CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	Secure E-auctions: a blockchain-based cluster consensus identity-based identification scheme
Authors	Vangujar, Apurva K.;Umrani, Alia;Chin, Ji-Jian;Palmieri, Paolo
Publication date	2025-01-22
Original Citation	Vangujar, A.K., Umrani, A., Chin, J.-J. and Palmieri, P. (2024) 'Secure E-auctions: a blockchain-based cluster consensus identity-based identification scheme', 6th International Conference on Blockchain Computing and Applications (BCCA). Dubai, United Arab Emirates: IEEE, pp. 189–197. https://doi.org/10.1109/BCCA62388.2024.10844472
Type of publication	Conference item
Link to publisher's version	https://ieeexplore.ieee.org/document/10844472 - 10.1109/BCCA62388.2024.10844472
Rights	© 2024, IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-01-17 23:05:16
Item downloaded from	https://hdl.handle.net/10468/17349

UCC
University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Secure E-Auctions: A Blockchain-Based Cluster Consensus Identity-Based Identification Scheme

Apurva K. Vangujar
Computer Science & IT
University College Cork
Cork, Ireland
a.vangujar@cs.ucc.ie

Alia Umrani
Computer Science & IT
University College Cork
Cork, Ireland
a.umrani@cs.ucc.ie

Ji-Jian Chin
Department of Computing
Plymouth University
Plymouth, United Kingdom
ji-jian.chin@plymouth.ac.uk

Paolo Palmieri
Computer Science & IT
University College Cork
Cork, Ireland
p.palmieri@cs.ucc.ie

***Abstract*—Electronic Auctions (e-auctions) must be secure and reliable in the fast-changing digital world. This paper proposes a Blockchain-Based Cluster Consensus Identity-Based Identification (BCCIBI) scheme to enhance e-auction security and privacy. To establish a secure e-auction environment, the BCCIBI scheme utilises advanced cryptographic primitives such as ElGamal encryption, the Boneh-Lynn-Shacham (BLS) signature scheme, Identity-Based Identification (IBI), Zero-Knowledge (ZK) proof, Commitment scheme, and Non-Interactive ZK proof. In the BCCIBI scheme, bids are secure due to the homomorphic properties of ElGamal encryption, which ensure *confidentiality* and *integrity*. The BCCIBI scheme, grounded in the Computational Diffie-Hellman assumption, accelerates multiple bid and bidder verifications by IBI, a crucial operation for clustered blockchains ensuring *authentication*. Meanwhile, ZK proof authenticates bidders anonymously, safeguarding bidder privacy. Our unique scheme assigns pseudo-identities to participants to provide *anonymity* and ensure *unlinkability*. The BCCIBI scheme is resilient against cryptographic attacks in an efficient clustered blockchain scenario using smart contracts, addressing blockchain performance issues. The BCCIBI e-auction scheme is based on timestamps to provide *aliveness* and records on the blockchain. This paper scrutinises the BCCIBI scheme's technical foundations and practical implications, laying the groundwork for more secure and transparent e-auction systems. The BCCIBI system advances safe e-auctions by employing a multi-faceted cryptographic technique.**

***Index Terms*—Identity-Based Identification, Cluster Consensus, Zero-Knowledge, E-auction**

I. Introduction

The development of the Internet has significantly changed the auction industry; Electronic Auctions (e-auctions) have become significant in many sectors, including bandwidth allocation and exquisite goods, surpassing conventional limitations of time, space, and audience accessibility. E-auctions have emerged as a means to promote transparent and secure online bidding processes, which encourage efficiency and fairness in the public interest. Significantly, e-auctions have better results compared to offline techniques in terms of efficiency, information accessibility, participant authenticity, and bid confidentiality. Blockchain technology and smart contracts have the potential to enhance transparency and integrity in this field, thereby ensuring a more secure and verifiable bidding approach. As per Technavio [1], the hard asset equipment segment alone is projected to hit USD 1.13 billion by 2023, indicating e-auctions as a significant economic catalyst.

At the present time, e-auctions can be broadly classified into two categories: sealed bid and open bid. Prior to the bid deadline, bidders in sealed-bid auctions submit their bids in secret. Subsequently, the auctioneer discloses the bids and determines the successful bidder in accordance with the auction rules [2]. Open-bid auctions are classified as either Dutch auctions or English auctions, and both are frequently employed for inter-individual commodity transactions. E-auctions offer a convenient means for potential bidders to engage in auction activities without the need to be physically present. During the specified duration for the auctioned item, potential bidders have the ability to effectively submit their bid from any location and at any time [3].

In the process of preserving information, the blockchain may need to receive sensitive data to execute a smart contract. Therefore, it is crucial to ensure the privacy and authenticity of the data that is sent to the blockchain so that everyone can verify the data without compromising sensitive information. However, in practice, the existing sealed-bid schemes continue to face major challenges, where building trust among auctioneers, sellers, and buyers is a primary one. Users might develop suspicions of one another due to the networks' anonymity and transparency. Multiple cryptographic primitives are used to overcome these challenges.

The scheme by Manimaran *et al*. [4] introduces a paradigm where smart contracts, as opposed to outside businesses, manage all bidding transactions. The integrity and confidentiality of the purchasing process are guaranteed by this scheme, given that blockchains are decentralised and tamper-proof by nature; thus, the auction's *integrity* is maintained, and a central authority is unnecessary. Scalability challenges may arise for the model in the presence of a large number of peers, which could result in possible inconsistencies and redundancy. Li and Xue [5] propose a blockchain-based sealed-bid e-auction scheme, integrating smart contracts and ZK proof technologies to enhance auction security, privacy, and *fairness*. Tyagi *et al*. [6] is a great addition to the field of decentralised

Apurva K Vangujar and Alia Umrani are supported by PhD scholarships funded by the Science Foundation Ireland ADVANCE Centre for Research Training, under Grant number 18/CRT/6222. This research has also been supported by Science Foundation Ireland under Grant number 13/RC/2077_P2 (CONNECT).

auction systems, and it uses peer-auction collateral to penalise cheating bidders. Chen *et al*. [7] introduces a smart contract based auction scheme that ensures bid privacy and public verifiability without needing trusted third party along with it achieves decentralization and preserving the privacy of losing votes. Recently, Tan and Heng [8] developed an e-auction system using cryptographic primitives. It satisfies the security requirements, which include but are not limited to privacy, *integrity, correctness, anonymity*, and *fairness*. Nonetheless, a number of privacy and security concerns arise, including the possibility of a malicious auctioneer, the *confidentiality* of the participants' information, the validity of the bid, and the *correctness* of the auction rule.

A. Related Work

1) Blockchain: Several e-auction systems based on blockchains have been proposed and developed. Chen *et al*. [9] developed a blockchain-based e-auction system using smart contracts. The smart contract, which includes the auctioneer's data, the auction's start and end times, the winner's address, and the highest bid, is implemented on Ethereum and activated when certain circumstances occur. Smart contracts are complex; hence, contract function calls are challenging. The authors also suggested adjusting authority levels for different functions, which is challenging to achieve.

2) E-auction Models: Recently, the need for privacy has been a factor of increasing importance in auction design, and various schemes to ensure the safe conduct of English auctions have been proposed by many. Earlier efforts by Chaum *et al.* [10] introduced the Group Signature (GS) scheme for the earliest iteration. Group members, group managers, and adversary group members are members of the GS scheme. Lee *et al.* [3] proposed a reliable sealed-bid e-auction system based on a GS scheme with the authenticated encryption function. Public cryptosystems are used to secure communication through a public channel, while the GS approach is implemented to safeguard confidential information. The proposed GS scheme involves four parties: bidders, a registration manager, an auction manager, and an identity manager. Meanwhile, a blockchain is used to access, verify, and transmit information through distributed nodes. It offers identity *authentication* to prevent counterfeiting attacks through public-key cryptosystems. The transactions stored in a block are verifiable and recorded in the same ledger.

Gao *et al.* [11] proposed an auction system called an enhanced privacy-preserving auction scheme that uses homomorphic encryption to guarantee that all bids are encrypted during the auction. The Paillier cryptosystem was used to appliance homomorphic encryption with a one-time pad. Vangujar *et al*. [12] proposed an identity-based authentication and key exchange scheme for message broadcasting and batch verification for VANETs in a cluster environment. The scheme utilizes Psuedo-Identity (PID) for *anonymity* while it uses Cluster Consensus Identity-based Identification (CCIBI) to provide *authentication*. We are adopting the CCIBI [12] scheme and PID generation algorithm for our e-auction scheme. The paper [13] introduces a secure sealed-bid auction model that ensures bid *confidentiality* and *integrity* using cryptographic techniques, along with a fair arbitration mechanism to efficiently resolve dispute. Omar *et al*. [14] ntroduces a publicly verifiable and privacy-preserving sealed-bid reverse auction algorithm that eliminates the need for auctioneers or trusted third parties and extends this scheme to first-price sealed-bid auctions without additional overhead. One of the primary issues is the significant computational and communication overhead incurred due to the bitwise method used to figure out the clearing price, requiring bidders to submit data multiple times.

B. ElGamal Encryption

ElGamal encryption [15], a cornerstone of modern cryptographic techniques, plays a crucial role in enhancing the security and privacy of e-auction systems. Its application in e-auctions allows for the secure encryption of bids, ensuring that only authorised parties can access bid information, thereby maintaining the *confidentiality* and *integrity* of the bidding process. By leveraging ElGamal's public key encryption framework, e-auction systems can protect against unauthorised access and manipulation while upholding the principles of transparency and fairness essential in online bidding environments. This paper explores the implementation of ElGamal encryption in e-auctions, highlighting its effectiveness in safeguarding participant privacy and bid authenticity in a digitally connected world.

C. Contribution

In this paper, we propose a cluster consensus identity-based identification scheme on the blockchain (BCCIBI) for secure e-auctions. The main contributions are as follows: First, we construct a novel cluster BLS signature scheme and incorporate it with ElGamal encryption and ZK proof. This constructed scheme is designed for e-auction processes, where BLS signatures authenticate and verify bidders, and ElGamal encryption ensures the *integrity* of the bidding process. Additionally, we assign a PID to each bidder, ensuring *anynomity* and *unlinkability* by utilising PID instead of the original ID. The bidding process is done by a commitment scheme, and payment happens using NIZK proof, keeping everything stored on the blockchain. Secure interaction between auctioneers, bidders, and smart contracts is secure under cluster settings. The proposed scheme guarantees the *fairness* and *correctness* of the e-auction process, *confidentiality* of bidding prices, *unlinkability* of bidder identities, and *aliveness* of the submission. Moreover, the BCCIBI scheme adopts a cluster structure that enhances efficiency. It records transactions and bids on a peer-to-peer network, enhancing transparency and resistance to tampering. The auctioneer functions as a cluster head, and bidders act as cluster members, thus organising the auction process more securely and efficiently. We provide a detailed security analysis of our scheme considering all stated security requirements.

D. Organization

The paper is structured as follows: Sec. II introduces the notations and preliminaries used in the rest of the paper. Sec. III presents our e-auction system, along with requirements, participants, and the e-auction process. Sec. IV provides a detailed construction of a novel CCIBI scheme. Sec. V provides the security analysis taking into account all of the requirements for our CCIBI scheme. Sec. VI provides efficiency analysis, and VII provides the conclusion.

II. Preliminaries

A. Computational Diffie-Hellman (CDH) Assumption

The security assumption of CDH is according to [16].

definition 1. *The CDH assumption holds given* $(g, g^a, g^b) \in \mathbb{G}$*, where* $a, b \in \mathbb{Z}_q^*$*, it is computationally infeasible for any Probabilistic Polynomial-Time (PPT) algorithm to compute* g^{ab}*.*

B. IBI Scheme

definition 2. *The definition of IBI scheme is given by Kurosawa and Heng [17] has three PPT algorithms IBI =* $(\mathsf{KeyGen}, \mathsf{Extract}, \mathsf{Identification})$ *defined as follows:*

1) KeyGen. *On input* 1^k*, it outputs public parameter* PP *and master secret key* msk.
2) $\mathsf{Extract}$. *It takes input as* $(\mathsf{msk}, \mathsf{ID})$ *and returns the private key* d.
3) $\mathsf{Identification}$. *In this phase, the prover* P *and the verifier* V *communicate with each other.* P *takes input as* $(\mathsf{PP}, \mathsf{ID}, \mathsf{d})$ *whereas* V *takes input as* $(\mathsf{PP}, \mathsf{ID})$. P *and* V *communicate with each other and give output in boolean decision* 0 *(rejects) or* 1 *(accepts). The canonical proof acts in four steps, as follows: (i)* P *sends commitment* CMT *to* V. *(ii)* V *provides challenge* CHA *which is randomly chosen. (iii)* P *calculates the response* RSP *to* V *as per challenge. (iv)* V *verifies* $(\mathsf{PP}, \mathsf{ID}, \mathsf{CMT}, \mathsf{CHA}, \mathsf{RES})$ *is DH tuple.*

C. BLS Scheme

Boneh-Lynn-Shacham (BLS) [18] signature scheme is used to set up the construction of our proposed scheme using CDH assumption.

- KeyGen takes a random integer $x \in \mathbb{Z}_q^*$, q is prime order, hash function H , and generator $g \in G$ as an input. The output is the private key x and the public key g^x.
- Sign algorithm in which prover signs the message with the input considering its private key x and the message m itself. Further, it hashes the message $h = H(m)$ where $H : \{0,1\}^* \rightarrow G$. Finally, with the help of the hash function, the output is signature $\sigma = h^x$.
- Verify is the last algorithm of the BLS scheme where verifier verifies the signature with the bilinearity property as follows:

$$e(g, \sigma) = e(H(m), g^x) \tag{1}$$

Kurosawa and Heng [17] converted the BLS digital signature scheme into IBI scheme. We use this basic IBI scheme to construct one with the cluster architecture.

D. ElGamal Encryption

The popular ElGamal cryptosystem [15] is secure under the discrete logarithmic hardness assumption and also homomorphic under multiplication. Let q be a prime and g be the generator of the cyclic multiplicative group $\mathbb{Z}_q^*$. Let Alice and Bob be the two parties who want to perform the computation. The ElGamal cryptosystem has three algorithmic steps $(\mathsf{KeyGen}, \mathsf{Encrypt}, \mathsf{Decrypt})$ as follows:

1) KeyGen. It takes the input as 1^λ and outputs the pair of $(\mathsf{sk}, \mathsf{pk})$ where sk is the secret key, randomly chosen from $\mathbb{Z}_q^*$ and $\mathsf{pk} = g^{\mathsf{sk}}$ mod q. Alice sends Bob (pk, g, q).
2) $\mathsf{Encrypt}$. By taking input (pk, m), Bob encrypts and provides the output as (m_1, m_2). Bob calculates $m_1 = g^k$ mod q and $m_2 = (\mathsf{pk}^k \cdot m)$ mod q, where k is a randomly chosen value. Bob then sends the tuple (m_1, m_2) to Alice.
3) $\mathsf{Decrypt}$. Alice finally takes (sk, m_1, m_2) as input and calculates (m_2/m_1^{sk}) mod q.

III. Our E-auction System

A. Requirements

Electronic auctions need to satisfy all the same requirements as traditional auctions, and our objective is to offer a higher level of security compared to conventional means. These security requirements are detailed in Table I, which also outlines the objectives for each requirement. *Integrity* and *Confidentiality* are guaranteed through the implementation of a collision-resistant hash function and ElGamal encryption. *Correctness* and *Authentication* are provided by ZK and NIZK proofs, allowing for validation and authorization of bidders and transactions within a cluster consensus setting. *Anonymity* and *Unlinkability* is ensured by the use of pseudo-identities (PID), allowing bidders to maintain anonymous identities within the cluster. *Aliveness* is achieved by assigning timestamps to all actions in the bidding process for the e-auction. IBI also requires both bidders and smart contracts to be online; this also addresses *aliveness*. *Fairness* is achieved by a random nonce for each bid, which is used only once.

B. Participants

1) **Blockchain**. Everything that occurs across a peer-to-peer network is recorded in this blockchain distributed ledger. Security and transparency are both ensured by the blockchain in the context of our e-auction. The inflexible nature of the records for transactions and bids, in addition to their PID, ensures that the process remains transparent and resistant to tampering. The consensus mechanism is essential for ensuring that all parties are in agreement regarding the state of the blockchain and that *integrity* is maintained.
2) **Smart Contract**. The smart contracts are self-executing, as they contain the provisions of the agreement recorded

TABLE I
REQUIREMENT FOR BCCIBI E-AUCTION SCHEME

Requirements	Description
Integrity	By employing ElGamal encryption for bid, our scheme ensures that no outsider or insider entity can alter the prices submitted by bidders, thus maintaining *Integrity*.
Fairness	Ensures that each participant has an equal and transparent opportunity and is treated without bias. *Fairness* in the auction is maintained by assigning a unique nonce to each bid.
Correctness	Auction results are accurately determined based on the chosen rules, using ZK and NIZK proofs and to ensure *correctness*.
Anonymity	Each cluster member (i.e., bidder) utilises Pseudo-Identity (PID) instead of their real identity (ID) for all interactions with the cluster head (i.e., auctioneer) and blockchain, thus providing *anonymity*.
Authentication	The blockchain and bidders verify the PID and e-auction results using IBI and ZK, which proves the *authentication*.
Confidentiality	The BCCIBI scheme uses ElGamal homomorphic encryption to encrypt the bidder price. While comparing the bidder's prices, other bidders cannot see the actual price, thus ensuring the *confidentiality* of the actual bid prices.
Unlinkability	In the BCCIBI scheme, the bidding prices are encrypted along with the PID. The participants cannot link the bidding price with the associated ID of the bidders.
Aliveness	Every bidder encrypts their bid, attaches a timestamp T_{submit} to it and sends the encrypted bid to the smart contract, which is recorded on the blockchain. The smart contract checks and verifies T_{submit}. If it is expired, the bid submission will be declined, thus ensuring the *aliveness* of the e-auction process.

directly in lines of code. Smart contracts automate numerous processes in our e-auction, including bid acceptance, time constraints, and the final awarding of the auction goods. They ensure compliance with the auction's regulations in place of a central authority.

3) **Auctioneer as Cluster Head**. The auctioneer is the owner of the items and is responsible for publishing goods and items on smart contracts for bidding. In our scheme, the auctioneer acts as a Cluster Head CH.
4) **Bidders as Cluster Members**. Each cluster member acts as a bidder interested in taking part in the e-auction. The bidders are responsible for generating their public and private keys and registering themselves with the smart contract for e-auction. Bidders are denoted by ID. Cluster consensus for our scheme is $\mathsf{C} = (\mathsf{CH}, \{\mathsf{ID}_1, \ldots, \mathsf{ID}_i, \ldots, \mathsf{ID}_n\})$ where $1 \leq i \leq n$.

C. E-Auction Phases

The proposed BCCIBI e-auction scheme includes three major phases described as follows and shown in Fig. 1.

1) **Initialization and Registration Phase**. In this phase, the auctioneer and bidders set up the keys for themselves, form consensus clusters using a modified BLS-based IBI scheme, share public components to smart contracts, and simultaneously store them on the blockchain along with their anonymous PID.

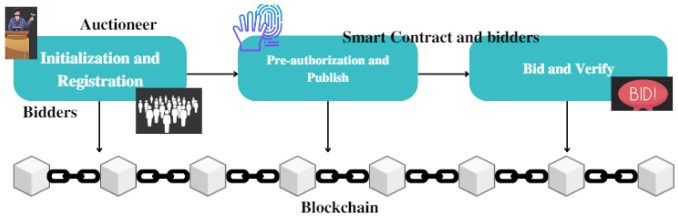

Fig. 1. Our E-Auction System

2) **The Pre-authorization and Publish Phase**. This phase covers the verification of bidders for e-auction participation and the granting of access to put in the bid. This involves each bidder's authentication using the ZK proof. This phase is responsible for publishing verified names and sharing reserve prices with the authorised bidders from the cluster.
3) **The Bid and Verify Phase**. This phase comprises the actual bidding on the goods items using a commitment scheme with Elgamal encryption. This may involve encrypting bids, timestamps, submitting bids, and validating the bid's legitimacy using the commitment scheme and payment by NIZK proof. During this phase, all bids will be encrypted, and the winning bidder will be notified of payment.

D. Definition of BCCIBI E-auction Scheme

The proposed BCCIBI scheme consists of eight PPT algorithms such that BCCIBI= $(\mathsf{KeyGen}, \mathsf{Join}, \mathsf{Extract}, \mathsf{Identification\ Protocol}, \mathsf{Sign}, \mathsf{Commit}, \mathsf{Reveal}, \mathsf{Verify})$ and is constructed using Def. 2, II-C, ElGamal from Def. II-D, commitment scheme from [5], and NIZK proof [19] run among CH, IDs, and smart contract over blockchain.

1) KeyGen. In this algorithm, the smart contract first generates the public parameters PP and the master public key mpk. Then, taking PP as input, the CH generates a pair of auctioneer cluster public and secret keys $(\mathsf{apk}, \mathsf{ask})$. Similarly, the ID_i output their bidder public and secret key pairs $(\mathsf{bpk}_i, \mathsf{bsk}_i)$ in the cluster. the
2) Join. Considering the generic scenario for the e-auction scheme, when a new bidder wants to join the cluster, the same KeyGen algorithm runs, and it becomes part of the cluster and also issues a PID for the newly joined bidder. This algorithm takes the real identity ID of the bidder as input and output and verifies the PID in two phases, as described below:
   a) *Phase 1*. Each bidder takes ID and apk as input and outputs the PID and submits it to the smart contract.
   b) *Phase 2*. The smart contract performs verification of the bidder's ID and verifies using the bpk. If it is valid, the PID is registered and assigned; otherwise, the smart contract defers the registration.

3) Extract. This algorithm takes (PID_i, mpk, bsk_i) as input and generates the user secret key d_i.
4) Identification Protocol. Following ZK proof between the bidder PID_i (acting as prover) and the smart contract (acting as verifier) $(\mathsf{CMT}, \mathsf{CHA}, \mathsf{RES})$ as defined below:
   a) CMT. A bidder with PID_i chooses a bpk_i as input and outputs a V_i as a CMT. The bidder sends CMT to smart contract.
   b) CHA. The smart contract generates a random challenge and forwards it to the bidder with PID_i.
   c) RES. The bidder with PID_i computes the U_i as RES based on the CHA and passes it to the smart contract. The smart contract accepts PID_i as an eligible bidder for the e-auction process if and only if it verifies using the CDH assumption.
5) Sign. In this algorithm, the CH issues goods and signs the reverse price for each authorised bidder. It generates a signature. The bidder verifies the signature. If it is valid, the bidder becomes ready for the bid.
6) Commit. The smart contract generates timestamps $(T_{\mathsf{start}}, T_{\mathsf{end}}, T_{\mathsf{bid}})$. Taking timestamps as input, the bidder commits to bid $\mathcal{B}_i$ by encrypting homorophically the bid prices using Elgamal encryption, generating $(\hat{P}_{1i}, \hat{P}_{2i}, \mathsf{PID}_i)$ and sending it to the smart contract. The smart contract checks $T_{\mathsf{start}} \leq T_{\mathsf{bid}} \leq T_{\mathsf{end}}$ if it is valid, the smart contract accepts the bid.
7) Reveal. Smart contracts decrypt bids $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$ and compare them against the reserve price $\mathcal{P}$. Valid highest bids exceeding the current highest bid $\mathcal{HB}_{\text{current}}$ and previous highest bid $\mathcal{HB}_{\text{previous}}$ and assigning the new winning highest bid as $\mathcal{HB}$ are updated on the blockchain, which maintains bidder anonymity in determining the winning bid with the new $\mathcal{HB}$.
8) Verify. The winning bidder creates a NIZK proof to confirm their payment without revealing transaction T_i details. The bidder computes a unique hash of their transaction function $f(T_i)$ and generates a NIZK proof, which they submit to the smart contract. It verifies this proof to ensure the existence of a valid, confidential transaction corresponding to the winning bid and forwards it to CH to close the e-auction for that good.

IV. Construction of Blockchain-based Cluster Consensus Identity-Based Identification Scheme

In the BCCIBI scheme, bidder authentication is efficiently managed by a smart contract employing a ZK proof of the BLS-based IBI scheme, which ensures *authentication*. This sophisticated approach ensures only verified bidders are eligible to participate in the e-auction process while preserving *anonymity* and *unlinkability* using a PID, which allows bidders to engage without revealing their true identities. ElGamal encryption is used to encrypt bid data, reinforcing both *integrity* and *confidentiality* as it prevents unauthorised access to bid prices. All operations occur over the blockchain with smart contracts acting as the trusted party, eliminating the need for a central authority, thus mitigating any single point of failure and trust issues. For secure payment transactions, NIZK proofs are employed, ensuring the *correctness* of each transaction within the system. Additionally, the e-auction process achieves *fairness* by tying each transaction to a nonce, a distinctive identifier that ensures the equity of each bid. Timestamps are the foundation of the entire bidding process and are essential for keeping the auction's *aliveness*. The BCCIBI scheme $\mathsf{BCCIBI} = (\mathsf{Auctioneer}, \{\mathsf{Bidder}_1, \ldots, \mathsf{Bidder}_i, \ldots, \mathsf{Bidder}_n\})$ is structured into three main phases using eight PPT algorithms, ensuring a secure e-auction environment.

A. Initialization and Registration Phase

1) KeyGen. This is the first algorithm run by all cluster members, including CH (auctioneer) and ID_i (bidders), where $1 \leq i \leq n$ generates self keys. Let q be a large prime number, and let g be a generator with order q of the cyclic group $\mathbb{G}$. The first hash function is $H_0 : \mathbb{G} \rightarrow \{0,1\}^*$. The second hash function is $H_1 : \{0,1\}^* \rightarrow \mathbb{G}$. The third hash function is $H_2 : \{0,1\}^k \rightarrow \mathbb{G}$ where k is the length of the good price and the fourth hash function is $H_3 : \{0,1\}^* \rightarrow \mathbb{Z}_q^*$. Smart contract takes the input as selects a random integer $x \in \mathbb{Z}_q^*$, generates $\mathsf{mpk} = g^x$, and outputs public parameters $\mathsf{PP} = \{\mathbb{G}, q, g, H_0, H_1, H_2, H_3, g^x\}$.
   - For auctioneer CH, chooses a random $\hat{x} \in \mathbb{Z}_q^*$ as auctioneer secret key ask and computes public key apk as $y = g^{\hat{x}}$, finally giving output as a pair of $(\mathsf{apk}, \mathsf{ask})$.
   - For bidder ID_i, it chooses a random $\hat{x}_i \in \mathbb{Z}_q^*$ as its bidder secret key bsk_i, calculates $y_i = g^{\hat{x}_i}$ as bpk_i and gives output as a pair of $(\mathsf{bpk}_i, \mathsf{bsk}_i)$.
   - Bidder public key bpk of all bidders in the cluster C are sent to the smart contracts and stored in the blockchain simultaneously.
2) Join. If a new bidder wants to join or register an auction in the cluster C, then it joins the rest of the identities in the cluster by performing KeyGen as described above and becomes a part of a cluster $\mathsf{C} = (\mathsf{CH}, \{\mathsf{ID}_1, \ldots, \mathsf{ID}_i, \ldots, \mathsf{ID}_n\})$ where $1 \leq i \leq n$. A new bidder is stored in the blockchain with their ID and bpk. Next, each bidder from the cluster provides real identity ID_i and bpk_i where $1 \leq i \leq n$ to the smart contract to generate PID_i for the anonymous e-auction process.
   a) *Phase 1* PID Generation.
      i) Each bidder ID_i has a real identity along with bpk_i.
      ii) Each bidder chooses a random $a_i \in \mathbb{Z}_q^*$ and computes $A_i = g^{a_i}$.
      iii) The PID_i is computed as $\mathsf{PID}_i = \mathsf{ID}_i \oplus H_0(\mathsf{bpk}_i^{a_i})$.
      iv) The bidder sends (PID_i, A_i) to the smart contract.
   b) *Phase 2* PID Verification.

i) Smart contract takes input as (PID_i, A_i) and verifies the identity by checking if $\mathsf{ID}_i = \mathsf{PID}_i \oplus H_0(A_i^{x_i})$ holds.
ii) If the verification is successful, the smart contract accepts the registration, confirms the PID_i, and stores it on the blockchain.
iii) If it fails, the smart contract discards the PID_i and cancels the registration.

A new anonymous cluster for auction is $\mathsf{BCCIBI} = (\mathsf{CH}, \{\mathsf{PID}_1, ..., \mathsf{PID}_i, ..., \mathsf{PID}_n\})$.

3) Extract. Consider PID_i from C, takes an input $(\mathsf{PID}_i, \mathsf{mpk}, \mathsf{bsk}_i)$ and calculates $Q_{\mathsf{PID}_i} = H_1(\mathsf{PID}_i)$. It outputs the user secret key $\mathsf{d}_i = \hat{x}_i Q_{\mathsf{PID}_i}$.

B. The Pre-authorization and Publish Phase

4) Identification Protocol. It is the communication between a bidder PID_i and a smart contract before the bidder starts putting the bid $\mathcal{B}_i$ on the goods. All bidder PID can be verified at the same time using the ZK proof. The identification protocol is used to verify a large number of bidders. It is easy to verify bidders under the BCCIBI scheme, which is given as follows:
   a) CMT. Bidder with PID_i takes bpk_i as input. It chooses a random number $r_i \in \mathbb{Z}_q^*$ and calculates $V_i = g^{r_i}$ and sends it to the smart contract as CMT.
   b) CHA. The random challenge $c \in \mathbb{Z}_q^*$ is generated by a smart contract and sent to each bidder.
   c) RES. Each bidder calculates a response based on the challenge $U_i = r_i + c\mathsf{d}_i$.
   d) Smart contract accepts the bidder for e-auction process if and only if $g^{U_i} = V_i + \mathsf{bpk}_i Q_{\mathsf{PID}_i} c$ is CDH-tuple.

   If the verification holds true, the smart contract confirms the bidder's eligibility for the e-auction process. Upon successful verification, the smart contract then issues the names of the goods and the reserve price to the authorised bidder.
5) Sign. The auctioneer now issues the goods name and signed reserve price for goods to all authorised bidders in the cluster using CDH-based blind signatures as follows:
   a) Let the auctioneer assign the price $\mathcal{P}$ for good and calculate the hash $\hat{\mathcal{P}} = H_2(\mathcal{P})$.
   b) The auctioneer selects a random $\bar{x} \in \mathbb{Z}_q^*$ (this is a one-time-use key or nonce for each signature) and computes the corresponding public key $K = g^{\bar{x}}$.
   c) The auctioneer computes the shared secret $S = K^{\hat{x}}$ using ask, which would be difficult for an attacker to compute without knowing $\hat{x}$, thus relying on the CDH assumption.
   d) The auctioneer creates the signature $\sigma = (K, S, \hat{\mathcal{P}}^{\bar{x}})$ for the bidder with PID_i.
   e) Bidders do not have $\mathcal{P}$, so they need to rely on $g^{\bar{x}\hat{x}}$ and $g^{\bar{x}}$ to confirm the authenticity of the signature.
   f) The bidders would have to solve the CDH assumption $(g^{\bar{x}}, y, \sigma) =$ True if and only if σ is a valid signature. Each bidder with PID_i will receive $\mathcal{P}$ and get ready for the bid.

C. The Bid and Verify Phase

Once the bidder is a proven authorised user, then PID_i puts the bid $\mathcal{B}_i$, and it stores in the blockchain. The bidder always has to put $\mathcal{B}_i$ such that $\mathcal{B}_i \geq \mathcal{P}$. Bids that are $\mathcal{B}_i < \mathcal{P}$ can not participate in e-auction. We are using a commitment scheme between bidders and smart contracts. For simplicity, we are assuming a PID_i wants to prove that their encrypted $\mathcal{B}_i$ is within a certain range without revealing the actual $\mathcal{B}_i$.

6) Commit. The bidding process starts with T_{start} and ends with T_{end}. Each bidder commits to a bid $\mathcal{B}_i$ and encrypts the bid using ElGamal encryption with the current timestamp T_{bid}.
   a) Select a random integer $b_i \in \mathbb{Z}_q^*$
   b) Compute $\hat{\mathcal{P}}_{1i} = (g^{b_i} || T_{\mathsf{bid}}) \mod q$.
   c) Compute the shared secret $\hat{S}_i = y_i^{b_i} \mod q$.
   d) Encrypt the price: $\hat{\mathcal{P}}_{2i} = (\mathcal{B}_i || T_{\mathsf{bid}}) \times \hat{S}_i \mod q$.

   The encrypted bid $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$ is sent to the smart contracts along with PID_i. Smart contracts record the submission time of bid T_{bid} and check its validity. If T_{bid} is valid only if $T_{\mathsf{start}} \leq T_{\mathsf{bid}} \leq T_{\mathsf{end}}$. All the timestamps are recorded on the blockchain.
7) Reveal. Smart contract compares $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i}, \mathsf{PID}_i)$ with other bids, and the highest will be sent to the auctioneer and also stored in the blockchain. The auctioneer will demand payment from the winning bidder, and the winning bidder will make the payment.
   a) Smart contracts takes input as $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i}, \mathsf{PID}_i)$
   b) Decrypts the bid $\mathcal{B}_i$ by calculating $\hat{S}_i' = (\hat{\mathcal{P}}_{1i})^{x_i} \mod q$.
   c) Only bids that meet or exceed the reserve price are accepted, so it compares $\mathcal{B}_i \geq \mathcal{P}$ and is considered valid.
   d) Assume $\mathcal{HB}$ is the current highest valid bid and has two components: $\mathcal{HB}_{\text{current}}$ is the value of the current highest bid, which is known only after the reveal phase, $\mathcal{HB}_{\text{previous}}$ previously stored highest commitment.
   e) For each new valid bid $\mathcal{B}_i$, check if $\mathcal{B}_i > \mathcal{HB}_{\text{previous}}$
   f) If true, update $\mathcal{HB}_{\text{current}} = (\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$.
   g) Maintain the anonymity of the bidder throughout the process by using PID_i and recording $\mathcal{HB}_{\text{current}} = \mathcal{HB}$ on the blockchain and storing same bid as $\mathcal{HB}_{\text{previous}}$ for next bidder.
8) Verify. The winner gets notified and gets a payment request for $\mathcal{HB}$. Let T_i be a private transaction for bidder PID_i. The bidder computes a unique identifier of the transaction T_i using a hash function H_3 where $f(T_i) = H_3(T_i)$. The bidder then generates a NIZK proof π_i that they know a transaction T_i for which the

statement is true $H_3(T_i) = f(T_i)$. The NIZK π_i is computed as follows:

$$\pi_i = \mathsf{Prove}(\mathsf{d}_i, H_3(T_i))$$

The bidder submits $f(T_i)$ and π_i to the auctioneer while keeping T_i secret. The auctioneer verifies the proof π_i using the verification key:

$$\mathsf{Verify}(\mathsf{bpk}_i, f(T_i), \pi_i)$$

If the proof π_i is valid, the auctioneer is assured that the bidder knows a valid transaction T_i corresponding to $f(T_i)$ without the auctioneer learning anything about T_i itself.

Correctness

1) $\mathsf{ID}_i = \mathsf{PID}_i \oplus H_0(A_i^{x_i}) = \mathsf{ID}_i \oplus H_0(\mathsf{bpk}_i^{a_i}) \oplus H_0(A_i^{x_i}) = \mathsf{ID}_i \oplus H_0(g^{x_i a_i}) \oplus H_0(A_i^{x_i}) = \mathsf{ID}_i \oplus H_0(A_i^{x_i}) \oplus H_0(A_i^{x_i}) = \mathsf{ID}_i$
2) $g^{U_i} = V_i + \mathsf{bpk_i} Q_{\mathsf{PID}_i}^c = g^{r_i} + g^{\hat{x}_i} Q_{\mathsf{PID}_i} c = g(r_i + Q_{\mathsf{PID}_i} \hat{x}_i c) = r_i + c\mathsf{d}_i$.
3) $\mathcal{B}_i = \frac{\hat{\mathcal{P}}_{2i}}{(\hat{\mathcal{P}}_{1i})^{x_i}} = \frac{(\mathcal{B}_i || T_{\mathsf{bid}}) \times \hat{S}_i \mod q}{(g^{b_i} || T_{\mathsf{bid}})^{x_i}} = \frac{(\mathcal{B}_i || T_{\mathsf{bid}}) \times y_i^{b_i} \mod q}{(g^{b_i} || T_{\mathsf{bid}})^{x_i} \mod q} = \frac{(\mathcal{B}_i || T_{\mathsf{bid}}) \times g^{x_i b_i} \mod q}{(g^{b_i x_i} || T_{\mathsf{bid}}) \mod q} = \mathcal{B}_i$

While our current proposal does not include implementation results, we plan to conduct a comprehensive evaluation in an extended version. All encryption operations are performed inside the smart contract, ensuring scalability by enhancing the efficiency and transparency of blockchain shown in following section. The scalability is further supported by the inherent parallel processing capabilities of blockchain networks. Even with many users, the system can manage the contract function execution time and gas costs through optimized smart contract design and efficient consensus protocols like Proof of Stake, which balance security, decentralization, and performance.

V. Security Analysis

Due to page limitations, we present a security analysis in the form of arguments addressing each security requirement from Table I. Furthermore, we mention a few adversary attacks by Adversary $\mathcal{A}$ to illustrate the robustness of our scheme against such threats. The complete reduction proof will be provided in the full version of the paper.

1) *Integrity*. During the Commit algorithm in our scheme, the bidder encrypts the price by computing $\hat{\mathcal{P}_{2i}} = (\mathcal{B}_i || T_{\mathsf{bid}}) \times \hat{S}_i \mod q$, where $\hat{\mathcal{P}}_{1i} = g^{b_i} \mod q$ is the price and $\hat{S}_i = y_i^{b_i} \mod q$ is the shared secret. $\mathcal{A}$ intercepting the $\mathcal{B}_i$ during transmission, but ElGamal encryption ensures that even if $\mathcal{A}$ intercepts the $\mathcal{B}_i$, they cannot decrypt and understand it without corresponding x_i. In the verify phase, the smart contract decrypts the bid and compares the prices. Since $\hat{\mathcal{P}}_{1i} = g^{b_i} \mod q$ where $b_i \in Z_q^*$ is the CDH assumption, the $\mathcal{A}$ cannot decrypt the price and bidding is unmodified and tamper-proof. In the Sign algorithm as well, we use H_2 to create σ, which keeps $\mathcal{P}$ tamper-proof and prevents eavesdropping.
2) *Fairness*. In the Commit algorithm of the BCCIBI scheme, the use of a random nonce b_i for each B_i is used only once. This ensures that each bid is unique and treated equally, without any bias. It also promotes the idea that no bidder is specifically favored. $\mathcal{A}$ attempts to reuse a valid bid transmission to influence the e-auction outcome. Nonce b_i makes Replay and Sybil attacks infeasible, thus providing *Fairness*.
3) *Correctness*. ZK and NIZK proof validate the authenticity of bidders PID, bids $\mathcal{B}$, and transactions T without revealing ID of the bidder. A smart contract ensures that all actions within the system are legitimate and verifiable. Smart contract verifies $\mathcal{B}_i = \frac{\hat{\mathcal{P}}_{2i}}{(\hat{\mathcal{P}}_{1i})^{x_i}}$ and gives the results. This correctness for Verify using π_i ensures the *correctness* and legitimacy of the winning bidder's $\mathcal{HB}$ and T_i, mitigating the risk of fraudulent payment.
4) *Anonymity*. After bidders join the cluster in the Join algorithm, they generate a $\mathsf{PID}_i = \mathsf{ID}_i \oplus H(\mathsf{cpk}_i^{a_i})$ by utilising their real- ID and then send a PID verification request to the smart contract. The smart contract verifies the $\mathsf{ID_i} = \mathsf{PID_i} \oplus H(A_i^{x_i})$. If it holds, the smart contract registers the corresponding for each bidder and notifies them. Each bidder utilises their PID for all interactions with the auctioneer and smart contract, thus providing *anonymity*. Moreover, the PID is generated using bpk and verified by $\mathsf{bsk} = x_i$. Since only the smart contract knows bsk, no other entity can verify the PID, hence providing *anonymity* and unlinkability and avoiding impersonation attack. Additionally, since the smart contract knows the real ID of the cluster bidder, it can reveal the real ID in the case of a dispute, thereby providing conditional privacy.
5) *Authentication*. In the Identification Protcol algorithm, the smart contract verifies the bidder's PID using $(\mathsf{CMT}, \mathsf{CHA}, \mathsf{RES})$ under the CDH assumption. This verification process uses the ZK proof that $g^{U_i} = V_i + \mathsf{bpk}_i Q_{\mathsf{PID}_i} c$ is a CDH tuple. This proof ensures that each bidder with PID is genuine and not a fictitious ID created to manipulate the e-auction process. ZK proof helps to prevent the Sybil attack and identity theft since the verification process is done by PID and not by revealing the bidder's ID, thus providing *authentication.*
6) *Confidentiality*. In Join, CH generates the signature σ for the price $\mathcal{P}$ using H_2 and one-time-key $\bar{x}$. H_2 is a hash function that ensures the $\mathcal{P}$ is kept confidential and only accessible to authorised bidder. The *confidentiality* requirement guarantees that any attempt to alter the bid price by CH is identifiable, and forgery is unachievable because of the H_2. In the Commit phase, the bidder uses ElGamal encryption to encrypt the $\mathcal{B}_i$. The output $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$ cannot be decrypted by the $\mathcal{B}_i$ or altered without detection. This prevents a man-in-the-middle attack. In the Reveal algorithm, while comparing the $\mathcal{B}_i$ with $\mathcal{P}$ later with $\mathcal{HB}_{\text{previous}}$ and $\mathcal{HB}_{\text{current}}$, the other bidder cannot see the actual price; however, it can only

TABLE II
COMPARISON OF BCCIBI WITH EXISTING SCHEME

Requirements	Li & Xue [5]	Tan & Heng [8]	Galal & Youssef [20]	Sharma *et al.* [21]	Our scheme
Anonymity	✓	✓	×	✓	✓
Authentication	✓	×	✓	×	✓
Unlinkability	✓	✓	×	✓	✓
Fairness	✓	✓	✓	✓	✓
Correctness	×	✓	✓	×	✓
Integrity	×	✓	×	✓	✓
Confidentiality	✓	✓	✓	✓	✓
Aliveness	×	×	×	×	✓

TABLE III
EFFICIENCY ANALYSIS FOR THE BCCIBI SCHEME.

Algorithm	$\mathbb{A}$	$\mathbb{E}$	H	$\mathbb{Z}_q$	P
KeyGen	0	1	0	1	0
Join	2	1	1	1	0
Extract	0	0	1	0	0
IdentificationProtocol	2	1	0	1	0
Sign	0	1	1	1	0
Commit	0	1	0	1	0
Reveal	0	1	0	0	0
Verify	0	0	1	0	0

Legends: $\mathbb{E}$ is exponentiation in $\mathbb{Z}_q^*$, $\mathbb{A}$ is addition in $\mathbb{Z}_q^*$, multiplicative $\mathbb{Z}_q$, P is bilinear pairing, and H is hash function.

see which bidder has selected the highest price, thus ensuring the *confidentiality* of actual bid prices by PID_i.

7) *Unlinkability*. Each bidder uses the distinct nonce $b_i \in \mathbb{Z}_q^*$ along with PID_i to homorophically encrypt the bidding prices $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$. The PID_i is generated using the ID_i and a random a_i, and correspondingly, the PID_i is updated by the Join algorithm. Since for each session the PID_i is different, the $\mathcal{A}$ cannot link the bidding price $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$ with the associated PID_i of the bidders. The BCCIBI scheme prevents $\mathcal{A}$ from linking malicious bids to specific bidders across different sessions. Since the PID is regularly updated and anonymous, it becomes infeasible for an $\mathcal{A}$ to track bidder's behaviour over time, hence providing *unlinkability*.
8) *Aliveness*. In the Commit algorithm, the bidding starts with T_{start} and ends at T_{end} while the bid is submitted at T_{bid} where $T_{\mathsf{start}} \leq T_{\mathsf{bid}} \leq T_{\mathsf{end}}$. During the Open phase, where each bidder PID_i commits the encrypted bid $(\hat{\mathcal{P}}_{1i}, \hat{\mathcal{P}}_{2i})$, it appends the T_{bid} and sends it to the smart contract, which is recorded on the blockchain. The smart contract verifies by checking $T_{\mathsf{start}} \leq T_{\mathsf{bid}} \leq T_{\mathsf{end}}$. If it is expired, the bid submission will be declined, thus ensuring the *aliveness* of the e-auction process in the BCCIBI scheme. The timestamps on all the actions in the bidding process help to prevent replay attacks and also ensure that old bids cannot be used maliciously.

VI. EFFICIENCY ANALYSIS

This section provides the efficiency analysis of our scheme in terms of computational cost along with a comparison with existing schemes regarding their requirements.

Table II illustrates the security criteria, showing that our scheme meets all the security requirements for e-auctions. It is also evident that *confidentiality* and *fairness*, essential for e-auction, are achieved by most schemes. Table III presents an efficiency analysis for a single bidder, denoted as ID_i, detailing the least amount of operations required by each algorithm. For evaluating the computational cost for n bidders in a cluster, these figures are scaled by a factor of n, showing the computational efficiency of our scheme while omitting pairing operations.

Although our scheme introduces computational overhead due to the integration of multiple cryptographic primitives, this unique combination ensures robust security, privacy, and public verifiability. Future performance evaluations will aim to quantify this overhead and validate our approach against existing schemes.

VII. CONCLUSION

The cluster structure in BCCIBI scheme significantly improves communication and consensus among bidders. This structure is crucial in a decentralised environment, especially when dealing with adversarial bidders, due to efficient management and coordination of the auction process. The IBI scheme simplifies the authentication process, allowing bidders to securely validate their identities anonymously. This is important in a decentralised setting to ensure that only legitimate participants are involved in the auction, making the IBI scheme a complement to BCCIBI. A key feature of ElGamal encryption is employed in the BCCIBI scheme for its homomorphic properties, enabling secure bid $\mathcal{B}$ encryption that aligns with the commitment scheme, guaranteeing the *integrity* and *confidentiality* where bid price remains concealed until the Reveal phase. PID protect bidder *anonymity* and *unlinkability* throughout the e-auction process. The nonce for each $\mathcal{B}_i$ further reinforces *fairness* provided that each bid is unique and treated unbiased. The use of ZK and NIZK proof maintains *authorization* and provides *correctness*. Timestamps for all actions provide *aliveness* e-auction process.

An additional strength of the BCCIBI scheme is its integration of blockchain and smart contracts, which provide a tamper-proof ledger for recording all e-auction transaction. The BCCIBI scheme further provides security against potential

attacks, including Sybil, replay, eavesdropping, man-in-the-middle, identity theft, and impersonation attacks. To address concerns about resource intensity and adoption barriers, the BCCIBI scheme employs optimized cryptographic algorithms and a cluster-based consensus mechanism to enhance efficiency and scalability. User-friendly interfaces and educational resources simplify the learning curve, while a robust governance model and regulatory compliance ensure reliability and sustainability.

The cluster structure in BCCIBI scheme significantly improves communication and consensus among bidders. This structure is crucial in a decentralised environment, especially when dealing with adversarial bidders, due to efficient management and coordination of the auction process. The IBI scheme simplifies the authentication process, allowing bidders to securely validate their identities anonymously. This is important in a decentralised setting to ensure that only legitimate participants are involved in the auction, making the IBI scheme a complement to BCCIBI. A key feature of ElGamal encryption is employed in the BCCIBI scheme for its homomorphic properties, enabling secure bid $\mathcal{B}$ encryption that aligns with the commitment scheme, guaranteeing the *integrity* and *confidentiality* where bid price remains concealed until the Reveal phase. PID protect bidder *anonymity* and *unlinkability* throughout the e-auction process. The nonce for each $\mathcal{B}_i$ further reinforces *fairness* provided that each bid is unique and treated unbiased. The use of ZK and NIZK proof maintains *authorization* and provides *correctness*. Timestamps for all actions provide *aliveness* e-auction process.

An additional strength of the BCCIBI scheme is its integration of blockchain and smart contracts, which provide a tamper-proof ledger for recording all e-auction transaction. The BCCIBI scheme further provides security against potential attacks, including Sybil, replay, eavesdropping, man-in-the-middle, identity theft, and impersonation attacks. To address concerns about resource intensity and adoption barriers, the BCCIBI scheme employs optimized cryptographic algorithms and a cluster-based consensus mechanism to enhance efficiency and scalability. User-friendly interfaces and educational resources simplify the learning curve, while a robust governance model and regulatory compliance ensure reliability and sustainability.

References

[1] Technavio, "Online auction market by product, platform, and geography - forecast and analysis 2023-2027," Technavio Research Report, January 2023, 156 pages, Report SKU: IRTNTR70208.

[2] Z. Guo, Y. Fu, and C. Cao, "Secure first-price sealed-bid auction scheme," Eurasip Journal on information security, vol. 2017, pp. 1–6, 2017.

[3] C.-C. Lee, P.-F. Ho, and M.-S. Hwang, "A secure e-auction scheme based on group signatures," Information Systems Frontiers, vol. 11, pp. 335–343, 2009.

[4] P. Manimaran and R. Dhanalakshmi, "Blockchain-based smart contract for e-bidding system," in 2019 2nd International Conference on Intelligent Communication and Computational Techniques (ICCT). IEEE, 2019, pp. 55–59.

[5] H. Li and W. Xue, "A blockchain-based sealed-bid e-auction scheme with smart contract and zero-knowledge proof," Security and Communication Networks, vol. 2021, pp. 1–10, 2021.

[6] N. Tyagi, A. Arun, C. Freitag, R. Wahby, J. Bonneau, and D. Mazières, "Riggs: Decentralized sealed-bid auctions," in Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security, 2023, pp. 1227–1241.

[7] B. Chen, X. Li, T. Xiang, and P. Wang, "Sbrac: Blockchain-based sealed-bid auction with bidding price privacy and public verifiability," Journal of Information Security and Applications, vol. 65, p. 103082, 2022.

[8] S. C. Tan and S. H. Heng, "Secure cryptographic e-auction system," International Journal of Technology, vol. 13, no. 6, p. 1222, 2022.

[9] Y.-H. Chen, S.-H. Chen, and I.-C. Lin, "Blockchain based smart contract for bidding system," in 2018 IEEE International Conference on Applied System Invention (ICASI). IEEE, 2018, pp. 208–211.

[10] D. Chaum and T. P. Pedersen, "Transferred cash grows in size," in Workshop on the Theory and Application of of Cryptographic Techniques. Springer, 1992, pp. 390–407.

[11] W. Gao, W. Yu, F. Liang, W. G. Hatcher, and C. Lu, "Privacy-preserving auction for big data trading using homomorphic encryption," IEEE Transactions on Network Science and Engineering, vol. 7, no. 2, pp. 776–791, 2018.

[12] A. K. Vangujar, A. Umrani, and P. Palmieri, "Identity-based cluster authentication and key exchange (id-cake) message broadcasting and batch verification in vanets," in International Conference on Applied Cryptography and Network Security. Springer, 2024, pp. 162–179.

[13] M. Zhang, M. Yang, and G. Shen, "Ssbas-fa: A secure sealed-bid e-auction scheme with fair arbitration based on time-released blockchain," Journal of systems architecture, vol. 129, p. 102619, 2022.

[14] I. A. Omar, H. R. Hasan, R. Jayaraman, K. Salah, and M. Omar, "Implementing decentralized auctions using blockchain smart contracts," Technological Forecasting and Social Change, vol. 168, p. 120786, 2021.

[15] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE transactions on information theory, vol. 31, no. 4, pp. 469–472, 1985.

[16] A. Boldyreva, "Threshold signatures, multisignatures and blind signatures based on the gap-diffie-hellman-group signature scheme," in PKC 2003, 6th Intl. Workshop on Theory and Practice in Public Key Cryptography, Miami, FL, USA, January 6-8, 2003, Proc., ser. LNCS, Y. Desmedt, Ed., vol. 2567. Springer, 2003, pp. 31–46.

[17] K. Kurosawa and S.-H. Heng, "From digital signature to ID-based identification/signature," in International Workshop on Public Key Cryptography, 2004, pp. 248–261.

[18] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the weil pairing," in International Conference on the Theory and Application of Cryptology and Information Security. Springer, 2001, pp. 514–532.

[19] S. Chakraborty, "Verifiable e-auction over a block-chain," 2021.

[20] H. S. Galal and A. M. Youssef, "Verifiable sealed-bid auction on the ethereum blockchain," in Financial Cryptography and Data Security: FC 2018 International Workshops, BITCOIN, VOTING, and WTSC, Nieuwpoort, Curaçao, March 2, 2018, Revised Selected Papers 22. Springer, 2019, pp. 265–278.

[21] G. Sharma, D. Verstraeten, V. Saraswat, J.-M. Dricot, and O. Markowitch, "Anonymous sealed-bid auction on ethereum," Electronics, vol. 10, no. 19, p. 2340, 2021.