Devising a Private Multi-Event Multi Authority E-Voting Scheme Using Secret Sharing

Ujan Mukhopadhyay
Dept. of Computer Science National Institute of Technology Karnataka Surathkal, India

***Abstract*—When the world, as we know it, is getting more digitalized every day, the relevance of ballots and papers are getting considerably reduced. Even the whole concept of being physically present at the site is being antiquated. Voting for a motion, or election of a candidate, has entered the E-Domain. And hence has raised the questions of privacy of the voters. In this paper, we discuss a protocol where the voters can choose more than one event, and the calculation would reveal the total number of votes for each party/event, and the identity of the voters will be private. No one would be able to relate a voter with his vote. And we do so utilizing the concept of secret sharing. Here a vote cast by a voter is mapped into a binary string. I t is then broken into shares by the voter himself, all of which are required to determine the vote with certainty. We thus offer the voter with the full flexibility of choosing as many numbers of events as he wishes, while keeping his identity secret.**

***Index Terms*—E-Voting, Secret Sharing.**

I. Introduction

E-voting schemes that have been initiated primarily focus on single event voting protocols. These systems have only one Boolean event. The voter has to either vote yes or no. Maintaining voter privacy in this case, though not trivial, is comparatively simple, and has been discussed by Sorin Iftene1. In this paper, we introduce a multi event system. Each of the events is Boolean, and one or more of them can be chosen by the voter. Subsequently, variations have been brought into the privacy protection part as well. Non-threshold secret sharing has been used for the purpose. Secret Sharing was a concept brought to light by Adi Shamir2. He devised a technique that disintegrates a secret into n components in way such that up to t components, when compromised would not reveal the original secret. One needs t + 1 components to successfully and accurately deciphers the secret. This constant t is called the threshold, and the scheme is known as threshold secret sharing scheme. But what we utilize here is non threshold in nature. The adversary has to obtain ALL the components in order to know the secret, in this case, the vote. Secret Sharing has been implemented in various ways for different purposes. Chinese Remainder Theorem, Polynomial Co-efficient are the major tools used in implementing Secret Sharing. But our system does not require such complex mathematics. Simple arithmetic can well ensure the privacy of a voter.

II. Our proposal

In our scheme, there would be three set of players; The Voters Xi , The Tallying Authorities Tj , and The Central Authority C. There would be a set of events Ei. The protocol is as follows:

i. There would be a predetermined number n, fixed by C. There would be n tallying authorities and n shares of a vote.

ii. Every Ei would be set to 0 by default.

iii. The voter Xi would form his vote Vi by flipping the value of the Ei he wants to choose. For example, if there are 4 events, and X wants to choose events 1 and 4, then his vote would be 1001.

iv. After the vote string is generated, Xi would make n shares of the vote Vi such that jVi,j = Vi , where j = 1 to n. Vi,j are the shares of the secret vote Vi.

v. The share Vi,j is sent to the tallying authority Tj. Thus Tj has the jth share of every vote. Ta doesnt know any values of Tb.

vi. After all voters complete the voting and share submission, each Tj calculates the sum i Vi,j. thus Tj gets the total sum of the jth share of each voter.

vii. The Tj s then submit their own totals to the central authority C.

viii. The C calculates the sum of the individual totals to find the gross total, and declares it.

Here is an example.

Voter	Event 1	Event 2	Event 3	String
V1	1	0	1	101
V2	0	1	1	011
V3	0	0	1	001

Hence, event 1 gets one vote, event 2 gets one vote, and event 3 gets 3 votes, as indicated by the string 113.

Voter	Share 1	Share 2	Share 3
V1	47	9	45
V2	3	17	-9
V3	22	-8	13

The central authority calculates 72 + 18 + 23 = 113.

When the number of voters becomes large, we use larger indices, i.e. up to 9 voters we can use decimal representation of the vote string as a share sum. When it exceeds, hexadecimal or higher indices can be used, depending on the number of voters.

III. Correctness Proof

Correctness proof is trivial. We are essentially calculating the sum of a number of integers, arranged in a matrix. So instead performing a row-major addition, we are doing a column-major addition. Ultimately they are equal.

IV. Proof of Security

There are some major issues to be addressed in this section. It includes the maintenance of the privacy of the voters, as well as the possibilities of security breach.

For the first issue, we need to prove that the voter cannot be connected to the vote string he has cast. Here we see that each share of the secret goes to a distinct tallying authority, and no tallying authority knows the content of others. But to determine the vote correctly, one needs to get hold of all the shares, which is logically hard. A security breach, however can happen, if all of the tallying authorities decide to collaborate and share their contents. But this situation is also highly unlikely, and the loyalty of just one authority can render the attack unsuccessful.

V. Future Work

There is, however, one other issue to be addressed. Although the privacy of the voter is absolutely secured, the integrity of the vote is not. If one dishonest tallying authority decides to tamper with a share, the actual result will vary. So we are planning to introduce a integrity check option like hashing or CRC, which will detect and prevent data manipulation.

VI. References

[1] Secret Sharing Schemes with Applications in Security Protocols (Ph.D. Thesis), Sorin Iftene, Universitatea Alexandru Ioan Cuza Iasi Facultatea de Informatic.

[2] A. Shamir, How to share a secret. Communications of the ACM, 22(11):612613.

[3] C. A. Asmuth and J. Bloom. A modular approach to key safeguarding. IEEE Transactions on Information Theory, IT-29(2):208210.

[4] A. Beimel, T. Tassa, and E. Weinreb. Characterizing ideal weighted threshold secret sharing. In J. Kilian, editor, Theory of Cryptography, Second Theory of Cryptography Conference, TCC 2005, Cambridge, MA, USA, February 10-12, 2005, volume 3378 of Lecture Notes in Computer Science, pages 600619. Springer-Verlag, 2005.

[5] J. Benaloh. Verifiable Secret-Ballot Elections. PhD thesis, Department of Computer Science, Yale University, September 1987.

[6] C. Boyd. Digital multisignatures. In H. Beker and F. Piper, editors, Cryptography and Coding, 1986, pages 241246. Oxford University Press.