Attacks and Modifications of CJC's E-voting Scheme *

Bennian Dou[1,#], Chun-hua Chen[2], Roberto Araujo[3]

[1] School of Science, Nanjing University of Science and Technology, Nanjing 210094, China

[2] Department of Electronic Engineering, Chienkuo Technology Unversity, Changhua 500, Taiwan

[3] Department of Computer Science, Darmstadt University of Technology, Darmstadt 64289, Germany

Abstract: In this paper, we point out the security weaknesses of Chen et al.'s e-voting scheme. We give a modification which satisfies the security requirements of a e-voting scheme.

Key words: Cryptanalysis, Election Schemes

1 Introduction

Electronic voting schemes have been proposed in the last two decades [1,2,3] as an alternative to paper-based voting systems.

Generally speaking, an e-voting scheme includes three phases:

Initialization phase: During this phase, the voters should register in some voting authority to get the right to vote and some parameters are also chosen.

Voting phase: The voters cast the desired ballots by using some communication technologies.

Counting phase: After the voting phase, the votes are counted and the result of the voting is published.

An e-voting scheme should offer at least the same security as paper-based voting systems. Although a standard of e-voting schemes is still not proposed, as described in [4], the following requirements are acceptable among the researchers:

Accuracy: All valid votes are counted correctly. A voter's vote cannot be altered, duplicated, or removed

Fairness: No one can learn the voting outcome before the tally.

Eligibility: Only eligible voters are permitted to vote.

Uniqueness: No voter is able to vote more than once.

Uncoercibility: No voter can prove how he voted to others to prevent bribery.

Anonymity: There is no way to derive a link between the voter's identity and the marked ballot. The voter remains anonymous.

Verifiability: A voting scheme is verifiable if every voter can independently verify that his ballot has been counted correctly.

There are some other requirements mentioned in [4] which are not so important in the discussion of this paper, so we do not consider such security requirements.

Recently, Chen et al. proposed a secure anonymous internet voting scheme [4]. They claimed that their scheme satisfied all the security requirements cited above. In this paper, we point out

* The first author is supported by the Research Foundation of Nanjing University of Science and Technology and CSC (Chinese Scholarship Council).

Corresponding author. E-mail: doubennian@yahoo.com.cn

that CJC's scheme does not satisfy the security requirements of accuracy, Fairness Uncoercibility, and verifiability.

The rest of this paper is organized as follows. In section 2 we briefly describe how CJC' voting scheme works. In section 3 we point out the weaknesses of their scheme. In section 4 we give a modification of their scheme. In section 5 we analyze the security of our scheme. We conclude the paper in section 6.

2 CJC's voting scheme

In this section we recall how CJC's scheme works.

2.1 Notations

V_i: Voter *i*.

v_i: A pseudonym chosen by V_i.

CA: Certificate authority which is a certificate service provider for all enrolled elections.
AC: Authentication center which is responsible for certifying all voters.
PS: A trusty public proxy server allows a voter to cast ballot without leaking his own IP address which can be used to link him.
TC: Tally center which is responsible for tallying the votes.
SC: Supervision center constructed by different politic parties which is responsible for supervising the *TC*.

2.2 CJC's scheme

Their scheme consists of three phases

Initialization phase

SC and *TC* use RSA encryption scheme, they publish N_π (a big number which is a multiplication of two big primes) and their common public key PK_π .They share the private key SK_π ($PK_\pi SK_\pi$ =1 mod $\phi(N_\pi)$) by using secret sharing $SK_\pi = S_{SC} + S_{TC}$. S_{SC} and S_{TC} are the secret shadows of *SC* and *TC* respectively. The following steps will be done.

$V_i \leftrightarrow CA$: An eligible voter V_i registers himself in *CA* and gets a "personal certificate".

$V_i \leftrightarrow AC$: V_i sends his "personal certificate" and the pseudonym v_i chosen randomly by V_i to *AC*, *AC* check the "personal certificate". V_i gets the signature s_i of *AC* from v_i by using blind signature scheme. *AC* allows each “personal certificate” to get a signature only once.

Voting phase

V_i chooses a random number β and encrypts a marked ballot *m* as $b = (\beta \oplus m)^{PK_\pi} \mod N_\pi$.

V_i sends (v_i, s_i, *b*, β) through a trusty public proxy server to both *SC* and *TC*. *SC* and *TC* verify the signature of v_i, they refuse multiple-voting by using the same pseudonym. Each of them records the (v_i, s_i, *b*, β) into their respective database.

Counting phase

After the voting deadline, each ballot will be decrypted with the cooperation of *SC* and *TC*. They compute $T_{SC} = (b)^{S_{SC}} \mod N_\pi$ and $T_{TC} = (b)^{S_{TC}} \mod N_\pi$ respectively. Then the marked ballot *m* can be decrypted as $(T_{SC}\ T_{TC}) \oplus \beta \mod N_\pi = m$. Under the supervision of *SC*, *TC* publishes the voting results.

2.3 Security analysis

Chen et al. claimed that their scheme satisfies all the security requirements which were described in section 1.

In the next section, however, we will point out that their scheme does not satisfy the security requirements of accuracy, privacy, and verifiability.

3 Weaknesses of CJC’s scheme

3.1 CJC’s scheme is not accurate

In the voting phase, voter V_i sends (v_i, s_i, *b*, β) through a trusty proxy server to *SC* and *TC*. As (v_i, s_i) is not encrypted and is open to anybody, an attacker can alter (v_i, s_i, *b*, β) by replacing (*b*, β) with his choice b', β', where b' is the ciphertext of his intended ballot. Then, he sends (v_i, s_i, b', β') to *SC* and *TC*, and keeps (v_i, s_i, *b*, β). (v_i, s_i, b', β') will be considered as a valid vote by *SC* and *TC* , so it can be counted in the result of the voting.

3.2 CJC’s scheme is not fair

All ballots do not remain secret while the voting is not completed. The number of different marked ballots *m*’s is very small. When an attacker get (v_i, s_i, *b*, β), it is possible to encrypt the several different marked ballots and compare the ciphertexts with *b*, thus, he can know which *m* is correspondent to *b*, for β and the public key PK_{π} are publicly known.

3.3 CJC's scheme is not uncoercible

During the decryption of a ballot, *TC* knows (v_i, s_i, *b*, β), thus, it can know which *m* links to v_i, on the other hand, v_i is bound to V_i, so voter V_i can prove to somebody how he has voted with the help of *TC*.

3.4 CJC's scheme is not verifiable

In the original paper of [4], the authors claimed that the requirement of verifiability can be realized by *SC*. In fact, *SC* cannot guarantee that each voter's ballot has been correctly counted.

4 Our modification

In this section we propose our modified scheme to CJC's scheme.

4.1 Cryptographic primitives

ElGamal public encryption To use an ElGamal cryptosystem, we need a cyclic group *G* = <*g*>, on which the discrete logarithm is difficult. Public key is $h = g^x$, and *x* is the private key. A message *m* is encrypted to (g^r, mh^r), where r is a random number. On getting a ciphertext (g^r, mh^r), the decryption is done by computing $(mh^r)(g^r)^{-x}$, *m* is thus recovered.

Re-encryption of ElGamal cryptosystem [5] A ciphertext (g^r, mh^r) which is under ElGamal scheme can be re-encrypted without influencing the decryption. The re-encryption is done like: (g^r, mh^r) is transferred to $(g^r g^z, mh^r h^z)$, where *z* is a random number. It is easy to verify that re-encryption does not influence the decryption. However, re-encryption can afford anonymity, which is needed in our proposed scheme.

Blind signature Blind signature can also afford anonymity. In our proposed scheme, we use RSA blind signature scheme. The signer's private key is *d*, and his public key is *e*, where $ed = 1 \bmod \phi(n)$, *n* is a multiplication of two big primes. The blinder want the signer to sign a document *m*. Firstly, he chooses a random number *r*, and he sends $r^e h(m)$ to the signer , where *h*(*m*) is a hash value of *m*, and then, the signer sends $(r^e h(m))^d$ to the blinder, in the end, the blinder can get the signature of *m* by computing $(r^e h(m))^d$ /*r*.

4.2 Our voting scheme

At first, we give some notations.

V_i : Voter *i*.

v_i : A pseudonym chosen by V_i .

CA: Certificate authority which is a certificate service provider for all enrolled elections.

AC: Authentication center which is responsible for certifying all voters.

PS: A trusty public proxy server allows a voter to cast ballot without leaking his own IP address which can be used to link him.

TC: Trust center which is trusty during its re-encryption the ciphertexts from the voters, outputting the re-encrypted ciphertexts in random order, and publishing the pseudonyms in some website.

SC: Supervision center constructed by *k* different politic parties which is responsible for tallying the votes.

Initialization phase

In our scheme, both *TC* and *SC* use ElGamal public cryptosystem over a cyclic group *G* = <*g*>, on which the discrete logarithm is difficult. *TC*'s public key is $h_{TC} = g^{x_{TC}}$, x_{TC} is its private key. The *k* parties have public keys $h_1 = g^{x_1}$, $h_2 = g^{x_2}$,⋯, $h_k = g^{x_k}$ respectively, and their common public key is $h = \prod_{i=1}^{k} h_i$, the private key of *h* is $x = \sum_{i=1}^{k} x_i$ shared secretly by the *k* parties [6]. *G*, *g*, the order of *G*, h_{TC} , and *h* are published. The following steps which are similar with CJC's scheme will be done.

$V_i \leftrightarrow CA$: An eligible voter V_i registers himself in *CA* and gets a "personal certificate".

$V_i \leftrightarrow AC$: V_i sends his "personal certificate" and the pseudonym v_i chosen randomly by V_i to *AC*, *AC* check the "personal certificate". V_i gets the signature s_i of *AC* from v_i by using the RSA blind signature scheme. *AC* allows each "personal certificate" to get a signature only once.

Voting phase

V_i chooses two random number r_i, y_i , and encrypt (v_i , s_i) and the marked ballot *m* as follows:

$$b = (g^{r_i} , (v_i , s_i , (g^{y_i} , m_i h^{y_i})) h_{TC}^{\ r_i})$$

V_i sends *b* to *TC* through a trusty public proxy server. On getting *b*, *TC* decrypts it to get v_i, s_i, (g^{y_i} , $m_i h^{y_i}$). *TC* will verify the signature s_i of v_i; it does not allow the "pseudonym v_i" to vote twice. *TC* then writes v_i to a website which can be written only by *TC*; this website is open to all voters, and the voters can access the website and check if his ballot has been correctly sent to *TC*. Meanwhile, *TC* will re-encrypt (g^{y_i} , $m_i h^{y_i}$) to ($g^{z_i}, m_1 h^{z_i}$), and records ($g^{z_i}, m_1 h^{z_i}$) to its protected database.

Counting phase

After the voting deadline, *TC* will send all the re-encrypted ciphertexts ($g^{z_1}, m_1 h^{z_1}$), ($g^{z_2}, m_2 h^{z_2}$),$\cdots$, ($g^{z_n}, m_n h^{z_n}$) in random order to each party of *SC*. To ciphertext (g^z , mh^z), the *k* parties compute $SC_1 = (g^z)^{-x_1}, SC_2 = (g^z)^{-x_2}, \cdots, SC_k = (g^z)^{-x_k}$ respectively. Under the cooperation of the *k* parties, the ballot *m* can be recovered by computing $mh^z(SC_1 SC_2 \cdots SC_k)$. Because each party does not know the corresponding ballot *m* to (g^z , mh^z) and if one of them is not honest, *m* cannot be correctly recovered, we can assume that $SC_1, SC_2, \cdots, SC_k$ are numbers which can be trusted. Each ballot can thus be recovered. In the end, *SC* will get all the ballots, and then, it counts the ballots and publishes the voting results.

5 Security analysis

5.1 Accuracy

Because the pseudonym and the ballot are both encrypted, *TC* is trusty, and *SC* consists of different parties, the decryption is done by all the parties, nobody can alter a vote, and a valid vote can not be eliminated. Moreover, *TC* and *SC* assure that a valid vote will be counted correctly.

5.2 Fairness

It is trivial that all ballots remain secret while the voting is not completed.

5.3 Eligibility and Uniqueness

By checking the signature of the pseudonym, only eligible voter can vote, and each eligible can vote only once.

5.4 Uncoercibility

The voter can only show the pseudonym he chose from the information on the website which is published by *TC* , and *SC* only decrypts the ciphertexts outputted by *TC*, so the voter cannot prove to somebody how he has voted.

5.5 Anonymity

By using pseudonym and proxy server, nobody can link a ballot with a voter.

5.6 Verifiability

This can be realized for a voter by accessing the website which has all the pseudonyms and checking if his pseudonym is on it.

6 Conclusions

We pointed out some security weaknesses of CJC's voting scheme. We gave a modification which satisfies the security requirements of a voting scheme. Moreover, in the voting scheme of our and CJC, the public proxy server can be replaced by a Mix-net [1] to improve the security.

References

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

[2] Fujioka, A., Okamoto, T., Ohta, K.: A practical secret voting scheme for large scale elections. Advances in Cryptology-AUCRYPT'92 Proceedings. Springer (1993) 15-19

[3] Okamoto, T.: Receipt-free electronic voting schemes for large scale elections. In Christianson, B., Crispo, B., Lomas, T.M.A., Roe, M., eds.: Security Protocols Workshop. Volume 1361 of Lecture Notes in Computer Science. Springer (1997) 25-35

[4] Chen, Y., Jan, J., Chen, C.: The design of a secure anonymous internet voting system. Computer & security 23() (2004) 330-337

[5] Park, C., Itoh, K., Kurosawa, K.: Efficient anonymous channel and all/nothing election scheme. In: EUROCRYPT. (1988) 177-182

[6] Pedersen, T.P.: A threshold cryptosystem without a trusted party (extended abstract). In: EUROCRYPT. (1991) 522-526