Blockchain-Based Sealed-Bid Domain Name Auction Protocol

Genhua Lu[1], Yi Zhang[1], Zhongxiang Lu[2], Jun Shao[1](✉), and Guiyi Wei[2]

[1] School of Computer and Information Engineering, Zhejiang Gongshang University,
Hangzhou, China
[2] School of Information and Electronic Engineering, Zhejiang Gongshang University,
Hangzhou, China
weigy@zjgsu.edu.cn

Abstract. Domain name system (DNS), mapping domain names to IP addresses, is critical to the Internet's running. However, the centralized architecture is one of the major criticisms of the current DNS. Many works suggest introducing the blockchain into DNS, but the existing blockchain-based DNSs do not support the domain name auction that is important for the domain name transfer. To solve this problem, we in this paper propose a blockchain-based sealed-bid domain name auction protocol by combining the smart contract, the Pedersen commitment, and zero-knowledge proof. Compared with the previous blockchain-based auction protocols, our proposal is the first one holding the tx-fairness, bidding-fairness, bid-guarantee, and fund-privacy at the same time. For showing its effectiveness, we also give an illustration of our proposal based on the smart contract system in Ethereum.

Keywords: DNS · Blockchain · Sealed-bid auction

1 Introduction

Domain Name System (DNS) is one of the crucial infrastructures of the Internet. With its help, people can use the human-readable domain name to visit websites instead of the hard-to-remember digital IP address [1]. The fundamental part of DNS is the management of root names and top-level domain names, which is principally maintained by the Internet Corporation for Assigned Names and Numbers (ICANN) [2]. This centralized architecture of the current DNS suffers from many attacks [3], such as the single-point failure and power abuse. One of the famous incidents of single-point failure on the current DNS is the 2016 Dyn cyberattack [4]. The failure of DNS provided by Dyn caused major websites, including Amazon.com, GitHub, Twitter, and Reddit, to convert unreachable

This work was supported by National Key Research and Development Program of China under grant number: 2019YFB1804500.

B. Chen and X. Huang (Eds.): AC3 2021, LNICST 386, pp. 25–38, 2021.
https://doi.org/10.1007/978-3-030-80851-8_3

via their corresponding domain names. To mitigate the above problems, it is natural to introduce the decentralized architecture into the current DNS.

Meanwhile, with over ten years of development, the blockchain has become the most famous and successful decentralized architecture. Recently, many researchers have suggested introducing blockchain into the DNS [5–10]. On the other hand, the domain name auction is one of the primary ways to obtain the DNS domain name [11]; however, the existing blockchain-based DNS solutions cannot provide the functionality of domain name auction as well as expected.

- Namecoin [5], ConsortiumDNS [6], EmerDNS [7], and Blockstack [8] mainly focus on domain name registration but ignoring the domain name auction.
- Ethereum Name Service (ENS) [9] provides the auction functionality, but it is only for cryptocurrency addresses, not IP addresses.
- Handshake [10] realizes a sealed-bid domain name auction, where the bid will not be revealed until all the bids are committed. The main advantage of a sealed-bid domain name auction is that bidders are more willing to bid according to the real value of the auction item [12]. However, the last bidder in Handshake can always win the auction with a reasonable bid since the current possible highest bid is predictable.

In this paper, aiming at the above challenges, we propose a sealed-bid blockchain-based domain name auction protocol to realize the domain name transfer in a fair, secure, and privacy-preserving way. The main contributions of this paper can be summarized as follows.

- We compile the security requirements for the blockchain-based sealed-bid domain name auction for the first time, including the tx-fairness, bidding-fairness, bid-guarantee, and fund-privacy.
- By integrating the account-based consortium blockchain with the anonymous fund, smart contract, the Pedersen commitment, and zero-knowledge proof, we propose the first blockchain-based sealed-bid domain name auction protocol.
- We also give a security analysis to show our proposal satisfies the tx-fairness, bidding-fairness, bid-guarantee, and fund-privacy simultaneously.
- At last, we give an illustration of our proposal by using the smart contract system in Ethereum to show the feasibility of our proposal.

The remainder of this paper is organized as follows. In Sect. 2, we give the system and security models, as well as the design goals in this paper. In what follows, we give some basic knowledge related to our proposal. After that, we present our domain name auction protocol and its analysis in Sect. 4 and Sect. 5, respectively. Section 6 reviews the related works. At last, we conclude this paper in Sect. 7.

2 Models and Design Goals

This section will present the system and security models for our proposed blockchain-based domain name auction protocol and identify the properties our proposal should have.

2.1 System Model

In our system model, we mainly consider a typical domain name auction scenario, where we have a domain name owner, multiple domain name bidders, and a blockchain system, as shown in Fig. 1.

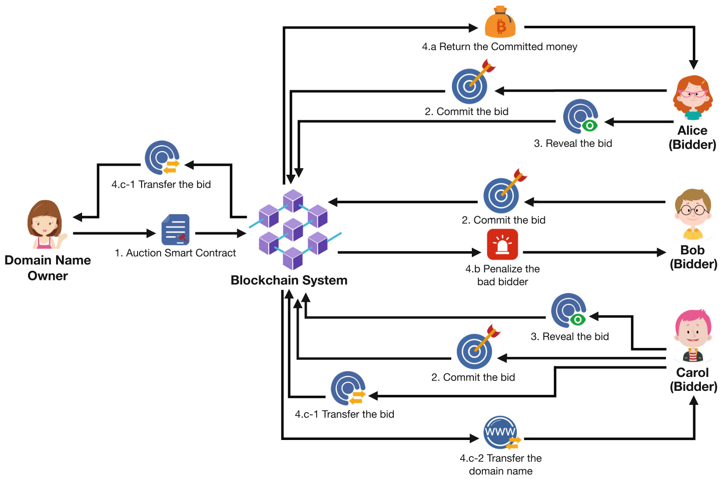

Fig. 1. The system model considered in this paper.

Our system is heavily based on the blockchain system that is an account-based consortium blockchain. The consensus nodes in the blockchain system could be the network authority in each government around the world. In other words, all the countries together, instead of the single entity ICANN alone, maintain the mapping from domain names to IP addresses. Hence, our system overcomes the problems caused by the centralized architecture.

All the participants in the blockchain system have at least one account. Every account has its corresponding fund of an encrypted format, such as $\mathsf{fund} = g^c h^r$, where c is the amount of the fund, r is a random number, and g and h are random elements from the underlying finite cyclic group. Furthermore, the blockchain also records the relationships between the domain name and the account.

When the domain name owner wants to sell his/her domain name, he/she deploys a smart contract in the blockchain. The domain name bidder can bid the domain name in a sealed-bid way, particularly the Vickrey auction. The Vickrey auction allows the bidder to submit the bid and keep the bid secret before all bids are submitted, and the winning bidder pays the second-highest bid instead of his/her bid. At the end of the auction, the domain name is transferred from the domain name owner to the winning bidder.

More detailed information about the underlying blockchain system and auction types can be found in Sect. 3.1 and Sect. 3.2, respectively.

2.2 Security Model

As with other blockchain-based systems [5–10], the blockchain in our system is assumed to be honest-but-curious. In particular, the blockchain system will faithfully execute the smart contract deployed by the domain name owner. However, it is also curious about others' secret information, including bids' values before the corresponding bidders open them and the values of funds the bidders have.

The domain name owner could be malicious. He/she will try his/her best to transfer to others the domain names not belonging to him/her or obtain the money from the bidder without transferring the domain name. The domain name bidders could also be malicious. They would launch active attacks, such as manipulating the communication data, to win the auction deviating from the rules of Vickrey auction or to obtain the domain name without payment.

2.3 Design Goal

This paper aims to design a blockchain-based domain name auction protocol with the following security properties.

TX-Fairness. The proposal should guarantee that once the domain name owner transfers the domain name to the winning bidder, he/she can obtain the corresponding money; vice versa.

Bidding-Fairness. The winning probability of the auction is not related to the time of bidding. In particular, the late bidding will not bring any advantage over the early bidding.

Bid-Guarantee. Anyone can verify that the bidder has enough money to cover his/her bid or not.

Fund-Privacy. No one can deduce how much money the bidder has in the blockchain system from the auction process.

3 Preliminaries

In this section, we will review some basic knowledge that will be used in our proposal, including the account-based consortium blockchain, auction types, commitment scheme, and zero-knowledge proof for an inequality.

3.1 Account-Based Consortium Blockchain

Blockchain is a technology enabling a group of users to build trust relationships in a decentralized way. Since the invention of the first blockchain system—Bitcoin [13], many blockchain systems have been proposed with different properties [14–16]. Generally speaking, the existing blockchain systems can be classified into three categories: public blockchain, consortium blockchain, and private blockchain, where anyone, only a specific group of users, and only a particular user can be the consensus nodes appending data to the blockchain, respectively. Unlike the existing blockchain-based DNS solutions [8,10,12], we adopt the consortium blockchain in our system due to its effectiveness and efficiency. Furthermore, we assume that the consensus nodes in our system are the network authorities in governments, which may be more realistic.

The consortium blockchain used in this paper belongs to the account-based style. Every user in this blockchain system has one or more accounts, and each account is recorded with its possessed fund and domain names. The fund recorded in the blockchain is with an encrypted format, such as $\mathsf{fund} = g^c h^r$, where c is the value of the fund, r is a random number, g and h are random elements in the underlying finite cyclic group, and no one knows $\log_h g$. In contrast, the domain names are recorded in plaintext format since they will be transferred from one account to another account through the sealed-bid auction.

At last, the consortium blockchain also supports smart contracts, which has been a fundamental functionality of a blockchain system since the invention of Ethereum [17]. Particularly, our consortium blockchain's consensus nodes can run the Turing-complete program instead of the program's creator.

Since we in this paper focus on the design of blockchain-based sealed-bid domain name auction, we omit the details of the underlying consortium blockchain. One may get such a blockchain from Ethereum with the anonymous fund and a consensus algorithm for consortium blockchains.

3.2 Auction Types

An auction is a process where people can buy items via the bidding method, and it usually involves an auctioneer and many bidders. The corresponding seller typically not appears but delegates the auctioneer to sell the items. Traditionally, there exist two main types of auctions [18]: open auction and sealed-bid auction. Every bidder can see others' bids in the former type and submit the bid to the auctioneer many times. In contrast, bidders in the latter type can submit their bids to the auctioneer only once in a private way. After all bids are submitted in a preset period, the auctioneer will open the bids and decide who the winner is. Since the sealed-bid auction can lead to a reasonable bid for the auction item with a higher probability [19], we adopt it in our system. Especially, we use the special sealed-bid auction named Vickrey auction in our system, where the bidder with the highest bid wins while he/she only needs to pay with the second-highest bid. Furthermore, we replace the auctioneer with the blockchain and assume the blockchain as an honest-but-curious party.

3.3 Pedersen Commitment

As we mentioned before, the auction we apply in this paper is the Vickrey auction, where the bid should be kept secret before the corresponding bidder opens it. To realize this process, we make use of the Pedersen commitment [20] that is widely used in many anonymous cryptocurrencies [21–23].

A commitment scheme usually contains two stages: `commit` and `reveal`. In the former stage, the sender commits a value to the receiver and reveals it in the latter stage. It requires that the receiver cannot reveal the committed value before the latter stage, and the sender cannot change the committed value after the former stage.

The Pedersen commitment goes as follows. In the `commit` stage, the sender sends $\mathsf{fund} = g^x h^r$ to the receiver, where x is the committed value, r is a random number, g and h are public parameters from a finite cyclic group with a big prime order, and no one knows $\log_h g$. Later on, the sender in the `reveal` stage sends (x', r') to the receiver. If $\mathsf{fund} = g^{x'} h^{r'}$ holds, the receiver accepts the committed value; otherwise, he/she rejects it. One of the properties of the Pedersen commitment is homomorphic addition. In particular, given $\mathsf{fund}_1 = g^{x_1} h^{r_1}$ and $\mathsf{fund}_2 = g^{x_2} h^{r_2}$, we have that $\mathsf{fund}_1 \cdot \mathsf{fund}_2 = g^{x_1} h^{r_1} \cdot g^{x_2} h^{r_2} = g^{x_1+x_2} h^{r_1+r_2}$. The value of $\mathsf{fund}_1 \cdot \mathsf{fund}_2$ is essentially a commitment to $x_1 + x_2$.

3.4 Zero-Knowledge Proof

The bid-guarantee and fund-privacy are the two security properties that our proposal should satisfy. In particular, the bidder in our proposal has the capability to prove that he/she has enough money to cover the committed bid without revealing how much fund he/she has. In this paper, we make use of the zero-knowledge proof for the inequality between two positive integers, which is original from [23].

Given $\mathsf{fund}_1 = g^{x_1} h^{r_1}$, $\{\mathsf{fund}'_{2i} = g^{a_i \cdot 2^i} h^{r'_{2i}}\}_{i=0}^{\ell}$, the prover proves that $x_1 \geq \sum_{i=0}^{\ell} a_i \cdot 2^i$ with the knowledge of x_1, r_1, and $\{a_i \in \{0,1\}, r'_{2i}\}_{i=0}^{\ell}$, where g and h are the same as that in the Pedersen commitment, and ℓ is a positive integer larger than the bit-length of any possible value of the fund in our system. The whole process contains the following two parts.

In the first part, the prover proves that $x_1 = \sum_{i=0}^{\ell} a_i \cdot 2^i + \sum_{i=0}^{\ell} b_i \cdot 2^i$ with the conditions $\mathsf{fund}_1 = g^{x_1} h^{r_1}$, $\{\mathsf{fund}'_{2i} = g^{a_i \cdot 2^i} h^{r'_{2i}}\}_{i=0}^{\ell}$, and $\{\mathsf{fund}'_{3i} = g^{b_i \cdot 2^i} h^{r'_{3i}}\}_{i=0}^{\ell}$ by providing a signature corresponding to the public key $(h, \mathsf{fund}_1/(\prod_{i=0}^{\ell} \mathsf{fund}'_{2i} \cdot \mathsf{fund}'_{3i}))$. Note that if $x_1 = \sum_{i=0}^{\ell} a_i \cdot 2^i + \sum_{i=0}^{\ell} b_i \cdot 2^i$, the prover knows the value of $\log_h \mathsf{fund}_1/(\prod_{i=0}^{\ell} \mathsf{fund}'_{2i} \cdot \mathsf{fund}'_{3i}) = r_1 - \sum_{i=0}^{\ell}(r'_{2i} + r'_{3i})$ and can generate the signature; otherwise, the prover does not know it and cannot generate the signature since $\log_h g$ is kept secret from everyone.

In the second part, the prover proves that all a_i's and b_i's belong to $\{0,1\}$ as follows. Let us take a_i as an example. The prover only needs to provide a ring signature corresponding to the public keys (h, fund'_{2i}) and $(h, \mathsf{fund}'_{2i}/g^{2^i})$. The ring signature scheme [24] allows the verifier to check the signature's validity without revealing the public key corresponding to the real signing key. Note that if $a_i = 0$, the prover knows $\log_h \mathsf{fund}'_{2i} = r'_{2i}$; and if $a_i = 1$, the prover knows $\log_h \mathsf{fund}'_{2i}/g^{2^i} = r'_{2i}$ instead; for other cases, the prover does not know either $\log_h \mathsf{fund}'_{2i}$ or $\log_h \mathsf{fund}'_{2i}/g^{2^i}$ since $\log_h g$ is kept secret from everyone. As a result, the verifier can check whether $a_i \in \{0,1\}$ via the ring signature.

4 Our Proposal

We are now ready to present our blockchain-based sealed-bid auction protocol, where we have the following four phases: Create the Auction, Commit the Bid, Reveal the Bid, and Close the Auction. In the first phase, the domain name owner deploys a smart contract in the blockchain to start the sealed-bid domain name auction. The bidder would submit his/her bid on the domain name via the smart contract in the second phase. After that, the bidder would open the bid in a predefined period. At last, the blockchain would finalize the auction according to the smart contract. Note that we have four functions in the auction smart contract, namely Create (Fig. 2), Commit (Fig. 3), Reveal (Fig. 4), and Finalize (Fig. 5), which are invoked in the four phases, respectively.

4.1 Create the Auction

After deploying the smart contract, the domain name owner invokes the function Create with $(T_1, T_2, \texttt{name}, \sigma)$, where T_1 and T_2 are the respective deadlines for committing and revealing bids, `name` is the domain name for bidding, and the last value is a signature on the smart contract under the public key corresponding to the account possessing the domain name. When the consensus nodes in the blockchain system receive the function call, they check the validity of $(T_1, T_2, \texttt{name}, \sigma)$ and whether there is no current active auction smart contract corresponding to `name`. If all of them are valid, the consensus nodes will run the rest of the function Create; otherwise, they stop running the function.

```
function Create($T_1$, $T_2$, name, $\sigma$)
    if $T_1$, $T_2$, name, and $\sigma$ are valid then
        if no active auction smart contract corresponding to name then
            initialize array bidder[]
            set state = 1
    return
```

Fig. 2. Pseudo-code for the function Create in the auction smart contract.

The pseudo-code of the function Create can be found in Fig. 2. The array `bidder[]` is used to record the information related to the bidder, including the public key `pk`, commitment $\{\mathtt{fund}'_i = g^{c'_i \cdot 2^i} h^{r'_i}\}_{i=0}^{\ell}$ for the bid, and the values $\{(c'_i, r'_i)\}_{i=0}^{\ell}$ for revealing the bid, where g, h, ℓ are the same as that explained in Sect. 3. The domain of `state` is $\{0, 1\}$, and 0 and 1 denote the inactive and active state of the smart contract, respectively.

4.2 Commit the Bid

When the bidder finds some auction smart contract on the domain name he/she is interested in, he/she invokes the function Commit with

$(\mathtt{pk}, \{\mathsf{fund}'_i\}_{i=0}^{\ell}, \sigma, \mathtt{ZKP})$, where $\mathtt{pk}$ is the public key corresponding to one of the bidder's accounts, $\{\mathsf{fund}'_i = g^{c'_i \cdot 2^i} h^{r_i}\}_{i=0}^{\ell}$ is the commitment for the bid $\sum_{i=0}^{\ell} c'_i \cdot 2^i$, σ is a signature on the smart contract under the public key $\mathtt{pk}$, and $\mathtt{ZKP}$ is the zero-knowledge proof for the statement $c \leq \sum_{i=0}^{\ell} c'_i \cdot 2^i \leq 0$. Here, c is the value of the fund corresponding to $\mathtt{pk}$, and it is recorded as $\mathsf{fund} = g^c h^r$ in the blockchain as mentioned in Sect. 3.1. It is easy to see that we can obtain $\mathtt{ZKP}$ as the steps in Sect. 3.4. When the consensus nodes receive the function call for Commit, they check whether the array $\mathtt{bidder}[\,]$ does not contain $\mathtt{pk}$, whether it is still in the valid period according to T_1, and the validity of $(\sigma, \mathtt{ZKP})$. If all of them are valid, the consensus nodes continue running the function Commit; otherwise, they stop running the function. The pseudo-code of the function Commit can be found in Fig. 3.

```
function Commit($\mathtt{pk}$, $\{\mathsf{fund}'_i\}_{i=0}^{\ell}$, $\sigma$, $\mathtt{ZKP}$)
    if $\mathtt{pk}$ does not exist in $\mathtt{bidder}[\,]$ then
        if current time and $(\sigma, \mathtt{ZKP})$ are valid then
            add $\mathtt{pk}$ and $\{\mathsf{fund}'_i\}_{i=0}^{\ell}$ into $\mathtt{bidder}[\,]$
    return
```

Fig. 3. Pseudo-code for the function Commit in the auction smart contract.

4.3 Reveal the Bid

When the bidding phase stops according to T_1, the bidder can start to reveal his/her bid by invoking the function Reveal with $(\mathtt{pk}, \{(\bar{c}'_i, \bar{r}'_i)\}_{i=0}^{\ell})$, where $\mathtt{pk}$ is the same public key the bidder used to invoke the function Commit, and $\{(\bar{c}'_i, \bar{r}'_i)\}_{i=0}^{\ell}$ are the values used to compute the commitment $\{\mathsf{fund}'_i\}_{i=0}^{\ell}$. When the consensus nodes receive the function call for Reveal, they check whether it is still in the valid period according to T_2, whether there exist $\{\mathsf{fund}'_i\}_{i=0}^{\ell}$ corresponding to $\mathtt{pk}$ in the array $\mathtt{bidder}[\,]$, and whether all $\mathsf{fund}'_i = g^{\bar{c}'_i \cdot 2^i} h^{\bar{r}'_i}$ for $i = 0, \cdots, \ell$ hold. If all of them are valid, the consensus nodes continue running the function Reveal; otherwise, they stop running the function. The pseudo-code of the function Reveal can be found in Fig. 4.

4.4 Close the Auction

After the third phase, anyone can deduce who is the winner according to $\{\bar{c}'_i\}$ recorded in the array $\mathtt{bidder}[\,]$. In this case, the auction winner will be the one who invokes the function Finalize. Note that anyone can invoke the function Finalize. In other words, the domain name owner can invoke the function for a quicker domain name transfer.

When the consensus nodes receive the function call, they check whether it is the time to finalize the auction according to T_2 and whether the state of the

```
function REVEAL(pk, $\{(\bar{c}'_i, \bar{r}'_i)\}_{i=0}^{\ell}$)
    if pk exists in bidder[] then
        if current time is valid then
            if $\mathsf{fund}'_i = g^{\bar{c}'_i \cdot 2^i} h^{\bar{r}'_i}$ for $i = 0, \cdots, \ell$ then
                add $\{(\bar{c}'_i, \bar{r}'_i)\}_{i=0}^{\ell}$ into bidder[] according to pk
    return
```

Fig. 4. Pseudo-code for the function Reveal in the auction smart contract.

smart contract is active. If both of them are valid, the consensus nodes continue running the function Finalize; otherwise, they stop running the function. The pseudo-code of the function Finalize can be found in Fig. 5, where we mainly deal with three cases for the bidders. The first one is that the bidder failed to open his/her bid in the third phase, and they will be punished by reducing the bid from the corresponding fund. The second one is for the winner whose money of the second-highest bid value will be transferred to the domain name owner, and the domain name is transferred to the winner from the domain name owner. The last case is for the ones following our protocol but losing this auction. We do not need to do anything in this case.

```
function FINALIZE( )
    if it's time for finalizing the smart contract then
        if state == 1 then
            for each item in bidder[] do
                if the bidder corresponding pk failed to reveal the bid then
                    punish the corresponding bidder by reducing his/her fund
                if the bidder winning the bidding then
                    reduce the winner's fund according to the second higher price
                    transfer name to the winner
                    set state = 0
    return
```

Fig. 5. Pseudo-code for the function Finalize in the auction smart contract.

5 Analysis of Our Proposal

In this section, we will analyze our proposal in terms of security and feasibility.

5.1 Security Analysis

This subsection shows our proposal satisfies the tx-fairness, bidding-fairness, bid-guarantee, and fund-privacy one by one as follows.

TX-Fairness. In this paper, we assume that the consensus nodes will faithfully execute the smart contract in the immutable blockchain. Hence, the codes in lines 8-11 in the function FINALIZE transferring the money and domain name are executed or not executed at the same time. In this case, we have the tx-fairness immediately.

Bidding-Fairness. To show the bidding-fairness of our proposal, we only need to show that no one can reveal the bid from the corresponding commitment. Fortunately, we can obtain this claim directly from the security of the Pedersen commitment.

Bid-Guarantee. It is easy to see that the zero-knowledge proof used in our system guarantees that the bidder has enough money to cover the bid.

Fund-Privacy. Firstly, $\mathsf{fund} = g^c h^r$ won't leak any information about c, since there are many pair (c', r')'s satisfying $\mathsf{fund} = g^{c'} h^{r'}$. Secondly, $\mathtt{ZKP}$ won't leak the concrete value of c due to the security property of the underlying zero-knowledge proof. Hence, we have the fund-privacy property.

5.2 Performance Evaluation

In this subsection, we evaluate our proposal's performance based on our experimental results in terms of off-chain cost and on-chain cost. The prototype we implement is based on the BigInteger class of Java language and the BigNumber library of Solidity language for the off-chain and on-chain parts. Furthermore, we run our prototype in a laptop with Intel(R) Core(TM) i5-1038NG7 CPU @2.00 GHz, 16 GB RAM, macOS 11.2 operating system, and Java 14, and the rinkeby test chain of Ethereum for obtaining the off-chain cost and on-chain cost, respectively. In our experiments, we set the bit-length of the order of g and h as 160, and all experiments are conducted 20 times and the average is recorded.

Off-Chain Cost. The main computational cost for the off-chain part of our proposal is due to the generation of commitments and zero-knowledge proofs. In particular, the bidder has to generate a commitment for the fund and a zero-knowledge proof for a statement. The statement shows that the bidder has enough money to cover his/her bidding. The experimental results can be found in Fig. 6. The abscissa represents the values of ℓ, and the ordinate denotes the computing time. From Fig. 6, "commitA" and "proofGen" denote the commitment commiting and the zero-knowledge proof of fund, respectively. And we can see that the computational cost is proportional to ℓ as expected, while the cost is acceptable.

On-Chain Cost. The on-chain cost is mainly due to the gas cost for the on-chain part of our proposal, especially, the verification of zero-knowledge proofs in the function COMMIT and the verification of commitments in the function REVEAL. Here, we ignore the gas cost due to the function of FINALIZE. The experimental results can be found in Fig. 7, where "commit", "proof_x",

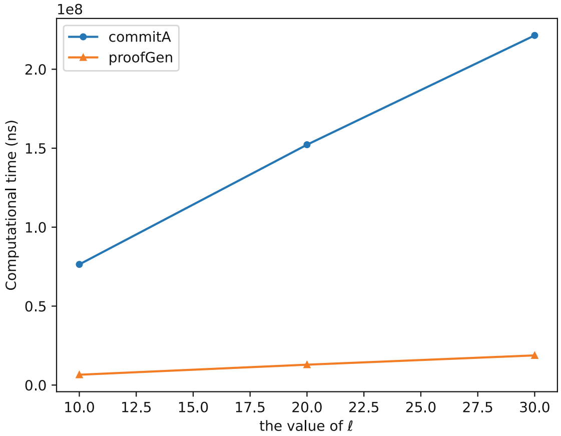

Fig. 6. The off-chain computational cost of our proposal.

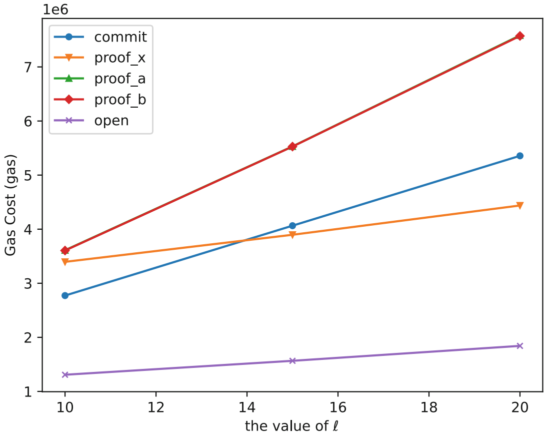

Fig. 7. The gas cost of our proposal.

“proof_a”, “proof_b”, and “open” denote the commitment commiting, zero-knowledge proofs of x, $a = \sum_0^\ell a_i \cdot 2^i$, and $b = \sum_0^\ell b_i \cdot 2^i$, and the commitment opening, respectively. From Fig. 6, we can see that the computational cost is also proportioanl to ℓ as the off-chain cost, while the cost is still acceptable. Note that the costs for “proof_a” and “proof_b” are the same since they require the same execution steps.

6 Related Work

As described in Sect. 4, our proposal is closed to blockchain-based DNS and blockchain-based auction. Hence, we give related works in two parts respectively.

6.1 Blockchain-Based DNS

To solve the traditional DNS' centralization problem, Namecoin [5], as the first solution, used blockchain's decentralized characteristics to realize the decentralized DNS. However, the resulting system is along with some side effects, such as cybersquatting. The Namecoin team proposed a new solution named Blockstack [8] by using the virtual chain, distributed hash table, cloud storage, and other techniques to tackle the above problems. Later on, ConsortiumDNS proposed another solution with better performance on data storage and domain name resolution. As we know, the domain name auction is one of the typical ways to obtain the domain name [11]. However, none of the above blockchain-based systems provide such functionality. In [9], Ethereum officials proposed a domain name system based on the smart contract to resolve domain names into Ethereum addresses but not the digital IP address. The Namebase team recently proposed a blockchain-based DNS, named Handshake, with a domain name auction protocol [10]. Their auction protocol follows the Vickrey auction, where the submitted bids are not revealed until all the bids are submitted, and the bidder with the highest bid wins the auction while paying with the second-highest bid. However, in their auction protocol, anyone can deduce the possible highest bid of the submitted bids, which may bring advantages over the latter bidders. In other words, the auction protocol cannot hold the bidding-fairness property. To the best of our knowledge, it is still challenging to design a fair, secure, and privacy-preserving blockchain-based sealed-bid domain name auction protocol.

6.2 Blockchain-Based Auction

Using the secure multi-party computing (MPC) technique, Kasba et al. [25] proposed the first blockchain-based auction protocol. However, the underlying secure MPC scheme is of a high level of interactivities, which leads to an inefficient solution, especially in the blockchain scenario. With the help of the trusted execution environment (TEE), Yuan et al. [26] proposed an efficient blockchain-based auction protocol without sacrificing the smart contract's confidentiality. Nevertheless, the TEE technique usually demands users to update their hardware, which not everyone can afford. What's worse, the security of implementation of TEE is still controversial [27,28]. Blass and Kerschbaum [29] proposed a new blockchain-based auction protocol still based on the secure MPC technique with a good performance. Nonetheless, the proposed protocol needs a pre-fixed bidding order and a semi-trusted auctioneer that is usually unnecessary in the blockchain scenario. Galal and Youssef [18] proposed an auction protocol over the Ethereum blockchain. However, it is not as efficient as expected due to the use of interactive zero-knowledge proof schemes. Furthermore, there is no mechanism to check whether the bidder has enough money to bid in the proposed auction protocol. Note that, although there is a deposit in their protocol, this deposit cannot be used as proof that the bidder has enough money to fulfill the bid. Recently, Nguyen and Thai [30] proposed a new efficient auction protocol based on multi-party state channels in terms of storage cost. However, the resulting protocol cannot support the sealed-bid auction. Qusa et al. [31] also proposed a new blockchain-based auction protocol, while the bidder colluding with some bidders can obtain some advantages over the rest of the bidders. It is fair to say that a fair, secure, and privacy-preserving blockchain-based auction protocol is still desired.

7 Conclusion

The blockchain-based DNS is considered one of the most promising decentralized DNSs. However, the current blockchain-based DNSs fail to provide a suitable domain name auction protocol essential for the domain name transfer. To fill this gap, we in this paper have proposed a new blockchain-based sealed-bid domain name auction protocol. Based on the security assumptions on the underlying blockchain system, the Pedersen commitment, and zero-knowledge proof, we have also shown that the proposed auction protocol can proceed in a fair, secure, and privacy-preserving way.

References

1. Mockapetris, P.V.: RFC1034: Domain Names-Concepts and Facilities (1987)
2. Mockapetris, P.V.: RFC1035: Domain Names-Implementation and Specification (1987)
3. Khormali, A., Park, J., Alasmary, H., Anwar, A., Saad, M., Mohaisen, D.A.: Domain name system security and privacy: a contemporary survey. Comput. Netw. **185**, 107699 (2021)
4. Verma, S., et al.: Stopping amplified DNS DDoS attacks through distributed query rate sharing. In: 11th International Conference on Availability, Reliability and Security, ARES 2016, IEEE Computer Society, pp. 69–78 (2016)
5. Loibl, A.: Namecoin (2014). https://www.namecoin.org/
6. Wang, X., Li, K., Li, H., Li, Y., Liang, Z.: ConsortiumDNS: a distributed domain name service based on consortium chain. In: 19th IEEE International Conference on High Performance Computing and Communications; 15th IEEE International Conference on Smart City; 3rd IEEE International Conference on Data Science and Systems, pp. 617–620. IEEE Computer Society (2017)
7. Emercoin NVS - Emercoin Community Documentation. https://emercoin.com/en/documentation/blockchain-services/emernvs
8. Ali, M., Nelson, J.C., Shea, R., Freedman, M.J., Blockstack: a global naming and storage system secured by blockchains. In: USENIX Annual Technical Conference, USENIX ATC 2016. USENIX Association 2016, pp. 181–194 (2016)
9. Johnson, N., Griffith, V.: Ethereum Name Service. https://ensuser.com/docs/readme.html
10. Roquerre, T.: Handshake project paper. https://handshake.org/files/handshake.txt
11. W. contributors: Domain Name Auction (2020). https://en.wikipedia.org/w/index.php?title=Domain_name_auction&oldid=975293048
12. Alvarez, R., Nojoumian, M.: Comprehensive survey on privacy-preserving protocols for sealed-bid auctions. Comput. Secur. **88** (2020)

13. Nakamoto, S., et al.: Bitcoin: A Peer-to-Peer Electronic Cash System (2008)
14. Poon, J., Dryja, T.: The Bitcoin Lightning Network: Scalable Off-Chain Instant Payments. https://lightning.network/lightning-network-paper.pdf
15. Luu, L., Narayanan, V., Zheng, C., Baweja, K., Gilbert, S., Saxena, P.: A secure sharding protocol for open blockchains. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 17–30. ACM (2016)
16. Gilad, Y., Hemo, R., Micali, S., Vlachos, G., Zeldovich, N.: Algorand: scaling byzantine agreements for cryptocurrencies. In: Proceedings of the 26th Symposium on Operating Systems Principles, pp. 51–68. ACM (2017)
17. Buterin, V., et al.: A Next-Generation Smart Contract and Decentralized Application Platform, white paper 3 (37) (2014)
18. Galal, H.S., Youssef, A.M.: Verifiable sealed-bid auction on the ethereum blockchain. IACR Cryptol. ePrint Arch. **2018**, 704 (2018)
19. Chandrashekar, T.S., Narahari, Y., Rosa, C.H., Kulkarni, D.M., Tew, J.D., Dayama, P.: Auction-based mechanisms for electronic procurement. IEEE Trans. Autom. Sci. Eng. **4**(3), 297–321 (2007)
20. Pedersen, T.P.: Non-interactive and information-theoretic secure verifiable secret sharing. In: Feigenbaum, J. (ed.) CRYPTO 1991. LNCS, vol. 576, pp. 129–140. Springer, Heidelberg (1992). https://doi.org/10.1007/3-540-46766-1_9
21. Miers, I., Garman, C., Green, M., Rubin, A.D., Zerocoin: anonymous distributed e-cash from bitcoin. In: IEEE Symposium on Security and Privacy. SP 2013, pp. 397–411. IEEE Computer Society (2013)
22. Ben-Sasson, E., et al.: Zerocash: decentralized anonymous payments from bitcoin. In: IEEE Symposium on Security and Privacy. SP 2014, pp. 459–474. IEEE Computer Society (2014)
23. Noether, S.: Ring Signature Confidential Transactions for Monero. IACR Cryptology ePrint Archive 2015, 1098 (2015)
24. Rivest, R.L., Shamir, A., Tauman, Y.: How to leak a secret. In: Boyd, C. (ed.) ASIACRYPT 2001. LNCS, vol. 2248, pp. 552–565. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-45682-1_32
25. Kosba, A.E., Miller, A., Shi, E., Wen, Z., Papamanthou, C.: Hawk: the blockchain model of cryptography and privacy-preserving smart contracts. In: IEEE Symposium on Security and Privacy, SP 2016, IEEE Computer Society, pp. 839–858 (2016)
26. Yuan, R., Xia, Y.-B., Chen, H.-B., Zang, B.-Y., Xie, J.: ShadowEth: private smart contract on public blockchain. J. Comput. Sci. Technol. **33**(3), 542–556 (2018). https://doi.org/10.1007/s11390-018-1839-y
27. Shaun Davenport, R.F.: SGX: The Good, The Bad and The Downright Ugly (2014). https://www.virusbulletin.com/virusbulletin/2014/01/sgx-good-bad-and-downright-ugly
28. Luis Merino, J.A.: SGX Secure Enclaves in Practice Security and Crypto Review (2016). https://www.blackhat.com/docs/us-16/materials/us-16-Aumasson-SGX-Secure-Enclaves-In-Practice-Security-And-Crypto-Review.pdf
29. Blass, E.-O., Kerschbaum, F.: Strain: a secure auction for blockchains. In: Lopez, J., Zhou, J., Soriano, M. (eds.) ESORICS 2018. LNCS, vol. 11098, pp. 87–110. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99073-6_5
30. Nguyen, T., Thai, M.T.: A blockchain-based iterative double auction protocol using multiparty state channels. ACM Trans. Internet Technol. **21**, 1–22 (2021)
31. Qusa, H., Tarazi, J., Akre, V.: Secure e-auction system using blockchain: UAE case study. In: Advances in Science and Engineering Technology International Conferences (ASET). IEEE 2020, pp. 1–5 (2020)