Analysis and Design of E-voting Protocol

JIANG SHAOQUAN, FENG DENGGUO, QING SIHAN
State Key Laboratory of Information Security
Engineering Research Center for Information Security Technology
Chinese Academy of Sciences, Beijing 100080 ,P.R.China
Jiangshq@sina.com

Key words: Voter, Security, Protocols, Key Distribution

Abstract: A number of drawbacks in previous electronic voting schemes are analyzed in this paper. A new voting protocol is proposed. As results, giving up voting rights is allowed, low computational complexity is achieved and wide application is embodied. In the end, the security of the protocol is proved.

1. INTRODUCTION

In cryptographic literature, electronic voting is known as multi-party computation. Lots of papers[1-5] discussed the problem in recent years. Under different design principles, kinds of protocols are proposed or improved. But the final purpose is common: to achieve a more practical, more efficient, and more secure protocol. We believe these following principles are important:

(1) completeness

If all the participants are honest, the votes will be counted correctly.

(2) Fast speed

For the practical purpose, all the participants only have to carry out the least computation.

(3) Privacy

All the votes are secret, by which the privacy of voters is protected.

(4) Security

Cheats from any person including voters, outsiders, administrators, and

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35515-3_53

S. Qing et al. (eds.), *Information Security for Global Information Infrastructures*
© IFIP International Federation for Information Processing 2000

the counter, or their collusion will not be successful under reasonable assumptions.

(5) Robust

Leak of minor information will not compromise the system.

(6) Verifiability

Every voter can check whether his vote is counted and the result is verifiable.

According to these principles, we analyze known e-voting protocols and point out their limitations. For example, the protocol in [11,9,10,5,22] did not consider the setting where voters can abstain from voting or the method used in it is not valid. As result, the administrator can handle their votes freely. The protocol in [9,10] can only apply to Yes/No voting. In fact, it's only a two-scale type appraisement of candidates. It's too rough in many settings. In [5], enough administrators can derive the recovery key and thus the voting is compromised. In [9,10,22] the complexity in the protocol is very high. In paper [10], for another example, proven-sum protocol and prove ± 1 protocol need to carry out many times for security. It is also a Yes/No voting protocol.

In this paper, drawbacks are noticed and with the frame of protocol in [11], a new protocol is proposed, which has the following feathers:

(i) Renunciation of voting rights is allowed;

(ii) Illegitimate votes can't be counted while legitimate votes can be surely counted;

(iii) Any cheat from voters or administrators and the counter or even their collusion will be frustrated on the assumption that at least two administrators do not take part in a given collusion. Further, dishonest persons will be found out.

(iv) Efficiency is investigated and confirmed by computing and comparing with that of the protocol in [9]. In the end, we conclude that our protocol is superior to known protocols.

In the following sections, this paper is organized as follows. The second section points out the drawbacks in previous voting protocols. The third section shows how to proof the equality of discrete logarithm. The fourth section proposes a new key distribution scheme and the fifth section proposes a new e-voting protocol and proves the security of the protocol and analyzes the complexity. The last section concludes this paper.

2. EQUALITY OF DISCRETE LOGS

This section we introduce methods that prove the equality of discrete logs.

2.1 Interactive Protocol [16]

p,q are large primes and $q \mid p-1$. g,h are elements of F_p with order q. Prover publishes $(x,y)=(g^{\alpha},h^{\alpha})$, where $\alpha \in Z_q$, and he will show the Verifier that $\log_g x = \log_h y$.

(1) Prover randomly selects $w \in Z_q$, computes $(a,b)=(g^w,h^w)$, and sends (a,b) to Verifier.

(2) Verifier randomly selects $c \in Z_q$ and sends it to Prover.

(3) Prover computes $r = w + c\alpha$ and sends it to Verifier.

(4) Verifier checks whether $g^r = ax^c$ and $h^r = by^c$. If yes, he believes $\log_g x = \log_h y$. Otherwise, he refuses to accept the fact.

More technical application of this method can be found in [17,9].

2.2 Non-interactive Verification

Many papers [6,18,19,20] use a hash function to make an interactive protocol non-interactive. We keep the same notations as above. Assume $H(\cdot)$ is a one-way hash function. Prover computes $c = H(a,b)$ and $r = w + c\alpha$. He sends (r,c) to Verifier.

Verifier checks whether $c = H(g^r x^{-c}, h^r y^{-c})$. If yes, he accepts the verification. Because c is a random to the prover, the security of the non-interactive verification is as secure as the interactive protocol above. Anyway, [21] has studied the security of using hash function to replace random oracle. And the conclusion is that such replacement is not always theoretically secure. So one must be careful of such replacement.

3. A NEW KEY DISTRIBUTION SCHEME

In this section, we proposes a new key distribution scheme for n share holders such that their share a_j's satisfies $\sum_{j=1}^{n} a_j t_j = 1$, where (t_1, t_2, Λ, t_n) is the first row of the matrix

$\begin{pmatrix} 1 & 1 & \Lambda & 1 \\ 1 & 2 & \Lambda & 2^{n-1} \\ & O & \Lambda & \\ 1 & n & \Lambda & k^{n-1} \end{pmatrix}^{-1}$ and a_j is the share of the jth share holder. p,q are large primes, with $q \mid p-1$; G is a group with order q over F_p; g is a generator of G.

(1) The jth share holder randomly selects $(c_{j1}, c_{j2}, \Lambda, c_{jn}) \in Z_q^{\ n}$, lets $f_j(x) = 1 + \sum_{i=1}^{n} c_{ji} x^i$, computes $a_{ji} = f_j(i), \beta_{ji} = g^{a_{ji}}$, publishes β_{ji} and sends a_{ji} to the ith share holder in a secret channel.

(2) The jth share holder verifies whether

$$\beta_{ij} = g^{a_{ij}}, i = 1,2,\Lambda, n \qquad \text{(I)}$$

$$\prod_{i=1}^{n} \beta_{ki}^{\ t_i} = g, k = 1,2,\Lambda, n \qquad \text{(II)}$$

If all the equations hold, he computes

$a_j = n^{-1} \sum_{i=1}^{n} a_{ij} \bmod q$, $\beta_j = (\prod_{i=1}^{n} \beta_{ij})^{n^{-1}}$, publishes β_j and keeps a_j as his secret key.

Theorem 1 The generation of the secret keys $a_1, a_2 \Lambda, a_n$ is secure under the assumption that no more than n-2 share holders collude.

Proof: Because (II) is publicly verifiable, it holds all k, therefore $\sum_{j=1}^{n} a_{kj} t_j = 1$ .Futhermore,for given$(a_{k1}, a_{k2} \Lambda, a_{kn}) \in Z_q^{\ n}$ with $\sum_{j=1}^{n} a_{kj} t_j = 1$, one can easily find a polynomial $f_k(x)$ over $Z_q[x]$ with degree n such that its constant item is 1 and $a_{kj} = f_k(j), j = 1,2,\Lambda, n$. So in fact, $(a_{k1}, a_{k2} \Lambda, a_{kn}) \in Z_q^{\ n}$ hidden in $(\beta_{k1}, \beta_{k2} \Lambda, \beta_{kn})$ is generated in the right way.

The generation of β_k is verifiable. On the other hand, for any given n-2 share holders, say 1,2,...,n-2, there are q polynomials $f_k(x)$ over $Z_q[x]$ with degree n such that its constant item is 1, $\sum_{j=1}^{n} a_{kj} t_j = 1$, and $a_{kj} = f_k(j), j = 1,2,\Lambda, n-2$ where k=n-1,n. Because of discrete problem,

they are equivalent to be a guess of the kth share holder's selection of $f_k(x)$. Therefore, the keys $a_1, a_2 \Lambda\ , a_n$ are safely generated.

4. DESIGN AND ANALYSIS OF A NEW PROTOCOL

We assume Ad1, Ad2,... Adk are k administrators. C is the counter, Vi is the ith voter. We assume one collusion set always includes no more than $k-2$ administrators. p, q are large primes, with $q \mid p-1$; G is a group with order q over F_p; g is a generator of G ; a_j is the private key of Adj; $\gamma_j = g^{a_j}$ is the public key of Adj, $j = 1,2,\Lambda\ k$. $(t_1, t_2, \Lambda\ , t_k)$ is the first row of the matrix $(i^{j-1})_{1\le i\le k, 1\le j\le k}^{-1}$ such that $\sum_{j=1}^{k} a_j t_j = 1$.(See section 4), $\xi(\cdot,\cdot)$ is a one-way hash function over finite field F_p. In addition, a bulletin board is used, on which only administrators and the counter can publish information in the entitled field. We assume the information on the board is just what the writer writes. And we also assume that before the election every voter has registered his ID and the public key that matches his private key by physical means and before the election every one's ID and his corresponding public key are both published on the bulletin board. We also assume that the communication between Adj and Vi, between Vi and C is encrypted. We use an anonymous channel [8,15] so that anyone can't relate the sender to the receiver.

4.1 Our Protocol

(1) Voter Vi fills in his vote v_i, computes the committed ballot[12,13,14] $x_i = \xi(v_i, k_i)$, where k_i is a random; then computes $e_i = g^{r_i} x_i$ to hide the ballot, generates his signature $S_i = Sign_i(e_i' \| EM)$,where e', EM are part bits and specific information of the election, respectively. Then he sends (ID_i, e_i, S_i) to Adj, $j = 1,2,\Lambda\ ,k$.

(2) Adj checks the status of Vi's voting. If Vi has voted, he refuses Vi's request to vote. Otherwise, Adj verifies Vi's identity and signature pair (ID_i, e_i, S_i) with Vi's public key on the bulletin board. If the verification is successful, then he computes $\alpha_{ij} = e_i^{a_j}$, sends it to Vi and records that Vi has voted and keeps the vi's signature. Otherwise, he sends failure information to Vi.

(3) If Vi receives failure information but he is sure of the validity of the data he provides, he can claim the case (seeing the signature is

publicly verifiable, so it is easy to adjudicate). Otherwise if he receives all the α_{ij}'s, he verifies whether

$$\prod_{j=1}^{k} \alpha_{ij}^{t_j} = e_i .$$

If it holds, he goes on to step (4), otherwise he asks the administrators to carry out the interactive protocol or non-interactive protocol in the third section to check whether $\log_{e_i} \alpha_{ij} = \log_g \gamma_j$.

(4) Vi computes $\beta_{ij} = \gamma_j^{-r_i} \alpha_{ij}$, sends (x_i, β_{ij}) to Adj via anonymous channel, $j = 1,2,\Lambda\ ,k$.

(5) Adj verifies that whether $x_i^{a_j} = \beta_{ij}$. If the verification holds, then adds data β_{ij} to the lth list of the bulletin board where x_i is (if the list does not exist, he just creates one and fills x_i and β_{ij} in.) and then lets β_{ij} appear green. Otherwise, red.

(6) If Vi finds all the β_{ij} 's $j = 1,2,\Lambda\ ,k$ appear green, he sends (l, k_i) to C through an anonymous channel. Otherwise, he claims the case (If Vi finds β_{ij} is red, then he can ask Adj to compute $\overline{\beta}_{ij} = x_i^{a_j}$. They carry out the protocol in the third section to check whether $\log_{x_i} \overline{\beta}_{ij} = \log_g \gamma_j$.

(7) When the counter C receives (l, k_i), he uses k_i to open v_i in x_i corresponding to l and counts correctly and adds v_i to the list in which x_i is. When the election is over, he publishes the result.

4.2 Appraisement of the Protocol

In this section we will evaluate our protocol according to the principles stated in the introduction.

(1) **Completeness**

Completeness means if all the participants are honest, every vote can be counted correctly. We only verify step (3). In fact,

$$\prod_{j=1}^{k} \alpha_{ij}^{t_j} = e_i^{\sum_{j=1}^{k} a_j t_j} = e_i .$$

(2) Security

(i) Legitimate voter can't cheat successfully

Although the legitimate voter can vote successfully, he perhaps wants to disrupt the election. And the possible ways are filling in a false vote at step (1), making a false claim at step (3) and

sending false β_{ij}. Now the committed ballot is opened with k_i and it's publicly verifiable, so the false vote is invalid. And the false claim and false β_{ij} will be detected and further will be made clear by the protocol in [6] or [9].

(ii) An illegitimate voter together with Adj and C can't cheat successfully.

We have assumed that the voter's signature will be verified, the size of the collusion set of administrators is no more than k-2 (thus we assume that Ad1 and Ad2 would not join the collusion) and the vote is different between different elections. Either the illegitimate voter can't get through the signature verification by Ad1 and Ad2 or e_i includes no information of this election. In the former case, to assure $\overline{\beta}_{i1}, \overline{\beta}_{i2}$ appears green, the voter must construct $\overline{\beta}_{i1} = \overline{x}_i^{a_1}, \overline{\beta}_{i2} = \overline{x}_i^{a_2}$, but a_1, a_2 are random to the cheats (but $\sum_{j=1}^{k} a_j t_j = 1$, so they are not independent). So the useful information is some pairs $(\beta'_{i1}, x'_i), (\beta'_{i2}, x'_i)$. To get a_1, a_2, the cheats must solve discrete logarithm over G, which is believed to be difficult. In the latter case, e_i has no vote information, to get the pairs $(x_i, x_i^{a_j})$, $j = 1,2$, he can only computes $\xi(\cdot,\cdot)$ with selected parameters because $\xi(\cdot,\cdot)$ is a one way hash function. Maybe he can make use of the known pairs $(x_i'^{a_1}, x'_i), (x_i'^{a_2}, x'_i)$, it also needs solving a discrete problem, which is difficult.

Now it's clear that any giving up voting rights is allowed, because others can't impersonate them without being caught.

(iii) Prevention of revote attack

We note that at the second step, Adj's will check the status of the voter's voting. See that the signature of the voter is publicly verifiable, any dishonest Adj who will disclaim the voter's voting status with mistake will be found out.

(3) **Privacy**

The vote is committed in x_i by a one-way function and further blinded in e_i. the blinding signature provides the separation between the identification and anonymous communication. Therefore, the commitment hides the vote until the deadline.

(4) **Robust**

Let's see what will happen if some information is leaked. Because the voters take part in voting independently, the leak of his parameters k_i, r_i, etc only will compromise his vote. The counter is only a public operator and he has no private information. And as to administrators, even if they collude, the voting scheme will as secure as the case that all the administrators are honest as long as the cheaters can not know all the private keys of administrators. That's, no less than two administrators' keys are not leaked out and kept secret from the cheaters. The reason is the same as (ii).

(5) **Verifiability**

x_i, v_i, k_i are published on the bulletin board. And so every vote can know whether his vote is counted correctly. Furthermore, when the result is published, any person can check the result by manually counting.

(6) Complexity

First, let's see the complexity of the election. Because the entire claim will be made clear, we are not going to count the cost of the interactive protocol in [9] or the non-interactive protocol in [6]. We will also compare our protocol with that of [9]. We will see our protocol is very efficient. **Calculation cost by voter:** 1 hash function $\xi(\cdot,\cdot)$, $2\lceil \log p \rceil + 2k\lceil \log p \rceil + 2 * 2\lceil \log p \rceil + 2\lceil \log p \rceil \approx 2(k+4)\log p$ times multiplications over F_p. **Calculation per voter byAdj:** One time signature verification, $2\lceil \log p \rceil + 2\lceil \log p \rceil \approx 4\log p$ multiplications over F_p. **Calculation cost per voter by counter C:** One time hash function $\xi(\cdot,\cdot)$ calculation. Now let's see the complexity of [9]. Let $\bar{k}, l$ denote the number of candidates, the number of voters, respectively, then **cost by voter Vi is:** $2^{\bar{k}} \cdot 10 \cdot \log p$ multiplications over F_p. **Cost per voter by administrator:** $(2\binom{l+\bar{k}}{\bar{k}-1} \cdot \log^{k-1} l + 2k \log p) l^{-1}$ times multiplications over F_p. See $l >> k \log p$, so it is $O(l^{\bar{k}-2})$. Because counter's cost is very low, our protocol is much more efficient than that of [9]

(7) **Wide application**

Although it is achieved in [11], it is deserve a mention again, because many papers neglect it by proposing yes/no voting schemes. Ours and that of [11] can apply to general elections.

(8) **Permanence**

Although voters get their private key by physical means, it's easy to see that they don't have to change their key in the next election. If they want to change their key, they don't have to do it manually, but on the

Internet with his pre-changed key as his modification password. If he fails to do it, he takes up the physical means because his private key has been amended by others. In addition, the administrators can change their private key at the same time but independent of the voters. They also don't have to change it if no cheat from them appears, because the only leaked information is some pairs (y, y^{a_i}), where $y \in F_p, i = 1,2,\Lambda, k$ and they will be found out whenever they cheat.

5. CONCLUSION

This paper analyzes and points out the drawbacks of known e-voting scheme. Then we come up a new protocol which allows renunciation of voting rights, carries low complexity and can apply not only to yes/no voting. We prove the completeness, security and show the efficiency by comparing it with the protocol of [9].

(* This work is supported by NSF(No. 19931010) and 973 Project(No. G1999035802).)

Reference

[1] J.Cohen and M.Fischer. A robust and verifiable cryptographically secure election scheme. In proc. 26th IEEE Symposium on Foundation of Computer Science (FOCS'85), pp 372-382. IEEE Computer Society, 1985.

[2] J. Benaloh and M.Yung. Distribution the power of a government to enhance the privacy of voters. In Proc. 5th ACM Symposium on Principles of Distributed Computing (PODC'86), pp52-62, New York,1986. A.C.M.

[3] J.Benaloh. Verifiable Secret-Ballot Elections. Ph.D. thesis, Yale University, Department of Computer Science Department, New Haven, CT, September 1987.

[4] J. Benaloh and D.Tuintra. Receipt-free secret-ballot elections. In Proc. 26th Symposium on Theory of Computing (STOC'94), pp544-553, New York, 1994. A.C.M.

[5] R.Cramer, M.Fraklin, B.Schoenmakers, and M.Yung. Multi-authority Secret ballot elections with linear work. In Advance in Cryptology-EUROCRYPT'96, Volume 1070 of Lecture Notes in Computer Science, pp72-83,Berlin, 1996. Springer-verlag.

[6] A.Fiat and A.Shamir. How to proof yourself: Practical solutions to identification and signature problems. In Advances in Cryptology-CRYPTO'86, Volume 263 of Lecture Notes in Computer Science, pp186-194,New York, 1987. Spring-verlag.

[7] R.Gennaro. Achieving independence efficiently and securely. In Proc. 14th ACM Symposium on Principles of Distributed Computing (PODC'95), New York, 1995. A.C.M.

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

[9] R.Cramer, R.Gennaro and B. Schoenmakes, A secure and Optimally Efficient Multi-Authority Election Scheme, Advances in Cryptology-Eurocrypt'97,pp103-118, Springer-Verlag.

[10] K. Sako and J.Kilian, Secure Voting Using Partially Compatible Homomorphisms, Advances in Cryptology-Crypto'94, pp411-424, Springer-Verlag.

[11] Fujioka,A., Okamoto,T., Ohta,K., A pracatical secret voting scheme for large scale elections, Advances in Cryptology-Ausocrypt'92, Spring-verlag,1993, pp.244-251.

[12] Schneier, B., Applied Cryptography: Protocols, Algorithms, and Source Code in C, John Wiley & Sons, New York, 2nd Edition,1996.

[13] Naor,M., Bit Commitment Using Pseudo-randomness,Advances in Cryptology-Crypt'89, Springer-verlag, 1990,pp128-136.

[14] Brassoud,G.,Chaum,D., Crepeau,C., Minimum Disclosure Proofs of Knowledge,Journal of Computer and System Science, 37(1988), pp156-189.

[15] C.Park, K. Itoh, and K. Kurosawa, Efficient Anonymous Channel and all/Nothing Election Scheme, Advances in Cryptology- Eurocrypt'93, Springer-verlag, 1994, pp248-259

[16] D. Chaum and T.P. Pedersen, Wallet databases with observers, In Advances in Cryptology-Eurocrypt'90, Volume 473 of Lecture Notes in Computer Science, pp89-105, Berlin,1993, Springer-verlag.

[17] Y. Frankel, Y. Tsiounis and M. Yung, Fair Off-line e-cash Made easy, advances in Cryptology-Asiacrypt'98, LNCS 1514, pp 257-270, Beijing, 1998, Springer-verlag.

[18] D. Pointcheval and J. Stern, Provably Secure Blind Signature Scheme, Advances in Cryptology-Asiacrypt'96 ,Lecture Notes in Computer Science 1163, pages 252-265, November 3-7, South Korea.

[19] C.P. Schnorr, Efficient Identification and Signature for Smart Cards. In G.Brassard, Editor, Advances in Cryptology-Proceedings of Crypto'89, LNCS 435, pp 235-251, Springer-verlag.

[20] F. Bao and R. Deng, An Efficient Fair Exchange Protocol with an Off-line Semi- Trusted Third Party, The International Workshop on Cryptographic Technique & E -Commerce, M. Blum and C.H.Lee, Editors, pp37-47, City University of HongKong Press, Hongkong

[21] R.Canetti, O.Goldreich and S.Halevi, The random oracle methodology, revisit, in he proceedings of STOC'98.

[22] Chun-I, Fan and Chin-Laung Lei, A Multi-recastable Ticket Scheme For Electronic Election, Advances in Cryptology-Asiacrypt'96 ,Lecture Notes in Computer Science 1163, pages 117-124, November 3-7, South Korea.