QUES-CHAIN: AN ETHEREUM BASED E-VOTING SYSTEM

Qixuan Zhang[1], Bowen Xu[1], Haotian Jing[1], Sicheng Zhang[2] and Zeyu Zheng[1]

[1]School of Information Science and Technology, Shanghai Tech University, China
[2]School of Information and Electronic Engineering, Zhejiang Gongshang University

ABSTRACT

Ethereum is an open-source, public, block chain-based distributed computing platform and operating system featuring smart contract functionality. In this paper, we proposed an Ethereum based electronic voting (e-voting) protocol, Ques-Chain, which can ensure the authentication can be done without hurting confidentiality and the anonymity can be protected without problems of scams at the same time. Furthermore, the authors considered the wider usages Ques-Chain can be applied on, pointing out that it is able to process all kinds of messages and can be used in all fields with similar needs.

KEYWORDS

Electronic voting, Ethereum, Smart contracts, Blind signature

1. INTRODUCTION

E-voting and web-based surveys are popular ways to collect opinions from citizens, clients, employees and sometimes organizations or companies. The conductors of elections, evaluations or questionnaires expect voters to give real and practical comments on their particular topics. However, e-voting systems face the threat of malicious manipulation by hackers. Questionnaire and poll service providers also struggle to prevent and eliminate scams, given the high costs to perform data cleaning. On the other hand, voters may doubt the integrity of the voting procedure and worry about anonymity failure. Thus, an e-voting system ought to be capable of conducting authentication, providing transparency, protecting anonymity, securing ballots, and yielding accurate statistics. Block chain, initially introduced as a distributed ledger and nowadays a computation vender, has proved its inherence in providing immutability, verifiability and decentralized consensus throughout its very first application, bitcoin the crypto currency. Ethereum (ETH), a block chain, and Smart Contract, a way to assign computing missions to the Ethereum network, have become the de facto standard of block chain-based trusted computing. Provided the outstanding features, it is promising to base future e-voting systems on block chain technology.

Natarajan Meghanathan (Eds) : CCSIT, NCWMC, DaKM - 2019
pp. 21-30, 2019. © CS & IT-CSCP 2019 DOI: 10.5121/csit.2019.90803

The paper is structured as follows: section 2 introduces existing works on e-voting and block chain, section 3 sums up our main contributions, section 4 presents two techniques which our work based on. Ques-Chain's mathematical model and scheme detail are provided respectively in section 5 and section 6. Ques-Chain's security properties are analyzed in section 7. Section 8 introduces some application scenarios of Ques-Chain.

2. RELATED WORK

There has been extensive research on adopting block chain into e-voting and surveys. Reference [1]introduced a multi-agent system, in which several kinds of intelligent agents act as block chain computing nodes, to ensure voters' right to audit and verify the voting process. Liu and Wang designed a feasible and flexible e-voting scheme with no dependency on time-triggered protocol (TTP), trusted third party, on the block chain and fulfilled criteria on general voting systems [3]. Liu and Wang also provided certain remedies for data neutrality deficiency and privacy risk in data transmission. Panja and Roy applied block chain technology to the existing DRE-ip e-voting system, which protects verified ballots from being modified before the tallying phase and provide a substitute to secure bulletin board [4]. Permissioned sidechains [5] can be adopted in e-voting systems for voter verification and voting operation recording, each computed and stored on its respective one-way pegged side chain but linked by the upper layer network. Hjalmarsson et al. conducted votes on private chains with the utilized smart contract to improve the processing speed and throughput, as well as to ease the load and save computational spends[6]. Block chain-based computing networks, by exploiting the consensus of nodes, are able to reject fraudulent ballots and guarantee the voting result un forgeable and transparent [7].

3. MAIN CONTRIBUTIONS

The main contributions of this paper are summed up as follows:

1. We have proposed a message authentication and transmission mechanism that allows permission checking while preserving anonymity. The mechanism can be utilized in various scenarios including vote, questionnaire, outcome assessment, opinion collection, complaint reporting and so on.
2. We have decoupled the blind signing and checking procedure into three steps, respectively processed by the organizer, the voters, and the Ques-Chain smart contract. With such a design, authentications can be carried out without sacrificing voters' anonymity and messages' confidentiality. In applications like opinion collection, it can also prevent spamming.
3. We have implemented the complete mechanism, featuring trusted computing technology based on Ethereum. Our well-designed architecture guarantees the integrity of all parties.

4. MAIN TECHNIQUES

In this session, we will introduce main techniques we used in our protocol.

4.1. BLIND SIGNATURE

Blind signature was firstly introduced by Chaum [8]. Same as digital signature, blind signature is used for validating the authenticity of a message. The difference between both methods of signature is that the messages are blinded and encrypted, which means that the data being signed varies from the original message. And the method assures that the one who receive the message can get the original message by un blinding and decrypting it. The process is same as applying a legacy digital signature to the original message. Therefore, the authenticity of the message can be verified by validating it with the public key of the signature. Blind signature can be used in e-voting system to perform better privacy protection for voters [9].

4.2. ETHEREUM

Ethereum acts as a general decentralized computing platform based on block chain for economic benefit and new kinds of calculating applications. It offers a decentralized Turing complete virtual machine called Ethereum virtual machine (EVM), where scripts for the platform can be run. The scripts are called smart contracts, which is mostly written in Solidity, a script language designed for EVM. After deployment, the scripts will automatically execute in decentralized network transparently. Transaction and deployment in Ethereum network require gas, a fraction of Ethereum token, which forms the justice and fairness of the block chain [9].

5. NOTATIONS

In this session, we will introduce notations which were used in our paper. All participants involved in the vote can be divided into three types - *voter*, *organizer* and *Ques-Chain Contract*.

- *Voter*, one who has the permission to vote.
- *Organizer*, one who initiated the vote, each e-voting only has one organizer.
- *Ques-Chain Contract*, a Smart Contract on EVM, which act as an inspector, contains
  - Public key, unchangeable key provided by the organizer to check the signature.
  - Judge function, a function to judge if the ballot is valid or not.
  - Ballot box a decentralized database to store valid ballots.

Let Voters be the set of all the voter,

$$\forall \text{voter} \in \text{Voters}, |\text{Voters}| \geq 1.$$

Let Users be the set of organizer, Ques-Chain Contract, and all voter,

$$\text{Users} = \text{Voters} \cup \{\text{organizer}\} \cup \{\text{Ques-Chain}\}.$$

All the elements in Users has its accounts in ETH.

Communications through ETH account during the e-voting, which will be recorded by ETH, can be represented like,

$$\text{a} \underset{\text{memo}}{\overset{c}{\rightarrow}} \text{Ques-Chain}$$

and c is the objects they what to send; memo is the explanation authors want to add.
Applying function f to object x can be represent like $f(x)$,

e.g.

$$m' = \text{Hash}(m).$$

Encrypt or decrypt message by function $\text{Enc}(key, message)$ and $\text{Dec}(key, message)$.

e.g.

$$m' = \mathrm{Enc}(pk, m)$$

where m' is the encrypted message of m under key pk.

Data (i.e. Public key) owned by different types of participants by Unified Modeling Language (UML) in Table 1.

Table 1: Data Owned by Different Types of Participants

organizer	voter
−sk	−r
+pk	+address/−accounts
+address/−accounts	−address/−accounts
	−uuid

1. Sk/pk: to sign/check the signature.
2. address/accounts: an ETH account, to prove who you are during communications.
3. address/accounts‘: an anonymous ETH account, which was randomly generated, destroyed immediately.
4. r: a key to blind the message, which was randomly generated, stored only in local.
5. uuid: to identify different ballots, which was randomly generated, stored only in local.

6. DETAILS OF PROTOCOL

In this session, we will introduce each stage of our protocol during the e-voting, which can be roughly divided into setup stage, sign stage, vote stage and count stage.

6.1. SETUP STAGE

The organizer should initialize the e-voting by following steps,

1. construct a Ques-Chain like Smart Contract with its pk, vote-start time st, vote-check time ct and vote-end time et.

$$\text{voter} \xrightarrow[\text{unchangable}]{pk,st,ct,et} \text{Ques-Chain}$$

2. construct and publish the voting page.
3. make a permission list of all voters' address like,

$$\text{Voters} = \{\text{voterA,voterB,voterC}, \dots\}$$

4. Applying function Chance() on voter to find out number of submissions the voter allowed.

Then the voters can construct their ballots, represented by ballot.

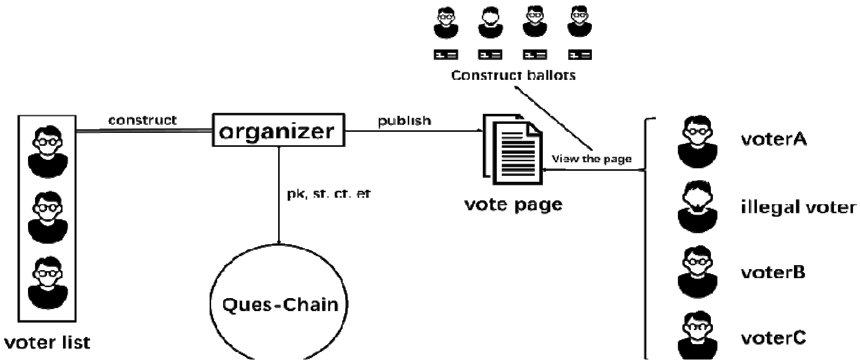

Figure 1: Setup stage in Ques-Chain protocol.

6.2. SIGN STAGE

In this stage, the voter will get a signed-blinded-ballot from the organizer. Before ct and after st, the voter should do the following steps.

Firstly, the voter need to get the hash of ballot.
Then, the voter randomly generated a key r and an uuid, stored only locally. Append the uuid into the end of m and encrypt them by r, get the BlindedBallot.

$$\text{BlindedBallot} = \text{Enc}(r,\text{Hash}(\text{ballot}) + \text{uuid})$$

where the operator $+$ means append into the end.

Algorithm 1: Decide whether to sign or not.
Require: address, BlindedBallot 1: **if** address in Voters **and** Chance(address)>0 **then** 2: SignedBlindedBallot = Enc(sk, BlindedBallot) 3: Chance(address) = Chance(address) - 1 4: **else** 5: SignedBlindedBallot = 0 6: **end if** 7: **return** Signed Blinded Ballot

Algorithm 2: Signature check in Ques-Chain contract.
Require: SignedBlindedBallot, BlindedBallot 1: **if**Dec(pk, SignedBlindedBallot) = BlindedBallot**then** 2: result = True 3: Chance(address) = Chance(address) - 1 4: **else** 5: result = False 6: **end if** 7: **return**result

After that, the voter sendBlindedBallot to the organizer through its ETH accounts which had the permission to vote.

$$\text{voter} \xrightarrow{\text{Blinded Ballot}} \text{organizer}$$

The organizer could decide whether to sign or not by Algorithm 1and then send the SignedBlindedBallot to the voter.

$$\text{organizer} \xrightarrow{\text{Signed Blinded Ballot}} \text{voter}$$

It should be noted that the organizer can't get any information of r and uuid for them only being stored locally.

And for the voter, to ensure the organizer give all the voters the same signature, he should send Signed Blinded Ballot and Blinded Ballot to Ques-Chain Contract to check the signature.

In the end, the voter send Signed Blinded Ballot and Blinded Ballot to Ques-Chain Contract for checking.

$$\text{voter} \xrightarrow[\text{BlindedBallot}]{\text{SignedBlindedBallot}} \text{Ques-Chain}$$

Ques-Chain Contract will check the signature and return result by Algorithm [algo:2].

If result is True, the voter can step into next stage, else he should communicate with the organizer to find out what's wrong with the SignedBlindedBallot or BlindedBallot.

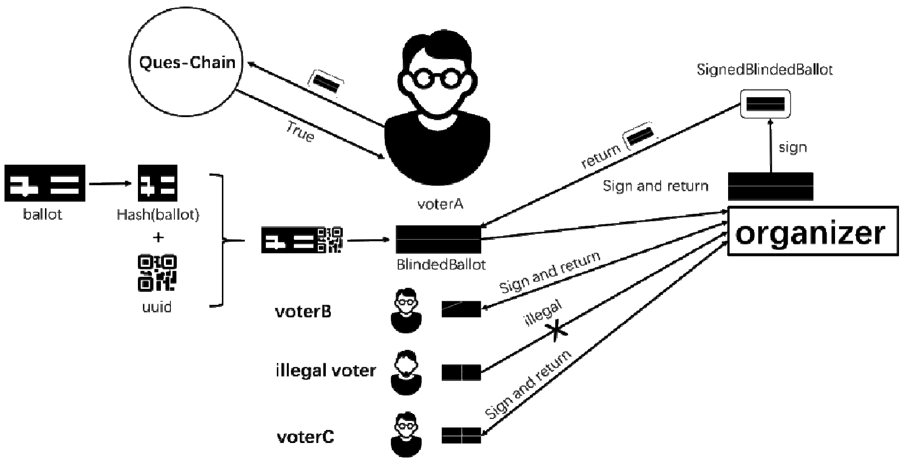

Figure 2: Sign stage in Ques-Chain protocol

Algorithm 3: Judge function.
Require: SignedBallot,Ballot,uuid 1: **if**Dec(pk,SignedBallot) = Hash(Ballot) + uuid **and not** (uuid in BallotBox.values()) **then** 2: Add map (Ballot, uuid) into BallotBox 3: result = True 4: **else**

5: result = False
6: **end if**
7: **return**result

6.3. Vote Stage

In this stage, the voter will vote. Before et and after ct, the voter should do the following steps.

1. decrypt the SignedBlindedBallot to SignedBallot.

$$\text{SignedBallot} = \text{Dec}(r, \text{SignedBlindedBallot})$$

2. randomly generated a new ETH account account′ with address′, stored only locally.
3. anonymously send the SignedBallot, Ballot, uuid to Ques-Chain Contract through account′.

$$\text{voter} \xrightarrow[\text{through account}']{\text{SignedBallot, Ballot, uuid}} \text{Ques-Chain}$$

It should be noted that account′ is generated randomly and only stored locally, so no one (except the voter) will know whom account′ belongs to, which made it untraceable.

6.4. Count Stage

In this stage, the voter will count legal ballots and publish the result.

Here we use a map BallotBox to store legal ballots and its uuid. Each time Ques-Chain Contract receive a ballot, Ques-Chain Contract will check the ballot by Algorithm [algo:3].

SignedBallot, Ballot, uuid Add map (Ballot, uuid) into BallotBox result = True result = False result

Every legal ballot will be stored in BallotBox while illegal ballots will be ignored.

After et, all users of the ETH can get the result of the vote by counting all ballots in BallotBox.

6.5. Publish Option

Due to the feature of ETH, the result of the vote will be published on ETH and everyone which everyone is accessible. However, in some scenery, the organizer may want to keep secret of the result or want to control whether push it. Here we give an option for the organizer to set whether publish or not by generating an extra $pk'' - sk''$ and use the Enc(pk,Ballot) replaces the Ballot on sign stage and vote stage. The organizer can publish the result by publishing sk''.

7. Security Analysis

According to the standard mentioned on the reference, our protocol equipped following security properties:

Correctness If all the election's participants, such as voters, authorities and so on are honest and behave as it is scheduled, then the final results are effectively the tally of casted votes.

Privacy No participant other than a voter should be able to determine the value of the vote cast by that voter.

Robustness Faulty behavior of any coalition of authorities can be tolerated. No coalition of voters can disrupt the election, and any cheating voter will be detected.

Verifiability Correct voting processes must be verifiable to prevent incorrect voting results.
Democracy There are two requirements to satisfy in democracy,

- *Eligibility*: only authorized voters are allowed to vote.
- *Prevention of multiple voting*: all eligible voters are allowed to cast the scheduled vote's number (function of the election system and his part in it) and not more, such that each voter has his intended power in deciding the outcome of the voting.

Fairness No participant can gain any knowledge, except his vote, about the (partial) tally before the counting stage (The knowledge of the partial tally could affect the intentions of the voters who has not yet voted.)

However, some attacks remain as follow.
Receipt-Freeness Voters must neither be able to obtain nor construct a receipt which can prove the content of their vote.

For the voters have key r, which was used to blind the ballots, to prove the content of their vote, our protocol doesn't equip Receipt-Freeness.

In general, the security properties of the protocol as shown inTable 2[10].

Table 2: Security Properties of the Protocol

<table>
<tr><th colspan="2">Requirement</th><th>Property</th></tr>
<tr><td colspan="2">Privacy</td><td>Correct</td></tr>
<tr><td colspan="2">Receipt-Freeness</td><td>Attacks Found</td></tr>
<tr><td colspan="2">Robustness</td><td>Correct</td></tr>
<tr><td rowspan="2">Verifiability</td><td>U</td><td rowspan="2">Correct</td></tr>
<tr><td>I</td></tr>
<tr><td rowspan="2">Democracy</td><td>E</td><td rowspan="2">Correct</td></tr>
<tr><td>PMV</td></tr>
<tr><td colspan="2">Fairness</td><td>Correct</td></tr>
<tr><td colspan="2">Correctness</td><td>Correct</td></tr>
</table>

8. USAGES

Ques-Chain can be applied to applications varying from the national referendum to internal evaluations conducted by companies.

In the case of election or referendum, Ques-Chain guarantees the consistency of the rule for all voters, making sure that every voter has chances exactly the organizer given to vote. Computing tasks can be distributed either throughout the Ethereum main network or to everyone in the country of election who volunteers to verify the computation with his or her computer.

The estimated cost of conducting the Brexit referendum using Ques-Chain on the Ethereum main network is 3.9 million pounds, 97% lower than the traditional way which cost 129.1 million pounds[11]. The latter way of using volunteer computation power may save even more. Online retailers may utilize Ques-Chain to collect customers' reviews. With full control of submission permission, shopping sites save money and time countering spam reviews, being able to show real ones to interested customers. Corporations can conduct employee evaluations using Ques-Chain. Anonymity is protected to encourage real feedback and prevent gossips or bullying. Non-governmental organizations (NGOs) may also evaluate projects they carried out or sponsored with the help of Ques-Chain.

Ques-Chain can be used in a situation which has a high information-security requirements. For the organizations want to hear real thoughts from their employees and clients. However, the latter may have concerns about whether their expression will bring them bad consequences. Such concerns may lead to distorted feedback results. Even if organizations don't care about the exact identity of feedback sources, with current technology, they are not equipped to prove or ensure that they won't do so.

Votes, questionnaires, outcome assessment, opinion collection, complaint reporting, etc., all applications which involve anonymous feedback can take advantage of Ques-Chain. While the permission right is held by organizers, voters' anonymity is still protected.

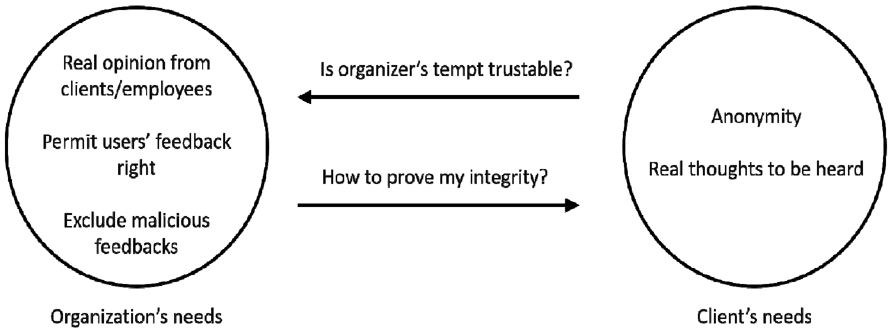

Figure 3: Applications of Ques-Chain protocol.

9. CONCLUSIONS

We introduced a new e-voting protocol (Ques-Chain), which is based on Ethereum (ETH) and implemented by blind signature. We showed the features of our protocol, including Correctness, Privacy, Robustness, Verifiability, Democracy, Fairness, etc. Based on these, we presented its serval advantages and further usages, like questionnaires, outcome assessment, opinion collection, complaint reporting. The real problem Ques-Chain solved is trusts between the organizer and voters.

REFERENCES

[1] M. Pawlak, A. Poniszewska-Marańda, and N. Kryvinska, "Towards the intelligent agents for blockchain e-voting system," *Procedia Computer Science*, vol. 141, pp. 239–246, 2018.

[2] Y. Liu and Q. Wang, "An e-voting protocol based on blockchain." *IACR Cryptology ePrint Archive*, vol. 2017, p. 1043, 2017.

[3] S. Panja and B. K. Roy, "A secure end-to-end verifiable e-voting system using zero knowledge based blockchain." *IACR Cryptology ePrint Archive*, vol. 2018, p. 466, 2018.

[4] F. Francesco, M. I. Lunesu, F. E. Pani, and A. Pinna, "Crypto-voting, a blockchain based e-voting system," in *10th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, 2018, pp. 223–227.

[5] F. .Hjálmarsson, G. K. Hreiarsson, M. Hamdaqa, and G. Hjálmtỳsson, "Blockchain-based e-voting system," in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*.IEEE, 2018, pp. 983–986.

[6] F. S. Hardwick, R. N. Akram, and K. Markantonakis, "E-voting with blockchain: An e-voting protocol with decentralisation and voter privacy,"*arXiv preprint arXiv:1805.10258, 2018*.

[7] D. Chaum, "Blind signatures for untraceable payments," in *Advances in cryptology*.Springer, 1983, pp. 199–203.

[8] S. Ibrahim, M. Kamat, M. Salleh, and S. R. A. Aziz, "Secure e-voting with blind signature," in *4th National Conference of Telecommunication Technology, 2003*. NCTT 2003 Proceedings.IEEE, 2003, pp. 193–197.

[9] G. Wood et al., "Ethereum: A secure decentralised generalised transaction ledger," *Ethereum project yellow paper*, vol. 151, pp. 1–32, 2014.

[10] L. Fouard, M. Duclos, and P. Lafourcade, "Survey on electronic voting schemes,"*supported by the ANR project AVOTÉ*, 2007.

[11] (2018) The costs of delivering the june 2016 referendum on the uk's membership of the european union. [Online]. Available: https://www.electoralcommission.org.uk/find-information-by-subject/elections-and-referendums/past-elections-and-referendums/the-costs-of-delivering-the-june-2016-referendum-on-the-uks-membership-of-the-european-union