THE JOURNAL OF DIGITAL FORENSICS, SECURITY AND LAW

Journal of Digital Forensics, Security and Law

Volume 11 | Number 2 Article 8

2016

Electronic Voting Service Using Block-Chain

Kibin Lee
Korea University

Joshua I. James
Hallym University, joshua@cybercrimetech.com

Tekachew G. Ejeta
Korea University

Hyoung J. Kim
Korea University

Recommended Citation
Lee, Kibin; James, Joshua I.; Ejeta, Tekachew G.; and Kim, Hyoung J. (2016) "Electronic Voting Service Using Block-Chain," *Journal of Digital Forensics, Security and Law*: Vol. 11 : No. 2 , Article 8.
DOI: https://doi.org/10.15394/jdfsl.2016.1383
Available at: http://commons.erau.edu/jdfsl/vol11/iss2/8

ELECTRONIC VOTING SERVICE USING BLOCK-CHAIN

Kibin Lee
Korea University
Graduate School of Information Security
Seoul, Seongbuk 02841
leekibin@korea.ac.kr

Joshua I. James
Hallym University
Legal Informatics and Forensic Science Institute
Gangwon, Chuncheon 24252
joshua.i.james@hallym.ac.kr

Tekachew Gobena Ejeta
Korea University
Department of Cyber Defense
Seoul, Seongbuk 20841
tekachew@korea.ac.kr

Hyoung Joong Kim
Korea University
Graduate School of Information Security
Seoul, Seongbuk 02841
khj-@korea.ac.kr

ABSTRACT

Cryptocurrency, and its underlying technologies, has been gaining popularity for transaction management beyond financial transactions. Transaction information is maintained in the block-chain, which can be used to audit the integrity of the transaction. The focus on this paper is the potential availability of block-chain technology of other transactional uses. Block-chain is one of the most stable open ledgers that preserves transaction information, and is difficult to forge. Since the information stored in block-chain is not related to personally identifiable information, it has the characteristics of anonymity. Also, the block-chain allows for transparent transaction verification since all information in the block-chain is open to the public. These characteristics are the same as the requirements for a voting system. That is, strong robustness, anonymity, and transparency. In this paper, we propose an electronic voting system as an application of block-chain, and describe block-chain based voting at a national level through examples.

Keywords: Electronic Voting System, Electronic Ballot, Block-chain, Ballot Authentication, e-voting Auditing:

1. INTRODUCTION

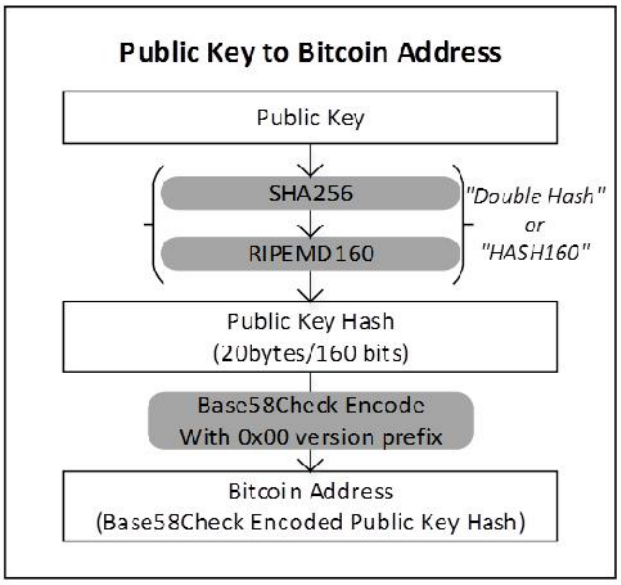

Figure 1. Method of generating a Bitcoin hash from a public key. [11]

Electronic voting systems have been of growing interest to many governments for the last several years [1]. This interest, however, has been followed closely by warnings of security issues [2-5]. While some methods for creating transparent voting system protocols have been proposed [6,7], these methods are both costly and have not been implemented on a large scale. Various pilot programs have been run [1, 8], though electronic voting systems have been fraught with security concerns and controversy [9]. Despite these concerns, electronic and remote voting continues to be developed. As more of a population uses the Internet regularly, electronic and remote voting becomes an incentive for greater participation in democracy [8]. In this work we discuss criteria of electronic voting, and how block-chain may be used as a transparent, cost-effective method to manage and verify transactions in large-scale voting.

2. BACKGROUND

Bitcoin, and specifically the block-chain, can be used to monitor and verify transactions. This section gives background into the underlying technologies that will be used in the proposed electronic voting system.

2.1 Public and Private Key and Bitcoin Address

Bitcoin uses public and private keys for addressing and transaction signing. A Bitcoin private key is a random 256 bits. Users use this key to sign their transactions every time they transfer Bitcoin. The private key is randomly generated by users. Since the key has 2^{256} bits of sample space, it is very unlikely to intersect with other private keys. The public key is derived from the private key through an elliptic curve crypto-algorithm, specifically secp256k1 [10]. The public key is an (x,y) pair resulting from the secp256k1 equation multiplied by the generator (G). This generator is fixed in Bitcoin systems. This means that public key uniqueness is not guaranteed by the generator (G), but is guaranteed by the uniqueness of the private key. A public key hash is produced using SHA256 and RIPEMD160 hashing algorithms

(Figure 1). The fingerprint of a public key, called the public key hash, has the size of 160 bits. The public key is Base58Check encoded to generate the Bitcoin address. Since this address is generated from a private key that contains no secret information, addresses can be known to the public.

2.2 The Block-chain

The block-chain is composed of time stamps which show at what time data (a block) was added. A block that contains transactions occurring at a certain time is similar to a time-stamped binary file. The hash value of the previous block and the current block will be the input of the hash value of the next block. Each hash value of a block is calculated from the hash value of the previous block, and transactions are recorded in the block. Since the hash of the previous block is used to produce the hash value of the next block, the next block is "chained" with its prior block, reinforcing the integrity of all the previous blocks that came before. Each anterior block contains information about the hash of prior blocks, as shown in Figure 2. Since the time stamping process forms a block-chain linked with the hash values of each block, each stamped block can be verified to be a valid transaction at a specific time. Time-stamped blocks are created, and blocks are linked forward to the next block. If any data in a block is modified, the hash value of the block will be changed. The result is that the hash of all blocks to the most recent will be changed. This forged chain will not be accepted as a consistent block chain and will be rejected.

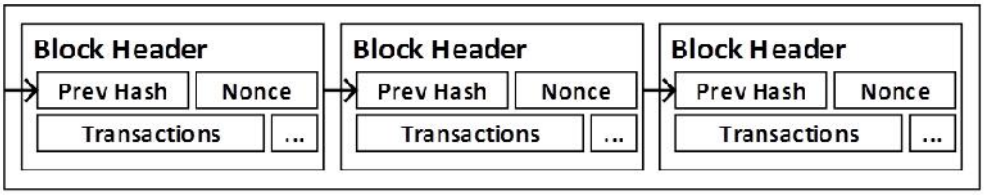

Figure 2. Structure of Block-chain blocks with hashing, nonce and transaction information linked to anterior blocks

2.3 Block-chain Safety

In a Bitcoin system, to make it difficult to forge context in blocks, a random number called a nonce is introduced to every block. A nonce is an arbitrary number used only once to help verify the hash. In order to produce a fingerprint - i.e., a hash of the block - miners use the header of the block which is a predetermined set of data. This set of data represents all transactions contained in the block, the date, time and some other data which can be fixed whenever a certain period time has passed. Miners do this to try to validate their proof of work. These header components and nonce will be put into a hash function to produce a block hash.

To add to the calculation difficulty, there is a condition that the block hash should be smaller than some given value. This means that the block hash should start with a certain number of zeros (based on difficulty). When we take a specific nonce found by a miner and the current block header, these two values should produce the fingerprint for the block hash. Fingerprints are 64 hexadecimal digits. Assume that the first 15 digits of a hash should be zeros, so 15 times 4 bits (i.e., 60 bits) at the beginning of the hash should be zero. The probability that corresponding 60 bits are zero is very low, about 2^{-60}. The current Bitcoin network requires 17 zeros at the beginning, so 68 bits must be zeros.

It requires massive compute power to operate until the nonce that produced the hash value satisfies the condition based on difficulty. We can estimate how many hash operations are needed before the correct hash is found. The Bitcoin network has a hash rate of around 1200 quadrillion (1,200P) hashes/s at this time [12], and it still takes 10 minutes on average to find the nonce. So 1200 Phash times 10 mins on average is how many hash operations the miner needs. There is no easier way to find this hash value because there is no (known) back door in the hash function. The only way to find the right nonce is by performing many hash operations. Since finding a specific nonce at every block is very difficult, attackers who try to forge the block-chain ledger need to find the corresponding nonce to the changed transactions.

Assume that an attacker tries to forge the context, such as the transaction, stored in a block located a few blocks away from the top of the block-chain. The change of a single letter of context will cause a change of the entire fingerprint of the block. The change of the block hash of a previous block will change the whole context of all blocks stacked upon this block. This means that an attacker must find the nonce to every block faster than the current hashing speed of the whole network, so the forged block-chain can be validated as the genuine block-chain. For the Bitcoin network, this work would require more than several times the hash power currently contributed. For personal-level processing power, forgery is impossible. Any transactions included in blocks are safe from being forged relative to the amount of hashing power being contributed to the network. In this case, integrity is well preserved.

3. PROPOSED METHOD

Neumann [2] proposed electronic voting criteria, that include:

- system integrity
- data integrity and reliability
- voter anonymity and data confidentiality
- operator authentication

As shown, the generation of addresses does not rely on personally identifiable information (PII), but allows transparent tracking of transactions. These transactions are verifiable, open to the public and are difficult to forge. Block-chains, then, can guarantee data integrity and reliability, voter anonymity, data confidentiality and – at least for the block-chain – system integrity. Operator authentication, however, is still required.

User authentication is necessary to ensure that the person voting has a right to vote. Once authenticated, a vote from one user must be tracked to one candidate. In this section we give a block-chain based voting system with government-based authentication systems.

3.1 Organization, Trusted third party, Voters, and Block-chain

There are four parts that are involved in this electronic voting model. An authentication organization refers to any institution that holds a voter registration list such as the National Election Committee or private companies. Electronic voting systems may be used for presidential elections, stockholders meetings, and so on. In a presidential election, only the National Election Committee will have the list of voters in their nation. Both Bitcoin and the proposed voting system are open to anyone to make any transactions, but the voting system restricts voters to only those who have right to vote in their own organization. As stated, this means authentication for a user is needed. There are three problems with this organization. The first is that an authentication organization should authenticate the voters, but should not be able to find who the voter voted for. The

second is that since the authentication organization has the voters list, they can potentially manipulate the number of voters in their nation. The third potential problem is that an authentication organization could potentially provide the majority of nonce mining. If so, they can potentially forge the block-chain ledger in the way that they want.

For this reason, a trusted third party (TTP) is introduced to authenticate voters, similar to a proxy. The TTP checks if the voter is authenticated by reporting declaration of vote to the authentication organization. A message hash is used for voter authentication without exposing identity information. A voter sends his or her own secret message hash to the authentication organization. Then the authentication organization, once it authenticates a registered voter, will link the message hash to each voter's identification when he or she is verified.

The authentication organization decides whether a voter has the right to vote or not, according to the voters list, through identification information such as Social Security Number (SSN). This ID information should not be exposed to the public, especially to the Trusted Third Party; otherwise it could be used to identify a voter. This means that when an ID is exposed, then the TTP would know exactly who voted for whom. Therefore, a secret message hash is needed to identify and authenticate valid voters between the organization and the TTP without directly providing identifying information.

In our proposed model, we keep vote transactions in the block-chain. There are many ways to manage a block-chain, and we introduce two ways that are useful for voting purposes.

1. Operating independent block-chain funded by the organization. Block-chain receives all transactions cast by anyone. Our block-chain receives all transactions cast and stamps them all no matter if it is authenticated by the organization and Trusted Third Party. It receives everything, and filters out unauthenticated or invalidated transactions. When voting is finished, the list of validated voters that is kept from the Trusted Third Party is used to filter out the transactions that are not validated. All transaction fees (block-chain processing charges) will be paid by the organization. A problem may occur when there are not enough miners providing hash power, in which case the independent block-chain will not be secure.

2. *Using current Bitcoin block-chain.* When a block-chain network is used, the organization does not need worry about having its own miners. There are multiple advantages to using the Bitcoin block-chain for voter transaction processing

1. A company or government does not need to operate an independent block-chain.
2. There is less risk for transactions to be forged.
3. Block-chain mining can incur a cost, but voters could receive tax benefits for voting, thus covering the costs of transaction fees while stimulating participation.

3.2 Declaration of a vote

In the voting system, there are individuals who can and cannot vote, so voters must be authenticated by an organization. We introduce declaration to solve this step. A voter declares a vote by sending a secret message hash to the authenticating organization. We assume that the authenticating organization has already registered a voter and provides a login for their account for authentication. The voter then registers their secret message hash to the organization. This hash should be unique to

each voter because this factor is going to be used as an authentication of votes in the block-chain.

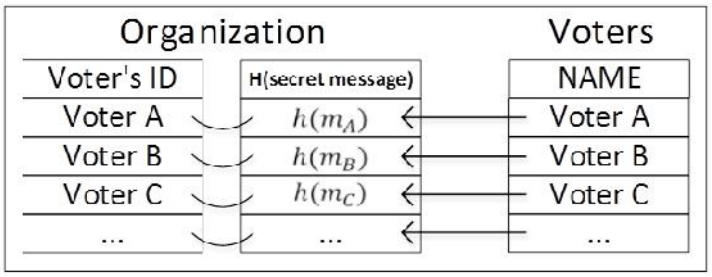

Figure 3. An authenticating organization matching a voter's secret message with the voter's ID.

When the message hash is sent to the organization holding the list of voters, if he or she is confirmed to have a vote, then they link the message hash with the voter's ID as shown in Figure 3. There are several IDs that can be used when voters login with their account. The reason voters cannot just register the address derived from their own private key is that the address will be written on every transaction that will be open to the public later in the block-chain. When this address is registered to the organization, then they can identify a vote as blocks being stamped. The information generated from a voter's private key, such as a public key, public key hash, or address, should not be registered to the organization, otherwise they will know who votes for whom. Since all transactions are stored in an open block-chain, when you give the organization your public key as an ID, they can know which user voted for whom. That is why the secret message hash is expected to be unique, which is also independent to the public key used as an ID.

3.3 Private Key and Votes

The right to vote is established when the secret message hash is declared to the organization. After, the right to vote is derived only from a private key and the message. Voting ownership does not belong to the public key, address, or digital signature. It only belongs to the private key. The only thing that voters should keep safe is the private key of the voter's account, and the message used during the declaration of their vote, when the message hash is made. Assuming that a voter secures the private key well, only the voter can access and have control of the right to vote linked to the address.

3.4 Casting a vote

The number of votes is defined as the number of transactions made to a candidate's address. Candidates will provide their addresses fixed and open to the public to receive transactions from voters. A person who runs an election will simply generate their private key, and open up their address which can be considered as a container of votes. Then voters make a transaction to the address of candidates.

3.5 Confirming votes

We propose a model to authenticate voters who have the right to vote and to assure secret voting. The layers between voters and the authenticating organization are composed of two parts. One is the trusted third party, and the other is block-chain. Overall process is shown in Figure 4.

When people who want to vote finish the declaration to vote - when their secret message hash has been linked with their ID (such as an SSN) that only the organization has - they need to make contact with the trusted third

party. Voters give the trusted third party their secret message hash and the trusted third party will ask the organization if they receive the same secret message hash from a voter. If the organization replies 'yes,' it means that a person who sent the hash value is registered as a valid voter who finished the declaration to vote. Then the trusted third party recognizes this person as a proper voter. The trusted third party saves the voter's public key hash once they confirm they are a valid voter through communication with the organization. Eventually, the trusted third party will have the list of confirmed public key hashes and the addresses which are confirmed to be registered in the organization. By using this registered address, transactions which are made by an invalid voter will not be counted but will be removed when voting is done. The voting protocol is shown in Figure 5.

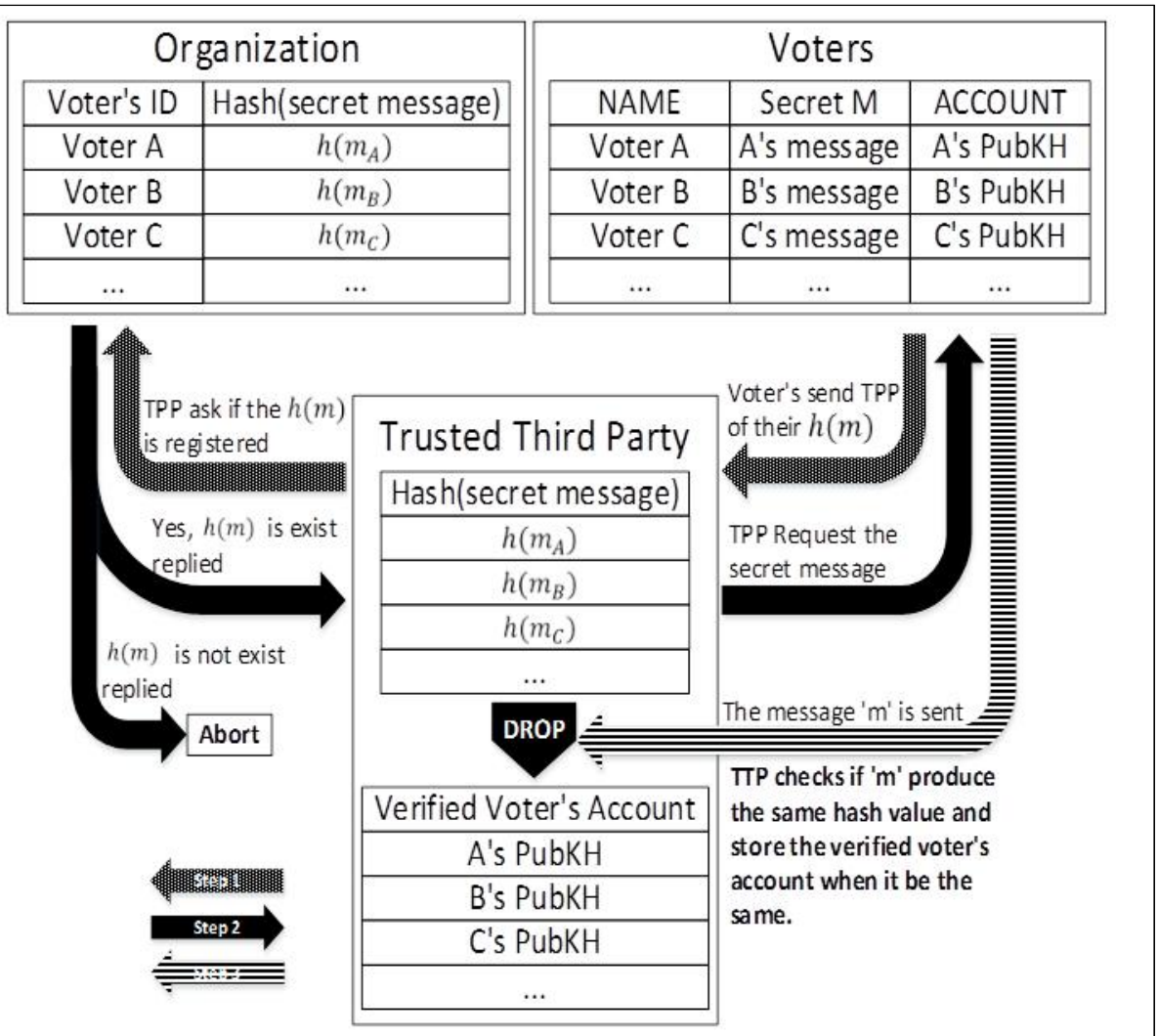

Figure 4. Process for authenticating organization validate a user and register a secret message later used to verify authentic voters

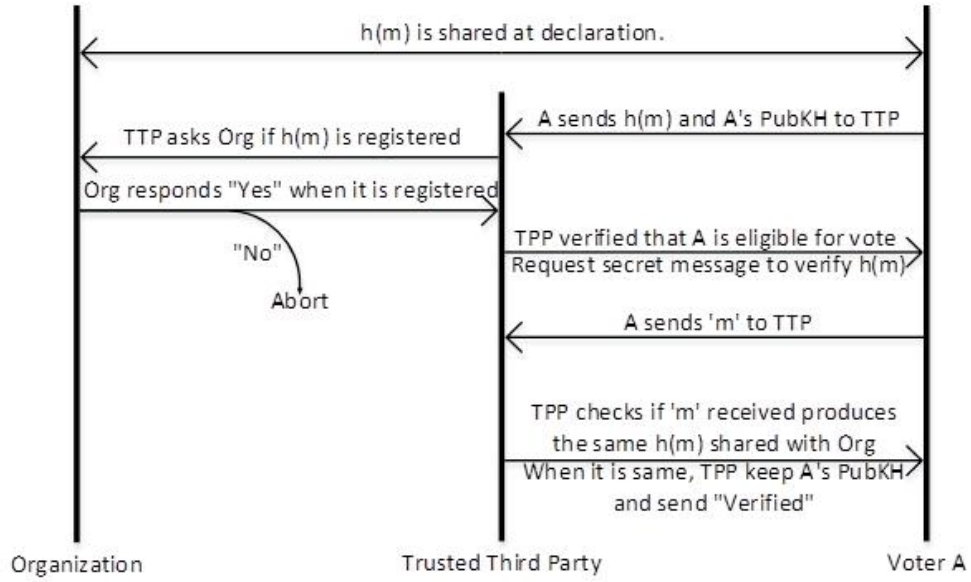

Figure 5. Voting protocol involving an authenticating organization, trusted third party, and voter

3.6 Overall Voting Example

Assume that Alice votes for candidate Bob. Alice generates a private key/public key pair and address for the public key hash based on the Bitcoin protocol. Alice should declare votes to the authenticating organization by registering a hash of her secret message. Alice needs to get authenticated through the trusted third party. Alice sends the trusted third party the same message hash used during registration. The trusted third party then sends the authenticating organization the same secret hash received from Alice, asking if the authenticating organization has the same hash value. The trusted third party never asks whose secret message they received; they only ask if this secret message is held by a verified voter listed in their voter roster. This step verifies that the voter is registered at the authentication organization and can claim the right to vote. When Alice is a human that belongs to the group, who appears to have a vote, and whose secret message hash is registered, then the organization will send yes-reply to the trusted third party. This means that the hash that the organization has is consistent with the one Alice sent to the trusted third party. Also, the organization never sends their real personal identifier so the trusted third party cannot know who made the voting transaction.

Assume Alice has a vote, and Charlie does not, because Charlie is not involved in the group having votes or not a human. At the beginning of the registering step, Charlie may not have logged in to the organization website and his name wouldn't have appeared on the list of voters held by the organization. Charlie may have failed to declare a vote.

The people who are not eligible to vote, who did not declare a vote, and cannot register the hash of their secret message, will never get a yes-reply from the authenticating organization to the trusted third party. Since the authenticating organization does not know of the addresses of voters, they just know the secret message hash that corresponds to each voter and there is no way for them to know who voted for whom, even when transactions are all open in Block-chain. They can only

reply yes or no depending on a whether a voter has declared a vote or not.

Assume that Alice is confirmed to have her vote declared so the trusted third party has received a yes-reply from authenticating organization. Then the trusted third party will ask Alice to send the secret message which generates the same hash registered at the authenticating organization, and send it to the trusted third party. The trusted third party then checks if the message is right, and saves the public key hash which is the address of Alice. This confirmed address will be used when filtering out unconfirmed transactions that are made without declaration and without conformation from the trusted third party. Voters could make several transactions to a candidate and all transactions will be stamped in the Bitcoin block-chain, but transaction cast from the same address will only be counted one time. A transaction itself is a vote.

3.7 Filtering Votes

A vote is defined by the number of transactions cast. In the counting off step, we calculate the number of votes for each candidate. Assuming that a voter can vote only one time, then the number of votes that a candidate received will be the same to the number of transaction made to the candidate. If there is a chance that voters can make several transactions, these transactions will only be counted one time. This is the reason that the trusted third party should keep the ledger of addresses that are confirmed during the authenticating step between the organization and the trusted third party itself. Several transactions cast from the same address to candidate A will be counted one time. Likewise, when several transactions are made from the same address to more than one address, for example, to candidate A and candidate B, this transaction will be invalidated, and will not be counted as a valid vote.

3.8 Hash Power and Mining Incentives

When the organization operates their own hash power, for a common election, the ballot may be open for 12 hours. If 6 blocks an hour are assumed, only 72 blocks need be stamped. If an incentive for mining one block is 100 thousand dollars, it will take 7.2 million dollars for all votes to be calculated and verified. That amount of incentive is likely to bring more than enough contributors to provide processing power to keep the block-chain from being forged. When the organization fails to bring enough miners, this could cause some problems. We discuss this at section 4.

When using the current Bitcoin Block-chain, we do not worry about mining and incentives, because it already provides enough hash power so as not to get disrupted by dishonest miners. The organization will give voters tax benefits as reimbursement of transactions fee taken from each transactions. Fortunately, since the organization has the voters' roster, they can tell who is eligible to receive reimbursement. The Bitcoin collected in the candidates account will be sent back to the Bitcoin address where it comes from.

4. AUDITING

We cannot follow the property of secret voting without a Trusted Third Party. If we give the authenticating organization the filtering task to do, they are able to track down who voted for whom since they also have the secret message hash to each ID. Because the TTP keeps the address and secret message hash for each verified voter based on the Yes-reply from the authenticating organization, then they are able to track down every vote including who voted for whom. This is the reason we implement a TTP and it is still dependent on

the authenticating organization's response, because we should not trust the organization that owns the list of voters. In fact, the TTP cannot determine whether a voter is valid without 'Yes' and 'No' responses from the organization. This means that the authority to validate a voter depends on the authentication organization. There is a possibility for the organization to always respond with 'Yes' or 'No' to manipulate the number of validated voters. To prevent those kinds of issues, both the authenticating organization and the Trusted Third Party need to publish a roster that consists of verified voter's account and hash value of the secret message. That is, cross auditing should be available and reasonable.

In the first step, the Trusted Third Party performs random permutation to the verified voter's account that is kept in order to filter out improper transactions, and publish the hash value of a secret message given by a voter. This random permutation process will remove the relation between h(m) and PubKH as shown in Figure 6. It delinks the ID, in this case h(m), from PubKH that each voter casts for. This means that no one knows who voted for which candidate. The permutation does not change the voting result, but only changes the order of the Verified Voter's Accounts column in Figure 6.

In the second step, the authenticating organization needs to simply publish both the voter's ID and the hash value of a secret message that the voter registered at the step of the declaration of vote. In this step, the voter's ID will be known to the public. It means that the public is able to know who voted or not, but it does not violate the property of secret voting since it is unknown who they voted for. Through these two published lists, voters can check their hash of the secret message by themselves after voting. When there is a voter whose secret message hash does not match both lists or is not listed at all, then we know that the organization manipulated the number of voters or the system of the Trusted Third Party has some fault. The hash value of a secret message published from the TTP must uniquely match one of the hash values owned by the authenticating organization. The number of hash values that are shared between the authenticating organization and the TTP must also be the same, and the hash values themselves should all be the same. The only thing that TTP must do is to perform permutation of the verified voter's account before they publish the list they keep, because the Public Key Hash is a main component of a transaction in a block-chain which is concurrently updated to the public. With these two published rosters, voters can verify that their votes were counted, and anyone can check whether any fraud has happened.

Case 1. When the organization issues more votes than the number of persons who declare their vote by using the IDs of those who did not. This forgery will appear on the roster published from the organization itself. In the pair of voter's ID and secret message hash they publish, when a person who did not declare votes, the secret message hash column should be empty. If the ID holder insists that he or she did not vote, and still the secret message hash is filled, then the vote is forged (or the voter is lying).

Case 2. When other groups of people who do not have the right to vote attempt voting transactions, transactions will still be received and verified to get stamped by the Block-Chain protocol. Transactions are also left in the open ledger, but since these groups of people do not have the right to vote, they must have failed at the declaration step. This means that their address can never be registered at the roster that the TTP holds so transactions made by them will be filtered out.

Case 3. When a person who is not relevant to the voting makes a transaction, this will be

considered as a valid transaction. But this person will not be verified as a legitimate voter from the authenticating organization nor by the filtering roster that the TTP holds.

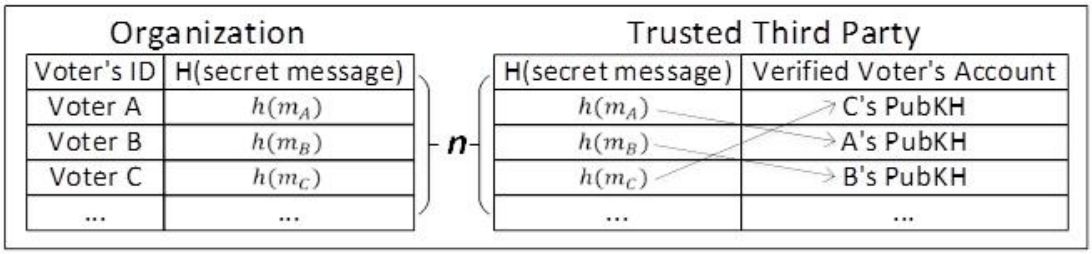

Figure 6. Rosters made public by the authentication organization and Trusted Third Party after voting is complete.

5. CONCENSUS ATTACK

Theoretically the block-chain consensus mechanism is vulnerable when sufficient hashing power is given to a dishonest miner. This miner can attack the consensus mechanism so as to disrupt the availability of the Bitcoin network. But still a consensus attack can only affect the blocks coming after the current block. The Block chain gets more robust as time passes. Practically, massive hash power is required to forge blocks at even a shallow depth because attackers have to solve multiple nonces while other honest miners solve only one current nonce. Additionally, consensus attacks do not affect the security of private key and signature algorithms. Consensus attacks cannot steal coins, consume it, or change past transactions and ownership records. These attributes are very important to voting systems, since mining attacks cannot (with enough hashing power) change the votes that have been cast. It can affect only the denial-of-service disruptions on the creation of future blocks. Since the Bitcoin network has a large amount of hash power, it would be very difficult to be disrupted by a few dishonest miners.

6. CONCLUSION

In this work, we introduced an electronic voting system that uses the Block-chain as a ledger of transactions, where authenticating and filtering are done by the authenticating organization and a trusted third party. The Bitcoin protocol has yet to have failed, and the block-chain open ledger has never been forged since it appeared in 2009. Further, the transparency of the block-chain enables more auditing and understanding of elections. These attributes are some of the requirements of a voting system. These characteristics come from a decentralized network, and can bring more democratic processes to elections, especially to direct election systems.

The proposed protocol changes the paradigm that we trust a single organization such as a government or a company. In current election systems, voters must trust the vote records provided by the voting organization and it is difficult, if not impossible, for a single voter to prove that there is no fraud. On the other hand, in the proposed method, the organization's only job is to send a reply based on the electoral roll they have, which is an immensely restricted job scope than before. With the proposed system, voters have to identify their right to vote by proving themselves to both authenticating organization and the TTP. Then, by publishing both sides of the roster, voters know that the given vote is uniquely validated and auditable.

ACKNOWLEDGEMENTS

This work was supported by the National Research Foundation of Korea (NRF-2015R1A2A2A01004587). This research was supported by Korea University.

REFERENCES

Madise, Ü. & Martens, T. (2006). E-voting in Estonia 2005. The first practice of country-wide binding Internet voting in the world. Electronic Voting, 86.

Neumann, P. G. (1993). Security Criteria for Electronic Voting. Baltimore, Maryland.

Rubin, A. D. (2002). Security Considerations for Remote Electronic Voting. Commun. ACM, 45(12), 39–44.

Kohno, T., Stubblefield, A., Rubin, A. D., & Wallach, D. S. (2004). Analysis of an electronic voting system (pp. 27–40).

Gritzalis, D. A. (2012). Secure Electronic Voting. Springer Science & Business Media.

Schoenmakers, B. (1999). A Simple Publicly Verifiable Secret Sharing Scheme and Its Application to Electronic Voting. In M. Wiener (Ed.), (pp. 148–164). Springer Berlin Heidelberg.

Kremer, S., Ryan, M., & Smyth, B. (2010). Election Verifiability in Electronic Voting Protocols. In D. Gritzalis, B. Preneel, & M. Theoharidou (Eds.), (pp. 389–404). Springer Berlin Heidelberg.

Braun, N., & Brändli, D. (2006). Swiss e-voting pilot projects: Evaluation, situation analysis and how to proceed.

Rubin, A. D. (2006). Brave New Ballot: The Battle to Safeguard Democracy in the Age of Electronic Voting. Crown/Archetype.

Secp256k1 - Bitcoin Wiki. (2016). Retrieved April 24, 2016, from https://en.bitcoin.it/wiki/Secp256k1

Andreas M. Antonopoulos, (2014), Mastering Bitcoin. O'REILLY (pp. 72)

Blockchain. (2011). Hashrate. Retrieved from https://blockchain.info/charts/hash-rate

