ELSEVIER

Computers & Security Vol. 15, No. 4, pp. 339–348, 1996

0167-4048/96 $15.00

PII: S0167-4048(96)00011-9

Refereed paper

A collision-free secret ballot protocol for computerized general elections

Wen-Shenq Juang and
Chin-Laung Lei

Room 246, New Electrical Engineering Building, Department of Electrical Engineering, National Taiwan University, Taipei, Taiwan, ROC
(lei@thunder.ee.ntu.edu.tw)

In secret ballot protocols, the unique voting property is crucial since without it a voter may vote more than once or his ballot may collide with others and be discarded by the authority. In this paper we present a collision-free secret ballot protocol based on the uniquely blind signature technique. Our proposed scheme can be used to hold large-scale general elections because it ensures independence among voters without the need for any global computation. This scheme preserves the privacy of a voter against the authority and other voters. Robustness is ensured in that no subset of voters can corrupt or disrupt the election. The verifiability of this protocol ensures that the authority cannot present a false tally without being caught. Copyright © 1996 Elsevier Science Ltd

Keywords: Privacy, Security, Secret ballot protocols, Uniquely blind signature schemes, Distributed systems.

1. Introduction

One of the hallmarks of democratic electoral systems is the institution of the secret ballot. Without ballot secrecy, the voters might be deterred from revealing their true opinions about the issues to be voted upon. In addition to ballot secrecy, every interested voter must vote exactly once. Voting more than once cannot be accepted by the authority and other voters. For ensuring that the authority cannot present a false tally without being caught, each voter can verify that his ballot has been counted and if not, he can ask the authority to recount his ballot. Since

electronic votes can be easily duplicated, there is a need to prevent malicious or careless voters from casting multiple votes. The naive approach of simply issuing a unique identification number to each voter would disclose the privacy of the voters. To overcome this difficulty many cryptographic protocols have been proposed [1–10].

In this paper we propose a cryptographic protocol for secret ballot elections with the following properties: (i) this protocol involves voters, the authority (so called the government); (ii) this protocol is collision free, i.e. a ballot of an eligible voter is always accepted by the authority; (iii) this protocol preserves the privacy of a voter against the authority and other voters; (iv) it is robust in that no voter can disrupt or corrupt the election; (v) when the authority publishes the votes, each voter can verify if his ballot has been counted and if not, he can ask the authority to recount his ballot. In this protocol the computations among voters are independent without the need for any global computation, so this protocol is a suitable scheme for large-scale general elections.

The remainder of this paper is organized as follows: in Section 2 previous work on secret ballot schemes is reviewed; in Section 3 we describe a uniquely blind signature technique and apply it to our proposed secret ballot protocol for achieving the uniquely voting property; our collision-free secret ballot protocol is presented in Section 4; the security considerations of this protocol are examined in Section 5; we discuss several issues in Section 6 and give a concluding remark in Section 7.

2. Related work

Some boardroom voting schemes [4–6] have been proposed in which voters openly send encrypted messages back and forth until they all are confident of the outcome of the election. The major problems of these schemes are that the computations of voters are not independent, and if any voter stops following the protocol during the voting the election is disrupted. Chaum [7] proposed a method of holding verifiable secret ballot elections similar to that of boardroom elections. A failure of a single voter can still disrupt the elections in that scheme, but it ensures that such failures can be traced. Nurmi *et al.* [3] proposed another secret ballot scheme based on ANDOS protocols [11,12]. For getting the authority's secrets as ballots, voters need to communicate with each other. Fujioka *et al.* [1] proposed a secret ballot scheme which is more suitable for large-scale elections since the computation and communication overhead is small even if the number of voters is large. The major problem of his scheme is that it requires all the registered voters to cast their votes and no voter can abstain from voting. Also, the failure of a single voter can disrupt the whole election process which makes the scheme impractical in real life.

Boyd has proposed a voting scheme based on the use of 'multiple key ciphers' [8,9]. It preserves the privacy of voters and ensures that ballots cannot be forged. The major problem with Boyd's scheme is that the tally is not verifiable, i.e. the authority can produce a false tally by adding votes of his own choice. This scheme uses random strings to distinguish each voter's ballot. In a distributed environment, voters may generate the same random strings via random number generators. This will result in some voter's ballot being discarded.

The schemes proposed in [3,8,9] are not collision free and [1,3–7] are not practical for large-scale elections.

Slessenger [13] proposed a socially secure cryptographic election scheme. It assures that all ballots of eligible voters have been correctly counted and the election result cannot be rigged by the authority. The major problem with his scheme is that ballots of the voters are public, i.e. everyone knows the intention of every other voter.

Iversen [2] proposed a voting scheme based on privacy homomorphism [14]. His scheme

preserves the privacy of the voters against the authority and other voters. Robustness is also ensured in his scheme. The verification of ballots can only be done by candidates. The essential drawback of this scheme is that if all candidates conspire, the privacy of the voters is violated. Moreover, this scheme is less practical for large-scale elections since it requires a great deal of communication and computation if the number of voters is large.

Benaloh *et al.* [15] proposed a receipt-free secret ballot protocol based on the probabilistic encryption method (PEM) and voting booths in which no more than two voters can stay at the same time. In their protocol, no one except the authority can coerce the voters into changing their intentions. This protocol is not a general election protocol since the intentions of voters are only either 'yes' or 'no'. In this scheme the privacy of any voter is preserved against all others except the authority.

3. Uniquely blind signature

The concept of blind signature schemes was proposed by Chaum [16,17]. Such systems have a party called the signer who is able to make certain digital signatures. The other parties, called requesters, would like to obtain such signatures on messages they provide to the signer. The major property of blind signatures is called 'unlinkability', i.e. the requester can prevent the signer from knowing the exact correspondence between the actual signing process performed by the signer and the signature which is later made public. In a distributed environment, assume that there are many persons requesting the authority for signing their blind messages. The signed blind messages can be thought as tickets in some applications, such as secret voting schemes [3,8–10]. If the contents of the signed messages are the same, these signed messages will be thought as only one ticket. Since these persons do not want to disclose their messages and the link between their identifications and the signatures, the blind messages may collide with each other. We call the above blind signatures which are collision free as *uniquely blind signatures*. The concept of blind signature and one-way permutation defined in this section will be used for constructing a uniquely blind signature scheme.

Let there be $n>1$ players in a distributed system and player i has his own secret s_i, where $1 \leqslant i \leqslant n$. A *secure computing protocol* for this system is a procedure for evaluating the function value $f(s_1, s_2, \ldots, s_n)$ jointly by the n players such that the output becomes commonly known while s_i remains secret. A secure computing protocol can be used to define blind signature schemes.

Definition 3.1. *Let M be the message to be signed which is owned by a requester, d be the secret key of the signer B, and S be the signature of M. A blind signature scheme is a digital signature scheme which can be processed in two phases: in phase 1 A and B compute the value $f(M,d)$ by a secure distributed computing protocol, where M is the secret of A and d is the secret of B; in phase 2 A computes the signature $S=g(f(M,d))$, where the function g is known only to A.* □

For example, the RSA blind signature scheme [16,17] is illustrated as follows. Let m be a message to be signed and s be the signature of m:

(1) The requester sends to the signer a message $m' = mR^e \bmod n$, where (e,n) is the public key of the signer and R is a random number chosen by the requester such that $\gcd(R,n)=1$.

(2) Upon receiving the message m', the signer generates its signature $s' = (m')^d \bmod n$ with his secret key d. Then he sends the message s' back to the requester.

(3) Upon receiving the message s', the requester can obtain signature s for m by computing

$$s = s'R^{-1} \bmod n$$

$$= ((mR^e \bmod n)^d \bmod n) R^{-1} \bmod n$$

$$= m^d RR^{-1} \bmod n$$

$$= m^d \bmod n.$$

The signer cannot derive m from m' since m' is transformed by the unknown random number R. On the other hand, the requester, knowing the value R, can compute the signature s of the message m from the message s'.

In the following we define the uniquely blind signature scheme which can be used to avoid the collision situation mentioned above.

Definition 3.2. *A uniquely blind signature scheme is a blind signature scheme such that: (i) this scheme involves one signer called the authority and n requesters R_i, where $1 \leqslant i \leqslant n$; (ii) the signing process, between any R_i and the authority, is a blind signature scheme whose signing function is injective; (iii) the messages to be signed are distinct.* □

Before giving an example of a uniquely blind signature scheme, one-way permutation functions must be defined. If the inputs of a one-way permutation function are the identifications of voters, it can hide the identification of each voter and ensure the uniqueness of the output values, since the identifications of voters are distinct.

Definition 3.3. *A one-way permutation function f is a bijective function such that: (i) there is a deterministic polynomial-time algorithm to compute f; (ii) for every probabilistic polynomial-time algorithm A, for every constant c and for any input x of size n, there exists a constant n_c such that the probability*

$$\text{prob}(f(A(f(x))) = f(x)) < n^{-c}$$

for all $n > n_c$. □

The existence of one-way permutations implies $P \neq NP$. Thus, no definitive examples have been found. The discrete logarithm function is an example of a candidate that is believed by some researchers to be a one-way permutation [18,20,21].

Assumption 3.1. *Discrete log assumption (DLA): let p be a prime, x be an integer and g be a generator (primitive root) of Z^*_p (i.e. $Z^*_p = \{g^x | x \, Z^*_p\}$). Define $DLP_{p,g}(x) = g^x \bmod p$, $1 \leqslant x \leqslant p-1$. DLA states that for every probabilistic polynomial-time algorithm A, for every constant c and for any input x of size n, there exists a constant n_c such that the probability*

$$\text{prob}(DLP_{p,g}(A(DLP_{p,g}(x)))) = DLP_{p,g}(x) < n^{-c}$$

for all $n > n_c$. □

Since Z^*_p is a cyclic group under the generator g, the function $DLP_{p,g}(x)$ is bijective and is easily computable. It follows that $DLP_{p,g}(x)$ is a one-way permutation function provided that DLA holds.

There are many famous cryptosystems [18–20] in which one of the underlying security assumptions is based on Assumption 3.1. For example, the underlying assumption of the Diffie and Hellman's public key distribution scheme [20] and the ElGamal's cryptosystem [19] are both based on the hardness of computing discrete logarithms.

Given a prime number p and the distinct prime factors of $p-1$, Lemmas 3.1 and 3.2 can easily randomly choose a generator g which makes $DLP_{p,g}(x)$ be a permutation function. When the function is implemented, distinct prime factors of $p-1$ can first be randomly chosen, and then p can be constructed from the known prime factors and tested to check that it is prime.

Lemma 3.1 [21]. *Suppose $p > 2$ is a prime and $p-1$ has k distinct prime factors which are known. Then g is a primitive root modulo p if $g^{(p-1)/q} \neq 1 (\text{mod } p)$ for every prime factor q of $p-1$. It can be done in time $O(k \log p) = O((\log p)^2)$ to determine if g is a primitive root mod p.* □

Lemma 3.2 [21]. *If p is a prime, then there exist $\phi(p-1)$ primitive roots modulo p, where ϕ denotes the Euler totient function and $\phi(p-1)$ is the number of positive integers $\leqslant p-1$ that are relatively prime to $p-1$.* □

Let n, e and d be the parameters in the RSA cryptosystem for the signer called the authority, f be a one-way permutation function and ID_A be the identification of a requester A. Let r and R_A be two random strings generated by A and M be the message to be signed. Without loss of generality, we assume that the identifications of voters are distinct. A uniquely blind signature scheme based on RSA blind signature scheme precedes as follows:

(1) The requester A first computes $Y=r^eM_A \bmod n$, where $M_A=H_A \cdot M$, $H_A=f(ID_A \cdot R_A)$ and $\cdot$ denotes the ordinal string concatenation operator, and then sends the message Y to the authority.

(2) Upon receiving the message Y, the authority computes

$$X=Y^d \bmod n$$

$$=(r^eM_A \bmod n)^d \bmod n$$

$$=rM^d_A \bmod n$$

and sends the message X to A.

(3) Upon receiving the message X, A computes the signature

$$S=r^{-1}X \bmod n$$

$$=r^{-1}(Y^d \bmod n) \bmod n$$

$$=r^{-1}(rM^d_A \bmod n) \bmod n$$

$$=M^d_A \bmod n.$$

The role of the random string R_A is to increase the security of the one-way permutation f. Since the entropy of user identifications is small, $H_A=f(ID_A \cdot R_A)$ is used to avoid the attack by an exhaustive search. It is clear that the above signature scheme is a uniquely blind signature scheme since this scheme is an RSA blind signature scheme [16,17] whose signing function is bijective and the signed message $M_A=H_A \cdot M=f(ID_A \cdot R_A) \cdot M$ is unique.

4. The collision-free secret ballot protocol

In this section a collision-free secret ballot protocol is presented. The protocol involves voters and the authority. The protocol consists of four phases: the initialization phase, the registration phase, the voting phase and the publication phase. During the initialization phase, the authority generates the system parameters: RSA keys, a public one-way permutation and a common public redundancy string for verifying ballots in the voting phase. In the registration phase, voters apply the uniquely blind signature technique described in Section 3 to get their blind ballots. In the voting phase, voters first generate their real ballots from the blind ballots received in the registration phase and then send them to the authority via an untraceable email. Finally, in the publication phase, the authority publishes the total ballots and checks if there exists any voter who votes more than once. If any voter finds that his ballot is misplaced or not counted, he has to ask the authority to recount his ballot.

The underlying assumptions of this protocol are:

(1) Every voter can communicate with the authority and he must follow the protocol properly.

(2) There exists a secure, untraceable electronic email system [22,23].

(3) The RSA cryptosystem is secure if factorization is intractable.

The *mix-net* approach is used in [22,24] to realize a sender-untraceable email system. In the mix-net approach the encrypted messages are sent to a mix agent who will disarrange all received messages and send them to the next mix agent. Finally, the last mix agent will send the encrypted messages to their destinations. The basic assumption of the

mix-net approach is at least one mix is honest. In [25] Pfitzmann shows several attacks on the anonymous channels proposed in [24]. The *dc-net* method is used in [23] to achieve a sender-untraceable email system which is unconditionally or cryptographically secure, depending on whether it is based on one-time keys or on keys generated by public key distributed systems, or pseudo random number generators. Both mix-nets and dc-nets can be applied to our protocol, but we recommend mix-nets since we need only an email system which performs periodic deliveries, not continuous deliveries.

Any secure, uniquely blind signature scheme is adequate for our protocol. For simplicity we adopt the RSA uniquely blind signature scheme in the following presentation. In our proposed protocol any sender authentication channel between the authority and voters is produced by the RSA signature system.

Let e_i, d_i and n_i be the RSA keys of eligible voter i, and every person can access the real public keys (e.g. e_i and n_i) of eligible voter i via an authentication channel.

First phase (the initialization phase):

(1) The authority randomly selects two large primes p, q and then chooses an integer e, $1 \leqslant e < n = p \times q$, such that $\gcd(e, \phi(n)) = 1$, where $\phi(n) = (p-1) \times (q-1)$, and computes d, $1 < d < n$, $e \times d = 1 \bmod \phi(n)$.

(2) The authority makes e and n public, and keeps d secret.

(3) The authority selects a public one-way permutation f and the public redundancy bits RD for verifying the validity of each ballot.

Second phase (the registration phase):

Let ID_i be the identification of voter i. Voter i chooses a random string R_i and determines his intention V_i. Voter i and the authority then perform the following protocol:

(4) Voter i generates a random value r_i, where $1 \leq r_i < n$ and $\gcd(r_i, n) = 1$, and computes the values $Y_i = (r_i^e M_i) \bmod n$, where $M_i = H_i \cdot RD \cdot V_i$ and $H_i = f(ID_i \cdot R_i)$, and $\text{Reg}_i = RD^{d_i} \bmod n_i$, and finally sends $\tilde{R}_i = Y_i \cdot \text{Reg}_i \cdot ID_i$ to the authority through an authentication channel.

(5) When receiving the message $\tilde{R}_i$ from the authentication channel, the authority checks if the received message is valid. If not, he will request voter i to retransmit the message $\tilde{R}_i$. After successfully receiving the message $\tilde{R}_i$, the authority checks the identification of voter i by verifying if $\text{Reg}_i^{e_i} \bmod n_i = RD$. If $\text{Reg}_i^{e_i} \bmod n_i \neq RD$, the authority rejects the registration of voter i. If $\text{Reg}_i^{e_i} \bmod n_i = RD$ and voter i has registered, the authority also rejects the registration of voter i. Otherwise, the authority records the fact that voter i has registered by keeping Reg_i in the registration database, computes $Z_i = Y_i^d \bmod n$ and sends Z_i to voter i.

Third phase (the voting phase):

(6) Upon receiving the message Z_i, voter i computes the value

$$\begin{aligned} X_i &= Z_i r_i^{-1} \bmod n \\ &= ((r_i^e M_i \bmod n)^d \bmod n)\, r_i^{-1} \bmod n \\ &= r_i M_i^d r_i^{-1} \bmod n \\ &= M_i^d \bmod n \end{aligned}$$

and sends (X_i, M_i) anonymously to the authority via an untraceable email.

(7) Upon receiving the message (X_i, M_i), the authority checks if $(X_i)^e \bmod n = M_i$. If this is true and RD is valid, the authority records (X_i, M_i).

Fourth phase (the publication phase):

(8) The authority sorts all ballots by their M_i, preserves only one copy of M_i and publishes all ballots (X_i,M_i) of voters and all registrations Reg_i of eligible voters.

(9) Voter i must check if his ballot has been properly counted by verifying that his ballot (X_i,M_i) is published by the authority. If voter i discovers that his ballot is not counted, he has to send (X_i,M_i) to the authority for recounting via an untraceable email. Every person can check if the total number of ballots is equal to the total number of the registrations to prevent the authority from adding extra ballots to the tally.

5. Security

The most important property of the secret ballot protocol is the privacy property. Moreover, the published tally must be equal to the actual result of the election, i.e. each voter must vote exactly once and the authority cannot add extra ballots to the total tally. We now show that our proposed scheme satisfies the above properties.

Based on the technique of uniquely blind signatures, we first show that the ballot of an eligible voter is always accepted by the authority.

Definition 5.1 (completeness). *A secret ballot protocol is said to be complete if the ballot of an eligible voter is always accepted by the authority.*

Theorem 5.1. *The secret ballot protocol of Section 4 is complete.*

Proof:
The proof is by contradiction. Assume that voter i follows the protocol properly and that his vote is rejected by the authority. In our protocol the ballot (X_i,M_i) of voter i can only be rejected by the authority in either Steps 5 or 8.

(1) If the ballot of voter i is rejected in Step 5, there are two possible cases:

Case 1: since every voter can communicate with the authority, the authority will receive the message $\tilde{R}_i$ in Step 5. It is impossible that $Reg_i^{e_i} \bmod n_i \neq RD$ since it violates the correctness of the RSA cryptosystem.

Case 2: assume that the authority finds that voter i has registered in Step 5. Then there exists a malicious person who can forge $Y_k \cdot Reg_i \cdot ID_i$, where $Y_k = (r_k^e M_k) \bmod n$ is chosen by this malicious person to impersonate voter i. It clearly contradicts to the assumption that the RSA signature scheme is secure since $Y_k \cdot Reg_i \cdot ID_i$ must be sent through an authentication channel and Reg_i can only be made by voter i.

(2) On the other hand, if the ballot of voter i is rejected in Step 8, since voter i must follow the protocol properly, the authority will receive the message (X_i,M_i) and record (X_i,M_i) in Step 7. In Step 8 assume that there exists another voter j such that $H_j = H_i$, and then his ballot (X_i,M_i) is rejected by the authority. Let ID_i be the identification of voter i, and f be the one-way permutation of the protocol. Since the identification of each voter is unique, we have $ID_i \cdot R_i \neq ID_j \cdot R_j$, $i \neq j$. Thus, $H_i = f(ID_i \cdot R_i) \neq f(ID_j \cdot R_j) = H_j$. Contradiction.

From the above, we conclude that the secret ballot of Section 4 is complete. □

Definition 5.2 (eligibility). *A secret ballot protocol is said to be eligible if only eligible voters can vote.*

Theorem 5.2. *The secret ballot protocol of Section 4 is eligible.*

Proof:
In our protocol an ineligible voter A can try to vote in the following ways:

(1) Send a used ballot of a previous election to the authority in Step 6.

(2) Impersonate an unregistered eligible voter i in Step 5 and send the received ballot (X_A, M_A), where M_A is chosen by A, to the authority in Step 6.

(3) Forge any valid ballot (X_A, M_A), where M_A is chosen by A, and send it to the authority in Step 6.

In every election, the authority chooses different RSA keys n, e and d. If the used ballots of a previous election can be used again, A can forge the signatures made by the authority. It clearly contradicts the assumption that the RSA signature scheme is secure, therefore the first method fails.

If A can impersonate an unregistered eligible voter i in Step 5, he can forge a valid message $Y_A \cdot \text{Reg}_i \cdot ID_i$ sent by voter i, where $Y_A = (r_A^e - M_A) \bmod n$ and M_A and r_A are chosen by A. It clearly contradicts the assumption that the RSA signature scheme is secure, since voter i sent $Y_i \cdot \text{Reg}_i \cdot ID_i$ to the authority through an authentication channel in Step 4 and Reg_i can only be made by voter i. Thus the second method fails.

If A can forge any valid ballot (X_A, M_A) in Step 6, he can forge signatures generated by the authority. It clearly contradicts the assumption that the RSA signature scheme is secure. Thus the third method fails.

From the above, the secret ballot protocol of Section 4 is eligible. □

We now describe how no voter can vote more than once.

Definition 5.3 (un-reusability). *A secret ballot protocol is said to be un-reusable if no voter can vote successfully more than once.*

By Theorem 5.2, only eligible voters can vote. In Step 8 of the protocol, the authority will sort the ballots by M_i and preserve only one copy of all duplicate votes. If any eligible voter casts his ballot more than once, only one vote will be counted to the total tally. So no voter can vote successfully more than once.

By Theorem 5.1 and the fact that the secret ballot protocol of Section 4 is un-reusable, we know that any interest voter will vote exactly once. Also, it is desirable that the authority cannot add extra ballots to the total tally.

Definition 5.4 (tally correctness). *The result of a secret ballot protocol is said to be correct if the published tally is equal to the actual result of the election.*

By Theorem 5.2, only eligible voters can vote. By Theorem 5.1 and the fact that the secret ballot protocol of Section 4 is un-reusable, a voter will vote exactly once. The authority must publish all registrations and ballots in the publication phase. In this protocol every voter must follow the protocol properly and then the total number of the ballots must be equal to the total number of registrations. Since every voter must check if his ballot has been counted properly and the total count of the registrations is equal to the total count of the published ballots, the authority cannot add any extra ballot to the tally. Therefore, the published tally is equal to the actual result of the election. It is clear that the result of the secret ballot protocol of Section 4 is correct.

Definition 5.5 (privacy). *A secret ballot protocol is said to be private if the privacy of voters is preserved.*

Theorem 5.3. *The secret ballot protocol of Section 4 is private.*

Proof:
In our protocol a malicious person can try to derive the intention of voter i only in the following ways:

(1) Derive the link between the message $\tilde{R}_i$ which is sent to the authority in Step 4 and the ballot (X_i, M_i) which is published in Step 8.

(2) Derive ID_i of voter i from his ballot (X_i, M_i) published in Step 8.

(3) Acquire the source address of the ballot (X_i, M_i) sent to the authority in Step 6.

To derive the link between the message $\tilde{R}_i$ which is sent to the authority in Step 4 and the ballot (X_i, M_i) which is published in Step 8 is computationally infeasible since it clearly contradicts the assumption that the RSA blind signature scheme is secure. Thus the first method fails.

To derive ID_i of voter i from his ballot (X_i, M_i) published in Step 8 is computationally infeasible since it clearly conflicts with the *DLA* assumption. Thus the second method fails.

To acquire the source address of the ballot (X_i, M_i) sent to the authority in Step 6 is computationally infeasible since it clearly conflicts with the availability of a secure, untraceable email. Thus the third method fails.

From the above, the secret ballot protocol of Section 4 is private. □

In our protocol, since every voter needs to communicate only with the authority, it is clear that no voter can corrupt or disrupt the election. On the other hand, if the authority does not disclose the partial result of the tally before the publication phase, the voting cannot be affected by anything since counting ballots is done after the voting phase and every voter only needs to communicate with the authority.

6. Discussion

6.1. Distributing the power of a single authority to several authorities

One of the basic assumptions of our protocol is that all the registered voters must cast their votes and no voter can abstain from voting. In real life registered voters may abstain from voting after the registration phase. To cope with this situation, some modifications of the secret ballot system in Section 4 must be made. The modifications are described below:

(1) Instead of a unique authority, the modified system consists of k authorities and at least one of them does not conspire with the others.

(2) The voting protocol between each authority and a voter is similar to the voting protocol in Section 4.

(3) During the initialization phase, every authority generates its system parameters: RSA keys, a public one-way permutation and a common public redundancy bits for verifying ballots in the voting phase.

(4) In the publication phase any interested voter must check if his vote has been properly counted. If his ballot is misplaced or not counted by any authority, he has to send his ballot to the authority for recounting via an untraceable email. To prevent any malicious authority from adding extra ballots to the tally, anyone can check that the total numbers of the ballots published by all authorities are the same.

From the above modifications, the power of a single authority is distributed among several authorities and all registered voters may abstain from voting after the registration phase.

6.2. Receipt-free secret ballots

It is clear that our protocol is not receipt free. If voter i wants to sell his vote to a buyer, he only discloses the value ID_i, and R_i generated in Step 4 to the buyer. The buyer can check if $H_i = f(ID_i, R_i)$ and the intention V_i contains his requirement.

It is clear if there is no voting booth in a voting system, then the voting system is not receipt free. The reason is that the buyer only has to participate the voting process when the voter is voting.

The receipt-free secret ballot scheme proposed by Benaloh *et al.* [15] is not a general election since the intention of voters is constrained to 'yes' or

'no'. It is still an open problem that there does not exist a receipt-free secret ballot protocol for computerized general elections.

7. Conclusion

In this paper we propose a secure and practical election scheme for computerized general election which provides completeness, privacy, robustness, verifiability, un-reusability and eligibility properties. The most important property of this scheme is the completeness property, i.e. the ballot of an eligible voter is always accepted by the authority. In addition, our protocol is suitable for large-scale elections since the communication and computation overhead is small, even if the number of voters is large.

References

[1] A. Fujioka, T. Okamoto and K. Ohta, A practical secret voting scheme for large scale elections, in *Advances in Cryptology: Proceedings of AusCrypt'92*, LNCS 718, pp. 244–251, Springer, Berlin, 1992.

[2] K.R. Iversen, A cryptographic scheme for computerized general elections, in *Advances in Cryptology: Proceedings of Crypt'91*, LNCS 576, pp. 405–419, Springer, Berlin, 1991.

[3] H. Nurmi, A. Salomaa and L. Santean, Secret ballot elections in computer networks, *Computers & Security*, 10 (1991) 553–560.

[4] C.P. Pfleeger, *Security in Computing*, Prentice-Hall, Englewood Cliffs, NJ, 1989.

[5] A. Yao, Protocols for secure communications, in *Proceedings of the 23rd Annual IEEE Symposium on the Foundations of Computer Science*, pp. 160–164, 1982.

[6] R. Demillo, N. Lynch and M. Merritt, Cryptographic protocols, in *Proceedings of the 14th Annual ACM Symposium on the Theory of Computing*, pp. 382–400, 1982.

[7] D. Chaum, Elections with unconditionally secret ballots and disruption equivalent to breaking RSA, in *Advances in Cryptology: Proceedings of EuroCrypt'88*, LNCS 330, pp. 177–182, Springer, Berlin, 1988.

[8] C. Boyd, Some applications of multiple key ciphers, in *Advances in Cryptology: Proceedings of EuroCrypt'88*, LNCS 330, pp. 455–467, Springer, Berlin, 1988.

[9] C. Boyd, A new multiple key ciphers and an improved voting scheme, in *Advances in Cryptology: Proceedings of EuroCrypt'89*, LNCS 434, pp. 617–625, Springer, Berlin, 1990.

[10] W.S. Juang, C.L. Lei and C.I. Fan, A collision free secret ballot protocol for computerized general elections, in *International Computer Symposium*, Taiwan, pp. 309–314, 1994.

[11] G. Brassard and C. Crepeau, All-or-nothing disclosure of secrets, in *Advances in Cryptology: Proceedings of Crypt'87*, LNCS 293, pp. 234–238, Springer, Berlin, 1987.

[12] A. Salomaa & L. Santean, Secret selling of secrets with many buyers, *EATCS Bulletin*, 42 (1990) 178–186.

[13] P.H. Slessenger, Socially secure cryptographic election scheme, *Electronics Letters 23rd*, 27 (11) (1991).

[14] K.R. Iversen, A novel probabilistic additive privacy homomorphism, in *Proceedings of the International Conference on Finite Fields, Coding Theory, and Advances in Communications and Computing*, Lecture Notes in Pure and Applied Mathematics, Dekker, New York, 1991.

[15] J. Benaloh and D. Tuinstra, Receipt free secret ballot elections, in *Proceedings of the 26th Annual ACM Symposium on the Theory of Computing*, pp. 544–553, 1994.

[16] D. Chaum, Blind signatures systems, in *Advances in Cryptology: Proceedings of Crypt'83*, p. 153, Plenum Press, New York, 1983.

[17] D. Chaum, Blinding for unanticipated signatures, *Advances in Cryptology: Proceedings of EuroCrypt'87*, LNCS 304, pp. 227–233, Springer, Berlin, 1987.

[18] S. Pohlig and M.E. Hellman, An improved algorithm for computing logarithms over GF(p) and its cryptographic significance, *IEEE Transactions on Information Theory*, IT-24 (1978) 106–110.

[19] T. ElGamal, A public key cryptosystem and a signature scheme based on discrete logarithm, *IEEE Transactions Information Theory*, 31 (1985) 469–472.

[20] W. Diffie and M.E. Hellman, New directions in cryptography, *IEEE Transactions on Information Theory*, IT-22 (1976) 644–654.

[21] G.J. Simmons, *Contemporary Cryptology—the Science of Information Integrity*, pp. 266–268, 1992.

[22] D. Chaum, Untraceable electronic mail, return addresses, and digital pseudonyms, *Communications of the ACM*, 24 (2) (1981) 84–88.

[23] D. Chaum, The dining cryptographers problem: unconditional sender and recipient untraceability, *Journal of Cryptology*, 1 (1988) 65–75.

[24] C. Park, K. Itoh and K. Kurosawa, Efficient anonymous channel and all/nothing election scheme, in *Advances in Cryptology: Proceedings of EuroCrypt'93*, LNCS 765, pp. 248–259, Springer, Berlin, 1994.

[25] B. Pfizmann, Breaking an efficient anonymous channel. *Advances in Cryptology: Proceedings of EuroCrypt'94*, LNCS 950, pp. 332–340, Springer, Berlin, 1995.