Project/Thesis No.: CSER-22-39

e-GP: A Blockchain-Based Tendering Technique

By

Md. Shahriar Karim

Roll: 1607051

&

Mahfujur Rahman

Roll: 1607036

Department of Computer Science and Engineering

Khulna University of Engineering & Technology

Khulna 9203, Bangladesh

April, 2022

e-GP: A Blockchain-Based Tendering Technique

By

Md. Shahriar Karim
Roll: 1607051

&

Mahfujur Rahman
Roll: 1607036

A thesis submitted in partial fulfillment of the requirements for the degree of
"Bachelor of Science in Computer Science & Engineering"

Supervisor:

Dr. M.M.A. Hashem
Professor
Department of Computer Science and Engineering
Khulna University of Engineering & Technology

_____
Signature

Acknowledgement

First of all, all the praise goes to the almighty Allah, who helped us with all His blessings and kindness for us to be able to finally see our thesis work come to fruition.

It is our honor to be guided in the process by our supervisor Dr. M.M.A. Hashem, Professor, Department of Computer Science and Engineering (CSE), Khulna University of Engineering & Technology (KUET). We will ever be grateful to him for constantly remaining supportive and encouraging throughout the process despite our own shortcomings.

Finally, we are thankful to our families, friends and all people connected to our lives who had played any kind of role in making this work possible.

Authors

Abstract

Blockchain technologies are currently transcending the realms of electronic currencies and making way into application space because of the security and reliability it provides. One of the key areas that can make use of this extra security in real human life situations is public procurement processes. Corrupted and biased tender processing have remained as an unaddressed problem and have resulted in large scale misuse of public property. Integrating blockchain technology can transform this situation. Despite introduction of digitized tendering systems, there is still opportunity to improve on the security and transparency metrics of the tendering process of our country. Our proposed technique incorporates the use of smart contracts in an Ethereum based platform to facilitate tender circulation and bid submission for the tender. This ensures the data keeps safe from the risk of tampering. The use of encryption using public and symmetric keys to control the access provides a window for auditing of the tendering process. That ensures the process is always secure, all at the same time maintaining the transparency that has been missing all along in the current systems.

Table of Contents

Page

LIST OF FIGURES

Page

LIST OF ALGORITHMS

Pages

CHAPTER 1

Introduction

1.1 Background

Blockchain is a new age technology that has transformed the landscape of modern day financial systems. Besides most common uses such as cryptocurrencies and NFTs, integration of blockchain into other aspects have resulted in applications that can provide security, transparency and accountability all within a trustless platform.

The basic idea behind blockchain technology came into existence following the publication of a white paper on a peer-to-peer electronic currency system named Bitcoin by the author Satoshi Nakamoto in 2009 [1]. The paper introduced the concept of using digitally signed hash values to keep track of digital transactions. This concept had no necessity for a centralized authority to oversee or verify transactions. The transactions on a blockchain were verified, immutable and hence secure from tampering. The introduced concept of distributed ledgers radicalized the digital currency and digital assets scheme. Built on the concepts popularized through bitcoin, Vitalik Butterin introduced another blockchain-based decentralized platform for application development in 2014 [2]. The platform was named Ethereum. Ethereum paved the way for blockchain technology to be integrated into applications other than electronic currency. It introduced concepts like smart contracts and gave birth to the first truly application-based blockchain approach.

1.2 Motivation

Bangladesh is a third world country riddled with corruption in almost every imaginable sector. Public procurement remains as one of the most important sectors among them. Lack of transparency and accountability in key stages of the process as lead to these large-scale

corruptions. Political influence also plays a key role in affecting the procurement process and tender evaluation. No data about the procurement details, evaluation metrics and spending details are disclosed or publicly available. This leaves a room for misuse of resources and information tampering. Such kind of transparency is necessary to be ensured if we want to truly move further towards open governance. The recently introduced eGP system by the Bangladeshi govt has simplified the process from a user perspective, but has done very little to deal with the problems lying within the depth of the tender processing and evaluation process. The tender processing part is the part that needs improvement in this respect. Establishment of a technique that can alleviate some of the problems with the use of blockchain technology and at the same time being within the regulations of the national laws was the motivation behind this proposed development.

1.3 Problem Statement

Tender submission, evaluation and awarding process has been digitalized in the recent past. But the lack of transparency or accountability has remained. It has still not been able to keep the process free from nepotism, political influence or simply personal interest. Tender evaluation often favors politically powerful candidates or candidates with high contacts, as opposed to choosing the rightful candidate. Sometimes the tender specifications are changed in the middle of the process to favor certain candidates. Sometimes public procurements involve huge amount of money-fraud by showing expenditure of unusual amount of money for the procurement. A mostly automated system made with blockchain can ensure the system is free from tampering risks, the data is secure, and transparency can also be ensured by public access of the data to the general people or the auditors. Relevant works have been attempted and achieved in the recent time within other nations' legal boundaries [3][4][5]. The main challenge is to set up a technique good enough to deal with this issue without contradicting the national procurement laws that must be followed in any tender evaluation process.

1.4 Objectives

The objectives of this thesis are as follows:

- To design and propose a blockchain-based model for tender processing
- To ensure that the data of the the bidders or their bids are not accessible to other bidders during the process
- To prevent data tampering during the tender processing stage
- To ensure tender specifications cannot be changed midway to favor any specific candidates
- To provide transparency of the total process to confirm no corruption took place
- To ensure that the proposed model is functional in the Bangladeshi context and within legal boundaries

1.5 Scope of the Thesis

This thesis focuses mainly on the tender processing part of the procurement process. This includes the tender circulation, bidding, information storing, bid evaluation and tender awarding. It explores the use of Ethereum platform in making such tender processing technique that can provide security and transparency at the same time.

1.6 Proposed System

The method or model proposed by us is built on the Ethereum platform using the concept of smart contracts. The proposed method is built on the method proposed on [3], but a modified version appropriate for use in our national legal context is devised.

The model uses smart contracts for advertisement of the initial tender circular and bid submissions. The first block of the chain is the tender advertisement as a smart contract. The initial block of the chain will contain the tender details, time limit and a public key. Potential bidders can see the tender advertisement and decide to submit bids according to the specifications announced in the tender advertisement. No tampering of the tender is possible after that otherwise a completely new blockchain will have to be initiated. The bid is also submitted through a smart contract. The smart contract holding the bid will be encrypted by

a symmetric key. The generated key will be encrypted by the initial tender's public key. The resulting complete key will not be revealed to any party before the tender submission deadline. This will prevent access to data during the submission process.

After the time limit is over and the deadline is reached, the blockchain stops accepting newer blocks hence no more submission is possible. At that stage, the key required to access the blockchain data will be revealed to the proper evaluation committee appointed to the task by the authority. The evaluation committee will now be able to download the blockchain and review all the bid submissions from the blockchain.

The evaluation committee will then properly evaluate the bid submissions and decide on the winning bid for the tender. After selecting the winner bid, the results will be published and the public key required to view the data of all the submissions in the blockchain will be published as well to ensure transparency in the process. This will allow individual auditors or the general public to conveniently check if any kind of corruption or biasness influenced the tender awarding process.

1.7 Contributions

The contributions made through this thesis work are more prominent in developing a contextually suitable technique for our national perspective. We proposed a blockchain-based tender processing model for adaptation in Bangladeshi legal context.

As part of this thesis work, we prototyped a functional version of our proposed model using the Ethereum platform. As ethereum-based blockchain applications rely on smart contracts, we also designed an appropriate smart contract which can aid the process of the tender processing by taking in inputs, storing the data of the tender bids and acting as the blocks in the blockchain.

1.8 Thesis Organization

Chapter 2: This chapter sheds light on earlier works regarding tender-processing and open-governance particularly using blockchain. This includes works done in other similar contexts and also works of otherwise different applications with potential applicability in our scenario.

Chapter 3: In this chapter, the theoretical considerations necessary for our development have been explained in details. The relevant technologies used and have connections with our proposed model have been discussed in details.

Chapter 4: The proposed model is discussed in details in this chapter. This includes how different components of the model works and how the technique operates as a whole. This also describes the implementation aspect of the proposed model. The prototyping we have developed to simulate the actual tender processing mechanism is described in details in this chapter.

Chapter 5: The chapter focuses on the comparative analysis of our proposed system against the existing system in use and explains the improvements we have made over the existing design.

Chapter 6: This chapter concludes the thesis with a summarization of the total work, the future potential developments and discussion about the limitations that can be further addressed.

Chapter 2

Literature Review

There are several research studies regarding the use of blockchain and blockchain-based smart contracts, as a potential technology to mitigate frauds in public procurement. Blockchain has a great potential to support public sector activities because of its features such as security, transparency. The basic perform of our system is to introduce a new approach in which the bidders will bid for the particular tenders online and after some processes will know that they have get their desire tenders or not [1]. They will also can get the information of the selected bidder so that there remains no unfair means and the whole process become transparent to the public level.

A study provides a blockchain administration hierarchy based on an analytic hierarchy approach for the adoption of blockchain technology for administrative reforms and its usage by the public administration [2]. The findings show that the main factors considered in the use of blockchain for public administration are the security related, economic and decentralized system. According to the study, blockchain's security feature enables it to strengthen government trust by ensuring procedural legitimacy by mutually oppressive transparent information sharing and rational choice.

Another study is presented about a blockchain-based smart contract prototype [3] for a specific benefit process from the Syddjurs Municipality government in Denmark about the deployment of blockchain-based smart contracts for municipal government processes. The authors show that there are some benefits to adopt that technology for the government processes such as integrity guarantees, verifiability, and direct collaboration of payment between parties. There are also some problems detected such as the cost of latency, peer to peer transaction charges, immutability of errors and a very concerning single point of failure the municipal government which is losing blockchain private keys resulting in losing control over government casework, with no recourse.

In South Africa, [4] gives a high-level overview of legal and practical obstacles that could impede the adoption of blockchain-based platforms as a viable solution to the problem of corruption in public procurement. As actual contracts, the authors emphasize the decentralized nature of blockchain-based smart contracts. Nevertheless, given the lack of remedies such as the ability to cancel tender procedures or contracts, the author underlines the legal challenges of adopting smart contracts as real contracts. [5] Proposes a framework employing the concept of smart contracts to the public procurement process. The authors concentrate on the tender process in order to use blockchain to consolidate open governance and e-government. The authors believe that blockchain-based technologies have the potential to promote fairness, transparency, and accountability. As a result, an independent and automatic auditing process is enabled, ensuring accountability and enables citizens to track their government's activities without unnecessary difficulty. The purpose of the framework is to use smart contracts to achieve a state of fair, transparent, and independently verifiable government tendering. This framework allows for open governance, preventing citizens from being left out of government transactions and promoting civil society participation in monitoring those transactions. The authors evaluate the performance and financial costs of the Ethereum platform, as well as the security and auditability aspects.

In China, the Chinese government successfully implemented the surgical procedure (Public-Private Partnership), a biological process game model was created to discuss the individual biological process stable techniques and obtain the dynamic replication part diagram. From the day of mercantilism to the day of settlement, it takes two operational days to complete the settlement. The project creates a victimization blockchain paradigm for clearing and settlement (in NSE). The model reduces the settlement process much faster, from two days to a few minutes. The proposed architecture has a distributed ledger that operates on rational go and easily transfers knowledge across all members' gifts in the network. The settlement time is reduced by the fast flow of knowledge between completely different entities. According to the study of the sport behavior selections between them, suggestions such as increasing the penalty intensity of rent-seeking, taking effective direction suggestions, and lowering the price of direction were made. It provides scientific basis for the direction of the surgical procedure project tender [6].

In Tanzania, public procurement seems to be a very ineffective process that results in inflated expenses, corruption, and delays in the implementation of governmental contracts. However

this paper examines how innovation could be used in public procurement to guarantee true fairness, aggressiveness, transparency, and value for money when it comes to public procurement. This paper explains the basic procure Management techniques used in Tanzania, the challenges, and how the procurement method promotes public procurement. However, e-evaluation as component of e-procurement will reduce the number of public procurements. When the budget of the public arrive, the Project sets up an associate degreed scope for the deployment of a cost-effective e-tendering system [7].

In 2018, the Mexican government advocated the use of blockchain technology for the first time in public procurement [8]. The Mexican solution used a blockchain governance paradigm, with stakeholders including government agencies, universities, and civil society organizations. The Mexican government created five smart contracts using Ethereum to handle the following phases: a) Tendering registries of public bodies, b) bidders' registration, c) bidders' prequalification procedure, d) bidding process and proposal evaluation, and e) selection.

The government of Seoul, South Korea, established a smart contract for the proposal review stage in order to improve transparency and impartiality in the decision-making process [9].

[10] Introduced blockchain based procurement system for the Open Government Partnership (A global initiative with participation from 71 countries either through national or local action.) They explore the government tendering process, a set of activities that have three distinct phases: a) government tender opening (publishing), b) bidding period, and c) tender closing and selection of the best bid. Morphing these activities as part of our proposal in such a manner that it enables;

1) A tender can be opened by the tendering organization, but once it is open, it cannot be changed. For this reason, it prevents the organization from altering the tender in order to favor a bidder. Each tender also contains evaluation criteria for picking the best potential bid.

2) Authorized bidding organizations can submit a bid with the guarantee that it will remain confidential (until the tendering deadline) and will not be changed (integrity protection). There's also some assurances that third parties won't be able to bid on account of other authorized bidding organizations. Furthermore, individual bidding organizations are not aware of the other organizations' bids during the bidding process. Furthermore, as a strict

privacy requirement, bidding organizations should not learn whether or not a specific organization has filed a bid.

3) Only once the tender is closed can the tendering organization open the bids. The winner of the best bid would be announced. Losing bidders might use the evaluation criteria to compare the winning proposal to their own in order to assess the decisions. Furthermore, all tender proposals can be made public if necessary, allowing citizens and other interested parties to assess the entire bidding process.

4) The technology provides non-repudiation, collision avoidance, confidentiality (time-dependent), privacy and integrity – along with independent auditability feature and evidential guarantees.

Chapter 3

Related Technologies

3.1 Blockchain:

Blockchain technology is based on the concept of a distributed ledger [11], which acts as a database that stores information about previous transactions involving those agents. It is audited on a regular basis by groups of agents (selected according to different policies, depending on the application domain) [12]. Each audit's result is saved in a block and broadcast to the network. Blocks are added to the ledger in a cryptographically linked chain in order. Attempts to change the sequence of the blocks or tamper with them are easily discovered. According to a set of rules, the entire community can accept or reject the reliability of any block. If an agent receives several legitimate additions to their local copy of the ledger, they always chose the longest chain of valid blocks (or the oldest one, if they are equal in length), discarding any other competing or less significant chains [13][14]. Even in cases where propagation is slow due to excessive network latency, this conceptually simple technique assures that consensus is finally attained. In the same way, ill-intentioned nodes may attempt to introduce unscrupulous entries into the ledger, but the community will simply reject their blocks and disregard their chains, thus forcing them to follow the rules. If the community approves auditing, the ledger is copied among the agents, potentially containing recent, previously unverified transactions [15]. Otherwise, the largest accepted section of the ledger is recreated with information about sonorities and the actions to be implemented – the results of which are then registered as new transactions to be audited in subsequent rounds of verifications [16]. As a result, a blockchain-based solution can be imagined to secure information accessibility in any large-scale system.

3.2 Ethereum:

Ethereum's goal is to bring together and improve on the concepts of scripting and on-chain meta-protocols, allowing developers to create arbitrary consensus-based applications with the scalability, interoperability, standardization, feature-completeness and integrated development environment that these different paradigms offer. Ethereum does this by constructing a blockchain with a built-in Turing-complete programming language, which is effectively the ultimate abstract fundamental layer. Ethereum enables developers to design smart contracts and decentralized apps with their own set of arbitrary ownership, transaction formats, and state transition mechanisms. On top of this platform, smart contracts can be written, which are cryptographic blocks that store value and can only be unlocked if certain conditions are met. Because of its Turing-completeness, value-awareness, blockchain-awareness, and state, Ethereum is extremely powerful.

The state of Ethereum is made up of objects known as "accounts," each of which has a 20-byte address and state transitions that are direct transfers of value and information between accounts. There are four fields in an Ethereum account: a) The nonce, a counter used to ensure that each transaction can only be completed once, b) The account's current ether amount, c) If applicable, the account's contract code d) The account's storage (default: empty) "Ether" is Ethereum's main internal crypto-fuel, and it is used to pay transaction fees. In general, there are two types of accounts: 1) externally owned accounts. They are controlled by private keys and 2) contract accounts, controlled by their contract code. An externally owned account has no code and one can send messages from an externally owned account by creating and signing a transaction; in a contract account, every time the contract account receives a message its code activates, allowing it to read and write to internal storage and send other messages or create contracts in turn.

3.3 Smart Contract:

In 1994, Nick Szabo coined the phrase "smart contract," which he defined as "a computerized transaction protocol that performs the conditions of a contract." [21] Every 210,000 blocks, Szabo proposes halving the payout. To reduce the need for trusted intermediaries between transacting parties and the occurrence of intentional or accidental exceptions, contractual

provisions (collateral, bonding, etc.) are translated into code and embedded into property (hardware or software) that may self-enforce them[17]. Smart contracts are scripts stored on the blockchain in the context of blockchain. (They're similar in concept to stored procedures in relational database management systems.) They have a distinct address because they are part of the chain. A smart contract is triggered by sending a transaction to it [18]. According on the data included in the triggering transaction, it then operates independently and automatically on every node in the network in a prescribed manner. (This means that every node in a smart contract-enabled blockchain is a virtual machine (VM), and the blockchain network is a distributed VM.) Smart contracts enable general-purpose computations to take place on the blockchain. When they're tasked with managing data-driven interactions [20] between network components, however, they thrive. A smart contract is deterministic, meaning that the same input will always provide the same result. When a non-deterministic contract is triggered, it will execute on every node on the network and may yield various random results, preventing the network from reaching a consensus on the contract's execution result. Non-deterministic smart contracts are either impossible to write in a properly built blockchain platform (by forcing contract developers to use a programming language that does not have any nondeterministic constructs), or they are possible but will be rejected if they are attempted to be deployed on the network [19]. Because all interactions with a contract take place through signed "messages" on the blockchain, all network members acquire a cryptographically verifiable record of the contract's activities.

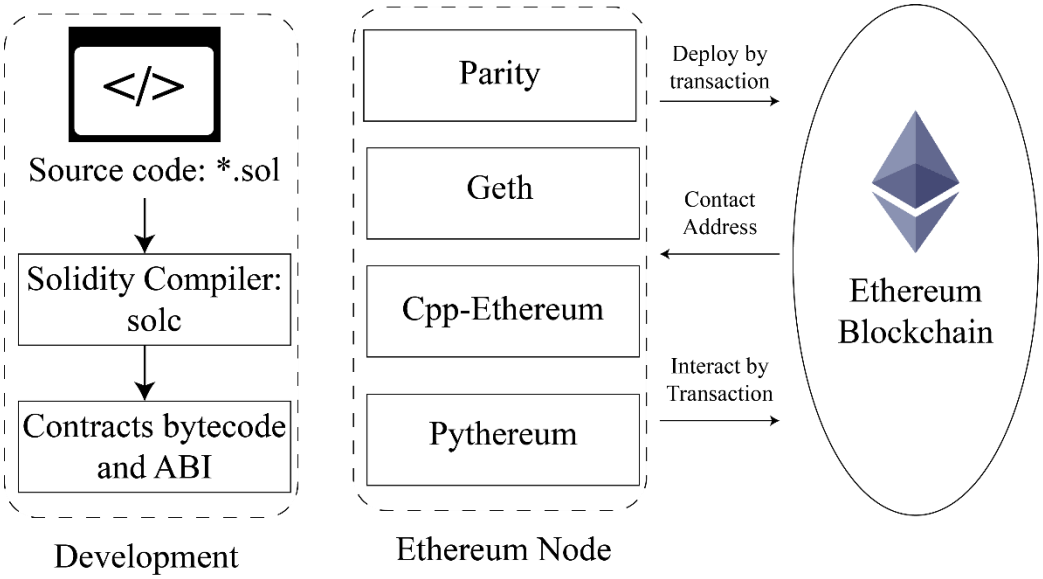

Figure 3.1: The process of smart contract's development, deployment, and interaction

3.4 Solidity:

Solidity is a Turing-complete high-level programming language with a syntax comparable to JavaScript. It allows inheritance and polymorphism, as well as libraries and complicated user-defined types, and is statically typed. Contracts are structured similarly to classes in object-oriented programming languages when use Solidity for contract development. Contract code, like traditional imperative programming, comprises of variables and functions that read and modify them. Solidity defines unique variables like as "msg," "block," "tx," and others that are always present in the global namespace and have properties that allows users to access information about an invocation-transaction and the blockchain. These variables, for example, make it possible to get the origin address, the amount of Ether, and the data delivered along with an invocation-transaction. Modifiers are a feature of Solidity that is very useful. Modifiers are enclosed code units that augment functions in order to change the flow of execution of the code. The purpose of this method is to eliminate conditional routes in function bodies using the condition-oriented programming (COP) paradigm. Modifiers are used to quickly change the behavior of functions and are specified after the function name in a whitespace-separated list. The modifiers body is the new function body, with '_' substituted with the original function body. Modifiers are commonly used to check for particular conditions before executing a function. Events are another useful and interesting aspect of Solidity. Smart contracts can fire events that are dispatched signals. User interfaces and applications can listen for those events on the blockchain and act on them without incurring significant costs. Aside from that, events can be used for logging. They save their arguments in a transaction's log, a particular data structure in the blockchain that maps all the way up to the block level, when they're called. These logs are linked to the contract's address and can be easily retrieved from outside the blockchain.

3.5 Merkle Tree:

A Merkle tree is a binary tree type [23]. The tree is comprised of a set of nodes with a huge number of leaf nodes at the bottom of the tree. It contains the underlying data, a set of intermediate nodes where each node is the hash of its two children, and finally a single root

node representing the “top” of the tree. The top of the tree is also consist of the hash of its two children. The Merkle tree’s goal is to allow data in a block to be sent piecemeal: a node can receive only the block’s header from one source, and a tiny portion of the tree relevant to them from another source, and still be satisfied that all of the data is valid. The reason this works is that hashes propagate upward: if a malicious user tries to swap in a fake transaction at the bottom of a Merkle tree, the change will cause a change in the node above it, and then another change in the node above that, eventually changing the root of the tree and thus the hash of the block, causing the protocol to register it as a completely different block (almost certainly with an invalid proof of work). The Merkle tree protocol is arguably essential to long-term sustainability.

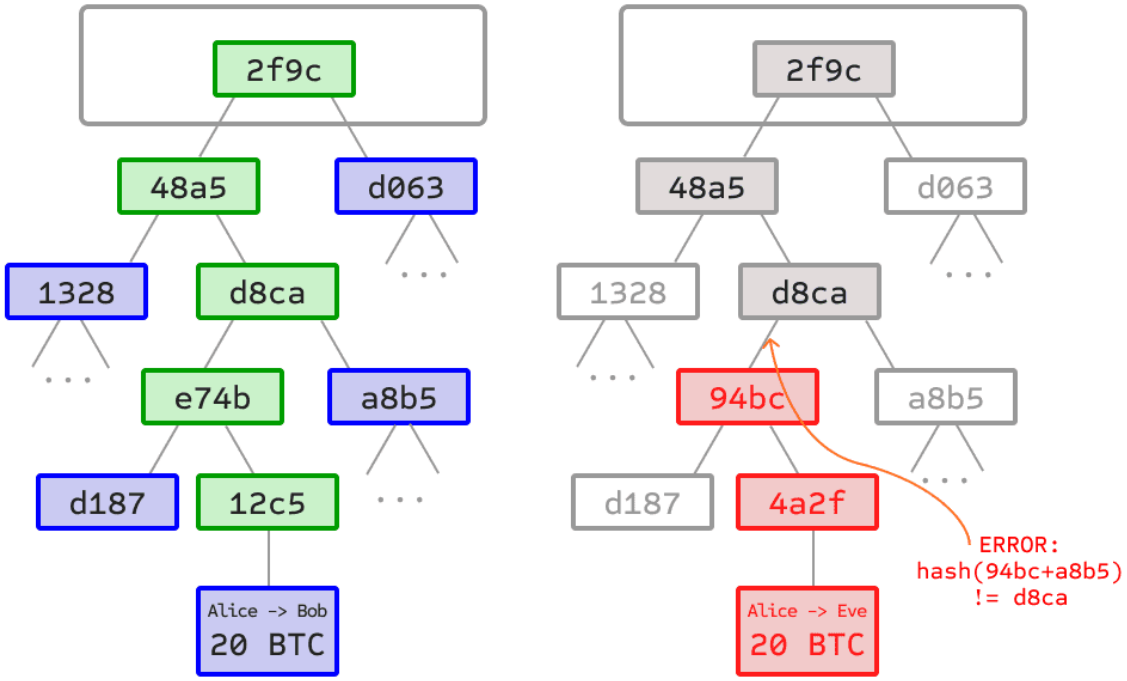

Figure 3.2: Merkle Tree from Ethereum whitepaper

3.6 Consensus Mechanism:

Blockchain solutions do not require a third-party trusted authority because they are decentralized. Instead, blockchain uses a decentralized consensus method to ensure the data and transactions are reliable and consistent. There are four basic consensus algorithms in existing blockchain systems [21]: PoW (Proof of Work), PoS (Proof of Stake), PBFT

(Practical Byzantine Fault Tolerance), and DpoS (Decentralized Proof of Stake) (Delegated Proof of Stake). The PoW mechanism is used by the two most popular blockchain systems (Bitcoin and Ethereum). The PoW process relies on the solving of puzzles to establish the data's authenticity. The problem is frequently computationally difficult but can be easily verified. A PoW puzzle must be solved when a node creates a block. As demonstrated in Fig. 3.3, after the PoW puzzle is solved, it will be disseminated to other nodes in order to reach consensus. The block structure may differ in detail amongst blockchain systems. Each block in Bitcoin typically includes PrevHash, nonce, and Tx [22]. Txs denotes the transactions included in this block, while PrevHash denotes the hash value of the previous created block. The value of nonce is obtained by solving the PoW puzzle. A correct nonce should satisfy that the hash value shown in Eq. (1) is less than a target value, which could be adjusted to tune the difficulty of PoW puzzle.

$$SHA256\ (PrevHash \,||\, Tx1 \,||\, Tx2 \,||\, \ldots \,||\, nonce) < Target \quad \ldots\ \ldots\ \ldots \quad (1)$$

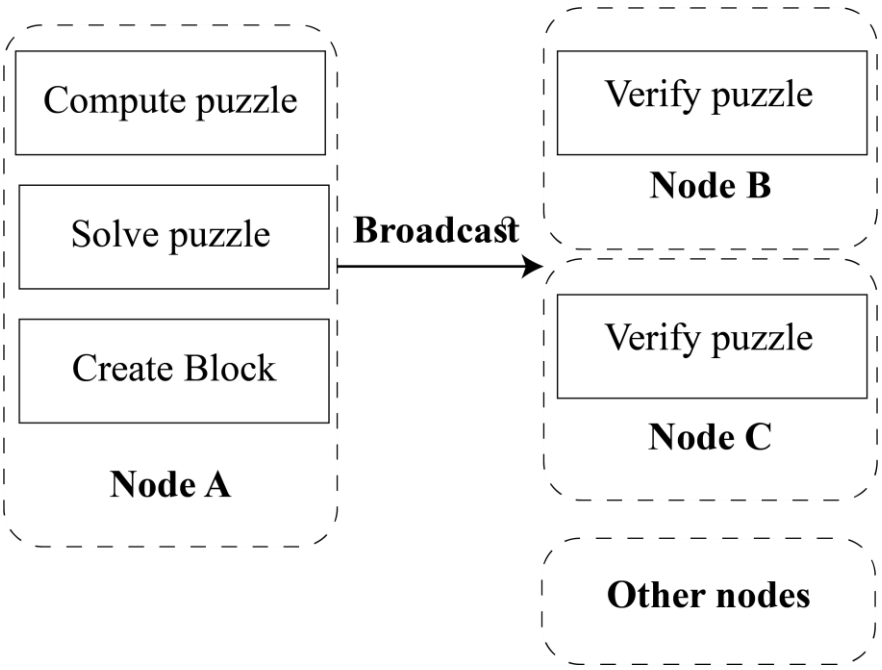

Figure 3.3: Proof-of-work consensus mechanism

3.7 Tender:

A tender is a regulated procedure in which an entity publicly reveals a contract, requests offers that meet the contract's requirements, evaluates the offers, and chooses one based on predetermined criteria. The internal phase, the external phase, and contract administration are the three basic steps of the public procurement process. The internal phase entails the establishment of records for the planning of goods and service acquisitions. Requests for bids and terms of reference including the participation of public agents in their creation are common records created during this stage. The most significant documents made at this stage include all of the terms and conditions for service and commodities acquisitions, as well as the guidelines for bids to participate in the process. The external phase entails making the records public in order to encourage fair competition. A call for bids is usually publicized alongside the terms of reference and a copy of the contract so that participants may examine their eligibility for the process and the terms of any agreements they may be associated with. During the external phase, the bidder with the most favorable proposal is chosen through a judging process. Following those two phases, the contract is awarded and monitored until it is completed in the third step.

3.8 e-GP:

The e-GP system or E-Government Procurement is a single web portal, from where and through which procuring agencies and entities will perform their procurement related activities like to publish Annual Procurement Plans, Invitation for Tender(IFT), Request for Proposal(RFP), Request for Quotation(RFQ), Tender/Application/Proposals submission, Opening, Evaluation, Contract Award Notices, Contract management, Payments, Procurement Management Information System with Key Procurement Performance Indicator Reports and other procurement related information as required by the PPA 2006 and PPR, 2008, using a dedicated secured web based dashboard, The e-GP system is accessible for their use to procuring agencies and entities of the government of Bangladesh. The objective of the e-GP is to enhance the efficiency and ensure transparency in public procurement through the implementation of a comprehensive e-GP solution to be used by any or all government organizations in the Country

All the stakeholders including general public, tenders/applicants/consultants, procuring entities, payment service providers, development partners, media, e-GP System administrators and auditors get access to e-GP System and information.

Though e-GP is a secured technology, it has some controversial aspects. In their privacy policy it is said that “In no event shall CPTU/IMED and/or third parties be liable for any damages including, but not limited to, direct or indirect or consequential damages or any damages including, but not limited to, errors or omissions, delays or incomplete transactions, planned or unplanned e-GP portal downtime or inaccessibility of the e-GP Portal, insufficient time to submit tender, loser user identities, session outages or accidental page closures, indirect or consequential damages or any damages whatsoever arising from use, loss of user data, whether in action of transaction, negligence or other action, arising out of or in connection with the use of the e-GP System.” (p. 1983).That means there is a chance for tender applications being damaged in this system and the authority is not responsible for this error. In another point it is stated “CPTU/IMED does not warrant that the functions contained in the e-GP system shall be uninterrupted or error free or that those defects shall be corrected or that this e-GP System or the server that those defects shall be corrected or that this e-GP System or the server that makes it available shall be free of viruses or bugs. CPTU/IMED does not warrant full functionality, accuracy or reliability of any material. CPTU/IMED may terminate, change, suspend or discontinue any aspect of the e-GP System, including the availability of any features of the system, at any time without notice or liability.”(p. 1984). This point indicates the software itself may not entirely error free. Moreover, there is a huge possibility of malware attack which can leak or tamper the bidder’s information and therefore hindering the whole bidding process.

3.9 Ganache:

A blockchain-based emulator used to execute several tests and commands. It controls the blockchain operation by inspecting the states of system. Formerly, its name was Test RPC, which was later renamed to ganache. It provides information such as; Visual MNEMONIC (key phrase for ganache accounts) and account addresses.

3.10 Metamask:

MetaMask is a web browser extension that allows you to access Ethereum-enabled distributed applications, or “Dapps.” The extension injects the Ethereum web3 API into the javascript context of every website, allowing Dapps to read from the blockchain. When a Dapp wishes to make a transaction and publish to the blockchain, MetaMask also enables the user create and manage their own identities (through private keys, local client wallets, and hardware wallets), so the user receives a secure interface to evaluate the transaction before approving or rejecting it. MetaMask requires access to read and write to any webpage because it adds functionality to the usual browser environment.

CHAPTER 4

Proposed System

4.1 Overview

Our proposed method consists of a few key components that are crucial to the model's functionality. Tender processing for public procurements usually involve three important steps – tender advertisement, tender evaluation and tender awarding. There are two key parties involved in the process – the tender authority and the bidders. The overall flow diagram of how the tender submission and evaluation mechanism works for the technique is demonstrated in figure 4.1 and figure 4.2.

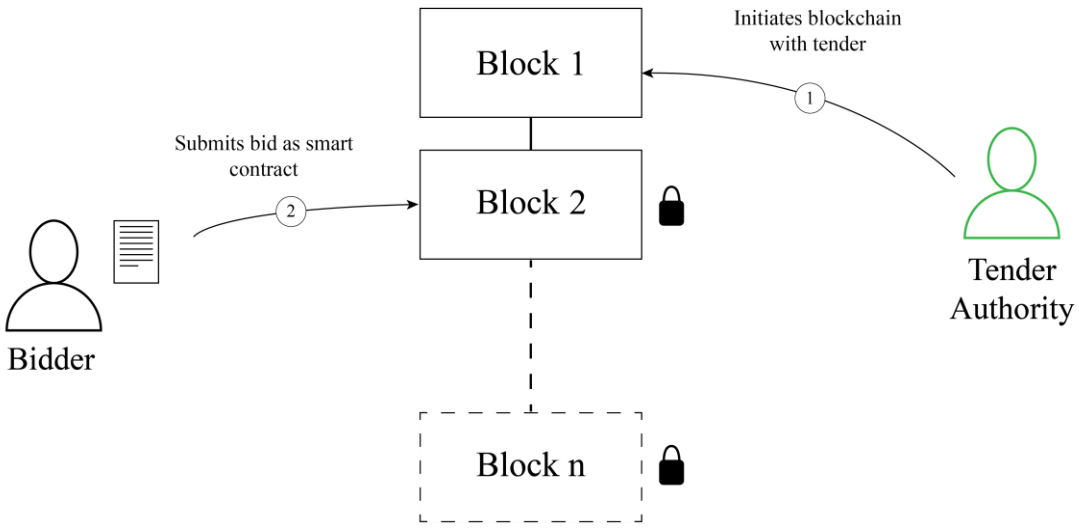

Figure 4.1: Tender submission process architecture in the blockchain

The first important part of the tender process is the tender submission by the potential bidders. To aid this process, the complete method works through smart contracts. All the bids

including the tender advertisement is done through smart contracts. The tendering authority initiates a blockchain by adding the first block that is the tender advertisement. The potential bidders interested to submit a bid submits a bid through a smart contract, which adds additional blocks to the blockchain. This process continues until the deadline is reached. Until which, all the blocks that contain the bid information are kept encrypted and the data are hidden from the authority or other bidders.

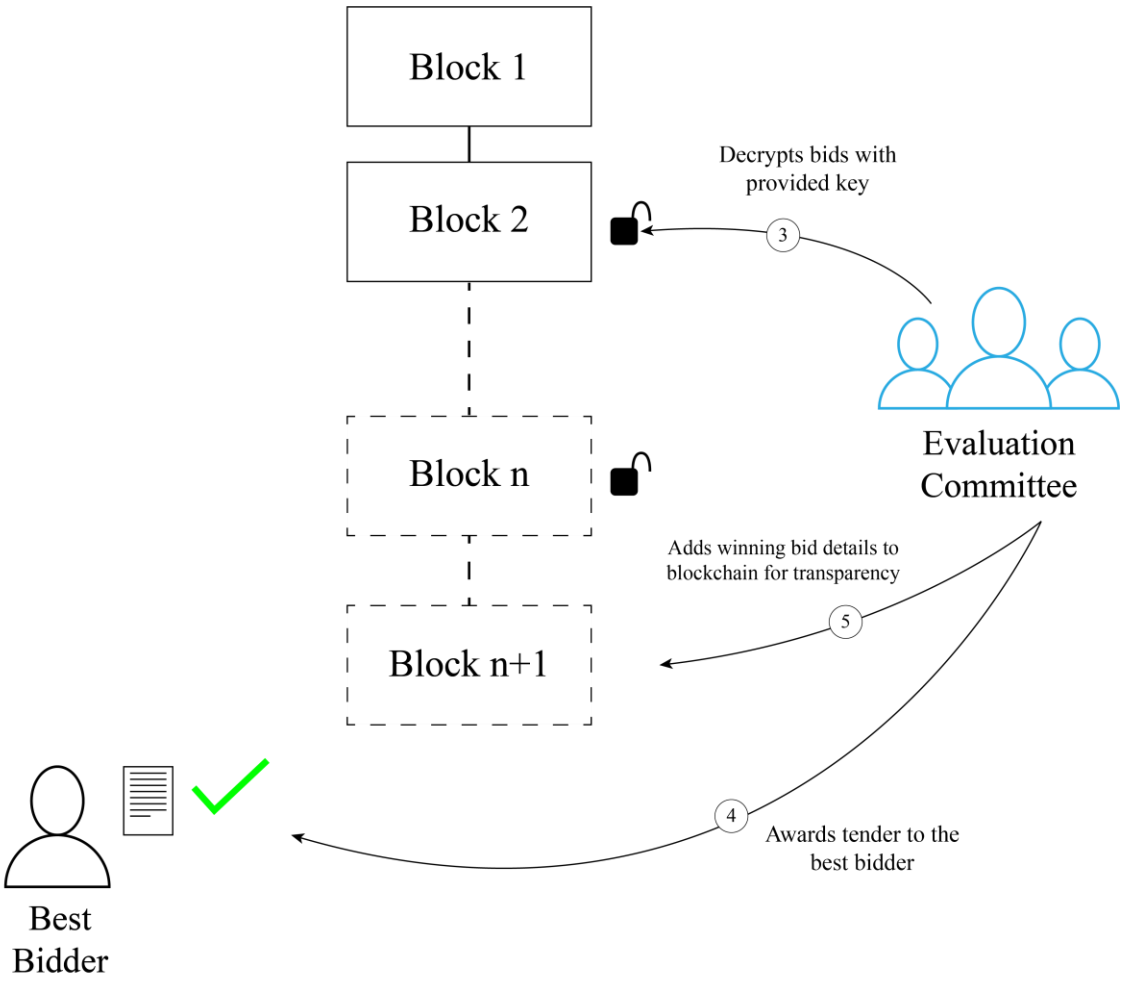

Figure 4.2: Tender evaluation mechanism by the evaluation committee after submission deadline

After deadline is reached, additional bid submissions are discarded. And the blockchain is now available for the selected evaluation committee personnel to view and download. They decrypt the blocks with the keys they receive after the deadline and proceed to review the

details and evaluate each bid details accordingly to the evaluation criteria set by the authority. After the evaluation process is complete and a best bidder is selected by the committee, the tender is awarded to the best bidder and the winning bid details are added at the end of the blockchain. This allows for ensuring transparency in the evaluation process.

4.2 System Design:

Data flow model: The main three functions of this system are authority committee, Bidders and the evaluation committee. These functions can easily be described by using context-level data flow diagram.

For the tendering authority committee, it deploys the tender to the blockchain and sets the deadline and evaluation time. The deadline can be changed if needed. On the other hand, evaluation time is an optional function for the tendering authority. It uses for the evaluation of the winning bid and making the whole blockchain public. Another function is maximum number of bids which helps to stop unnecessary bidding by the bidders and prevents DDS attack. The output of this function is tender deployment.

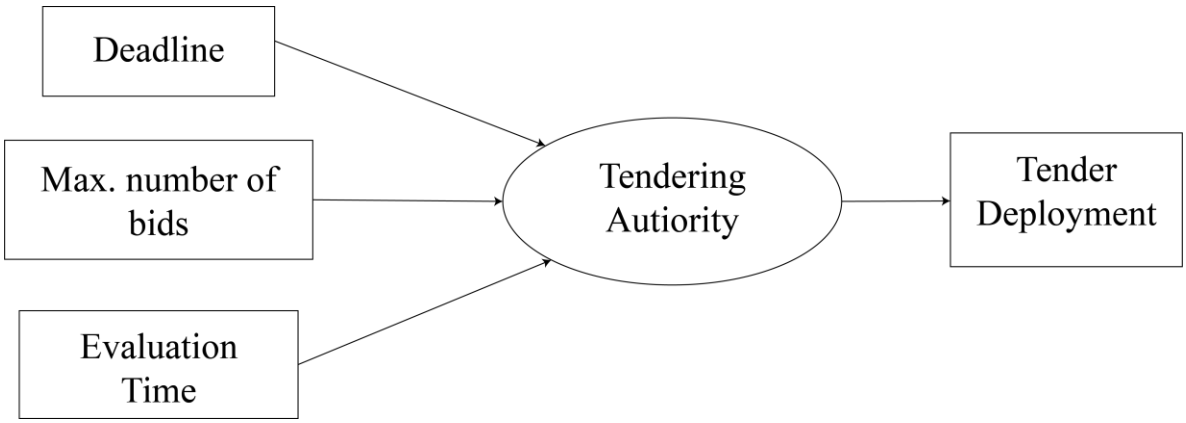

Fig 4.3: Context-level DFD for the Tendering Authority Function

The Bidders function is the mid-level and the most important functions among these three. The inputs of this functions are Maximum number of bids and tender deployment which are discussed earlier. Another input is required time which helps the bidders to check whether

the bidding window is available or not. The output of this function is submit bid which adds at the end of the blockchain as a block.

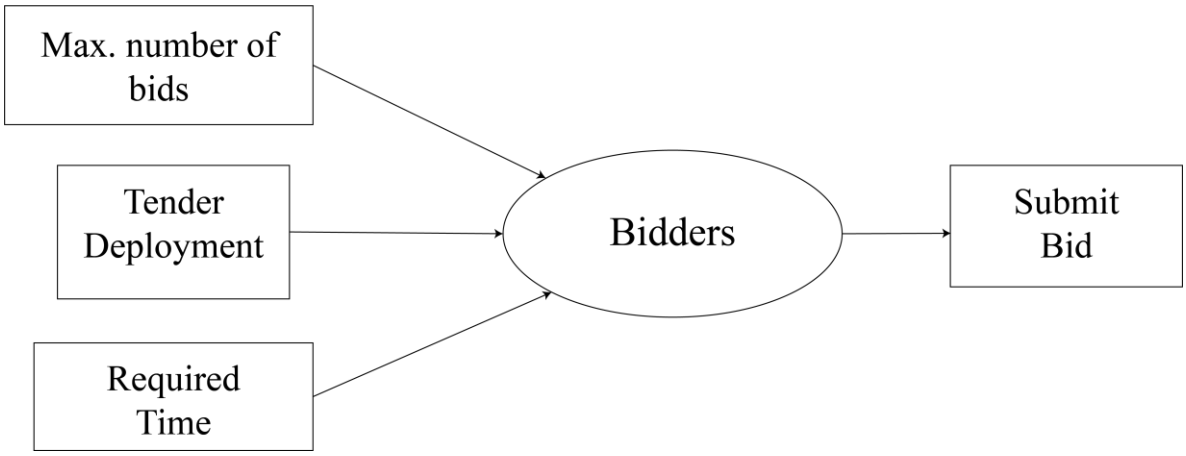

Fig 4.4: Context-level DFD for the Bidder function

The last function is evaluation committee, after the termination of bidding time it activates. It receives all the submitted bids as inputs. Another optional input is evaluation time which will be under consideration if the tendering authority sets it. The outputs of this functions are get number of bids, get all bids, evaluates winning bid and selects top bid, which all will help to decide which bid is the better of all the submitted bids and become winner.

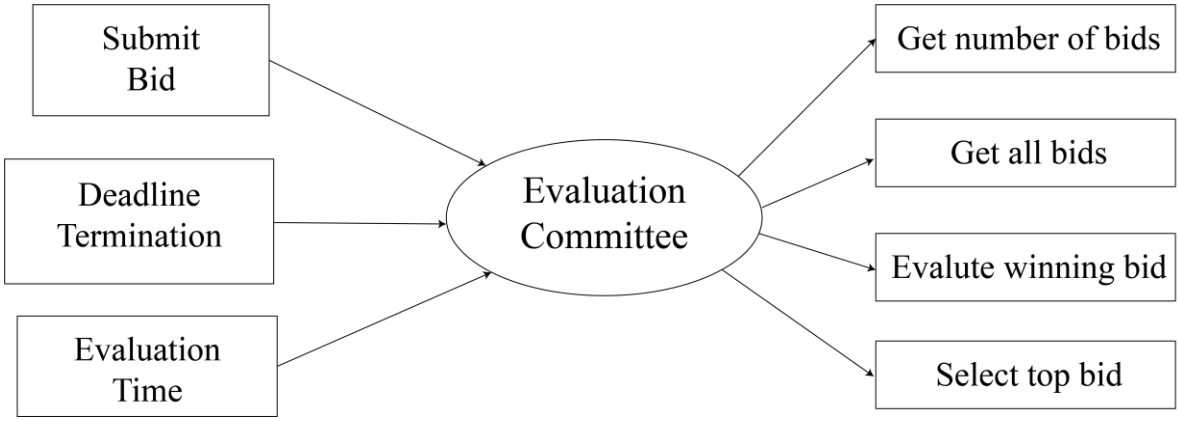

Fig 4.5: Context-level DFD for the Evaluation Committee function

Next to the context diagram is the level 1 data flow diagram for the Login and Registration process of the existing e-GP system. The content of Login and Registration DFD level 1 must be single process node from the context diagram and is broken down into sub processes in this level, the system must display or reveal further processing information.

The following are essential data to accommodate:

- User Records
- User Information
- Date of Logs
- Transaction Records

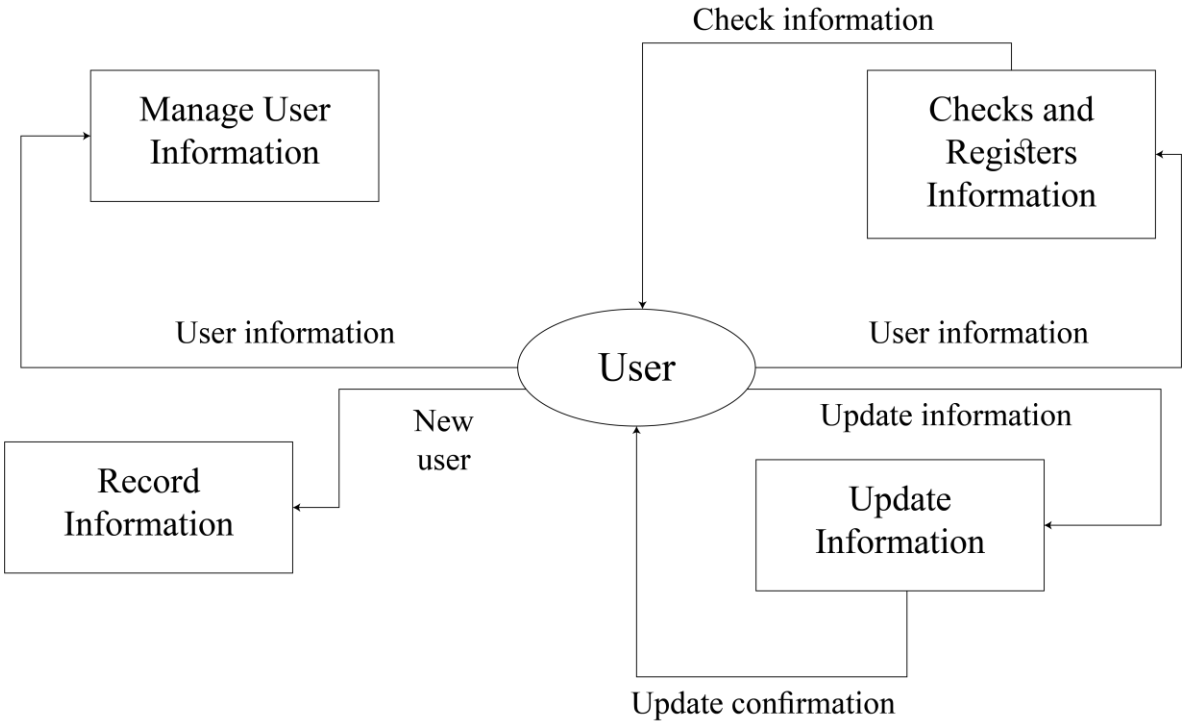

Fig 4.6: Level-1 DFD for the login and registration functions for e-GP system

The next topic is the data flow of the whole tendering process which can be illustrated easily by using level 2 DFD. First of all, the tendering authority deploys a tender by stating data such as description, bidding period, and so on. The tendering firm additionally supplies documentation such as its balance sheet and necessary license so that the bidding company

can check the tendering company's reliability before proceeding with the bidding. Then the tender is sent to the blockchain in the form of a smart contract. Once the tender details are finished and the documents are provided. Only companies that are willing to bid have access to the documents. The bidder registers in the system. The valid bidder is then shown all the tenders available for bid with tender details. The bidder then searches for appropriate tender for bidding. A prospective bidder can see the notice of the required tender from the blockchain. After that, the bidder submits a bid on the tender by submitting a quote with quotation clauses.

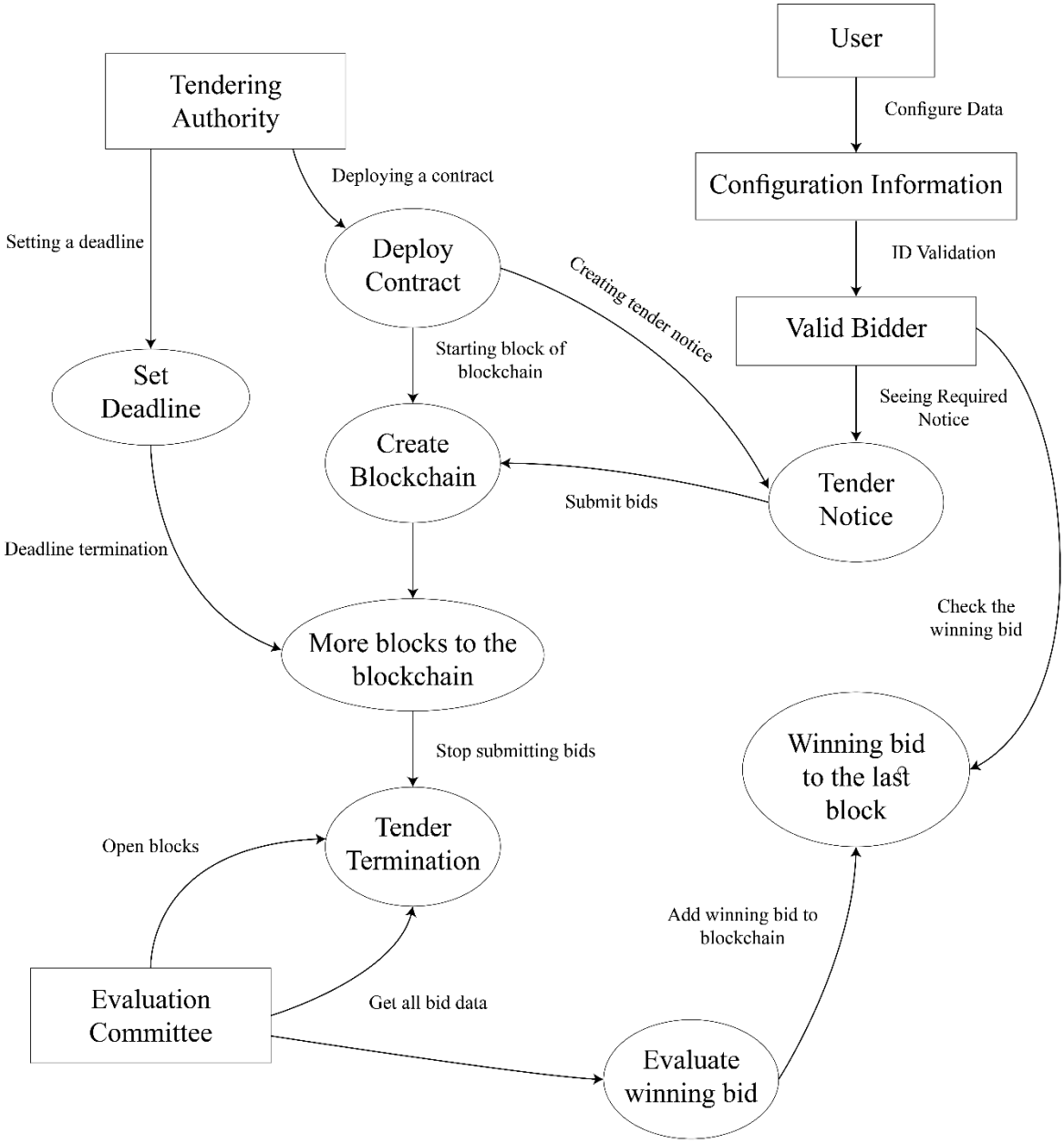

Fig 4.7: Level-2 DFD for e-GP tendering system

The bidder additionally establishes its legitimacy by submitting a balance sheet and a RoC (Registrar of Companies) license. The bidder can re-bid for the same tender if he can do any mistake for a fixed number of times. The bidder can only bid during the bidding period, and no additional bids will be accepted once the deadline has passed. When the deadline for bid submission expires, the smart contract on the blockchain stops accepting new bids. Then the evaluation can get the access of the all submitted bids, and the bid is evaluated and the credibility and capability of the bidding company are checked using the documents provided in the bid. The tendering organization can approve or reject the bid. The tender organization will push the results of the bid evaluations to the blockchain and make it public for the participated bidders and other users.

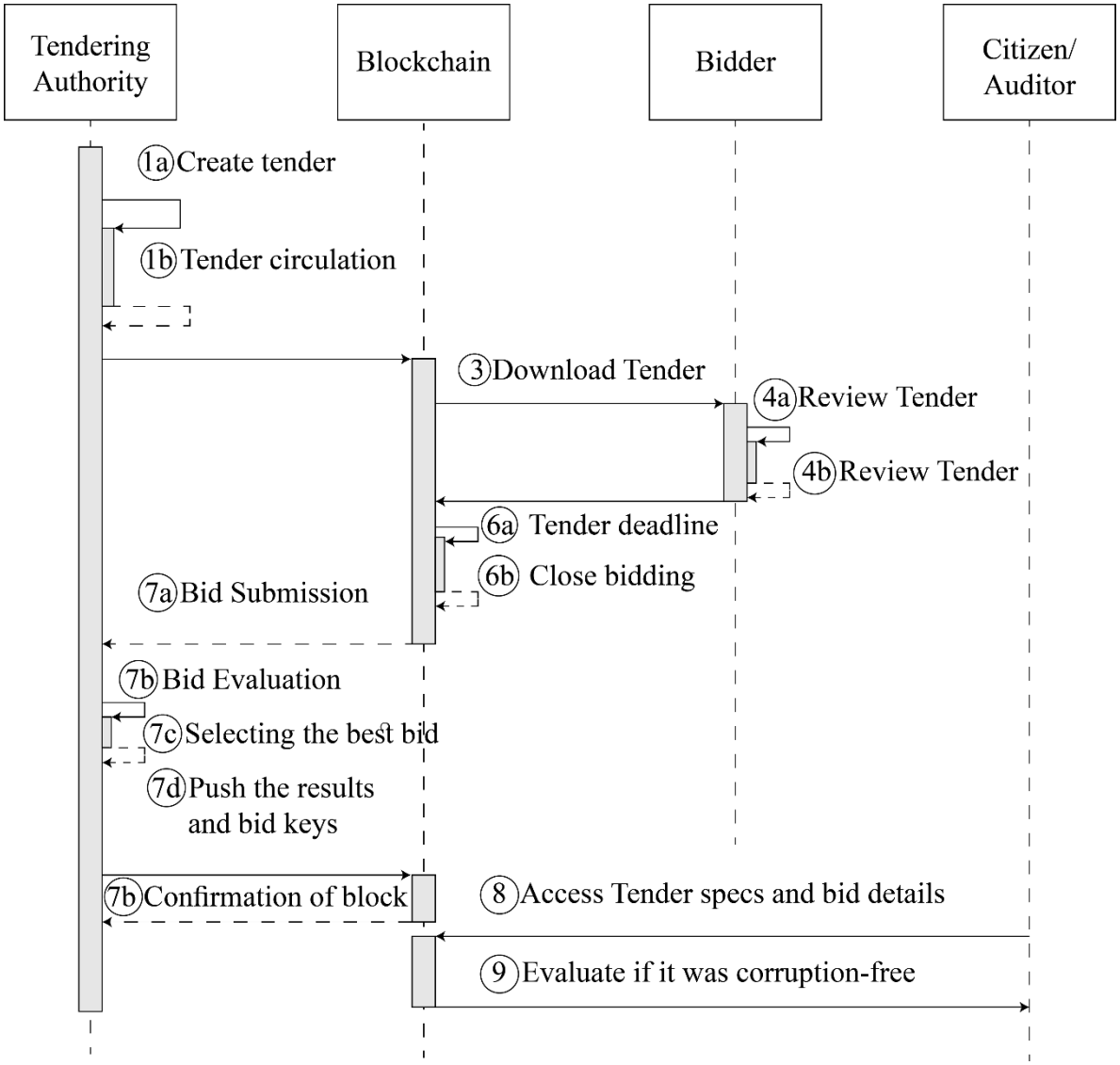

Figure 4.8: e-Tendering System Architecture

1. (a) Creating Tender: The tendering organization creates a tender by stating data such as description, bidding period, and so on. The tendering firm additionally supplies documentation such as its balance sheet and a RoC (Registrar of Companies) license so that the bidding company can check the tendering company's reliability and risk of default before proceeding with the bidding.

(b) Tender Circulation: Tender stipulation occurs, in which the tendering company explains its requirements as well as the evaluation criteria that will be used to evaluate a bid.

2. Opening the Tender: The tender is then sent to the blockchain in the form of a smart contract once the tender details are finished and the documents are provided. Only companies that are willing to bid have access to the documents, which are stored in an encrypted format in a database.

3. Tender Download: A prospective bidder can download the tender from the blockchain.

4. (a) Reviewing the Tender: The bidder registers asynchronously with the system. The bidder is then shown all the tenders available for bid with tender details. The bidder then searches for appropriate tender for bidding.

(b) Preparing Bid: After that, the bidder submits a bid on the tender by submitting a quote with quotation clauses. Bids are encrypted twice: first with the bidder's symmetric key and again with the tender's public key (address of tender on the blockchain).

5. Submitting Bid Details: The bidder additionally establishes its legitimacy by submitting a balance sheet and a RoC (Registrar of Companies) license, both of which are encrypted using the same key as previously stated.

6. (a) Tender Deadline: The bidder can only bid during the bidding period, and no additional bids will be accepted once the deadline has passed.

(b) Close Bidding: When the deadline for bid submission expires, the smart contract on the blockchain stops accepting new bids.

7. (a) Bids Collection: The tendering organisation can download the submitted bids, and they can decrypt the bids if they have full private key.

(b) Bid Evaluation: Then the bid is evaluated and the credibility and capability of the bidding company are checked using the documents provided in the bid. The tendering organization can approve or reject the bid.

(c) Selecting the Best Bid: At the tender closing date, tendering organisation will run the evaluation code and select the best bid.

(d) The tender organization will push the results of the bid evaluations along with bidder's keys to the blockchain. This information is crucial for independent auditing of the tendering process.

8. Access Tender Spec and Bids: Citizens or auditors can access the tender details from the blockchain (where this data will reside in perpetuity) along with the bid evaluation criteria.

9. Downloading Evaluation Criteria: Citizens or auditors can download the tender contract and details about the evaluation criteria. And can analyze if the tendering process was clean.

4.3 System Components

4.3.1 Smart Contracts

Smart contracts are at the heart of any Ethereum-based blockchain application system. Smart contracts are the non-deterministic, verifiable and self-running pieces of code that keep the blockchain running and functional. Smart contracts can also keep storage. Our proposed model relies on the functionality of smart contracts to do most of the work in the blockchain. As stated before, both the tender advertisement and the individual bids both will work through the smart contracts. This necessitates two different types of smart contracts to be used – one for tender which will initiate the blockchain, another for bids which will be added as extra blocks later in the chain.

As per the basis of our design, the smart contract for initiating the blockchain will have to contain certain details such as – time limit, a public key, and a limit for number of submissions by a bidder. The last detail is important to ensure attacks like DdoS cannot be carried out by overloading the blockchain with unusual amounts of bid submissions.

Algorithm 1 Initiating a tender by smart contract

1: procedure ReqForTender(*_length, _publicKey, _limit*)
2: *biddingEnd* ← *TimeNow*( ) + *length*
3: *limit* ← *_limit*
4: *publicKey* ← *_publicKey*

This smart contract will initiate the blockchain with the tender details from the tendering authority. The length will determine how long the bid submission process will go on. The TimeNow() takes the UNIX epoch time for the start value and adding an appropriate length value will determine the deadline within which all the bids will need to be submitted. After this time is crossed, no other bids will be accepted and no more blocks will be added to the blockchain.

For facilitating the bidding process by individual bidders, a separate design of smart contract is needed. This smart contract will enable the bidding process by receiving bids and adding the bid details to the blockchain. This smart contract will store data about the details necessary to identify and evaluate a bid for the tender. Now, a tender can be of many different varieties depending on the kind of procurement it is part of. That leaves a room for a lot of customization necessary in the design of this smart contract as the smart contract will carry and store the data about that particular bid. This means this smart contract can be wildly different based on the use case but a rather generalized version based on our proposed model will be something along the lines of this:

Algorithm 2 Smart contract for placing a bid on the blockchain

1: procedure PLACEBID(*id; data; msgHashed, v, r, s*)
2: *bidValidity* ← *ValidBid(id,msgHashed,v,r,s)*
3: if *bidValidity* then
4: *bidCount[id]* + = 1
5: *bid* ← new Bid(*id,data,bidValidity,bidsPlaced,biddingEnd*)
6: *bidsPlaced.add(bid)*
7: return *bid*
8: procedure VALIDBID (*id, msgHashed, v, r, s*)
9: *validHash* ← *verify*(*msgHash, v, r, s*)
10: *validTime* ← *timeNow*() < *biddingEnd*
11: *allowedBid* ← *bidCount[id]* < *limit*
12: return *validHash* and *validTime* and *allowedBid*
13: procedure BID(*_id, _data, _validity, _bidsPlaced, _biddingEnd*)
14: *id* ← *_id*

```
15:  data ← _data
16:  validity ← _validity
17:  bidsPlaced ← _bidsPlaced
18:  biddingEnds ← _biddingEnd
```

This smart contract does a few functionalities such as verifying if a placed bid is valid, then proceeding to submit the bid if it verifies, and finally storing the data about the specific bid in the blockchain.

4.3.2 Encryption

Encryption is an important part of our proposed model as it will play a key role in maintaining the security of the bidders' data during the bid submission process. The key functionality of not revealing the bid details until the deadline is reached will keep incidents like modifying the tender details to favor certain candidates.

The encryption method proposed to be used in the model is fairly simple. As stated before, each initial block of the blockchain which is the tender advertisement itself, will hold a public key of the tendering authority. After each bid is placed on the system, a symmetric key will be generated. The private key will be encrypted by the public key to generate a new encrypted version. This resultant key will be needed to access the bid details on the blockchain.

This is where the necessary functionality will be achieved, by withholding the complete key needed to access a block or bid from any of the authority or the evaluation committee until the deadline time has been reached. Only half of the key will be stored directly with the bids in the chain. The other half will be communicated to the suitable evaluation committee only prior to their evaluation phase. The recommended encryption method to meet the standard of cryptography configuration is to follow standards such as at least 256 bits.

4.3.3 Blockchain Network Nodes/Miners

Any blockchain application on the Ethereum platform uses a proof-of-work mechanism to verify and maintain the validity of the blockchain. This requires a network of nodes or miners to keep the system running through sharing resources. This is an important part of any system built on blockchain platforms. The nodes participate in finding the proof-of-work consensus in exchange of Ethereum cryptocurrencies which is commonly referred to as Ethereum GAS cost in running the blockchain.

Blockchain provides incredibly secure and easily scalable systems. But all this comes at a cost. The computational cost of a blockchain is determined by how much GAS cost is necessary to run the system without dropping any blocks. Too high amount of GAS cost can cause the nodes to reject a block before being processed which keeps it from being added to the blockchain. This is an important consideration before developing the technique how viable it would be to process the amount of data we need to process for the tendering process to be fruitful.

4.4 System Entities

4.4.1 Tendering Authority/Organization

The tendering authority or the tendering organization (TO) is the authority that launches the tendering process as part of any procurement. The procurement may be of any kind including of any products, services or projects. This may be a govt or private organization. The tendering authority holds the ability to initiate a tender with their desired specifications and hence launch the blockchain.

4.4.2 Bidders

The bidders are the people who participate in the tendering competition by submitting bids matching or even beating the specification standards set by the tendering authority. Bidders aim to win the tender competition by providing the best possible service or products within the cost constraints set by the tendering authority. Generally the bidder offering best cost-to-

performance ratio wins the tender. In our system the bidders will place bids through the smart contract which will then be added to the blockchain as the bidding process carries on.

4.4.3 Evaluation Committee

The Public Procurement Act of our Bangladeshi constitution states the necessity of evaluation of procurement tender process by human interaction. Evaluation has to be done by human conduct to stay within the legal boundaries. The evaluation committee will come into the picture in the stage after the deadline has been crossed. They will be able to access and and download the blockchain with all the bid details after the submission deadline expires. They will then follow a guideline for evaluation criteria and choose the best bidder to award the tender.

4.4.4 General Citizens/ Auditors

To ensure transparency we need to ensure that information about how the tendering process went on should be publicly accessible or at least accessible to reliable auditors who can analyze the data and detect biasness or corruption in the simplest way possible. Hence the final piece of our big picture will be a third party completely disconnected from the other parts of the tender processing, but with access to the data regarding the tender evaluation and awarding. In our proposed system, they will have access to the data of the winning bids as well as other bids to compare and determine if proper evaluation criteria have been followed during the process. They will gain access once all the steps of tender evaluation are finished and tender is awarded to someone.

4.5 Implementation

Based on the proposed model after all considerations, we prototyped a functional tender processing model having the entities. The prototype we devised can function as a complete tender processing system having functionalities such as tender submission, adding to the blockchain as new bids, tender evaluation. The prototype also features an encryption system to aid the tendering process. The choice of platform for developing the blockchain application is Ethereum for its open source nature. We need the open source nature for the latter part of our proposed model to work as the transparent publicly accessible data source.

Because Ethereum is a well-known open source publicly available blockchain technology with the capability of creating and running smart contracts in a specific virtual machine, it was chosen to implement the tender prototype Ethereum Virtual Machine (EVM). For the prototype, Ethereum test environment Ganache is used from the Truffle Suite. This is a local runtime environment simulating an Ethereum blockchain and the codes are written in the Solidity language. Ether is used as cryptocurrency. The miners who run the Ethereum nodes must be paid in "Gas" for the costs of deploying and running smart contracts. The amount of gas consumed is determined by the transaction's computational cost. The reason for calculating the amount of gas consumed in each transaction is that every node in the blockchain validates it. Verification on the network would be slow and result in a processing bottleneck if transactions were allowed to be arbitrarily complex. Miners will utilize the cost of gas, as placed on each transaction by the node that pushed it, to assess whether or not it is worth include the transaction in the block that they are mining. Trying to push a transaction that is too complex or has a low gas cost will cause the miners to disregard it when deciding which transactions to include in their block. This is irrelevant for the prototype because a local blockchain (Rinkeby Ethereum Network) is used.

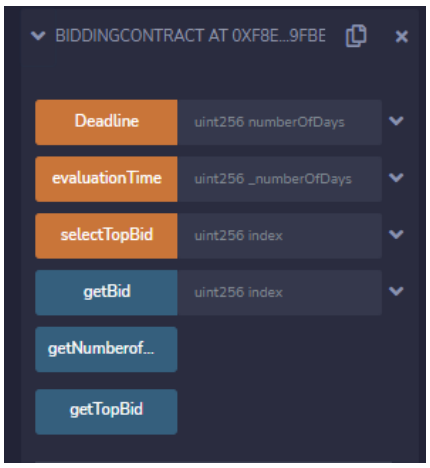

Fig 4.9: Tendering Authority Interface

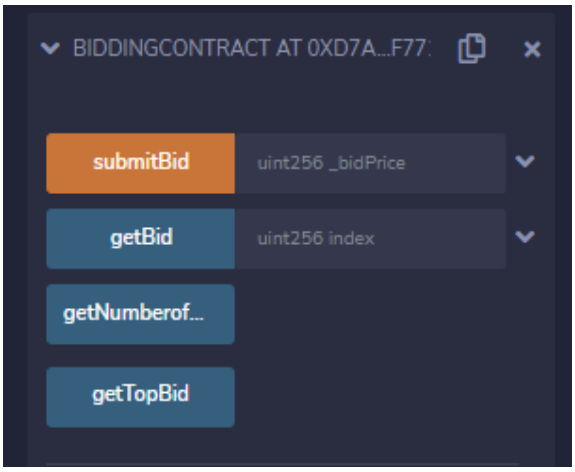

Fig 4.10: Bidder Interface

The testing network is used for all operational performance tests. This network was chosen because, according to public documentation, it was the most similar to the Ethereum production environment at the time of testing. "Rinkeby Faucet" is the blockchain explorer used to interface with the contracts, with an account created via the MetaMask browser plugin. However, these are costs that should not be overlooked for productive use on the

Ethereum Mainnet. Furthermore, the Proof of Work consensus algorithm's use raises serious concerns about its energy consumption. The node attempting to push a block into the larger chain must exert some effort due to this limitation. This prevents a node from adding any number of blocks to the chain because doing so would be computationally impossible. A malicious node would have to recompute every subsequent block with their new cryptographic hash quicker than every other participating node is working on their active chains in order to corrupt a block halfway down the chain and present it as the valid active chain. The figures 4.9, 4.10, 4.11 and 4.12 demonstrates the interfaces of the authority and the bidders and the details of an example transaction for each type of operation in Etherscan.

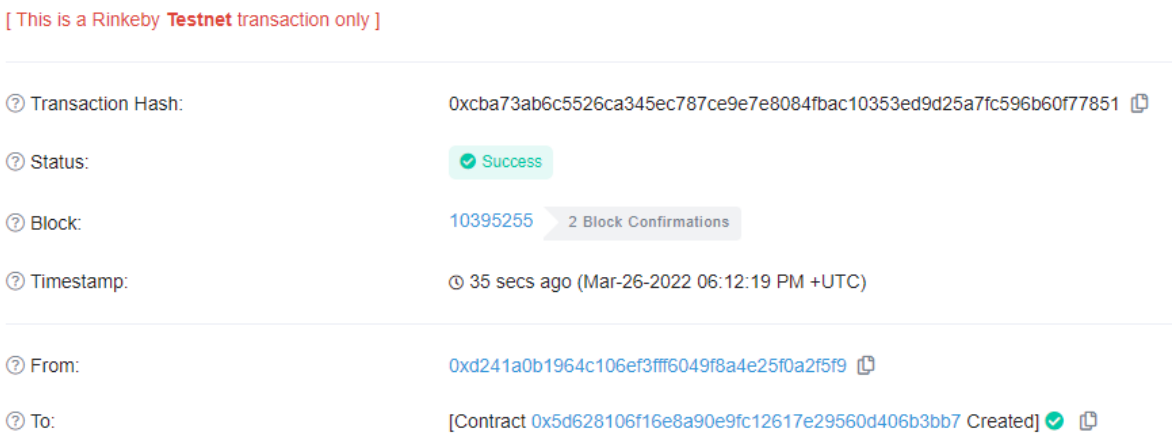

Fig 4.11: Deployment transaction details in Etherscan

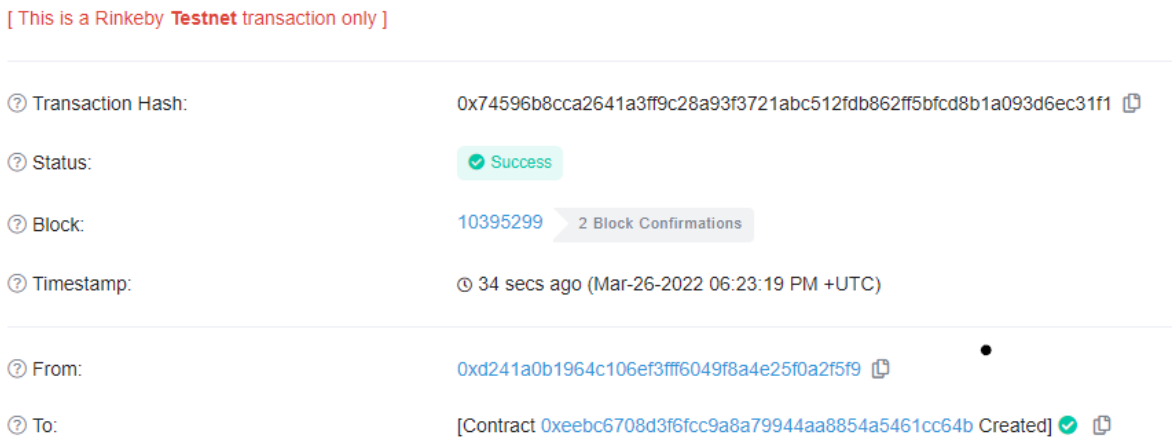

Fig 4.12: Bid submission transaction details in Etherscan

Chapter 5

Comparative Analysis

Our proposed model of tendering process is built on ideas from similar approaches on relevant works in this same domain. The main contribution was to develop something that is appropriate for application in our own national context without contradicting the constitutional laws. And at the same time improving on any of the existing systems that are functioning currently.

The performance metrics of a blockchain-based application model is generally drawn from the computational cost known as the GAS cost of hosting the blockchain. But this does not give a complete image of the impact because it is not actually a performance measurement rather a cost analysis, and the added security always comes at a little extra cost.

To focus on the actual real-life impact our proposed model can be expected to have, a comparative analysis can be drawn against the existing tender processing system and how our model improves on various sectors.

- The traditional tendering system has the possibility of data tampering to occur in between the tendering process. This traditionally happens in order to change tender details midway to favor certain candidates. Our blockchain-based model keeps this out of question as the tender cannot be modified after initiating in the blockchain. To change the details a totally new blockchain has to be initiated with new tender details.
- The existing system runs the risk of political influence or personal biasness to affect the tender evaluation process because there exists to transparency of the process. Our model emphasizes this issue particularly well by providing the means of publishing the tender evaluation related details.
- Any kind of cyber-attacks can alter the data of an entire tendering process to hamper the entire process. A blockchain-based model is free from the risk of data alteration as the proof-of-work consensus scheme will ensure the data is never altered during the process.

Chapter 6

Discussion and Conclusion

6.1 Summary

In this thesis work to find out a proposed tender processing model that uses blockchain, we explored concepts in the areas of Ethereum, smart contracts and the use of encryption technologies to aid in access control. The proposed technique used Ethereum smart contracts as a means for posting tenders and accepting bids. The model uses a symmetric encryption system along with the public key of the Ethereum account to aid in the access control portion of the model. Solidity was used as the programming language to write the smart contracts for the Ethereum network. The prototyping of the model was done using tools such as Ganache from Truffle and Metamask.

6.2 Conclusion

Change from any existing traditional system is always challenging and involves risk. And changing human lives directly by integrating newer technologies in real life human works is the best way to use technology. Tender processing from the recent and far past involved multi-million taka scams, political power-practice and ridiculous amounts of frauds. Transition to a newer but more secure system, although cumbersome, should be welcome nonetheless. Our work was a prototype of a solution for part of the problem. But in the bigger picture, much more work remains to be done for technology like this to be truly integrated into our national tendering system. We will hope we have been able to set the foundation to inspire future works to be built on this to make it truly a reality.

6.3 Future Work Recommendations

The prototyping of the model attempted by us has been proved to be functional and there is potential for larger scale version of this to be integrated into the mainstream tendering system of our national procurements. It is possible to build an actual functional version of this that can integrate with the existing digitized procurement platforms. This work would need a more in-depth study into the computational costs, and the additional challenges presented in a large scale scenario.

Ethereum 2.0 is to be a newer version of the Ethereum platform which will use a proof-of-stake consensus mechanism instead of the current proof-of-work consensus. Ethereum 2.0 will introduce more security technologies and give more control on the application development side of the blockchain platform. This can transform the work done in this thesis into an even better version with improved performance metrics and can potentially address the existing situation even better.

References

[1] Pramod, D.; Zachariah, B.; Salim, T. Moving Beyond Paperwork: Blockchain in Public Sector. Telecom Bus. Rev. 2019, 12, 50–55.

[2] Myeong, S.; Jung, Y. Administrative Reforms in the Fourth Industrial Revolution: The Case of Blockchain Use. Sustainability 2019, 11, 3971. [CrossRef]

[3] Krogsboll, M.; Borre, L.H.; Slaats, T.; Debois, S. Smart Contracts for Government Processes: Case Study and Prototype Implementation. In Financial Cryptography and Data Security; Springer: Kota Kinabalu, Malaysia, 2020; pp. 676–684.

[4] Williams-Elegbe, S. Public Procurement, Corruption and Blockchain Technology in South Africa: A Preliminary Legal Inquiry. In Regulating Public Procurement in Africa for Development in Uncertain Times. Available online: https://ssrn.com/abstract=3458 877 (accessed on 29 June 2021).

[5] Hardwick, F.S.; Akram, R.N.; Markantonakis, K. Fair and Transparent Blockchain based Tendering Framework—A Step Towards Open Governance. In Proceedings of the 17th IEEE International Conference onTrust, Security and Privacy in Computing and Communications (TrustCom), New York, NY, USA, 1–3 August 2018.

[6] Na, Li, And Chao, Ma. "Evolutionary Game Analysis on Supervision Of PPP Project Tender" 978-1-5386-1329- 0/18/$31.00 ©2018 IEEE.

[7] A. Dello, C. Yoshida. "Online Tendering and Evaluation for Public Procurement in Tanzania" 978-1-5090- 5504-3/17/$31.00 ©2017 IEEE SNPD 2017, June 26-28, 2017, Kanazawa, Japan.

[8] México: Pionero en Licitar con Blockchain. Bitcoin Mexico. 2018. Available online: https://www.bitcoin.com.mx/primeralicitacion-con-blockchain-en-mexico/ (accessed on 29 June 2021).

[9] Ledger Insights. Seoul District Using Blockchain for Public Procurement. Ledger Insights. 2019. Available online: https: //www.ledgerinsights.com/seoul-district-using-blockchain-for-public-procurement/ (accessed on 17 April 2021).

[10] Fair and Transparent Blockchain based Tendering Framework - A Step Towards Open Governance Freya Sheer Hardwick, Raja Naeem Akram, and Konstantinos Markantonakis ISG-SCC, Royal Holloway, University of London, Egham, United Kingdom.

[11] D. Jayasinghe, K. Markantonakis, and K. Mayes, Optimistic FairExchange with Anonymity for Bitcoin Users. IEEE Computer Society, 11 2014, pp. 44–51.

[12] R. Beck, J. S. Czepluch, N. Lollike, and S. Malone, "Blockchain-the gateway to trust-free cryptographic transactions." in ECIS, 2016, p. 153.

[13] S. Yin, J. Bao, Y. Zhang, and X. Huang, "M2m security technology of cps based on blockchains," Symmetry, vol. 9, no. 9, p. 193, 2017

[14] T. Jacobs, Blockchain: A Step-By-Step Guide For Beginners To Implementing Blockchain Technology And Leveraging Blockchain Programming (Volume 1). USA: CreateSpace Independent Publishing Platform, 2017.

[15] C. Cachin, "Architecture of the hyperledger blockchain fabric," in Workshop on Distributed Cryptocurrencies and Consensus Ledgers, 2016.

[16] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou, "Hawk: The blockchain model of cryptography and privacy-preserving smart contracts," in Security and Privacy (SP), 2016 IEEE Symposium on. IEEE, 2016, pp. 839–858.

[17] K. Bhargavan, A. Delignat-Lavaud, C. Fournet, A. Gollamudi, G. Gonthier, N. Kobeissi, N. Kulatova, A. Rastogi, T. Sibut-Pinote, N. Swamy et al., "Formal verification of smart contracts: Short paper," in Proceedings of the 2016 ACM Workshop on Programming Languages and Analysis for Security. ACM, 2016, pp. 91–96.

[18] L. Luu, D.-H. Chu, H. Olickel, P. Saxena, and A. Hobor, "Making smart contracts smarter," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 254–269.

[19] A. J. Menezes, Elliptic curve public key cryptosystems. Springer Science & Business Media, 2012, vol. 234.

[20] Blockchains and Smart Contracts for the Internet of Things KONSTANTINOS CHRISTIDIS, (Graduate Student Member, IEEE), AND MICHAEL DEVETSIKIOTIS, (Fellow, IEEE).

[21] N. Szabo. (1994). Smart Contracts. [Online]. Available:http://szabo.best.vwh.net/smart.contracts.html

[22] Z. Zheng, S. Xie, H.-N. Dai, H. Wang, Blockchain challenges and opportunities: A survey, Internat. J. Web Grid Serv. (2016).

[23] L. Luu, Y. Velner, J. Teutsch, P. Saxena, Smart pool: Practical decentralized pooled mining, in: USENIX Security Symposium, 2017.

[24] V. Buterin, Ethereum: A Next-Generation Smart Contract and Decentralized Application Platform, 2014.

[25] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf, 2008.