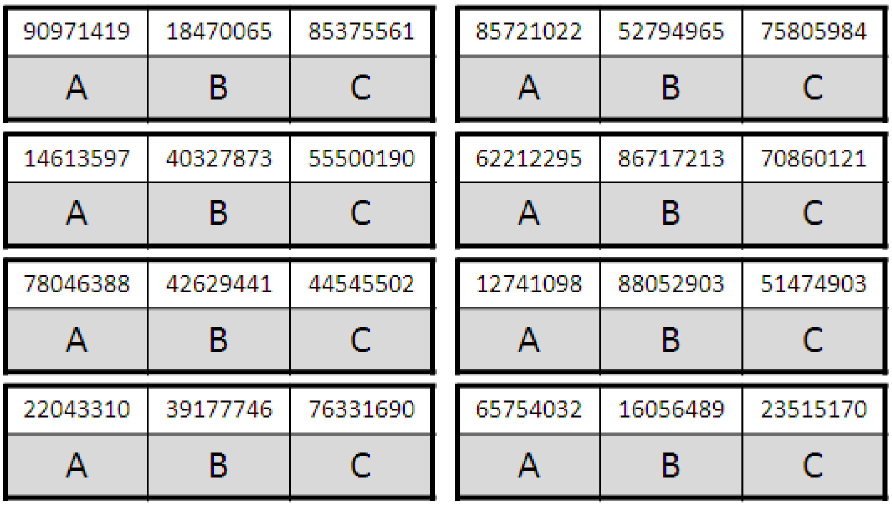
To begin with, the EAC creates the voting cards. Figure 3 shows one possible set of voting cards. These consist of unique prime numbers and an encrypted value that uses the TF's public key. These values are kept secret until the voting deadline has passed.

The EAC creates a voting card for each voter. A voting card consists of an encrypted prime number for each candidates. The EAC does not include the unique prime numbers in the card. Figure 4 shows all 8 voting cards in this example. Here, we assume that an RSA cryptosystem is used.



Unique Prime	8377	4969	2957
Encrypted prime	90971419	18470065	85375561
For Candidate	A	B	C

Fugure 3. Data for the voting cards.



90971419	18470065	85375561
A	B	C

85721022	52794965	75805984
A	B	C

14613597	40327873	55500190
A	B	C

62212295	86717213	70860121
A	B	C

78046388	42629441	44545502
A	B	C

12741098	88052903	51474903
A	B	C

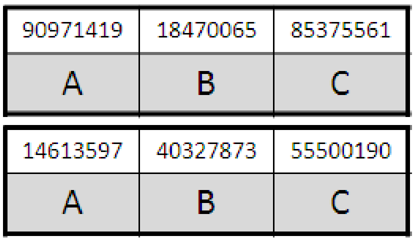
22043310	39177746	76331690
A	B	C

65754032	16056489	23515170
A	B	C

Figure 4. The voting cards.

The EAC distributes the voting cards via an anonymous distribution system, so that no one can know where each voting card is sent. In this example, each voter receives 2 cards. As shown in Figure 5 one of the voters elects Alice and Carol, and therefore picks up 90971419 for Alice and 55500190 for Carol. The calculation 90971419 × 55500190 is made and then sent to Mix-net.

The voter cannot pick up 90971419 and 85375561 because both values are in the same card.



90971419	18470065	85375561
A	B	C

14613597	40327873	55500190
A	B	C

Figure 5. One voter’s voting cards.

Therefore, the voter sends:

$$
\begin{aligned}
&90971419 \times 55500190\\
&\quad = E(8377) \times E(9067)\\
&\qquad = E(8377 \times 9067). \qquad (3)
\end{aligned}
$$

The TF collects all of the voting cards from Mix-net and keeps them all secret. Once the voting period expires, the TF decrypts these values and factors them into unique prime values.

If the TF received the following decrypted values: {

8377×9067,

8707×4549,

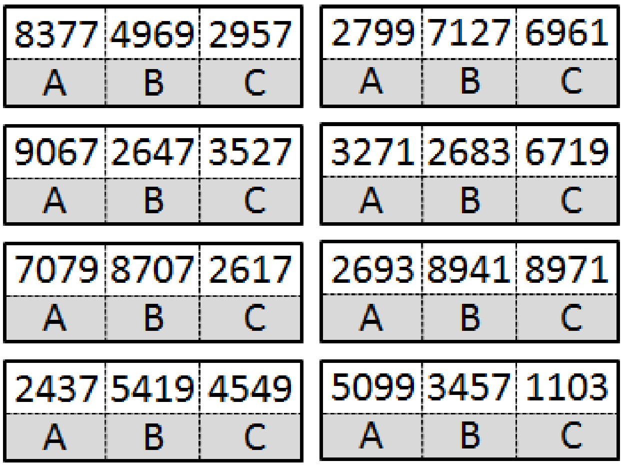
7127,

8971×3457 },

then the TF can factor them all and obtain the following prime values:

{8377, 9067, 8707, 4549, 7127, 8971, 3457} .

The TF makes them open on BBS.



8377	4969	2957	2799	7127	6961
A	B	C	A	B	C
9067	2647	3527	3271	2683	6719
A	B	C	A	B	C
7079	8707	2617	2693	8941	8971
A	B	C	A	B	C
2437	5419	4549	5099	3457	1103
A	B	C	A	B	C

Figure 6. Raw data for the voting cards.

The EAC then makes all data for the voting cards public on the BBS, without their encrypted values. Figure 6 shows the prime numbers that correspond to each candidate. The EAC must be truthful about this data because the hash values of cards have previously been made public.

We can easily compute from Figure 6 that:

Alice : 2 ballots (card: 8377 and 9067)
Bob : 3 ballots (card: 8707, 7127, and 3457)
Carol : 2 ballots (card: 4549 and 8971)

Therefore Bob has won, and the election is over.

We can check whether or not two ballots were cast on a single voting card, or if any other prime number has been used.

7. SECURITY

A set of requirements on the electronic voting system must be satisfied by any secure voting protocol [22]. These requirements can be grouped and summarized as follows:

- *Eligibility*: Only eligible and authorized voters can vote.
- *Privacy*: All votes must be kept secret. No one should be able to determine the value of the vote cast by any given voter.
- *Fairness*: Nothing must affect the voting. No participant is allowed to gain any knowledge about the tally until the election deadline has passed.
- *Robustness*: A coalition of voters or authorities cannot disrupt the results.
- *Universal Verifiability*: A voting system is verifiable if anyone is able to verify that all votes have been counted correctly. Any participant or observer can check that the final tally is indeed the correct sum of all votes.
- *Receipt-Freeness*: No voter should be able to prove his vote to any other participant.

We will check that our scheme satisfies these requirements.

Eligibility: All voters need to be authenticated when sending ballots to Mix-net, therefore double voting cannot occur as each voter can only connect to Mix-net once. Any ineligible individual cannot participate in the vote, because he cannot receive any voting cards and cannot connect to Mix-net. Only an eligible voter can receive a voting card from the EAC and send it to Mix-net.

If the TF is corrupt, and attempts to change the voting cards so that their preferred candidate is victorious, they will be thwarted. This is because the TF cannot change the prime numbers on a voting card as they cannot associate them with any specific candidate. Every prime number that corresponds to each of the candidates is kept secret by the EAC. We therefore assume that the EAC and the TF do not conspire together.

Privacy: Anonymity and privacy are achieved by using an anonymous distribution and Mix-net. The safety of these can be increased by increasing the number of servers.

Fairness: If the TF cannot decrypt the prime product values which the voter has cast, then the TF cannot decipher the tally. To achieve this, the encryption function consists of multiple functions whose key is divided into multiple pieces, and distributed to multiple authorities. If the TF cannot decrypt the votes, then only the number of voters can be known, not the number of ballots. If we also need to keep the number of voters secret, then we have to divide the ballots into multiple authorities.

If the TF can decrypt and factor the values of the ballots, then the TF would know how many ballots have been cast, but the TF still could not know which candidate is winning or losing. This is because the TF cannot know how the prime numbers map to each candidate without conspiring with the EAC.

Robustness: A dishonest voter cannot disrupt the voting. He cannot send more than one product of prime numbers because he cannot connect to Mix-Net more than once. If he disrupts his own product of prime numbers, then he could only send the products of random primes. However, he does not know which candidates, if any, correspond to those random primes. The TF can easily ignore these. If by chance a random prime does correspond to a candidate, it is equivalent to having correctly cast a vote for that candidate, but the voting itself cannot be disrupted.

Universal Verifiability: Universal verifiability is achieved through the use of the BBS. Anyone is able to check the calculation for the total number of ballots, because the data is publicly available on the BBS. Our scheme is very simple, and the calculations are trivial. One simply has to get the data from the BBS and perform the count for each candidate, which is something that can be done by anyone.

Receipt-Freeness: No voter can claim that a specific value on the BBS is the result of his own vote, because he only knows the encrypted prime value and not the plain prime value. To satisfy Receipt-Freeness, we recommend that the encryption function be probabilistic, because if the public key of encryption function E is known, then we could easily encrypt the prime value on the BBS, and check whether or not it is the same as the encrypted prime number on our voting card.

One other security issue is now described. If the EAC were indeed corrupt, then he could change the voting cards that have previously been created. For example, he could exchange the prime value B_a in voting cards that correspond to Mr. A with value B_b that corresponds to Mr. B. If a voter casts a ballot for Mr. A, and sends prime value B_a to Mix-net, then the corrupt EAC could exchange these two values before making it public on the BBS. We will then tally the vote for Mr. B, with the value B_a, which were originally intended for Mr. A.

To avoid this attack, we need to obtain the hash values of all the voting cards that correspond to candidates before the start of the election. If the EAC wants to change any of the values used in the voting cards, then this would easily be detected because anyone can calculate the hash value of vote cards on a public BBS, and any voter can calculate the hash value of their own vote cards. If the hash value is not same, then the EAC is suspect.

8. DISCUSSION

We will now discuss the efficiency of our proposed scheme. Products of prime values have been used here; however, we could easily do the same thing by placing voting cards in order and then encrypting and submitting that complete bundle, without needing any multiplications. Figure 7 shows such an encrypted bundle. We have to add dummy values into the submission, because otherwise it’s very easy to determine how many ballots have been cast by the length of the bundle. This scheme is easier than ours as there are no multiplications to perform.

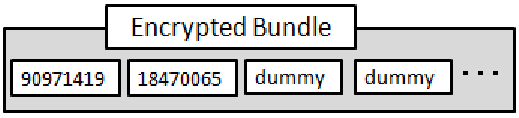


Figure 7. Encrypted bundle.

However, our scheme is especially effective when we divide the task into layers. In multiple e-voting elections, votes can be cast not merely by an individual, but also by an organization. For example, assume that voting is performed in 5 groups. Each group, labeled A, B, C, D and E, have 20 rights each for their votes. However, group A might have 20 people, whereas group B might have only 10 people. That is, the number of people in each group may differ. Each group can freely decide how to divide up their rights to vote. A representative from each group could decide how the ballots will be divided up, and then distribute and collect them. The representative then adjusts all of the ballots in his group and casts it to the voting system as a single voter. Figure 8 illustrates this process.

When this system is used, we need to add another homomorphic encryption function *F* for the TF, such that for any prime values *a* and *b* on a voting card:

$$F(E(a))\times F(E(b)) = F(E(a)\times E(b))=F(E(a\times b)) \qquad (4)$$

That is, the product is homomorphic over function *E*. However, since the public key for *F* is available to any voter, it must also be probabilistic.

Actually, when a voter at the bottom of the system casts their vote $E(a\times b)$ to the representative of his group, the representative can in fact know which candidate he voted for. Therefore, the voter must encrypt his ballot using function *F* before casting his vote to the representative. A representative can then collect all of the ballots, but cannot know which candidate he has voted for. He adjusts them into a single product together with the other products from the other voters in the same group, then casts only one value to Mix-net as a single voter. The TF decrypts the ballot using a private key of functions *E* and *F*, from which the plain prime value is obtained.

We can do the same thing using bundles, by putting cards in order, however at that time the representative cannot adjust the bundle, and anyone is therefore able to determine how that group is organized internally.

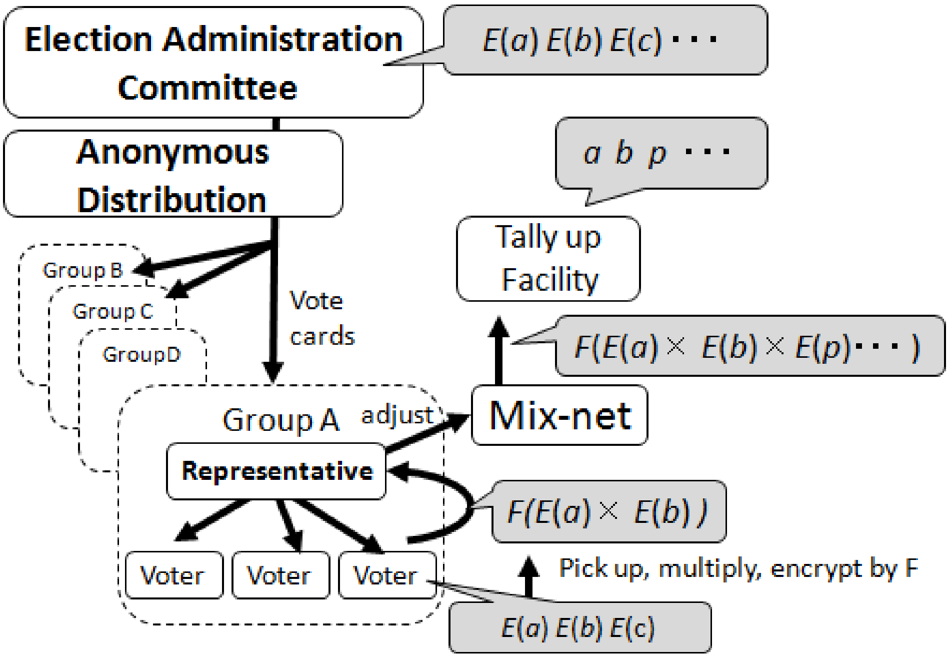


Figure 8. A layered voting system.

Finally, we describe an important point in regards to our proposed method. We denote N as the number of modulo bases in the crypto function. For example, in RSA, it is $N=p \times q$ for prime p and q. In this scheme, we have to ensure that the maximum value of the product of prime numbers is smaller than N. When there are a large number of voters, with many rights to vote, and many candidates to vote for, we have to use a lot of prime values, and therefore have to use large primes whose product will also be very large. Thus, we need to make N a very large number and need to ensure that sufficient computation time is allowed for. So our proposed scheme is good for small election which consists of many small sub-groups.

9. CONCLUSION

We have proposed a new multiple ballot election scheme using an anonymous distribution. In this scheme each voter would need NOT to perform multiple actions. Each voter requires only one action to cast multiple ballots. It would contribute to reducing the burden on voters. Voters can cast ballots by only one e-mail or Web-access. It is a very simple scheme that supports eligibility, privacy, fairness, robustness, universal verifiability, and receipt-free operation. We do not need the big calculation. All a voter has to do is to multiply values in voting cards he got.

Our proposed scheme is also very effective when organization is divided into layers. We can divide a vote act into some groups. It would contribute to reducing the burden on voting-system.

REFERENCES

[1] T.Saisho, T.Saito, H.Doi, and S.Tsuji, “Note on Voting Schemes Suitable for Multiple Ballots per Voter : To Construct Electronic Decision Systems at Stockholder's Meetings ,” PSJ SIG Notes 2002(43), pp.13-18, 2002.

[2] ISHIDA Natsuki , MATSUO Shin'ichiro, and OGATA Wakaha ,” Efficient Divisible Voting Scheme,” IEICE transactions on fundamentals of electronics, communications and computer sciences E88-A(1), pp.230-238, 2005.

[3] D. Chaum, “Untraceable electronic mail, return address, and digital pseudonyms,” Communications of the ACM, vol.24, no.2, pp.84-88, February 1981.

[4] M. Abe, “Universally verifiable mix-net with verification work independent of the number of mix-servers,” Advances in Cryptology - Eurocrypt 1998, Lecture Notes in Computer Science, vol.1403, pp.437–447, Espoo, Finland, May 1998.

[5] J.Fujikawa and k.Sako, “an efficient scheme for proving a shuffle,” Proc.Crypto 2001,LNCS 2045, pp.368-387,2001.

[6] M.Jakobsson and A.Juels, “Mix and match: Secure function evaluation via ciphertext,” Proc.Asiacrypt 2000,LNCS 1976, pp.162-177,2000.

[7] C.Park, K.Itoh, and K.Kurosawa, “Efficient anonymous channel and all/nothing election scheme,” Proc.Eurocrypto’93, LNCS 765,pp.248-259,1993.

[8] K.Sako,”Electronic voting scheme allowing open objection to the tally,” IEICE Trans.Fundamentals,vol.E77-A,no.1,pp.24-30,Jan.1994.

[9] K.Sako and J.Kilian,”Receipt-free mix-type voting scheme,” Proc.Eurocrypto’95,LNCS 921,pp.393-403,1995.

[10] D. Chaum, “Blind signatures for untraceable payments,” Advances in Cryptology - Crypto 1982, Plenum Press, pp.199-203, 1983.

[11] J.Furukawa and K.Sako,”An efficient scheme for proving a shuffle,” Proc.Crypto 2001,LNCS 2045,pp.368-387,2001.

[12] T.Okamoto,"An electronic voting scheme," Proc.IFIP'96, Advanced IT Tools,1996.

[13] M.Ohkubo,F.Miura,M.Abe,A.Fujioka, and T.Okamoto,"An improve-ment on a practrical secret voting scheme," Proc.ISW'99,LNCS 1729,pp.225-234,1999.

[14] J.Benaloh and D.Tuinstra,"Receipt-free secret-ballot elections," Proc.STOC'94,pp.544-553,1994.

[15] J.Benaloh and M.Yung,"Distributiing the power of a government to enhance the privacy of voters," Proc.PODC'86,pp.52-62,1986.

[16] J.Cohen and M.Fischer,"A robust and verifiable cryptographically secure election scheme," Proc.FOCS'85,pp.372-382,1985.

[17] R.Craimer,M.Franklin,B.Schoenmakers, and M.Yung,"Multi-authority secret-ballot elections with linear work," Proc.Eurocrypt'96,LNCS 1070,pp.72-82,1996.

[18] R.Craimer, R.Gennaro, and B.Schoenmakersm "A secure and optimally efficient multi-authority election scheme," Proc.Eurocrypt'97, LNCS 1233,pp.103-118,1997.

[19] K.R.Iversen,"A cryptgraphic scheme for computed general elections," Proc.Crypto'91,LNCS 576,pp.405-419,1991..

[20] K.Sako and J.Kilian, "Secure voting using partially comatible homomorphism," Proc.Crypto'94, LNCS 839,pp.411-424,1994.

[21] M.Okamoto and Y.Tanaka, "Electronic Voting Schemes with Anonymous Distribution," The Transactions of the Institute of Electronics, Information and Communication Engineers. A J87-A(7), pp.958-966, 2004.

[22] M.Okamoto and Y.Tanaka, "Secret Information Distribution," The Transactions of the Institute of Electronics, Information and Communication Engineers. A J89-A(8), pp.662-670, 2006.

[23] Hayam K. Al-Anie, Mohammad A. Alia and Adnan A. Hnaif, "E-Voting Protocol Based On Public-Key Cryptography," International Journal of Network Security & Its Applications (IJSNA), Volume 3, Number 4, 2011.

[24] Jaydeep Howlader, Vivek Nair, Saikat Basu and A. K. Mal, "Uncoercibility In E-Voting And E-Auctioning Mechanisms Using Deniable Encryption," International Journal of Network Security & Its Applications (IJSNA), Volume 3, Number 2, 2011.

Authors

Manabu Okamoto

received the B.S. and M.S. degrees in Mathematics from Waseda University in 1995 and 1997, respectively. In 2010, he received the Doctor's degree of Global Information and Telecommunication from Waseda University. He is now an Associate Professor at Kanagawa Institute of Technology.