electronics

MDPI

Article

Distributed E-Voting and E-Bidding Systems Based on Smart Contract

Raylin Tso [1,*], Zi-Yuan Liu [1] and Jen-Ho Hsiao [2]

[1] Department of Computer Science, National Chengchi University, Taipei 11605, Taiwan; yad50968@gmail.com

[2] National Center for Cyber Security Technology, The Executive Yuan, Taipei 10674, Taiwan; peter087744982@gmail.com

* Correspondence: raylin@cs.nccu.edu.tw; Tel.: +886-0229393091 (ext. 62329)

Received: 4 March 2019; Accepted: 8 April 2019; Published: 11 April 2019

Abstract: Traditional voting and bidding systems largely rely on paperwork and human resources throughout the voting process, which can incur high costs in terms of both time and money. Electronic voting and electronic bidding systems can be used to reduce costs, and many new systems have been introduced. However, most systems require a powerful and trusted third party to guarantee system integrity and security. With developments in blockchain technology, research has begun to highlight the core concept of decentralization. In this study, we introduce the first decentralized electronic voting and bidding systems based on a blockchain and smart contract. We also use cryptographic techniques such as oblivious transfer and homomorphic encryptions to improve privacy protection. Our proposed systems allow voters and bidders to participate in the opening phase and improve participant anonymity, the privacy of data transmission, and data reliability and verifiability. Moreover, compared with other electronic voting and bidding systems, our systems are safer and more efficient.

Keywords: blockchain; oblivious transfer; smart contract; E-voting; E-bidding; homomorphic encryption

1. Introduction

In areas such as the financial and banking service industries, private information is transmitted through a trusted third party. However, this process involves many problems and complicated procedures. For instance, users may want to know if the trusted party is really honest or their sensitive information is safely protected. How to maintain trust, privacy, and evaluation without a trusted authority becomes an important issue. In 2010, Dolev et al. showed how multi-users trust each other without the help of trusted authority [1]. Their scheme also significantly reduced the number of message exchanges, and therefore, it is more suitable for real environments. Additionally, many studies on multi-party computation have also been proposed in the past decade [2,3].

Recently, with the development of blockchain technology, researchers pay more and more attention to the core concept of decentralization, which is a main feature of blockchain and smart contracts (SCs), whereby an application uses a blockchain as its core technique. Due to the feature of decentralization, current researchers began to analyze the structure of electronic voting (E-voting) systems currently in use. In turn, they discovered that the blockchain and SC in the application could improve data validity and lower costs while maintaining the openness and transparency of the application [4–6].

1.1. E-Voting

The Twenty-First Century is an era of democracy in which the majority rules, which is an indispensable element for democratic countries. When using a voting system to identify the candidate with the majority of votes, everyone must respect the final results. Furthermore, the voting process for elections must be fair, just, transparent, and prevent the unlawful obtainment of votes. According to the provisions of Article 63 of the Civil Servants Election And Recall Act of Taiwan, if the final result reveals a gap of less than three thousand votes between the candidates with the highest and second highest number of votes, the second-place candidate can submit a request to the Legislative Yuan for a ballot recount within seven days of the election. Furthermore, the Central Election Commission will execute the relevant procedures in accordance with the law. Many preparations must be undertaken before a traditional election, including the publication of the electoral bulletin, legal vote notice, and various ballot documents. A great deal of manpower and material is required, which incurs substantial costs. To solve this problem, most research has focused on electronic voting to improve voting efficiency and reduce relevant expenses.

In the most well-known U.S. presidential election [7], some states used an E-voting system. The E-voting mechanism in the United States uses an electronic voting machine (EVM), and all voters participating in E-voting must personally visit a polling station to vote. Voter eligibility is first confirmed manually, and voters whose identities have been confirmed receive a personal password for them to enter into the voting machine manually. After voters have entered their password, they can vote. They can use candidate information available on the electronic voting machine to vote, and the ballot information is stored in the voting machine. Unless hackers install malware on the voting machine, they cannot manipulate the voting machine or tamper with ballot information through the Internet because the machine is offline. However, the voting machines are monitored, meaning that using electronic voting machines to store ballot information is a fair and secure approach.

Although the E-voting mechanism in the United States can satisfy most security requirements, its utility rate is only 29%–31%, which means that the reliability of voters on E-voting remains low. We believe that the utility rate would substantially increase if voters were permitted to participate in the opening phase in the future.

1.2. E-Bidding

The mode of government procurement of votes has gradually transitioned to an electronic system. The emergence of electronic bidding (E-bidding) enables the relevant authority to upload its tender documents to the tender website system of the Public Construction Commission (PCC) of the Executive Yuan; bidders can then read or download the document from the system after paying a tender document fee. Additionally, if bidders also want to join, then they can upload their bidding documents after paying the deposit, and the document will be stored in the tender website system of the PCC until the opening date. On the opening date, the tender authority will determine and notify the bidding winner after reviewing the price, as well as bid bond and corporate certificates. Next, the tender authority will refund bid bonds to all bidders except the winner, whose bid bond certificate is retained for review. E-bidding has become a convenient service in past years; it aims to provide vendors with a fair bidding environment and develop a public and transparent procurement procedure to protect public benefits. Current bidding systems [8] continue to employ a trusted third party, namely the tender authority, to open and verify the bidding documents; however, the vendors who participate in bidding can only trust the credibility of the tender authority.

1.3. Related Works

1.3.1. E-Voting

Hereunder, we introduce the other start-of-the-art schemes for E-voting. First of all, in 2012, Buccafurri et al. proposed a light-weighted voting system relying on existing social networks [9]. In this system, the participants use their social network profile to perform voting. In addition, Buccafurri et al. also provided an improving scheme, which was more secure [10]. In this scheme, in order to keep users anonymous, they are related to a number of attributes that are chosen by themselves.

Additionally, in 2013, Chen et al. proposed an electronic voting system [11] with an oblivious signature protocol, which was mainly implemented by the RSA digital signature mechanism with the one-out-of-n oblivious signature protocol. Although the system enables voters to participate in verifying the ballots at the opening phase, it is still not confident enough since voting initiators with a private key are able to know the trend of the ballots, which are encrypted by a public key of the voting initiator before being stored in the record center, even if the election has not ended. Moreover, there are two more defects in this electronic voting system. One is that obtaining the results of the election in advance is still possible because the ballots only use a single public key for encryption. Besides, whoever owns the corresponding private key has the opportunity to decrypt the information of the ballot beforehand, causing the fairness of the election to be compromised. Another defect is that when obtaining the ballot signature, the signer cannot verify whether the document sent by the voters is correct. If the document is not relevant to the election and voters attempt to bilk signatures, the signer must still sign the document, meaning that the signer's privacy is left unprotected.

In 2015, Nair et al. proposed a program [12] that uses a secret sharing scheme as the core structure of the E-voting mechanism and secure multiparty computation to improve the efficiency of the opening phase. This program converts the binary value, which is generated from the number of votes a candidate received, to a decimal value, and this decimal value is used as the secret value of the vote. The voting machine scatters the secret value into five point coordinates through secret sharing and stores them on five data servers in different environments. After the opening phase has begin, any three of the five servers restore the secret information and obtain a multinomial of the total votes. By removing the constant of the multinomial and converting it to a binary value, the actual number of votes for each candidate can be obtained directly, thereby significantly improving the efficiency of voting in the opening phase.

1.3.2. Blockchain-Based E-Voting

With the development of the blockchain technique, many blockchain-based E-voting schemes have also been proposed. For example, in 2017, McCorry et al. proposed a self-tallying Internet voting protocol based on blockchain [13]. Their protocol was based on the decentralized two-round protocol, called the open vote network, which was designed for supporting small-scale boardroom voting. Unfortunately, we found that there was an important drawback in their scheme, i.e., the Open Vote Network required all the registered voters to finish the vote. Concretely speaking, if there is one registered voter that does not finish the voting, the tally calculation cannot be performed. Additionally, in 2018, Hjálmarsson et al. also proposed a new voting scheme based on blockchain [14]. Their scheme uses a smart contract to tally the result. However, because everyone has access to the blockchain, the voters can obtain the voting information during the voting. Therefore, how to construct using blockchain, which can protect the privacy of the voters and the correctness of the result, is still an important issue.

1.4. Contributions

In this study, we analyzed the architecture of currently-used electronic voting (E-voting) and E-bidding systems and discovered that both systems employ a trusted third party to complete the opening phase and are required to meet security requirements such as confidentially, undeniability, and anonymity. However, the blockchain and SC have features of decentralization that can improve the aforementioned shortcoming. Replacing the third party with a smart contract that is based on the Ethereum blockchain is a promising method for achieving the goal of lower costs and improved data verifiability.

The basic security requirements of elections include voter anonymity, as well as ballot confidentiality and verifiability. Among these, anonymity and confidentiality can be ensured by adopting a public key cryptosystem with a secret sharing scheme. With the properties of public access, transparency, and nonrepudiation, SCs can be used to ensure ballot verifiability. Voters on the blockchain network can obtain ballot information from the SC and then compute and verify the election result without the trusted third party during the billing phase.

The basic security requirements for the procurement of bidding results by public authorities include verifying the vendor's identity and ensuring that the bidding document remains confidential and verifiable. Among these, anonymity and confidentiality can be ensured by using a public key crypto system; the properties of public access, transparency, and nonrepudiation of the SC ensure the verifiability of bidding documents. The vendors on the blockchain network can obtain bidding documents in the cipher type from the SC and then compute and verify the bidding result without a trusted third party in the opening phase.

The emergence of blockchain technology and SCs has enabled E-voting and E-bidding to operate without a trusted third party. Any voter who participates in an election or vendor who participates in bidding can respectively obtain ballot information or bidding documents from the SC and then independently compute the result during the opening phase. This approach not only strengthens overall trust in the electronic process, but also reduces expenditure on human and material resources, thereby improving the efficiency of the administration.

This study combined blockchain technology with privacy-protection cryptography to produce and distribute an E-voting and E-bidding application that does not involve a trusted third party. User anonymity, data transmission privacy, and data verifiability are universally ensured and transparent during the opening phase. We believe that our mechanism can replace the existing voting and tendering process. More concretely, in our scheme, a voter can perform voting through the network. In addition, by oblivious transfer, the ballot is anonymous. Besides, by taking advantage of the secret sharing technology, the ballots are encrypted and stored on different servers. Moreover, the integrity of the ballot is protected through the blockchain.

1.5. Study Structure

Our research is divided into six sections. In Section 1, we introduce our research motivations and contributions. Section 2 presents a review of related research and knowledge, and Sections 3 and 4 present the results of implementing our E-voting and E-bidding systems. We explain the security analysis of our schemes in Section 5 and offer conclusions and directions for future research in Section 6.

2. Preliminaries

2.1. Notation

For the sake of simplicity and readability, we let $\mathbb{Z}$ be the set of integers, $\mathbb{Z}_N$ be the additive group of integers modulo N, and $\mathbb{Z}_N^*$ be the multiplicative group of integers modulo N. We defined $h(\cdot)$ as a secure hash function (e.g., SHA-256 or SHA-3) and $\varphi(\cdot)$ as a Euler's phi function. $E_k(m)$ denotes the asymmetric encrypt in the message m using key k; $D_k(c)$ denotes the asymmetric decrypt in the ciphertext c using key k; and $Sig_k(m)$ denotes the sign in the message m using key k. We use LCM and GCD to represent lowest common multiple and greatest common divisor, respectively.

2.2. Electronic Voting

The E-voting process can be categorized into the following three phases:

- Registration phase: During this phase, the voting qualification of each voter must be verified offline before the election begins. Next, verified voters will receive a unique virtual identification code (*PID*), which they can use to obtain a unique voting certification (*Cert*) from the registration server. Voters should use this certification to apply their ballot signature and personal key pair on the voting day and undergo the normal voting process.

- Voting phase: On the polling day, voters must use their unique ballot certifications obtained in the registration stage to confirm their voting qualifications through the secure transmission channel with the verification server. Voters who have been verified by the server can obtain a ballot signature with their ballot certifications and personal key pairs. After obtaining ballot certification, the voter will be able to vote either on voting websites or using relevant applications. Ballot information will be recorded and stored in the smart contract by the voting website or relevant server. Voters can review current ballot information at any time before the opening stage to ensure that information has been correctly recorded in the ballot box. This improves the reliability of the system.
- Opening phase: When the voting stage is closed, voters may browse the voting website and review ballot information stored in the SC. Additionally, they may verify the correctness of certifications and information on the ballot and compare the results with those announced at the record center (RC). If any conflicts or inconsistencies are observed, voters can directly request ballot verification to ensure the fairness of the election.

2.3. Electronic Bidding

E-bidding must follow the standards detailed in the articles of the Government Procurement Act. The process is categorized into the following six phases:

- Tender inviting phase: The Government Procurement Department authorizes the E-bidding center (referred to as the tender website) to announce the tender items and content to establish an open tender. There are three types of tender: open tenders, selective tenders, and limited tenders. Open tenders are the subject of this research.
- Tender obtaining phase: Several different tender cases are posted on the tender website. Companies or manufacturers wishing to participate are required to prepay a service fee (referred to as the tender receiving fee); after obtaining the tender receiving certificate, the suppliers can then download the tender document from the website. According to the provisions of Article 34 in the Government Procurement Act, the relevant authorities shall not disclose the reserve prices, names, or number of tenderers or any related information that would compromise the scope or fairness of the competition.
- Tender submitting phase: After receiving an invitation to tender, suppliers should complete the submission document and pay the bidding bond (should not exceed 10% of the marked price) according to the submission provisions specified on the invitation. The correctness of the paid bid amount from suppliers will be verified with a fair external banking system; if correct, then the supplier will receive a certification of proof that the bid bond has been paid, and if not correct, then the supplier will be notified that the payment amount is incorrect and that the submission has failed. The bid bond certification is the determining factor for whether prepaid bid bonds will be returned to suppliers who have failed to win the bid after the tender deciding stage. Thus, this certification is critical and should not be released at will.
- Tender opening phase: According to the provision of Article 34 of the Government Procurement Act, the government reserve price shall remain confidential until the award is granted; moreover, in special circumstances, the reserve prices may be disclosed after the award stage. However, the relevant authority may, if necessary, disclose the government reserve prices in tender documentation. Unless otherwise required for official use, or provided by relevant laws and regulations, tenders submitted by suppliers shall remain confidential. In the opening stage, tender authorities should conduct an anonymous and public tender opening. However, if the proposed price provided by suppliers does not meet the upset price, then the bargaining stage will be opened. Suppliers may then modify the proposed price to participate in second-round bidding.

According to the provision of Article 50 of the Government Procurement Act, if the tender authorities have already conducted tender opening on the opening day, but the qualifications of the supplier do not meet the requirements or the proposed price still does not meet the upset price, then the supplier is not permitted to enter the tender deciding stage and will be considered a null tender.

- Tender deciding phase: In this stage, the tender authorities compare and analyze every proposed bidding price provided by legal suppliers. The authorities must determine whether the proposed price is reasonable and lower than the government estimate. If the proposed prices provided by suppliers are all higher than the estimate, then the authorities must select one of the most favorable suppliers and announce it as the winner.
- Contract management phase: In the deciding stage, tender authorities will select a winning supplier and send the winner a notification. The winning supplier must accept the result; if the supplier wants to retract after winning the bid, then authorities directly confiscate the bid bond. The bid bond paid by the winning supplier is then directly transferred to a performance bond, and the winning supplier also received a performance bond certification. Suppliers who did not win the bidding may apply for the return of their bid bond by providing authorities with their bid bond certification.

2.4. Ethereum and Smart Contracts

Blockchain is a continuous ledger that is connected by multiple blocks. Each block contains hundreds to thousands of transactions, which are verified by miners on the blockchain network and then packaged and sent to the end of the main chain; this process is referred to as mining.

Mining is a consensus algorithm that is used to enable miners to verify transactions and package them for transfer to the end of the main chain. It can record the time-generated order of each block in detail, achieve consensus between miners by using the algorithm, and prevent attackers from tampering with the block. No one can tamper with any component of the block to cheat or cancel their transactions. Each block contains a hash value of the previous block; therefore, after the block has been generated, its internal content cannot be easily tampered with. If a block is tampered with, then subsequent blocks must also be affected, meaning that achieving this goal requires a great deal of computing power. In other words, all attempts to engage in double spending will fail.

Because a blockchain maintains a public and transparent ledger, any participant in the blockchain network can question or verify the content of transactions to ensure nonrepudiation. Ethereum has an improved Bitcoin architecture and solved the problem of limited flexibility. The main contribution of Ethereum is an SC, which enables participants to operate applications on a private chain. Additionally, only authorized participants can participate in reaching a consensus on the private chain or alliance chain. Ethereum is a platform that has improved the architecture of Bitcoin, and its main application is an SC. Because the SC retains the core concept of a blockchain, it can be regarded as a proxy for processing transactions and procedures.

SCs are programmed using high-level programming languages, such as Solidity or Serpent. Through the corresponding response and processing of outside messages from prewritten program logic, they can reduce the burden of processing insurance claims [15] and salary payments, as well as improve operating speeds and efficiency. SCs use blockchain as the core technique, and any information on the SC is made public and transparent. Therefore, including too much external logic and confidential information in them is inappropriate. If confidential information is required to be included in the SC, then confidential information should be encrypted outside of the contract before inclusion. Ethereum also generates the address of the contract, and the person who knows the address is able to communicate and deliver messages with the contract.

However, transactions processed with SCs are not limited to money transfers; any action taken using an SC is considered a transaction, even simple searches or the addition of information. For elections, a trustworthy electronic voting process must provide a public environment that can withstand verification by voters. Based on the verifiability and nonrepudiation of the blockchain and SC, the present study uses an SC to replace the existing bulletin board to simplify the election process and enable voters to review voting progress and information at the time of registration, billing, and ticketing.

2.5. Paillier Cryptosystem

Paillier proposed a public key cryptosystem [16] in 1999. It has the key feature of additive homomorphic encryption [17], which ensures that the sum of two ciphertexts produced through computation after decrypting is equivalent to the sum of two plaintexts. This feature can be widely used in applications that require sum computations, such as E-voting and E-bidding, and also meets the requirement of data confidentially. Paillier's public key cryptosystem consists of the key generation phase, encryption phase, and decryption phase, which are detailed as follows:

- Key generation phase
  1. Select sufficiently large prime numbers p and q, where $GCD(pq, (p-1)(q-1)) = 1$
  2. Compute $N = p \times q$ and $\lambda = LCM(p-1, q-1)$
  3. Select a random number $g \in \mathbb{Z}^*_{N^2}$
  4. Define function $L(u) = \frac{u-1}{N}$
  5. Compute $\mu = (L(g^\lambda) \bmod N^2)^{-1} \pmod N$
  6. Generate public key (N, g) and private key (λ, μ)
- Encryption phase
  1. Select a random number $r \in \mathbb{Z}^*_N$
  2. Compute ciphertext $c = g^m \times r^N$
- Decryption phase
  1. Compute plaintext $m = L(c^\lambda \bmod N^2) \times \mu \pmod N$

2.6. Additive Homomorphic Encryption

Additive homomorphic encryption [18] is the result of using a certain computation method to add two cipher texts and is equivalent to the sum of two plaintexts. Assume that we encrypt two plaintexts, m_1 and m_2, using the same public key (N, g) to obtain two ciphertexts c_1 and c_2, respectively.

$$E_{pk}(m_1) = g^{m_1 \times r_1^N} = c_1$$
$$E_{pk}(m_2) = g^{m_2 \times r_2^N} = c_2$$

Next, multiplying c_1 by c_2 yields C. By decrypting C using corresponding private key sk, we can obtain a result equal to m_1 plus m_2 in plaintext.

$$c_1 \odot c_2 = E_{pk}(m_1) \odot E_{pk}(m_1) = E(m_1 + m_2) = C$$

2.7. Shamir's Secret Sharing Scheme

Shamir proposed the earliest concept of (k, n) secret sharing [19] in 1979, in which secret S is divided into n pieces of data $S_1, \cdots, S_n$. Recovering the secret S requires that at least k pieces of data be combined. Based on this concept, Figure 1 illustrates a secret S held by n people who can only have one fragment (represented as S_1–S_n). The secret can only be recovered if k or more participants work together.

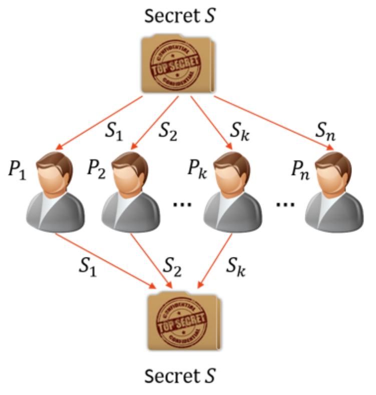

Figure 1. Shamir's secret sharing scheme.

2.8. Oblivious Transfer

Rabin et al. proposed the first oblivious transfer (OT) protocol in 1981 [15]. The OT protocol is as follows: when a receiver obtains a message sent by the sender, the sender remains oblivious regarding whether the message was received, knowing only that the probability is 1/2.

The study of the OT protocol has been separated into three types; one-out-of-two, one-out-of-N [20], and T-out-of-N [21]. This study focuses on the one-out-of-N OT protocol, which includes two parties, namely the sender and receiver. Assuming that the sender has n messages $m_1, m_2, \cdots, m_n$ and the receiver wants to obtain a specific message m_c, the OT protocol guarantees the correctness of the content and privacy of the sender and receiver.

- Correctness: The receiver can obtain the specific message m_c after executing the protocol.
- Privacy of the sender: The receiver can only obtain the specific message m_c and has no knowledge regarding the other messages to protect the privacy of the sender.
- Privacy of the receiver: The sender cannot know which message has been obtained by the receiver.

3. Proposed E-Voting System

This study combines an SC and privacy-protection cryptography to produce a distributed electronic voting system that enables voters to participate in the billing phase and improves the efficiency of the election process. Because information on the blockchain is completely transparent and public, voter ballot information must be fully confidential before the billing phase begins. Table 1 provides a system description, and Figure 2 displays its complete architecture. All transmitted information is stored in SC for further verification.

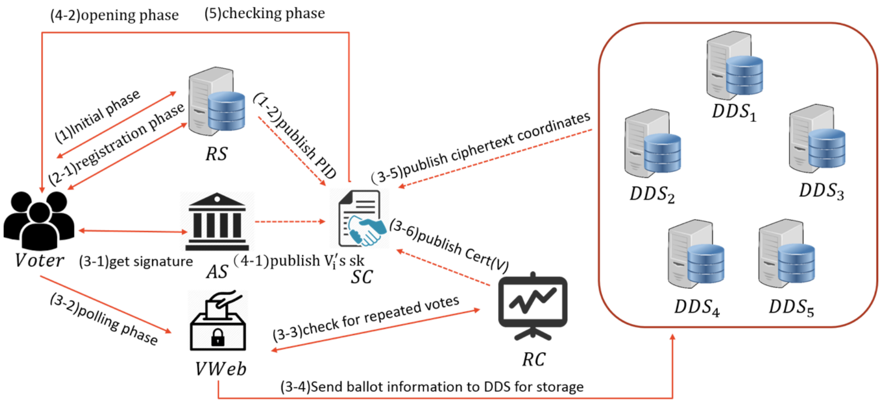

Figure 2. Architecture of our E-voting system.

Table 1. Description of our E-voting system.

Term	Description
V_i	Voters with voting qualification
RS	Registration server
AS	Authentication center
$VWeb$	Voting website
RC	Recode center
DDS	Distributed data server
SC	Smart Contract
$Cert_{V_i}$	V_i's voting certificate
pk_{RC}, sk_{RC}	RC's key pair
pk_{V_i}, sk_{V_i}	V_i's personal key pair
PID_i	V_i's personal identification code
SSN_{V_i}	V_i's social security number
$Sig_k(m)$	Signature of m that is signed by k

3.1. System Description

Our system has seven roles:

1. Voter (V_i): V_i is voter i with voting eligibility; after undergoing identity verification through RS to obtain a voting certificate $Cert_{V_i}$, voters can identify their PID_i on the SC. With the voting certificate, voters can request AS for their ballot signature in the voting phase. After voting, they can review the ballot information that has been published onto SC to confirm that their vote has been correctly counted. If the voters discover that the ballot has not been properly counted for any reason, then they should respond to the election committee immediately to confirm whether the certificate is abnormal and determine whether they should vote again.
2. Registration server (RS): This is responsible for verifying voter identity and generating and sending voting certificates $Cert_{V_i}$ and personal key pairs (pk_{V_i}, sk_{V_i}) to legal voters.
3. Authentication server (AS): This is responsible for verifying voter identity, as well as generating and sending ballot signatures to legal voters through one-out-of-n oblivious transfer.
4. Voting website ($VWeb$): The system $VWeb$ is part of the electoral organization. After a voter has voted, the voting website determines whether the voter has cast more than one vote. Next, it encrypts the ballot, first using pk_{V_i} followed by a secret sharing scheme, and then transmits the ballot coordinates to DDS.
5. Record center (RC): After voters have cast their votes, $VWeb$ sends both the voting certificate and ballot signature to the RC, which then confirms that they have not cast more than one vote. If the voter passes this test, then the RC will store the voting certificate and inform $VWeb$ that the voter's ballot information can be delivered to DDS. At the end of the process, RC records the voter's PID_i on the SC for further confirmation. After DDS receives the coordinates, it uses the RC's public key pk_{RC} to encrypt the coordinates and then records them on the SC for further confirmation if necessary.
6. Smart contract (SC): The smart contract is dynamic and enables voters to review their ballots and count the votes at the billing stage. It replaces the function of a traditional bulletin and increases public trust in elections.

3.2. Processes and Steps

In this section, we provide further explanations of the E-voting process. Our system had n_1 voters, n_2 candidates, and five DDSs. Additionally, all transmission processes were performed with Hypertext Transfer Protocol Secure. The voting process was divided into five stages: (1) initial phase, (2) registration phase, (3) voting phase, (4) opening phase, and (5) verification phase.

3.2.1. Initial Phase

Before the protocol, the *RS* and *AS* must generate their RSA-based public/private key pair, (e', N_1) and $(d', N_1)/(e, N_2)$ and (d, N_2), in which (d, N_2) and (d', N_1) are the signing key and (e, N_2) and (e', N_1) are the public key used for signature verification. The *RC* must generate its Paillier-based encryption and decryption key pair (pk_{RC}, sk_{RC}).

- The process of generating the *RS* signature key is as follows:
  1. Select two sufficiently large prime numbers p_1 and q_1.
  2. Compute $N_1 = p_1 \cdot q_1$.
  3. Compute $\varphi(N_1) = \varphi(p_1)\varphi(q_1) = (p_1 - 1)(q_1 - 1)$.
  4. Select a value e' that satisfies $GCD(e', \varphi(N_1)) = 1$.
  5. Determine a value d' that satisfies $e'd' \equiv 1 \bmod \varphi(N_1)$.
- The process of generating the *AS* signature key is as follows:
  1. Select two sufficiently large prime numbers p_2 and q_2.
  2. Compute $N_2 = p_2 \cdot q_2$.
  3. Compute $\varphi(N_2) = \varphi(p_2)\varphi(q_2) = (p_2 - 1)(q_2 - 1)$.
  4. Select a value e that satisfies $GCD(e, \varphi(N_2)) = 1$.
  5. Determine a value d that satisfies $ed \equiv 1 \bmod \varphi(N_2)$.
- The process of generating the *RC* key pair is as follows:
  1. Select two sufficiently large prime numbers p_3 and q_3 that satisfy $GCD(p_3q_3, (p_3 - 1)(q_3 - 1)) = 1$.
  2. Compute $N_3 = p_3 \cdot q_3$.
  3. Compute $\lambda = LCM(p_3 - 1, q_3 - 1)$.
  4. Select a random number $g \in \mathbb{Z}^*_{N_3^2}$.
  5. Define a function $L(u) = \frac{u-1}{N_3}$.
  6. Compute $\mu = (L(g^\lambda) \bmod N_3^2)^{-1} \pmod{N_3}$.

3.2.2. Registration Phase

In this stage, the *RS* confirms the identity of voters and sends voting certificates $Cert_{V_i}$ to voters, $1 \le i \le n_1$. The procedures are conducted offline.

1. The generation of user personal identification code proceeds for each V_i as follows:
   (a) Select a random number $t \in \mathbb{Z}^*_N$.
   (b) Generate the unique user personal identification code $PID_i = h(SSN_{V_i} \| t)$, where SSN_{V_i} is the voter's social security number.
   (c) Send PID_i to the *RS* for verification.
2. To verify the identity of V_i, where $1 \le i \le n_1$, the *RS* proceeds as follows:
   (a) Verify the correctness of PID_i.
   (b) If PID_i is correct, then the *RS* accepts it and issues a voting certificate $Cert_{V_i} = \{PID_i, Sig_{d'}(PID_i)\}$ to V_i. The certificate is the signature of PID_i signed by the *RS*.
   (c) Issue the key pair (pk_{V_i}, sk_{V_i}) belonging to voter V_i.
   (d) Publish the PID_i of eligible voters on the bulletin board. The bulletin board is in the form of an SC in this study.

3.2.3. Polling Phase

Assume that we have V_i voters, $1 \le i \le n_1$, and m_j candidates, $1 \le j \le n_2$. The following are thus the procedures that V_i must implement to obtain a ballot signature from the *AS* to be able to vote.

1. V_i provides $Cert_{V_i}$ to the AS and requests a ballot signature.
2. Assume that V_i wants to vote for candidate λ, where $\lambda \in \{1, \cdots, n_2\}$; V_i proceeds as follows:
   (a) Compute $h(\lambda)$.
   (b) Generate $E_{pk_{V_i}}(h(\lambda))$, using pk_{V_i} to encrypt $h(\lambda)$.
3. Select n_2 random numbers $r_j \in \mathbb{Z}_N^*$, and generate $c_j = m_j \| r_j$, for $1 \le j \le n_2$, where m_j denotes the ballot corresponding to candidate j. Send $E_{pk_{V_i}}(h(\lambda))$ and c_j to AS for $1 \le j \le n_2$.
4. After receiving c_j, for which $1 \le j \le n_2$, and $E_{pk_{V_i}}(h(\lambda))$ from V_i, the AS proceeds as follows:
   (a) Review each m_j from c_j, $1 \le j \le n_2$, to avoid the signing of incorrect or unrelated documents.
   (b) Compute the hash value of each $c_j = h(c_j)$, for which $1 \le j \le n_2$.
   (c) Encrypt the signatures and the hash values of λ_j using V_i's public key pk_{V_i} to obtain $X_j = E_{pk_{V_i}}(h(c_j)^d)$ and $E_{pk_{V_i}}(h(\lambda_j))$, for which $1 \le j \le n_2$.
   (d) Select n_2 random numbers k_j, where $k_j \in \mathbb{Z}_N^*$, and compute $E_{pk_{V_i}}(h(\lambda))^{k_j}$ and $E_{pk_{V_i}}(h(\lambda_j))^{k_j}$, for which $1 \le j \le n_2$. Notably, according to Paillier's additive homomorphic property, $E_{pk_{V_i}}(h(\lambda))^{k_j} = E_{pk_{V_i}}(h(\lambda)^{k_j})$ and $E_{pk_{V_i}}(h(\lambda_j))^{k_j} = E_{pk_{V_i}}(h(\lambda_j)^{k_j})$.
   (e) Compute $M(i,j) = E_{pk_{V_i}}(k_j(h(\lambda) - h(\lambda_j)) + h(c_j)^d)$, for which $1 \le j \le n_2$, and send $M(i,j)$ to V_i.
5. After receiving $M(i,j)$, for which $1 \le j \le n_2$, from the AS, V_i undertakes the following actions:
   (a) Obtain $M(i,j)$, for which $1 \le j \le n_2$, and then, use the private key sk_{V_i} to decrypt $M(i,j)$ to obtain n_2 ciphertexts.
   (b) Verify ciphertext λ using the AS public key to obtain only the ballot signature λ, for which $h(c_\lambda)^d$.

By following the procedures detailed in Figures 3 and 4, V_i obtains the voting ballot signature $h(c_\lambda)^d$ and then continues with the voting procedure. When V_i casts a vote, *VWeb* sends both the certificate and ballot signature $h(c_\lambda)^d$ of V_i to the *RC*, which then confirms that V_i has not already voted. If V_i has not voted, then the *RC* stores both the certificate and ballot signature and informs *VWeb* to transfer the voting information of V_i to *DDS*.

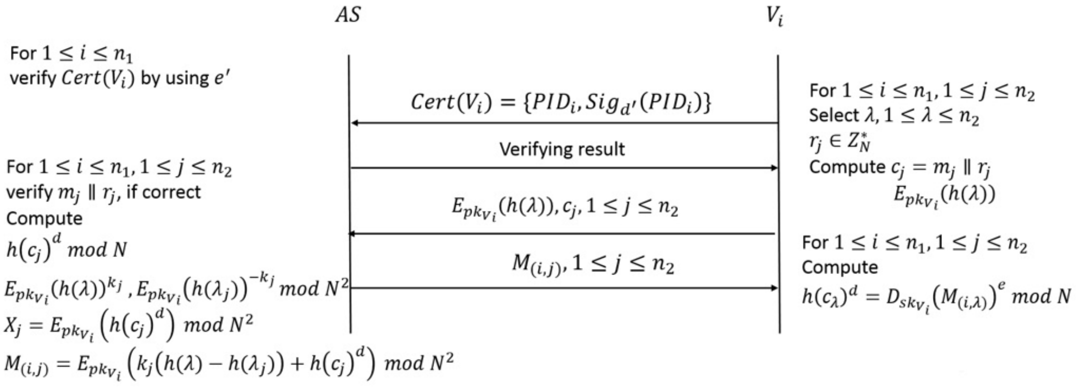
AS
V_i
For $1 \le i \le n_1$
verify $Cert(V_i)$ by using e'
$Cert(V_i) = \{PID_i, Sig_{d'}(PID_i)\}$
For $1 \le i \le n_1, 1 \le j \le n_2$
Select $\lambda, 1 \le \lambda \le n_2$
$r_j \in Z_N^*$
Compute $c_j = m_j \| r_j$
$E_{pk_{V_i}}(h(\lambda))$
Verifying result
For $1 \le i \le n_1, 1 \le j \le n_2$
verify $m_j \| r_j$, if correct
Compute
$h(c_j)^d$ mod N
$E_{pk_{V_i}}(h(\lambda)), c_j, 1 \le j \le n_2$
$M_{(i,j)}, 1 \le j \le n_2$
$E_{pk_{V_i}}(h(\lambda))^{k_j}, E_{pk_{V_i}}(h(\lambda_j))^{-k_j}$ mod N^2
$X_j = E_{pk_{V_i}}\left(h(c_j)^d\right)$ mod N^2
$M_{(i,j)} = E_{pk_{V_i}}\left(k_j(h(\lambda) - h(\lambda_j)) + h(c_j)^d\right)$ mod N^2
For $1 \le i \le n_1, 1 \le j \le n_2$
Compute
$h(c_\lambda)^d = D_{sk_{V_i}}(M_{(i,\lambda)})^e$ mod N

Figure 3. Ballot signature obtained with one-out-of-noblivious transfer (OT).

After polling, the candidate number λ selected by V_i and the ballot signature will be encrypted by *VWeb*, which uses the public key of V_i, denoted as pk_{V_i}. The cipher of the ballot is denoted as C_i, which is divided into k plaintext coordinates $PC_{(i,k)} = (x_k, y_k)$, in which $1 \le k \le 5$, using the $(3,5)$ secret sharing scheme, which can be recovered from three-out-of-five plaintext coordinates. *VWeb* stores these coordinates together with PID_i in the *DDS*.

Because the voter's key pair is issued by *RS*, if ballots are only encrypted by the voter's key pair, then the *RS* can obtain the results in advance. To prevent this from happening, after the *DDS* receives the coordinates, it will use the *RC* public key pk_{RC} to encrypt the coordinates, ultimately announcing the coordinates and PID_i, for which $1 \leq i \leq n_1$, through the *SC* to enable V_i to determine whether their ballot has been correctly counted.

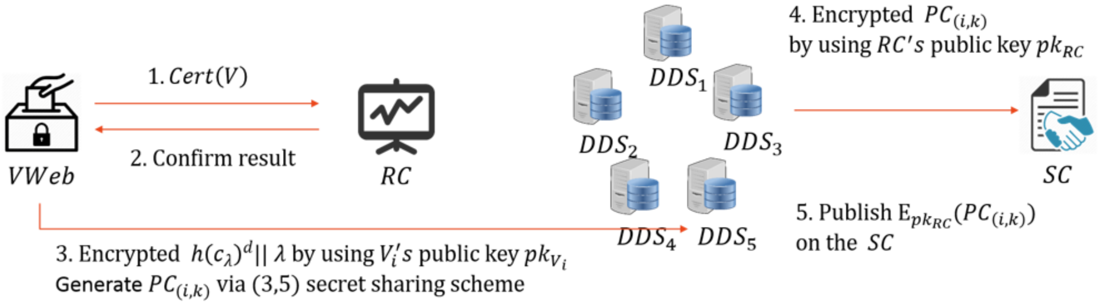
1. $Cert(V)$
2. Confirm result
$VWeb$
RC
DDS_1
DDS_2
DDS_3
DDS_4
DDS_5
SC
4. Encrypted $PC_{(i,k)}$ by using $RC's$ public key pk_{RC}
5. Publish $\mathrm{E}_{pk_{RC}}(PC_{(i,k)})$ on the SC
3. Encrypted $h(c_\lambda)^d || \lambda$ by using $V_i's$ public key pk_{V_i}
Generate $PC_{(i,k)}$ via (3,5) secret sharing scheme

Figure 4. Polling phase.

3.2.4. Opening Phase

After the voting period has ended, the *SC* informs all the voters that it is about to open and count the ballots. The procedures of this process are as follows:

1. The *AS* proceeds as follows:
   (a) Publish the V_i private key sk_{V_i} and random number $r_j, 1 \leq j \leq n_2$ selected in the voting phase on *SC*.
2. The *RC* proceeds as follows:
   (a) Publish its own private key sk_{RC} onto the *SC*.
3. V_i proceeds as follows:
   (a) Use sk_{RC} to decrypt all ciphertext coordinates, $CC_{(i,k)} = (E_{pk_{RC}}(x_k, E_{pk_{V_i}}(y_k))), 1 \leq k \leq 5$, published by the *DDS*.
   (b) Set $PC_{(i,k)} = (x_k, y_k), 1 \leq k \leq 5$.
   (c) Use sk_{V_i} to decrypt all C_i and thereby obtain the ballot containing the candidate number λ selected by V_i and the ballot signature.
   (d) Use m_j and random number $r_j, 1 \leq j \leq n_2$ to verify the ballot signatures, and determine whether the value of λ is consistent with the information on the ballot signatures.

3.2.5. Checking Phase

If a mistake occurs, all voters reserve the right to ask to review their ballot. The verification process uses sk_{V_i} and sk_{RC} to decrypt the ballots and thus confirm that the signature and candidate number is correct.

3.3. Experiment Setting

3.3.1. Hardware/Software

- Processor: 2.4-GHz Intel Core i5
- Memory: 4-GB 1600-MHz DDR3
- Operating system: OS X EI Capitan 10.11.4
- Python 2.7.10

- Node. JS 3.10.10
- Solidity 0.4.0

3.3.2. System Parameters

- Digital signature module: RSA-512 bits/RSA-1024 bits
- Encryption module: Paillier-512 bits
- Decryption module: Paillier-1024 bits
- Secret sharing scheme: Shamir's (k, n) secret sharing scheme
- 5 voters, 5 candidates, and 5 DDSs

3.3.3. Operating Procedures

1. Voters should enter their social security number to obtain their *PID*, which is generated by the RS; the RS then records it in the SC.
2. Voters should enter their *PID* to obtain their voting certificate, which is generated by the voting website; the voting website then records it in the SC.
3. Voters should enter their voting certificate and candidate number to obtain their ballot signature.
4. Voters should enter their *PID*, ballot signature, voting certificate, and candidate number to be able to vote. The voting website records the poll results ($PID||$BallotInfoSecretSharing) in the SC.
5. When voters want to review the number of votes currently completed, they can click the check button on the page to retrieve data from the SC.
6. After voters have learned the number of voters from Step 5, they can click the check button to obtain the *PID* list of voters who have voted.
7. Voters can click the check button on the page to review the ballot information if they so desire.
8. Voters can review the ballot information in the SC at the billing stage; the back-end program will decrypt and recover the ballot and then return the election results for voters to view on the page.

3.3.4. Experimental Results

Throughout the experiment, we used RSA-512 bits/Paillier-512 bits and RSA-1024 bits/Paillier-1024 bits for performance comparison. As indicated in Table 2, when on of the bit numbers is twice as large as the other, it grows five-times larger during encryption and decryption and ten-times larger during secret sharing. Therefore, we can deduce that when the number of bits greater, the time required is also greater.

Table 2. Performance comparison.

	Encryption and Decryption	Secret Sharing
512 bits	18 s	8 s
1024 bits	90 s	75 s

Because the voting phase contains encryption and decryption, the secret sharing scheme and other processes require a more sophisticated computational algorithm. We compared the following two scenarios to identify the method with favorable performance:

- Scenario 1: Conduct secret sharing on the ballot first, followed by encryption
- Scenario 2: Encrypt the ballot first, and then conduct secret sharing

The n values represent the number of servers. In Scenario 1, the input bit of secret sharing is the sum of the binary value of candidates and the security parameter. In Scenario 2, the input bit of secret sharing is only the security parameter. In our experiment, the number of servers was five, meaning that the value of n was also five. Because the number of candidates was five, the security parameter was 512 or 1024. Therefore, the input bits of secret sharing were 515 or 1027 in Scenario 1 and 512 or 1024 in Scenario 2. Table 3 demonstrates that both the numbers of encryptions, decryptions, and input bits of secret sharing in Scenario 2 were all less than those in Scenario 1; we thus inferred that Scenario 2 outperformed Scenario 1 and consequently adopted Scenario 2 as our experimental model.

Table 3. Scenario analysis.

	Scenario 1	**Scenario 2**
Encryption	$2n$	$n+1$
Decryption	$2n$	$n+1$
Secret sharing	1	1
Recover secret	1	1

4. Proposed E-Bidding System

The study combined an SC with privacy-protection cryptography to produce a distributed electronic bidding system that enhances bidding efficiency and enables vendors to participate in the opening phase. Because the information on the blockchain is completely transparent and public, the bidding documents of vendors must be fully confidential before the opening phase begins. Table 4 provides the system description, and Figure 5 displays the system architecture. Notably, all transmitted information is stored in the SC for further verification.

Table 4. Description of our E-bidding system.

Term	**Description**
V_i	Voters with voting qualification
GCA	Government Certificate Authority
$Bank$	Financial authority
TA	Tender authority
$TWeb$	Tender website
SC	Smart contract
$BCert_{V_i}$	V_i's bid bond certificate
$Cert_{V_i}$	V_i's tender certificate
$GCert_{V_i}$	V_i's corporate certificate
PID_i	V_i's personal identification code
SSN_{V_i}	V_i's social security number

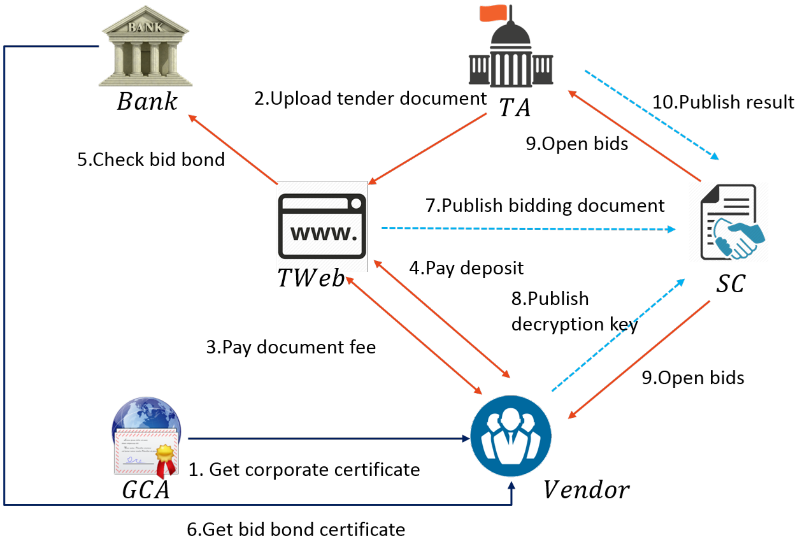

Figure 5. Architecture of our E-bidding system.

4.1. System Definition

The following describes the six roles of the system:

1. Vendor (V_i): V_i is vendor i with a bidding qualification. After undergoing identity verification through GCAto obtain a tender certificate ($Cert_{V_i}$), the vendor pays for bid bonds to obtain a bid bond certificate ($BCert_{V_i}$). After following the aforementioned steps, the vendor can then request a personal identification code (PID_i) from the SC. After bidding, vendors can review their bidding documents, which have been published on the *SC*, to confirm that their bidding has been correctly counted. If the bidding participants discover that the bidding has not been properly counted for any reason, the vendor should respond to the tender authority immediately, confirm that the certificate is abnormal, and determine whether the vendor should bid again.
2. Government Certification Administration Center (*GCA*): The *GCA* is responsible for verifying the identity of vendors and issuing corporate certificates ($GCert_{V_i}$) to legal vendors. Vendors can then review their PID_i, which is published on the SC, to confirm that their corporate certificate is correct.
3. Financial authority (*Bank*): The bank is responsible for confirming that the amount of the bid bond (replaced by the deposit) is correct; if it is correct, then the bank issues a bid bond certificate ($BCert_{V_i}$). Vendors can then review the PID_i, which is published on the SC, to confirm that their bid bond certificate is correct.
4. Tender authority (*TA*): The TAis responsible for writing the tender documents and publishing them on the tender website. The tender authority and vendors open the bid together, and the bidding result is announced by the tender authority, who also sends a hard copy of the letter to notify the winning vendor. Vendors who participate in the bidding, with the exception of the winning vendor, can ask the tender authority to return their bid bond by presenting their bid bond certificate.
5. Tender website (*TWeb*): The TWeb is responsible for transferring deposits from vendors to the financial authority. When vendors complete bidding, the TWeb encrypts their bidding documents C_{V_i}) and publishes them in the SC.
6. Smart contract (*SC*): The SCreplaces the function of a traditional bulletin board. It is dynamic, enables vendors to verify their bidding information, and increases the credibility of the bidding and the confidence of vendors.

4.2. Processes and Steps

This section provides further explanation of the E-bidding process; our system included n_1 vendors, 1 *bank*, 1 *GCA*, 1 *SC*, and 1 tender authority. Additionally, all transmission processes were performed through a secure channel. The bidding process comprised seven phases: (1) initial phase, (2) invitation phase, (3) obtaining phase, (4) bidding phase, (5) opening phase, (6) decision phase, and (7) contract management phase.

4.2.1. Initial Phase

Before implementing the protocol, the *Bank* and *GCA* must generate their RSA-based signature key pair $(e', N_1)/(d', N_1)$ and $(e, N_2)/(d, N_2)$, respectively, in which (d, N_2) and (d', N_1) are the signing key and (e, N_2) and (e', N_1) are the public key used for signature verification. Vendors must generate its encryption and decryption key pair pk_{V_i}, sk_{V_i}.

1. The process of generating the signature key of the *Bank* is as follows:
   (a) Select two sufficiently large prime numbers p_1 and q_1.
   (b) Compute $N_1 = p_1 \cdot q_1$.
   (c) Define $\varphi(N_1) = \varphi(p_1)\varphi(q_1) = (p_1 - 1)(q_1 - 1)$.
   (d) Select a value e' that satisfies $GCD(e', \varphi(N_1)) = 1$.

   (e) Identify a value d' that satisfies $e'd' \equiv 1 \bmod \varphi(N_1)$.

2. The process of generating the signature key of the *GCA* is as follows:
   (a) Select two sufficiently large prime numbers p_2 and q_2.
   (b) Compute $N_2 = p_2 \cdot q_2$.
   (c) Define $\varphi(N_2) = \varphi(p_2)\varphi(q_2) = (p_2 - 1)(q_2 - 1)$.
   (d) Select a value e that satisfies $GCD(e, \varphi(N_2)) = 1$.
   (e) Identify a value d that satisfies $ed \equiv 1 \bmod \varphi(N_2)$.

3. The process of generating the key pair for V_i is as follows:
   (a) Select two sufficiently large prime numbers p_3 and q_3.
   (b) Compute $N_3 = p_3 \cdot q_3$.
   (c) Define $\varphi(N_3) = \varphi(p_3)\varphi(q_3) = (p_3 - 1)(q_3 - 1)$.
   (d) Select a value pk_{V_i} that satisfies $GCD(pk_{V_i}, \varphi(N_3)) = 1$.
   (e) Identify a value sk_{V_i} that satisfies $pk_{V_i}sk_{V_i} \equiv 1 \bmod \varphi(N_3)$.

4.2.2. Tender Inviting Phase

In this stage, the *GCA* confirms the identity of vendors and sends corporate certificates $GCert_{V_i}$, for which $1 \leq i \leq n_1$, to legal vendors. The procedures are conducted offline.

1. The user code is generated for each V_i as follows:
   (a) Select a random number $t \in \mathbb{Z}_N^*$.
   (b) Generate its personal identification code $PID_i = h(SSN_{V_i} \| t)$.
   (c) Send PID_i to *GCA* for verification.

2. To verify the identity of V_i, for which $1 \leq i \leq n_1$, *GCA* proceeds as follows:
   (a) Confirm the correctness of PID_i.
   (b) Issue a bidding certificate $GCert_{V_i} = \{PID_i, Sig_d(PID_i)\}$ to V_i if PID_i is correct. The certificate is the signature of the PID_i signed by *GCA*.
   (c) Publish the PID_i of legal vendors on the bulletin board. In this system, the SC serves as the bulletin board.

3. To upload tender documents, the *TA* proceeds as follows:
   (a) Upload the tender document to *TWeb*.

4.2.3. Tender Obtaining Phase

Vendors can browse through various cases announced by the tender authority in the government procurement tender system; vendors can also download and read complete tender documents if they pay the tender documentation fee. The tender documents are confidential, but vendors who pay the tender documentation fee are permitted to browse through them.

4.2.4. Tender Submitting Phase

Vendors must first pay a deposit and obtain a certificate of deposit. They then follow instructions to complete the tender document and use their own private key to sign the bidding document to prove that the document has been completed by them. Because the use of digital signatures can help the bidder to verify the correctness of information on the document, the bidding documents completed by vendors are included in the digital signature of vendors.

1. $V_i, 1 \leq i \leq n_1$ proceeds as follows:
   (a) Transmit the corporate certificate $GCert(V_i)$ and deposit to the bank for verification of the correct deposit amount.

2. The bank proceeds as follows:
   (a) Verify the validity of $GCert_{V_i}$.
   (b) Verify the correctness of the deposit amount.
   (c) Issue a certificate of deposit $BCert_{V_i} = \{GCert_{V_i}, Sig_{d'}(PID_i)\}$ to V_i if both $GCert_{V_i}$ and the deposit amount are valid.
3. V_i, for which $1 \leq i \leq n_1$, proceeds as follows:
   (a) Encrypt document $C_{V_i} = E_{pk_{V_i}}(PID_i \| GCert_{V_i}) \| BCert_{V_i} \| betting_i)$.
   (b) Upload bidding document C_{V_i} onto the TWeb.

   After completing the aforementioned steps, vendors pay the deposit and upload their bidding document to the tender website.

4.2.5. Tender Opening Phase

The bidder uses the public keys of each vendor, the bank, and the *GCA* to verify the validity of the bidding documents, the corporate certificate, and the certificate of deposit. After confirming their validity, vendors begin bidding. After bidding, all V_i, for which $1 \leq i \leq n_1$, are notified through the SC that bids are ready to be recorded. The procedures are as follows.

1. V_i proceeds as follows:
   (a) Publish private key sk_{V_i} onto the *SC*.
   (b) Decrypt bidding documents C_{V_i} individually.
   (c) Verify the correctness of the bidding document C_{V_i} using the public key of the V_i.
   (d) Verify the correctness of the corporate certificate $GCert_{V_i}$ using the public key of the *GCA*.
   (e) Verify the correctness of the certificate of deposit $BCert_{V_i}$ using the public key of the *Bank*.
   (f) Bid on all vendor betting.
2. The *TA* proceeds as follows:
   (a) Compare the betting of vendors and identify the winner; if no objections arise after the opening phase, then the winner will be notified by the *TA*.

4.2.6. Tender Deciding Phase

This phase determines who the winner is, and the winner is notified by the *TA*. The vendor must fully accept the result of the electronic notice. If the winner retracts his/her bid at this time, then the bid bond is confiscated by the *TA*.

4.2.7. Contract Management Phase

Vendors participating in the bidding, with the exception of the winning vendor, can ask the tender authority to return the bid bond by presenting his/her bid bond certificate.

4.3. Experiment Setting

4.3.1. Hardware/Software

- Processor: 2.4-GHz Intel Core i5
- Memory: 4-GB 1600-MHz DDR3
- Operating system: OS X EI Capitan 10.11.4
- Python 2.7.10
- Node. JS 3.10.10
- Solidity 0.4.0

4.3.2. System Parameters

- Digital signature module: RSA-1024 bits
- Encryption/decryption module: RSA-2048 bits
- 1 TA, 1 Bank, 1 GCA, 1 SC, and n_1 vendors

4.3.3. Operating Procedures

1. Vendors must pay the tender documentation fee to view the tender documents.
2. Vendors should enter their social security number to obtain their PID, which is generated by the GCA. The GCA then records it in the SC.
3. Vendors should enter their PID_i to obtain a corporate certificate $GCert_{V_i}$, which is generated by the GCA.
4. Vendors must pay a deposit and obtain a certificate of deposit $BCert_{V_i}$ to bid.
5. Vendors must submit their PID_i, $GCert_{V_i}$, $BCert_{V_i}$, and $betting_i$ to bid. The bidding document $C_{V_i} = (PID_i \| GCert_{V_i} \| BCert_{V_i} \| betting_i)$ is recorded in the SC by the tender website.
6. If bidding participants want to review the current number of bids completed, they can click the check button on the web page to retrieve data from the SC.
7. After identifying the number of vendors from Step 6, bidding participants can click the check button to obtain the PID list of vendors who have completed bidding.
8. Bidding participants can click the check button on the web page to review the bidding information if they desire.
9. Bidding participants can decrypt bidding information in the SC in the opening phase; the backend program decrypts and recovers the bidding document and then returns the bidding results to the page for participants.

4.3.4. Experimental Results

- Performance: Throughout the experiment, the digital signature module used RSA-1024 bits; the encryption and decryption module used RSA-2048 bits. Both modules in the implementation were highly time-efficient, requiring less than one second.
- Contract information: Contracts must record the identity of each vendor's PID_i, corporate certificate $GCert_{V_i}$, certificate of deposit $BCert_{V_i}$, and bid amount $betting_i$ for bidding participants to query. In the opening phase, the contract is used to publish the private key associated with the decryption.

In our experiment, we did not transfer payments from vendors to the bank or download the tender document; rather, bidding documents were encrypted and stored in the SC to enable all participants involved in the bid to not only confirm, but also compute the bid price in the opening phase.

5. Security Analysis

This section provides a brief description and analysis of the proposed E-voting and E-bidding systems. We firstly describe how the blockchain satisfies public verifiability, nonrepudiation, and being non-tamperable. Then, we show that the correctness of our scheme, the privacy of AS, and the privacy of each voter V_i.

5.1. Blockchain Technology

Blockchain technology has the following three security properties:

- Public verifiability: Each valid node on the Bitcoin network can confirm and verify all content in the block.

- Nonrepudiation: Because the blockchain maintains a public and transparent ledger, each valid node on the Bitcoin network can confirm and verify all content in the block, which demonstrates the characteristic of nonrepudiation.
- Non-tamperable: Because the blockchain maintains a public and transparent ledger, each valid node on the Bitcoin network can confirm and verify all the content in the block. Each block contains the hash value of the previous block. Therefore, when a transaction within a block is tampered with, all subsequent blocks are affected, requiring manipulators to possess a large amount of computing power and thus rendering this task remarkably difficult to complete. That is, tampering with the content in the block is virtually impossible.

This study replaced traditional bulletin boards with *SC*s, which are based on blockchain technology. The nodes on the blockchain network participated in verification and calculation to increase user anonymity, the privacy of data transmission, and the trustworthiness and verifiability of the opening phase.

5.2. Correctness and Privacy

- Correctness: When voters V_i receive $M_{(i,j)}, 1 \leq i \leq n_1, 1 \leq j \leq n_2$, they use their private key sk_{V_i} to open $M_{(i,j)}$ and obtain n_2 ciphertexts. They then use the public key of AS e to verify ciphertext λ of the signature, in which λ is the number of voter signatures obtained during the voting phase.
- Privacy of AS: The privacy of AS ensures that the voter V_i can only obtain one signature λ of ballot $h(c_\lambda)^d$ and have no knowledge regarding other $n_2 - 1$ ballot signatures. Suppose that V_i is a semi-honest voter who attempts to obtain the other $n_2 - 1$ signatures of the ballot. Because $h(\lambda) \neq h(\lambda_i)$ and the content of the ciphertext contain the random number k_i selected by AS, V_i cannot obtain the signature of any other ballot except signature λ, thus protecting the privacy of AS.
- Privacy of voter V_i: The privacy of voter V_i ensures that AS cannot obtain any information of the ballot chosen by voter V_i. In the voting phase, voter V_i selects a ballot number λ and encrypts it to $E_{pk_{V_i}}(h(\lambda))$ by the Paillier cryptosystem. Then, voter V_i sends it to the AS. Even the AS knows that the range of λ is from 1–n_2 and can thus also compare the ciphertext after encrypting from 1–n_2. However, it still cannot identify the corresponding ballot because the Paillier encryption process for each c_i contains the random number $r_j \in \mathbb{Z}_N^*$ selected by V_i, thus protecting the privacy of V_i.

5.3. Properties of the E-Voting System

This section uses the following six properties to analyze the security of the E-voting system. The security of our work is mainly based on blockchain technology and the oblivious transfer scheme:

- Eligibility: This property ensures that only voters with legal voting qualifications can vote to protect the fairness of the voting process. In the registration phase of our scheme, RS verifies the identity of V_i, and only those who pass the identity verification stage can obtain a voting certificate $Cert_{V_i}$.
- Non-repeatability: This property ensures that each voter is limited to one vote, and it is forbidden to repeat the vote or other malicious votes, or to protect the fairness of the voting process. In our scheme, people must present their own voting certificate $Cert_{V_i}$ of the voting stage. The RC first determines whether the certificate is legal. If it is legal, then the process of storing the ballot is completed, and the certificate is marked as voted. When voters attempt to vote more than once, the RC refuses the repeated vote to prevent ballot stuffing. Because the RC is semi-honest, the stored $Cert_{V_i}$ will not be tampered with or imitated. Therefore, when voters attempt to vote more than once, the RC can carefully inspect the result and send it to the *VWeb*.
- Rationality: This property ensures that no internal or external attackers or voters have the opportunity to tamper with other people's votes maliciously, thereby ensuring the legitimacy of the voting process. In our scheme, voter ballot information was directly recorded in the SC

through the voting website and related applications for access and inquiry. No one can tamper with the ballot information because the blockchain is undeniably difficult to manipulate, thus ensuring that the rationality of the voting process is maintained.

- Completeness: This property ensures that each voter can check whether the ballot information is correctly counted and checked at the billing stage. In our scheme, during the ballot opening phase, all voters can verify the ballot information independently, and the ballot information is stored in the server in a decentralized environment for candidate number λ to determine the number of votes that they have received and the ballot signature value $h(c_\lambda)^d$. By evaluating the information on the ballot signature, voters can verify that their ballots have been correctly recorded. If the result is incorrect, then they can report it for further verification.
- Fairness: This property ensures that no internal or external attackers or voters can know the election trend and results before the billing stage, thereby ensuring the fairness of the voting process. In our scheme, because the key pairs of voters pk_{V_i}, sk_{V_i} were issued by RS, if the ballots were encrypted only by these keys, then RS can ascertain the election results before the opening phase. Therefore, after completing encryption using the public key of the voter pk_{V_i}, a second encryption is performed using the public key of the RC pk_{RC} to prevent ballot information from being decrypted by attackers attempting to obtain the election results in advance. When voters complete the polling process, the ballot $h(c_\lambda)^d \| \lambda$ is first encrypted by the voter's public key pk_{V_i} and then saved to distributed data servers through (k, n) secret sharing. After receiving the encrypted ballots, the DDS servers then encrypt the ballots again using the public key of the RC pk_{RC} and record $E_{pk_{RC}}(PC_{i,k})$ in the SC for voter query. Based on the oblivious transfer protocol and privacy-protection cryptography, neither internal nor external attackers can discover the results of the election in advance by decrypting the ballots or retrieving data from the server side; this preserves the fairness of the election.
- Anonymity: This property ensures that no internal or external attackers or voters can know which voter information actually corresponds to each voter, thereby protecting the confidentiality and security of the voter identity. In our scheme, when voters create their personal identity, they will combine their social security number with a random number they have selected to generate their unique identity $PID_i = h(SSN_{V_i} \| t), 1 \le i \le n_1$. No one can connect the PID_i to the real person V_i because no one has knowledge of this system, thus protecting voter anonymity.

A comparison of the study and related E-voting mechanisms is as follows (Table 5).

Table 5. Comparison of security properties.

	Johnson's [7]	**Nair's [12]**	**Chen's [11]**	**Our's**
Eligibility	✓	✗	✓	✓
Non-repeatability	✓	✗	✓	✓
Rationality	✓	✓	✓	✓
Completeness	✗	✗	✓	✓
Fairness	✓	✓	✗	✓
Anonymity	✓	✗	✓	✓

5.4. Properties of the E-Bidding System

This section provides a brief description and analysis of the proposed E-bidding system:

- Eligibility: The GCA first verifies the identity of V_i, who must obtain a corporate certificate issued by the GCA to obtain tender qualification.
- Nonrepudiation: Because bidding documents are stored in the SC and the blockchain has the property of nonrepudiation, they cannot be tampered with.
- Public verifiability: All vendors who participated in bidding can verify their corporate certificates $GCert_{V_i}$, bid bond certificates $BCert_{V_i}$, and bidding amount $betting_i$ of corresponding bidding

documents in the opening phase. If they pass the verification stage, then they can participate in the price competition; otherwise, they will not be permitted to compete.

- Secrecy of bidding price: bidding documents are encrypted using the vendors's public key pk_{V_i} to prevent anyone from obtaining the bidding result before the opening phase.

6. Conclusions

This study aimed to design a distributed E-voting and E-bidding system that is different from the currently-used electronic system, which still employs a trusted third party, by replacing the third party with an SC, which has public and transparent properties. The core idea is to combine blockchain technology with privacy-protection cryptography to enable all participants who used the application to be involved in the opening phase.

Because we adopted Shamir's secret sharing scheme, we discovered that first encrypting the ballot and then conducting secret sharing resulted in favorable performance compared with first conducting secret sharing and then encryption. For this reason, we adopted Scenario 2 for our experimental model. However, the order differs for different secret sharing schemes. Therefore, future research must analyze numerous secret sharing schemes to identify the optimal solution.

To satisfy the security requirements for electronic applications, this study enabled all participants who used the application to be involved in the opening phase. Additionally, when a dispute was encountered, the traditional mechanism requires a substantial amount of processing time; conversely, our system enables the court to extract the entire vote and tender information directly from the blockchain, thereby improving the efficiency of the E-voting and E-bidding systems.

Author Contributions: R.T. and J.-H.H. developed the idea, protocol design, and wrote the original draft. Z.-Y.L. improved the scheme, provided useful advice, and performed the experiment. All authors discussed the results and contributed to the final manuscript.

Funding: This research was funded by the Ministry of Science and Technology, Taiwan (ROC), under Project Numbers MOST 107-2218-E-004-001, MOST 105-2221-E-004-001-MY3, and by Taiwan Information Security Center at National Sun Yat-sen University (TWISC@NSYSU).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Dolev, S.; Gilboa, N.; Kopeetsky, M. Computing Multi-party Trust Privately: In O(N) Time Units Sending One (Possibly Large) Message at a Time. In Proceedings of the 2010 ACM Symposium on Applied Computing (SAC '10), Sierre, Switzerland, 22–26 March 2010; ACM: New York, NY, USA, 2010; pp. 1460–1465. [CrossRef]
2. Dimitriou, T.; Michalas, A. Multi-Party Trust Computation in Decentralized Environments. In Proceedings of the 2012 5th International Conference on New Technologies, Mobility and Security (NTMS), Istanbul, Turkey, 7–10 May 2012; pp. 1–5. [CrossRef]
3. Dimitriou, T.; Michalas, A. Multi-party trust computation in decentralized environments in the presence of malicious adversaries. *Ad Hoc Net.* **2014**, *15*, 53–66. [CrossRef]
4. Kshetri, N.; Voas, J. Blockchain-Enabled E-Voting. *IEEE Softw.* **2018**, *35*, 95–99. [CrossRef]
5. Andrew, B.; Christopher, B.; Thomas, P. Digital Voting with the Use of Blockchain Technology. Available online: https://www.economist.com/sites/default/files/plymouth.pdf (accessed on 18 December 2018).
6. Agora. Agora Whitepaper. Bringing Our Voting Systems into the 21st Century. Available online: https://static1.squarespace.com/static/5b0be2f4e2ccd12e7e8a9be9/t/5b6c38550e2e725e9cad3f18/1533818\968655/Agora_Whitepaper.pdf (accessed on 17 February 2019).
7. Johnson, N.; Jones, B.M.; Clendenon, K. e-Voting in America: Current Realities and Future Directions. In *Social Computing and Social Media. Human Behavior*; Meiselwitz, G., Ed.; Springer International Publishing: Cham, Switzerland, 2017; pp. 337–349.
8. Liao, T.; Wang, M.; Tserng, H. A Framework of Electronic Tendering for Government Procurement: A Lesson Learned in Taiwan. *Autom. Constr.* **2002**, *11*, 731–742. [CrossRef]

9. Buccafurri, F.; Fotia, L.; Lax, G. Allowing Continuous Evaluation of Citizen Opinions through Social Networks. In *Advancing Democracy, Government and Governance*; Kő, A., Leitner, C., Leitold, H., Prosser, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 242–253.
10. Buccafurri, F.; Fotia, L.; Lax, G. Allowing Non-identifying Information Disclosure in Citizen Opinion Evaluation. In *Technology-Enabled Innovation for Democracy, Government and Governance*; Kő, A., Leitner, C., Leitold, H., Prosser, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 241–254.
11. Chen, Y.S.; Tso, R. *An E-Voting System Based on Oblivious Signatures*; National Chengchi University: Taipei, Taiwan, 2013.
12. Nair, D.G.; Binu, V.P.; Kumar, G.S. An Improved E-Voting Scheme Using Secret Sharing Based Secure Multi-Party Computation. CoRR 2015, abs/1502.07469. http://xxx.lanl.gov/abs/1502.07469 (accessed on 25 March 2019).
13. McCorry, P.; Shahandashti, S.; Hao, F. A Smart Contract for Boardroom Voting with Maximum Voter Privacy. In *Financial Cryptography and Data Security*; Springer: Berlin/Heidelberg, Germany, 2017.
14. Hjalmarsson, F.; Hreiarsson, G.K.; Hamdaqa, M.; Hjalmtysson, G. Blockchain-Based E-Voting System. In Proceedings of the 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), San Francisco, CA, USA, 10–13 December 2018; pp. 983–986. [CrossRef]
15. Rabin, M.O. *How To Exchange Secrets with Oblivious Transfer*; Harvard University Technical Report 81 talr@watson.ibm.com 12955 received 21 Jun 2005; Harvard University Press: Cambridge, MA, USA, 2005.
16. Paillier, P. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In *Advances in Cryptology—EUROCRYPT '99*; Stern, J., Ed.; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.
17. Yi, X.; Bouguettaya, A.; Georgakopoulos, D.; Song, A.; Willemson, J. Privacy Protection for Wireless Medical Sensor Data. *IEEE Trans. Depend. Secur. Comput.* **2015**, *13*, 369–380. [CrossRef]
18. Ugus, O.; Westhoff, D.; Laue, R.; Shoufan, A.; Huss, S.A. Optimized Implementation of Elliptic Curve Based Additive Homomorphic Encryption for Wireless Sensor Networks. CoRR 2009, abs/0903.3900. http://xxx.lanl.gov/abs/0903.3900 (accessed on 5 October 2018).
19. Shamir, A. How to Share a Secret. *Commun. ACM* **1979**, *22*, 612–613. [CrossRef]
20. Corniaux, C.L.F.; Ghodosi, H. A Verifiable Distributed Oblivious Transfer Protocol. In *Information Security and Privacy*; Parampalli, U., Hawkes, P., Eds.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 444–450.
21. Chu, C.K.; Tzeng, W.G. Efficient K-out-of-N Oblivious Transfer Schemes with Adaptive and Non-adaptive Queries. In Proceedings of the 8th International Conference on Theory and Practice in Public Key Cryptography (PKC'05), Janeiro, Brazil, 25–29 March 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 172–183. [CrossRef]