5-1-2023

SECURITY RESEARCH FOR BLOCKCHAIN IN SMART GRID

Lanqin Sang
Southern Illinois University Carbondale, lanqinsang@msn.com

SECURITY RESEARCH FOR BLOCKCHAIN IN SMART GRID

by

Lanqin Sang

B.E., National University of Defense Technology, China, 1986
M.S., Southern Illinois University Carbondale, 1998
M.S., Southern Illinois University Carbondale, 2000

A Dissertation
Submitted in Partial Fulfillment of the Requirements for the
Doctor of Philosophy Degree

School of Computing
in the Graduate School
Southern Illinois University Carbondale
May 2023

DISSERTATION APPROVAL

SECURITY RESEARCH FOR BLOCKCHAIN IN SMART GRID

by

Lanqin Sang

A Dissertation Submitted in Partial

Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in the field of Computer Science

Approved by:

Henry Hexmoor, Chair

Bidyut Gupta

Ahmed Imteaj

Koushik Sinha

Ning Yang

Graduate School
Southern Illinois University Carbondale
March 27, 2023

AN ABSTRACT OF THE DISSERTATION OF

Lanqin Sang, for the Doctor of Philosophy degree in Computer Science, presented on March 27, 2023, at Southern Illinois University Carbondale.

TITLE: SECURITY RESEARCH FOR BLOCKCHAIN IN SMART GRID

MAJOR PROFESSOR: Dr. Henry Hexmoor

Smart grid is a power supply system that uses digital communication technology to detect and react to local changes for power demand. Modern and future power supply system requires a distributed system for effective communication and management. Blockchain, a distributed technology, has been applied in many fields, e.g., cryptocurrency exchange, secure sharing of medical data, and personal identity security. Much research has been done on the application of blockchain to smart grid. While blockchain has many advantages, such as security and no interference from third parties, it also has inherent disadvantages, such as untrusted network environment, lacking data source privacy, and low network throughput.

In this research, three systems are designed to tackle some of these problems in blockchain technology. In the first study, Information-Centric Blockchain Model, we focus on data privacy. In this model, the transactions created by nodes in the network are categorized into separate groups, such as billing transactions, power generation transactions, etc. In this model, all transactions are first encrypted by the corresponding pairs of asymmetric keys, which guarantees that only the intended receivers can see the data so that data confidentiality is preserved. Secondly, all transactions are sent on behalf of their groups, which hides the data sources to preserve the privacy. Our preliminary implementation verified the feasibility of the model, and our analysis demonstrates its effectiveness in securing data source privacy, increasing network throughput, and reducing storage usage.

In the second study, we focus on increasing the network's trustworthiness in an untrusted network environment. A reputation system is designed to evaluate all node's behaviors. The reputation of a node is evaluated on its computing power, online time, defense ability, function, and service quality. The performance of a node will affect its reputation scores, and a node's reputation scores will be used to assess its qualification, privileges, and job assignments. Our design is a relatively thorough, self-operated, and closed-loop system. Continuing evaluation of all node's abilities and behaviors guarantees that only nodes with good scores are qualified to handle certain tasks. Thus, the reputation system helps enhance network security by preventing both internal and external attacks. Preliminary implementation and security analysis showed that the reputation model is feasible and enhances blockchain system's security.

In the third research, a countermeasure was designed for double spending. Double spending is one of the two most concerned security attacks in blockchain. In this study, one of the most reputable nodes was selected as detection node, which keeps checking for conflict transactions in two consecutive blocks. Upon a problematic transaction was discovered, two punishment transactions were created to punish the current attack behavior and to prevent it to happen in future. The experiment shows our design can detect the double spending effectively while using much less detection time and resources.

ACKNOWLEDGMENTS

First and foremost, I give thanks to God who gave me the courage, wisdom, and strength to start and continue my PhD study journey. I also deeply appreciate my parents, who valued our education dearly and supported us with all they could.

I would like to express my deepest gratitude to my advisor, Dr. Henry Hexmoor, who gave me the opportunity to join his team for my PhD research. Throughout the years of working together on the research, he has constantly provided professional guidance, advice, and continuous support. I would not have been able to finish my research projects and dissertation without Dr. Hexmoor's help and encouragement.

I am grateful to Dr. Bidyut Gupta for serving on my committee. I took many classes with Dr. Gupta during my PhD study, and I learned so much from him. I would also like to express my deep appreciation to Dr. Ahmed Imteaj, Dr. Koushik Sinha, and Dr. Ning Yang for serving on my graduate committee and for their valuable feedback on my proposal and dissertation.

I am also thankful to Dr. Norman Carver, who gave me the opportunity to pursue my PhD study and gave me the needed help whenever I asked. Ms. Andi Russell has been very helpful and kind to me, which is also greatly appreciated.

Lastly, I would like to thank my husband, Yuqing Hou, who has been very supportive through the years during my study, and my three wonderful daughters, who are forever my inspiration for personal growth.

DEDICATION

GOD, my mother, and my father, my husband, my three daughters, and myself for not GIVING UP.

TABLE OF CONTENTS

LIST OF TABLES

TABLE PAGE

LIST OF FIGURES

FIGURE PAGE

LIST OF ALGORITHMS

ALGORITHM PAGE

CHAPTER 1

INTRODUCTION

1.1 Smart Grid

Smart grid is an infrastructure that uses digital computation and communication technologies to transform the conventional centralized grid into a more responsive, efficient, and intelligent energy access and delivery network system [1]. People have realized that our dependence on fossil fuel energy and accelerating climate change must be stopped to keep us on a sustainable path. Thus, it is essential that various renewable energy sources be developed. The goal of the transformation is to reform the energy landscape by integrating and utilizing such renewable energy resources and reducing the dependence on fossil fuel energy. While the conventional legacy grid serves consumers via centralized long-distance transmission lines, the future smart grid paradigm brings energy producers and consumers together in a distributed manner. See Figure 1.

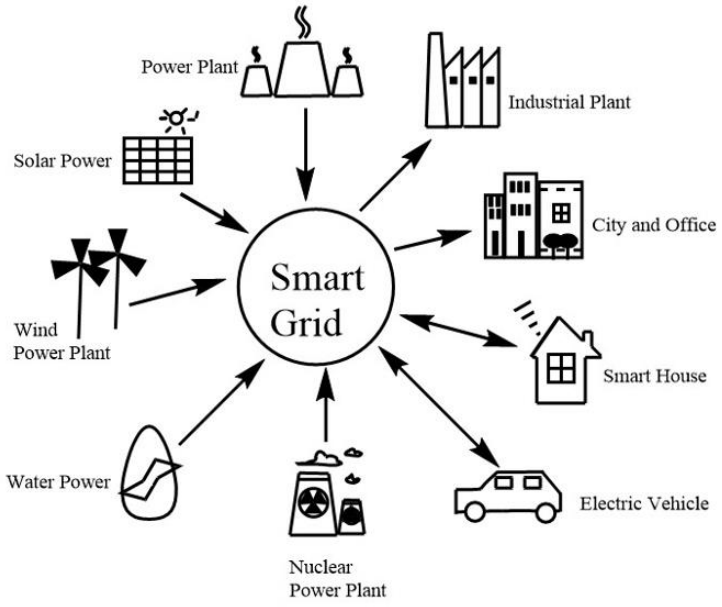

Figure 1. Smart Grid

1.2 Blockchain

1.2.1 Blockchain Definition

Blockchain technology, introduced in [2], is a distributed ledger or distributed database technology [3], [4], [5], which is a consensus of replicated, shared, and synchronized digital data [1]. Blockchain technology is a secure, decentralized, trusted cyber infrastructure, where multiple entities in the network can create, maintain, and store a chain of blocks. It includes fault-tolerant consensus algorithms, smart contracts, public-key cryptography, peer-to-peer networks, and database management technologies that form a low-cost, secure, and efficient operational management system [6].

In blockchain, as shown in Figure 2, each block registers different records of data or transactions. A record of data or a transaction is the message that is sent by one node to other nodes. Each transaction contains the address of the recipient, transaction data payload, and a transaction value. Transactions are signed by the sender's private key and the state of the network is changed only by such transactions [5].

A block may include timestamp, nonce, a hash tree named Merkle tree [7], hash, smart contract scripts, and so on [8],[9]. Merkle Tree is used to efficiently summarize and verify the large sets of data in a block. Markle Trees are suitable for devices that do not have enough space to store the entire blockchain to search and verify data quickly. The timestamp in each block, which shows the time of block creation [10], creates a source of variation for block hash, and makes tampering a blockchain more difficult. The hash is the cryptographic hash of the block, all blocks are attached together, and each block references the hash of the previous blocks. The hash and Merkle Tree are to ensure data integrity by allowing the verification that the content of the blocks is intact [4].

Each node in the network continuously adds newly validated blocks to its local chain at regular intervals. Therefore, the blockchain's data is spread across numerous sites, countries, and institutions geographically and can be accessed by all participants of the network. To alter any block's content, all the block's corresponding data must be changed due to the mechanism that each block has the previous block's hash. In a linear chain-based structure, each bock only connects to the previous block. If each block references multiple previous blocks, it is called Directed Acyclic Graph (DAG) [1]. Blockchain technology creates redundant systems, which is why it is resilient to single point failure and cyber-attacks [11].

1.2.2 Blockchain Types

There are primarily two main types of blockchains - private blockchain and public blockchain, and one major variation - consortium blockchain.

1. Public Blockchain. Public blockchain is a non-restrictive distributed ledger system. Anyone

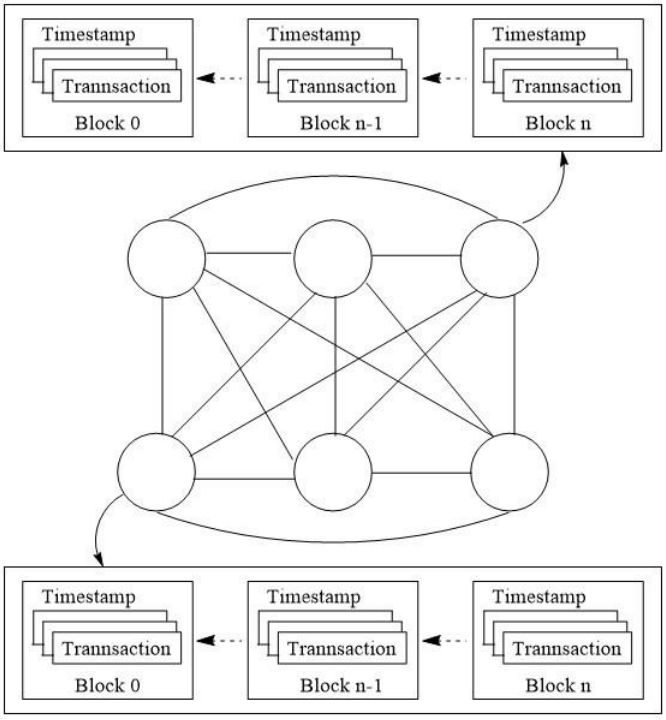

Figure 2. Schematic Representation of Blockchain

can join the blockchain network and the nodes do not need authentication by others [10]. All nodes are authorized to create blocks, access any records in the network, verify transactions for an incoming block, and save blocks, which makes the entire blockchain system completely open and transparent to all participating nodes. The nodes in a public blockchain is not trusted. However, a public blockchain is trustable, and it is much safer than private ones due to thoughtful cryptogenic encrypting methods, the consensus mechanism that make every transaction and block legitimate, a larger network, and distribution of records, which thwarts hackers from invading the network. A public blockchain is typically a huge network with a lot of nodes. For every node to verify a transaction and do proof-of-work is time-consuming, which causes low TPS (Transactions Per Second) and higher energy consumption [4]. Public blockchain's slow rate of processing and completing transactions also causes scalability issues. The most basic use of public blockchain is for mining and exchanging cryptocurrencies, such as Bitcoin and Ethereum [12].

2. Private Blockchain. Private blockchain is a restrictive blockchain that is usually created within an organization and only selected members can join the network. The controlling organization is in charge of the security, authorization, and accessibility. Because of the limited number of nodes, a private blockchain has a much higher TPS rate and consumes less energy. Both the consensus process of a block and adding new transactions to a block are fast [13]. Because private blockchains are relatively small, they have good scalability. Furthermore, since the nodes need to build trust in order to transmit confidential information within the network, the network runs a higher risk of security breach. Private blockchains need a control unit and Identity Access Management (IAM) system to monitor

and administer rights, which defies the concept of decentralization and contradicts the idea of blockchain technology. Private blockchain networks are deployed for voting, supply chain management [12], such as Hyperledger Fabric [14].

3. Consortium Blockchain. Consortium blockchain is semi-decentralized and managed by more than one organization [12]. Consortium blockchain integrates the features of public and private systems, with open consensus for public systems and centralized control for private systems. Consortium blockchains are typically used by companies to create a broad-based platform for sharing data with stakeholders [15], [16]. The key features of these blockchain taxonomies are summarized in Table 1 [4], [10], [13], [17].

1.2.3 Blockchain Features

Blockchain has the following desirable characteristics [1], [13], [18], [19].

1. Decentralization. Blockchain networks are managed by decentralized nodes through consensus protocols. Such networks run in a peer-to-peer manner without trusted centralized management units.
2. Scalability. The decentralized nature of blockchain network dictates that many nodes can join the system to scale up the network.
3. Trustless but Secure System. Blockchain network is a trustless but secure system. Individual nodes in blockchain network are not trusted. All transactions are encrypted with asymmetric cryptography. A block can only be permanently saved if it passes the consensus mechanism.
4. Immutability. Since blockchain technology utilizes cryptographic techniques, and a block-hash connects the chain structure and maintains a global ledger that is synchronized among

the nodes, it is extremely difficult to alter the contents of the blocks unless the majority of the nodes have become malicious.

5. Transparency and Auditability. The blocks in blockchain are open to any nodes in the blockchain network. The nodes can audit the blocks and verify the authenticity of the transactions to assure that the blocks are not altered.
6. Resiliency. The resiliency comes from blockchain's decentralization, which can avoid single point failure. Storing the entire chain by every node in the blockchain network makes blockchain a fault-tolerant network. Any fault or malicious activities can be identified and corrected quickly.
7. Smart Contracts. Introduced by Nick Szabo [20], [21], smart contract is a computerized protocol that can execute the terms and conditions of an agreement [1]. They are securely deployed inside the blockchain in the same fashion as transactions. As a computer script, a smart contract records the conditions and events, such as an asset's target value, an ending date, or transaction information. When a smart contract is deployed in the blockchain, if the condition is reached or an event triggers, the contract will be executed automatically without any centralized intervention[22]. Ethereum [8] is the most popular smart contract platform based on blockchain [1], [5]. Smart contract's ultimate goals include elimination of trusted intermediaries, less human intervention, reduction of enforcement costs, and eradication of intentional or unintentional fraud and security risks.

1.2.4 The Attacks and Problems of Blockchain

Blockchain, although considered a safe technology, is still vulnerable to certain types of threats that are associated with PoW and PoS protocols. [1], [10], [23], [24], [25].

Table 1. Different Features of blockchain Classifications

Item	Public	Private	Consortium
Join	Permissionless	Strictly Permissioned	Permissioned
Governance	None	Managed by one administrator	Managed by a set of participants
Access right	Any node can read, write, leave, and join	Only authorized nodes	Only a set of authorized nodes
Anonymity	Yes	No	No
Nodes	Not trusted	Trusted	Trusted
Consensus mechanism	Proof of Work (PoW), Proof of Stake (PoS)	Multi-part voting	Strictly pre-approved node's voting
Cryptocurrency	Bult-in cryptocurrency	No	No
Security performance	High security. 51% attack tolerance, hard to tamper, no finality	Low security. 33.33% attack tolerance, could be tampered, enabled finality	High-end security
Transactions per second (TPS)	Bitcoin 7 TPS; Ethereum network 15 TPS	VISA 24,000 TPS	1000 - 2000 TPS
Computation complexity	High	Low	Middle
Decentralization level	Highly decentralized	Highly centralized	Semi-centralized
Transaction Validation	Incentive-based mining by any node	List of authorized validators, no incentive required	
Energy consumption	High	Low	Middle
Scalability	Low	High	Middle

1. Attack of 51%. 51% attack occurs when one or a group of malicious nodes take control of 51% of processing power in the blockchain network.

   Distributed Denial of Services (DDoS) attack. Denial of Services (DoS) attacks aim to overwhelm the network services by inundating them with requests [26]. A protection

mechanism against in DDoS attack is to limit size of a block up to 1 MB and the size of each script up to 10000 bytes. Up to 20000 of the signatures can check and at maximum of the multiple signatures is 20 keys.

2. Double Spending attacks. Double Spending is the risk that a digital currency can be spent twice. It is a potential problem unique to digital currencies because digital information can be reproduced easily by savvy individuals who understand the blockchain network and the computing power necessary to manipulate it.
3. Sybil's attack. It is possible that one node has several fake identities, because the blockchain network cannot authentically distinguish the physical machines. The malicious node may fill the blockchain with fake users under its control, which can lead to 51% attack and/or double-spending attack.
4. Replay attack. Replay attack attempts to reuse transactions and replay them in order to increase the impact of the same transaction. By carrying out such attack, a malicious participant can claim it has been involved.

in a transaction that is profitable for itself multiple times. It can also be used to undermine honest participants.

5. Selfish mining attack. A selfish mining attack or block withholding attack attempts to discredit blockchain network integrity. Selfish mining attacks occur when an individual attempts to withhold a successfully validated block from being broadcasted to the rest network.
6. Eclipse attacks. In eclipse attack, an attacker capable of delaying information that a victim expects to receive can launch double spending attacks and selfish attacks.

7. Flash attack. In flash attack, an attacker is able to obtain a temporary majority of computing power by renting enough mining capacity. This would break the security assumption of classic PoW-based systems.
8. Content Poisoning Attack. A node's information was changed.
9. On-off attack. On-off attack refers to irregular behaviors of attackers, which means that malicious nodes can behave well or badly alternately in order to remain undetected while causing damage.
10. Cracking encryption. Quantum algorithms, such as 'Shora', make it possible to break the RSA encryption.
11. Blockchain consistency and system liveness. Distributed systems, such as blockchains, have a concept of correctness, which includes two parts: safety and liveness. Liveness means that something good will happen. In a blockchain consensus mechanism, liveness is the guarantee that all validators will agree on a value eventually. Safety is the guarantee that nothing bad will ever happen in the system. In terms of consensus, this means that no two processes/validators/actors will ever come up with different values.

1.2.5 Blockchain Challenges

Blockchain has the following challenges [27], [28], [29], [30].

1. High energy consumption. The main disadvantage of blockchain is its high energy consumption. Keeping a real-time ledger, all nodes communicating with each other, all miners validating transactions, all nodes giving extreme levels of fault tolerance and ensuring zero downtime, all nodes storing data on the blockchain, all of these actions burn electricity. Each node repeats the achievement of consensus, signing each transaction with a

cryptographic scheme needs high computing power, and creating more blocks due to fixed block size also cause high energy consumption.

2. Latency. For security reason, the blocks in blockchains are kept to a fixed size, around 1 MB. Therefore, some transactions must be processed later, which causes processing latency for these transactions.
3. Huge storage requirement. Every node in the blockchain network needs to store a copy of the blockchain, which causes high storage demanding.
4. Scalability. The blockchain consensus involves all nodes and takes a long time to accomplish. The more node joins the network, the longer time will be needed to finish the consensus procedure, which, in turn, seriously lowers the network throughput and performance.
5. A fault-tolerant network. A network that has resistance against availability attacks needs to be developed.
6. Privacy-preserving capacity. Blockchain lacks the capacity to leverage advanced privacy-preserving techniques to protect information disclosure and increase trust, transparency, and democracy among all the entities.

1.2.6 Consensus Mechanism

Consensus mechanism ensures the trust in the network, which is why it is a key component of blockchain technology. During consensus process, a set of validators reach an agreement whether a block is valid or not. Consensus algorithms are in both public and private blockchains. Every node can participate in the procedure in public blockchain, whereas only selected nodes can be a part of the process in private blockchain. Due to the limited number of validators, private blockchain has simpler consensus mechanism and better network performance

[1]. Depending on the specifications of blockchain applications, the components of a transaction block and consensus algorithms may vary. Below are some of the most popular consensus mechanisms.

1. Proof of Work (PoW). PoW is introduced in Bitcoin and is the first public blockchain consensus [2]. In PoW, the consensus nodes compete to solve a computationally expensive puzzle, named as PoW problem, which is hard to solve but easy to verify. Whoever solves the problem will attach the solution to a new block and broadcast the block across the network so the other nodes can verify its correctness. The process of cracking the puzzle and verifying the block is called mining. PoW attempts to prevent various kinds of cyber-attacks, but is vulnerable to 51% attack, where one or a group of malicious nodes may take control of 51% of the network's processing power. PoW suffers from inefficient throughput, high latency, and high energy consumption, which reduces its utility in other blockchain applications [1], [31], [32].
2. Proof of Stake (PoS). PoS [33], [34], [20] is an alternative mechanism of PoW, intending to overcome PoW's common limitations. In PoS-based blockchain, the blocks are commonly validated rather than mined. The algorithm randomly assigns validators to create new blocks, and the probability of a node selected to validate the next new block is proportional to the stakes/assets it owns. Since the wealthiest validators administer the blockchain, it inherently makes PoS mechanism unfair [1], [32].
3. Delegated Proof of Stake (DPoS). DPoS [35] is a variant of PoS. The main difference between PoS and DPoS is that PoS is directly democratic, while DPoS is delegated democratic. In DPoS, only a number of selected delegates can generate and validate the blocks. Since fewer nodes engage in the validation process, DPoS is faster than PoS.

4. Leased Proof of Stake (LPoS). LPoS [36] allows nodes to lease their own assets to others. This leasing can increase the probability for the borrowing nodes to be selected as validators. More voteable participants can reduce the chance that a single group of nodes dominate the blockchain network.
5. Proof of Activity (PoAc). PoAc [37] is developed based on PoW and PoS. The new block creators initially work as miners using PoW mechanism to defend security attacks, and hence, they start to receive the rewards. Once the miners collect enough coins (assets), they utilize PoS mechanism to publish new blocks [38], [39]. PoAc ensures that tokens offered as rewards are on time.
6. Proof of Burn (PoB). PoB [40] is an alternative of PoW and PoS. Once the nodes burn their own coins/assets by delivering them to verifiable, public, and un-spendable addresses, PoB allows the nodes to become authorized validators so they can create a new block and get rewarded This spent coin is considered an investment. In contrast to PoW and PoS, PoB does not require high energy consumption. Slimcoin [41] is developed based on PoB.
7. Proof of Inclusion (PoI). Merkle tree root [7] in the block can be used as a proof of the inclusion of records. Merkle tree root enables nodes to verify individual records without reviewing and comparing the entire chain. If a blockchain has an identical Merkle tree root as another node's blockchain, it can be said that the blocks of two nodes are verified and consistent. Ethereum blockchain presented in [8] utilizes PoI.
8. Proof of Elapsed Time (PoET). PoET [42] is designed to address the challenges of expensive investment of energy in PoW. In PoET, the computationally expensive work is replaced with the proof of elapsed time. PoET randomly chooses the next leader among the entire population of the validators to publish the block, and the validators request a random

wait time from their enclaves. The validator having the shortest wait time for a particular block is elected as the leader, which can publish the new block after the wait time has expired. The trust is established in the hardware that produces the time. Hyperledger Sawtooth [43] is based on PoET.

9. Proof of Authority (PoA). PoA [44] is designed particularly for permissioned blockchain and considers a participant's identity as a stake. Before becoming an authority to publish a block, the participant must confirm its identity in the network. The authorities are pre-selected and trusted to publish a block. It is also convenient to detect malicious authorities and inform other nodes about the malicious activities. Parity Ethereum [45] is developed based on PoA.
10. Proof of Reputation (PoR). PoR [46] is an upgraded, stronger, and more secure form of PoA, whose blockchain consensus mechanism depends on the reputation of the participants to keep the network secure [47]. In PoR, if any nodes were to attempt to cheat the system, they would face significant financial and brand consequences. PoR was implemented in

    [48].
11. Practical Byzantine Fault Tolerance (PBFT). PBFT [49] provides a solution to the Byzantine Generals Problems [50] for asynchronous environment. One validator can bundle multiple transactions to create a new block. This consensus mechanism uses game theory to verify blocks among professional miners. PBFT works on the assumption that at least two thirds of the total number of nodes are honest. PBFT was used in [51].
12. Verifiable Random Function (VRF). A VRF is the public-key version of a keyed cryptographic hash. Only the holder of the private VRF key can compute the hash for any

given data. Anyone with the corresponding public key can verify the correctness of the hash without knowing the actual data [52]. The VRF algorithm is:

(a) A VRF's key generation algorithm generates a pair of public-private keys <*pk*,*sk*>

(b) A VRF hashes an input message using the private VRF key *sk* to obtain an output hash:

$$hash = VRF\ hash(sk,message) \quad (1)$$

(c) The VRF hash algorithm uses the private *sk* to construct a proof that the hash is the correct hash output:

$$proof = VRF\ prove(sk,message) \quad (2)$$

(d) The prover sends out the <*hash*,*proof*> to the verifier.

(e) The verifier gets <*hash*,*proof*> and calculates the hash directly from the proof as:

$$hash = VRF\_proof2hash(proof) \quad (3)$$

(f) The verifier will determine if there is a unique correspondence between the message and the hash:

$$True/False = VRF\_verify(pk,message,proof) \quad (4)$$

If the last equation is true, the verification is determined to be successful; otherwise, it has failed. Since VRF's security properties are unique, including collision resistance and pseudo randomness, VRF is often used in blockchain consensus mechanism to select the master node and verify transactions [53].

13. Other Consensus Mechanism. In Proof-of-Importance (PoImportance), the chance a node can participate in verifying transactions depends on the stakes it has and the number and the quality of the transactions it has processed in the past [54]. Proof-of-Capacity (PoC) is used especially for decentralized storage as it utilizes the availability and capacity of storage space on a user's drive [55]; Proof-of-Weight (PoWeight) assigns each user a

certain “weight”, which is relative to a selected value that represents a user’s contribution to the network. All weighted users play an integral role in the process of achieving consensus [56].

1.3 Cyber Security

The NIST Computer Security Handbook [NIST95] defines the term computer security as ”The protection afforded to an automated information system in order to achieve the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications)”. This definition introduces three key objectives of computer security – confidentiality, integrity and availability, which forms the CIA triad. Besides the CIA triad, authenticity and accountability are commonly included in security fields [57].

1. Confidentiality includes data confidentiality and privacy. Data confidentiality assures that private or confidential information is not made available or disclosed to unauthorized individuals; Privacy assures that individuals control or influence what information related to them may be collected, stored, and by whom and to whom that information may be disclosed.
2. Integrity consists of data integrity and system integrity. Data integrity assures that information and programs are changed only in a specified and authorized manner, while system integrity assures that a system performs its intended function in an unaffected manner, free from deliberate or unauthorized manipulation of the system.
3. Availability assures that systems work immediately, and service is not denied to authorized users.

4. Authenticity is “the property of being genuine and being able to be verified and trusted, confidence in the validity of a transmission, a message, or message originator. This means verifying that users are who they say they are and that each input arriving at the system came from a trusted source.”
5. Accountability is “the security goal that generates the requirement for actions of an entity to be traced uniquely to that entity. This supports nonrepudiation, deterrence, fault isolation, intrusion detection and prevention, and after-action recovery and legal action”.

The security features applied in different types of applications include authentication, authorization, encryption, logging, and application security testing. Developers can also code applications to reduce security vulnerabilities[58].

1. Authentication. Authentication procedures ensure that users are who they say they are, and only authorized users have access to the application. This can be accomplished by requiring the user to provide a username and password when logging into an application. Multi-factor authentication is a common practice now.
2. Authorization. After a user has been authenticated, the user may be authorized to use the application and its data.
3. Encryption. All the sensitive data are encrypted to maintain data security. Only authorized users can decrypt the sent message and see the data.
4. Logging. Logging can help identify who has access to the data and how. Application log files provide time-stamped records about application accessing information, such as who, when, what, and activities. Logging can identify who made a security breach in an application.

5. Application security testing. A process to ensure that all these security controls work as intended.

1.4 Trust and Reputation

Trust plays an especially important role in our daily life. Very often we say we trust a person or a company's products. Trust is also a required character in many organization's operations, such as nuclear power plant and electricity delivery. Trust is defined as someone or something is good, honest, and not harmful to you [59], [60], [61]. In [62], trust is defined as assured reliance on the character, ability, strength, or truth of someone or something. In [63], "Trust is a subjective expectation an agent has about another's future behavior based on the history of their encounters". Reputation is the opinion that people have about somebody/something, based on what has happened in the past [59], [60] or based on overall character [61] or quality as judged by people in general [62]. In [63], reputation is defined as "perception that an agent creates through past actions about its intentions and norms". In blockchain network, the trust we have on any nodes must build on their ability, strength and reputation, which comes from their past actions or behaviors.

$$Trust = Direct\ Experience + Indirect\ Experience \quad (5)$$

$$Business\ Action = trust + resource + ability \quad (6)$$

1.5 Motivation

To meet the demand for power in modern society, smart grids have been developed to trade energy via central systems in traditional solutions [64]. However, besides the single-point failure issue, smart grids are susceptible to cybersecurity attacks through scanning IP addresses, hacking communication devices, visible pores, and soft viruses [65]; these systems also lack the trust of customers and privacy [66]. Due to blockchain's nature of trustworthiness, transparency,

built-in security features, and immunity of records, with its distributed platform and smart contracts, blockchain has the potential to replace centralized power systems, providing an opportunity against security threats and hacking to serve consumers in a secure way [67], [68]. Applying blockchain to smart grid will enhance the trustworthiness and preserve the integrity of the data - two major challenges that currently threaten the security of electrical infrastructure [69]. Blockchain brings its strength to smart grid, along with its intrinsic problems, such as trustless network environments and risking violating user's privacy due to its transparency.

The future smart grid not only needs to connect traditional power consuming devices, it also needs to connect the power-thirsty appliances, ranging from electric vehicles [70] to smart phones and wearable devices in our society. Thus, smart grid must face security, privacy, trust concerns and various innovative technologies that embody them [66], [71], [72], [73], [74], [75]. [76] The Advanced Metering Infrastructure (AMI) is one of the key technologies of smart grid. AMI achieves two-way communication between the utility provider and the customers. Smart meter (SM) is an important part in AMI and large quantities of SMs have been installed in users' homes to collect near real-time electricity consumption data on a requested or scheduled basis [77] and power consumption requests. The data collected by SMs may disclose user privacy. With non-intrusive appliance load monitors (NALM), an adversary could analyze the user's electricity consumption profile and the power consumption requests to infer user's behaviors. Also, because the blocks are transparent to all the nodes in the network, user's identity and behavior can be inferred from the collected information, which further threatens user's identity privacy [78] [79]. Keke Gai implemented a consortium blockchain-based technique to solve privacy issue and data leakages in smart grids, which can lead to leakage of the customer locations [80]. Adversaries can use this information to attack the users, such as stealing their

power or meddling with their smart meters [81]. Due to the trustless nature of the nodes in blockchain network, consensus mechanism is used to ensure data integrity. Consensus mechanism assumes that most nodes are honest, which may not be true. Therefore, having a strategy to encourage nodes to behave honestly is critical. In this study, we have designed three systems to protect user privacy and enhance the security of blockchain network.

CHAPTER 2

LITERATURE REVIEW

2.1 Related Work in Privacy

A customer's privacy includes user or object identity and user data [82]. The proposed techniques to address the problem can be classified into two categories – protecting user identity and protecting user data. Common privacy protection techniques are pseudonyms, data aggregation, and anonymity. Common data privacy techniques are data encryption, data obfuscation, and anonymity. This section discusses the techniques used in these fields.

2.1.1 Protecting Identity and Data Privacy

Preserving identity with pseudonyms and anonymity. A method was presented in [76] to protect user privacy by dividing the users into separate groups according to their electricity consumption profiles. Each group has a private blockchain for data recording. The user data was held in a group protected from others in the same group through creating and assigning multiple pseudonyms to each user. However, an attacker can use cluster and time analysis to estimate the relationship between the input and output addresses, mapping pseudonyms to uncover the user's real identity. [83] proposed a blockchain-based V2G payment method that is capable of sharing user data without revealing the identity. The RA registers users by hiding their identities and using their public keys to create signatures for them. The users create their accounts with their signatures to hide their true identities. The auditing of payments is performed by privileged users. [84] introduced two approaches (token-based and Pederson Commitment Scheme-based) to preserve the privacy of EVs during their charging process. These two methods combine zero knowledge proofs with blockchain and smart contracts to protect the privacy of the charging EV when authenticating for charging. [85] aimed to protect the privacy of the users in a smart grid

by incorporating a permissioned blockchain with smart contracts and edge computing in which group signatures and covert channel authorization techniques are utilized. [86] addressed the problem of uncoordinated charging of Energy Storage Units (ESUs) by constructing a decentralized charging coordination mechanism to which ESUs can perform anonymous authentication.

Preserve data privacy. a) Preserving data privacy with data encryption. The commonly used encryption to protect data is homomorphic encryption, which allows the intermediary agent to operate the encrypted data with no information about the plaintext. The property of additive homomorphism is often used to calculate the sum of electricity consumption data. A typical homomorphic encryption is Paillier encryption, which can be used in electronic voting and electronic cash. Paillier is a type of key pair-based cryptography. Unlike other key pair cryptosystems, Paillier provides “additive homomorphism”, which means that messages can be added together while they are encrypted, and they will decrypt correctly. The Paillier method also includes a zero-knowledge proof property, which can verify an encrypted message that follows a specific format without de-encrypting the message. [87] proposes a secret sharing method to protect user data privacy. TCA creates private-public keys and send them to the users and GW. The users use the private keys to encrypt their data and create signatures to protect data privacy and for GWs to do authentication. The GWs use their private keys to create signatures so CC can validate them. b) Preserving data with data obfuscation: Data obfuscation adds noise into the data to obfuscate (i.e., hide) the original data. The noise can be random [88] or user specific [89]. The noise will be managed by the control center so it can be canceled out properly later. c) Preserving data with group signature. In [53], any member of a group can sign a message on behalf the group and no one knows who really has sent out this information, except the control

center. The group signature protects user privacy and data traceability, which can be used as a privacy-preserving scheme. Our design uses the concept of group signature; however, our groups are information groups, not user groups.

Preserving privacy with other methods. [90] addressed the problem of the privacy and security of consumption and trading data using a blockchain-based energy scheduling model whose optimization is separated into trivial scheduling problems, which are then solved by consensus algorithm and smart contracts. [91] devised a framework to provide data privacy and security in smart power networks with two modules. The first is a two-level module that is dedicated to verifying data integrity using proof of work blockchain along with applying a variational autoencoder to transform data, and the second is an anomaly detection module for training and validating the output of the first module. The second module uses deep learning techniques. Experiments show its competitiveness against state-of-the-art techniques in protecting data and identify anomalies. [92] proposed to not only help users secure their data against the service providers but also share credits with other blockchain users. Electric vehicle's information may also be linked to its owner, which reveals the identity of the owner. Exposing this kind of information may lead to PII being leaked, including physical locations, times of movement, and extrapolated information. Their design aimed to protect user's privacy whilst charging Electrical Vehicles (EV) in a Vehicle-to-grid (V2G) environment.

2.1.2 Research Justification

Security and privacy are the key concerns for every system. Many strategies have been proposed to meet these needs but they all with some limitations. We would like to make our contributions in these areas. We hereby propose to ensure user's privacy by encrypting the transactions to protect user data, encrypting user's public keys to protect user's identity, and

using group signature to hide the data source.

2.2 Related Work in Trust

Consensus is the key mechanism to keep the data integrity. Many general or application-specific consensus mechanisms have been proposed. Reputation based consensus is quite different from other consensus mechanisms on how to select the block leader and consensus group members, how to produce reputation scores, and how to calculate reputation scores.

2.2.1 Related Research

In RepuCoin [93], the leader is randomly selected from the most reputable miners and the members of the consensus group are selected from the miners with the highest reputation scores. A miner's reputation score is based on the correctness of its behavior and is related to its computing power. The key block creator obtains a fixed mining fee and a share of the transactions according to its reputation. The selected leader takes the remaining transaction fees. The consensus group validates the blocks but receives no benefit from their work, which is unfair. A Proof of Reputation method was proposed in [94]. The node with the highest reputation score is selected as the leader to create a new block, the top 20% nodes in the reputation list are high-reputation nodes and participate in the consensus process. A node's reputation score is updated according to historical transactions, current age, participation in consensus, and illegal behavior. Another Proof of Reputation consensus mechanism was designed by [95]. The node with the highest score will be the leader, which constructs and publishes a block. Other nodes will verify that the block's sender is valid and consensus the block's transactions. A node's reputation is based on good behavior and block publication. At the end of each interaction, the service requester will generate a rating for the service and broadcast it. Other nodes will verify it and save the rating locally. A reputation-based neighborhood-watch mechanism [96] is used to

detect and counteract the impact of data integrity attacks. The reputation is defined as trustworthy level that one controller could put on another controller. This design can detect colluding attacks that use false praise/false accusation. However, this mechanism would fail if multiple neighbors shared elaborate information to help each other to pass the block validation phase. A dynamic reputation–based consensus mechanism was proposed by Cai *et al.*[97]. In this design, a monitoring node selects consensus nodes according to their reputation ranking. Among the consensus nodes, the monitoring node randomly selects a primary node, which creates transactions and blocks. A node's reputation is based on its hardware memory, accessing system time, and storage performance. However, the functions of nodes were not considered.

A Proof-of-Review was designed in [98]. The node with the most positive reviews will be selected as the round leader and can publish new blocks to earn more positive reviews. The nodes with the longest online time will be selected as consensus nodes. A node's trust value is based on the reviews it receives on previous transactions and interactions it had with other nodes. Trust-management toolkit combines reputation-based trust with network-flow algorithms to identify and migrate faulty protection nodes [99]. The protection nodes monitor common power-grid variables and send them to the designated central node, which is selected from all the protection nodes. This design increases the robustness and lowers the risk of faulty node power outages. However, the central node selection and function defeats the basic purpose of blockchain. In R-CoDEMS framework, each agent monitors the correctness of received consensus estimation and assesses the reputation of its neighbors [100]. This design overcomes single and coordinated profit-driven attacks but cannot handle a local majority collusion. RBT is a distributed reputation system which is designed to improve blockchain consensus and peer-to-

peer energy trading fairness [101]. The reputation is based on three roles: consensus node, energy seller, and energy buyer. However, this design only took the functions of nodes into consideration.

2.2.2 Research Justification

In the reported research, only a few dimensions have been used to evaluate the reputation of a node, whereas in some cases, some nodes are selected as judges/monitors to evaluate other nodes. Such designs are vulnerable and have inherent flaws. For example, a node that provides good services might have attacked other nodes. A node with many resources may launch selfish attacks. The process of monitor selection might be biased. Overall security can be greatly enhanced by adding additional security measures [102]. In this research, we attempt to measure a node's reputation in multiple dimensions and provide all nodes with equal opportunities in order to overcome these flaws and potential biases. We name our reputation measurement system R360, in which we evaluate every node by its function, defense capability, offense, quality of service, availability, and resources. Our system also takes time decay into account. A node's behavior and ability determine its reputation scores, which in turn renders qualifications and privileges to the node. Penalties will be assessed for a node's incorrect decisions or inaction with free rides. Any serious offenses will wipe out all of a node's positive reputation scores. If a node makes a correct decision on a block, it will be rewarded even if the block failed to be validated. All nodes in the blockchain network form a self-managed dynamic system to provide a relatively trusted network environment.

2.3 Related work on double spending countermeasure

A secure blockchain network depends on the safety and security of the participating nodes. As more nodes join the blockchain network, the ingenuity of attacks in the chain have

progressively become worse. Among all the security attacks, double spending is one of the most concerned attacks. Double spending usually targets sellers or vendors. Double spending is a single digital token was spent more than once [103]. A successful attack would be that the money and service are taken by the attacker, leaving the seller with nothing. This would cause honest nodes to stop working due to the lack of security in their transactions. Much research has been done to find solutions to mitigate or eliminate double spending attack in blockchain network.

2.3.1 Related Research

A method that blocks incoming connection requests as a countermeasure [104]. This essentially prevents one kind of double spending that requires that the attacker connect to the vendor directly. By blocking incoming connection requests, the attacker cannot establish a direct connection to the vendor to send the vendor the offensive transaction. However, newly joined vendors must request connections to other peers to ensure they have the latest block chain information. The attacker can use this opportunity and create several malicious nodes which will be distributed throughout the network, and vendors could randomly connect to one of them and become victims.

A forwarding framework in [105] increases the amount of confirmation to make it harder to attack. Increasing confirmation would require more authentications to be made in the system in order to confirm a transaction. Based on the hash rate of a sender, the amount of confirmation was calculated, which would be adequate to mitigate double spending attacks. The researcher concluded that when their probability methods to combat attacks is applied, if an attacker controls more hash rate than the honest mining network, the success rate of the attack will still be 100%. A forwarding mechanism in [106] uses peer monitoring techniques to alert the nodes in

the system that there are attacks on the blockchain. If the nodes configure the alert system to avoid receiving alerts, they will be vulnerable to attacks. A method proposed in [104] requires the vendor to wait for a transaction to propagate a number of steps before accepting it. The idea is that if more nodes have seen the transaction, it is more likely trustworthy such that greater depth is assumed to be better. However, with a chain of malicious nodes, an attacker could simply move offensive transactions along until the propagation reaches the required depth.

A dynamic observation method in [107] proposed the ENHOBS (enhanced observers) method, which used active observers with indistinguishable traffic patterns for valuable transaction inspection. To detect double spending attack on the network, a one-time scan was run on the blockchain to find duplicate transactions. When matching transactions were detected, an alert would be sent through the network. Once the alert was received and was seen as having verifiable proof of an attack, any transactions matching the same input value would be dropped from the memory pool immediately. A method proposed in [108] requires peers to conduct a deeper investigation of conflicting transactions and broadcast alerts to all peers if a double-spending attack is detected. This approach can catch double spenders only after an attack has occurred, and there is no prevention for future occurrence. Even if the attacker was put on a blacklist, the attacker could create a new pseudonym easily and attack again.

A listening period was used in [109] to monitor all transactions that have been previously received and checked if there were attempts to double spend. If there were, an alert would be sent out to the network. This will not be effective in detecting attacks because the attacker can delay sending the attacking transactions until the monitoring window has expired. Another technique proposed in [109] is to randomly insert observers across the P2P network, which forwards all transactions in the monitoring period to help detect double spending because at least

one of the observers will receive conflicting transactions, if there are any. If an attack is detected, an alert message will be sent to the network. This approach is somewhat effective. However, it does not directly prevent the double-spending attack or the propagation of the offensive transaction. Plus, the observer's traffic patterns can be easily analyzed by an attacker [110], who can carry out DDoS attacks against the observers and re-enable double spending.

A broadcasting programming strategy in [111] proposed a mechanism to construct special transaction outputs to combat double spending. The output of a bitcoin transaction includes two fields. The first one indicates the amount of bitcoins that will be deposited. The second field, named FR-P2PK (fixed-rpay-to-pubkey), defines the conditions under which this output could be spent. Such output can be spent with a single signature but has the property that if two different signatures have the same output, which indicates a double spending attack, the private key used to sign the transaction is revealed. Then the observer can generate a third transaction spending the same output and send the amount himself.

A detection method in [112] uses blind signature cryptography with a publicly verifiable time-based payment transcript as double spending countermeasure. In order for the coin to be cashed by the client, the vendor must present a NIZK (non-interactive zero-knowledge) proof, which will bind the payment transcript to the target client and time. Another solution presented in [112] is a coin renewal protocol which provides a coin with three stages. Before reaching the dates, the coins can be cashed or renewed. If the coin reaches the first date, it can only be renewed. If it reaches the second date, the coin will be totally void.

2.3.2 Research Justification

In the strategies discussed above, the conflicting transactions were not handled, the resources were wasted because all peers were checking double-spending transactions, the

monitoring window was short, and the observers inserted into the network required management overhead, demanding increased network traffic and CPU utilization. The observers may also cause DDoS attack due to its special traffic pattern. These strategies do not explain how conflicting transactions will affect the consensus results and how to prevent the offending nodes to implement double spending attack again. Another issue with blockchain is even a block passed the consensus, it only means that all the nodes received the same set of transactions. It dose not mean that all the transactions in the block are truly accurate, even when all the nodes are honest. This is because there IS no mechanism to test if these transactions are correct from their sources, e.g., one node's evaluations on another node, the money a node needs to pay. Incorrect transactions will cause disputes among users and damage the network reputation. In this research, we propose a design to address these issues.

CHAPTER 3

INFORMATION-CENTRIC MODEL

3.1 Preliminaries

3.1.1 Information Centric Orientation

The consumers in any network are mainly interested in the information rather than the network locations of the data sources or destinations. This is commonly phrased as Information-Centric Networking (ICN) [113]. ICN places data at the center of the networking landscape where information is published, resolved, delivered, and stored [114]. ICN names schemes based on the content and hides sender's and receiver's IP addresses, which reduces the risk of network-borne attacks, especially for the measurement devices, such as Phaser Management Unit (PMUs) and smart meters that have limited resources to defend themselves against attacks [52]. PMUs are used for recording synchronized measurements across a wide area. This is a part of the effort to develop a new communication network in order to provide real-time monitoring of the PMUs. When there are many parties exchanging and sharing information in the blockchain environment, ICN provides more flexibility than traditional host-centric solutions. In our paper, we apply this ICN concept and divide the transactions into different information groups to hide data sources and preserve user privacy.

3.1.2 Group Signature

The group signature [53] is used to protect the privacy and traceability. In the group signature scheme, each member of the group can send transactions on behalf of the group without disclosing its own identity. The nodes with the group public keys can verify whether a signature is from someone in the group and determine if the received transaction is valid.

3.1.3 Information Classification

An important function of smart grid is to communicate and share information in the network, which is paramount for smart grid to operate effectively and efficiently. We are going to classify the information into the following categories:

1. Energy consumption. This information will be created by smart meters.
2. Energy generation. This information will be produced by PMUs.
3. Energy storage. This information will be sent out by IoT.
4. Energy trading is buying and dispatching. This information will be put into computers manually.
5. Control. This information will be sent out by the control center.
6. Others.

All nodes in the blockchain network will be assigned to at least one of these groups. A node can be in more than one group. For simplification, we assume each node only belongs to a single group. If a node participates in more than one group, each group's transactions will be saved in distinct locations. In our discussion, we only cover two kinds of user cases: energy consumption and energy trading. There are three parts in each information group. a) The publishers that generate the transactions, such as the smart meters, PMU (Phasor Measurement Unit), and computers for information input in the data input layer. b) The relays that consent to verify, store, and broadcast the transactions, such as the nodes in the blockchain layer. c) The information consumers, also called subscribers, such as the billing center, control center in the receiver layer, and the users who receive data from others, such as available energy data and energy trading related information.

3.1.4 Data Encryption and Verification

1. Suppose the transactions in the network are grouped into m groups and each group is assigned a group message, group i's group message will be m_i.
2. Public keys

$$Priv,Pub = CreateKeyPair() \tag{7}$$

3. Encryption

$$Cipher = Encrypt(msg,Pub) \tag{8}$$

4. Decryption.

$$msg = Decrypt(Cipher,Priv) \tag{9}$$

5. Create Signature

$$Sign = CreateSign(m_i,Priv) \tag{10}$$

6. Signature validation

$$Err = V erifySign(m_i,Pub,Sign) \tag{11}$$

No Err means success.

7. Block validation. Suppose there are two blocks, b_1 and b_2, if

$$hash(b_1) = hash(b_2) \tag{12}$$

we will say b_1 is the same as b_2.

3.2 Privacy-Preserving Information Centric Scheme

3.2.1 System Model

There are four layers in our design - data input layer, blockchain layer, which is for data collection, verification, and storage, communication layer, and receiving layer, which includes control center, billing center, and common receivers. The Public Key Infrastructure (PKI) spans

across all the four layers, see Figure 3. The colored circles in the blockchain layer represent different information groups.

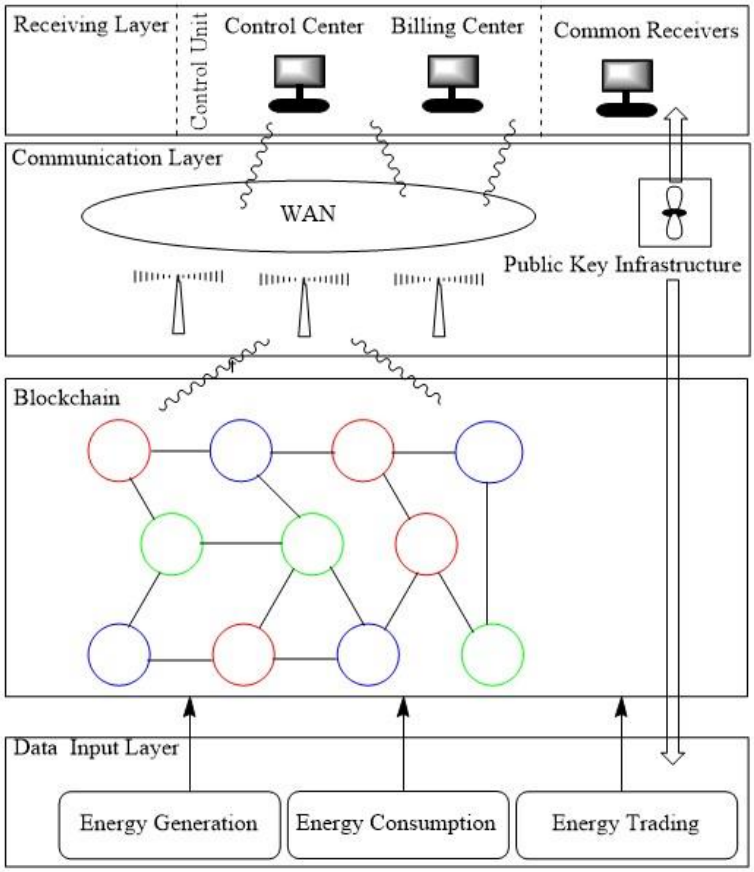

Figure 3. System Model

1. Data input layer. All publishers are in this layer, such as smart meters, PMUs, and computers used by users to enter their energy buying information. This layer is where all new transactions are created. Each new transaction includes the encrypted core message, group signature, encrypted sender's public key, and the information type. After a new transaction is created, it will be broadcasted to the entire network.
2. Blockchain layer. This layer is for transaction audit, validation, and blockchain storage. A valid block will be broadcasted to the network for subscribers/receivers to process.

3. Communication layer. In this layer, all needed information passes through the entire network.
4. Receiving layer. There are two parts in this layer, the general customers, such as users receiving their trading information, and the management departments, such as the control center and billing center. The control center will be responsible for each group's registration, and power dispatching.
5. Public Key Infrastructure (PKI). PKI creates public-private keys for all the participants, including each information group. Every group will get a group message and two pairs of public-private keys, one for the publishers PUgp-PRgp and one for the subscribers PUgs-PRgs.

3.2.2 System Initialization

Key Generation and Group Membership

1. All the participants in the system must acquire public-private key pairs. For receivers, we use PUr-PRr.
2. All publishers will get control center's public key, PUc , the group message, the group publisher's private key, PRgp, receiver's public key, PUr, and subscriber's public key, PUgs.
3. All receivers will get the group message and the group publisher's public key, PUgp, and the group subscriber's private key, PRgs.

3.3 System Smart Contract

3.3.1 Transaction Creation

The smart contract will be installed on every device in the data input layer and executed according to their schedules, such as time interval, a signal, or as requested. There are four parts

in a newly created transaction. a) Group type. b) The core message encrypted by the group's subscriber's public key PUgs, and the receiver public key PUr if the data is for a specific receiver. The core message includes energy data and time stamp, and the energy data can be power consumption data, available power, appliance states, or energy order information. c) The sender's public key PUs encrypted by the control center's public key, PUc. d) The transaction signature created by the group message and the group publisher's private key PRgp. After the transaction is created, it will be broadcasted to the network. See Figure 4.

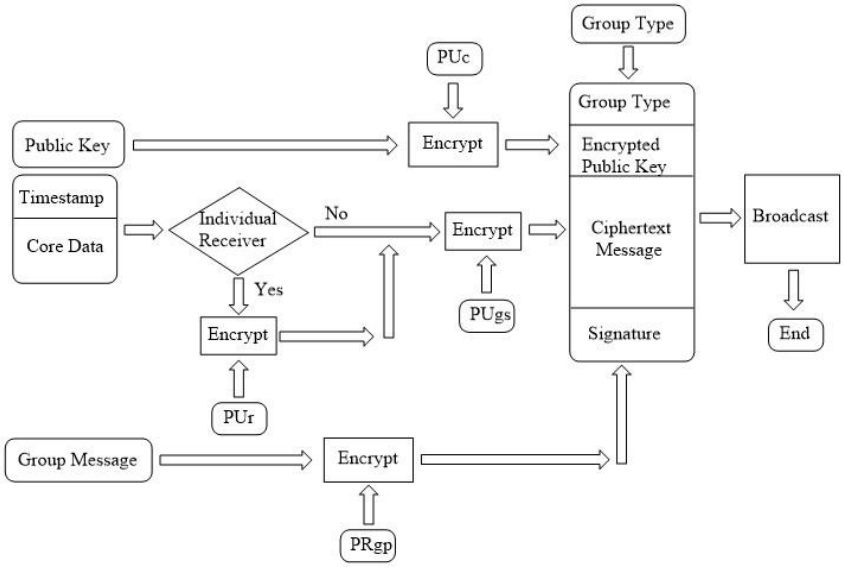

Figure 4. Transaction Creation

3.3.2 Consensus Mechanism

The consensus smart contract will be installed on all the nodes in the blockchain layer because it is responsible for verifying and storing various information groups. The VRF function will be used to select a master node and the consensus mechanism will be based on PBFT as shown in Figure 5.

1. When a node receives a new transaction, if the transaction belongs to the right group and the publisher is valid, it will save the transaction locally; otherwise, the node will drop this transaction.
2. Iteratively, every 15 minutes, the group nodes start to vote for a master node using VRF. Each group starts at a different time, e.g., at a 2-minute interval, in order to avoid conflict among information groups.
3. When a master node is selected, the master node will pack all this group's transactions in its local pool in the past 15 minutes and broadcast the block to the entire network.
4. When receiving the block, non-master nodes will use the VRF function to compare the transactions in the block with the transactions in their local pools. If the verification fails, the block will be discarded. Otherwise the block will be saved and a message "confirm" will be broadcasted.
5. When a non-master node receives more than two-thirds of the group nodes agreeing on the candidate block, it will write the new block to the local blockchain and delete the transactions in its transaction pool. Otherwise, both the block and the local transactions will be discarded.
6. When the master node receives the results from at least two-thirds of the group nodes agreeing on that candidate block, the master node will write the block to the blockchain and broadcast it to the entire network to subscribers or receivers. If the block failed to pass, the master will delete its local transactions and broadcast a failed block notification. If there is no transactions during this time period, the master will broadcast a null-transaction notification to prevent an adversary from creating a fake block [115].

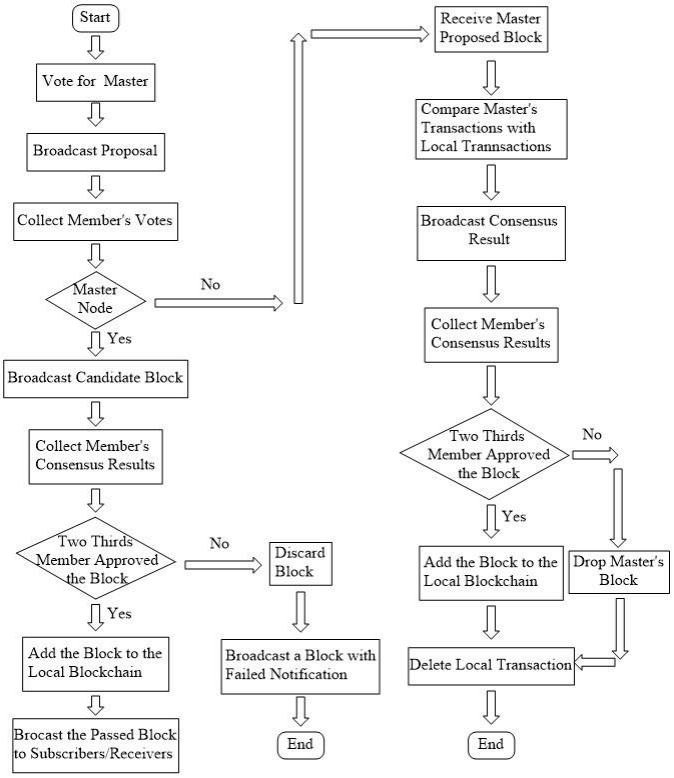

Figure 5. Information Centric Consensus Mechanism

3.3.3 Transaction Receiving

When the subscribers/receivers receive a block, they will go through the block. If a transaction is not in the right information group or from a valid publisher, the transaction will be discarded. Otherwise, the transaction will be first decrypted with the group-subscriber private key PRgs, then decrypted with the receiver's private key PRr, if the transaction is for a specific user. The decrypted transaction will be processed according to its group. See Figure 6.

3.4 System Analysis

3.4.1 Security Analysis

The three basic security requirements are confidentiality, integrity, and availability. In our design, the user's public key and core data are encrypted and only the legal users with the corresponding private keys can decrypt them. For example, information group's nodes can verify

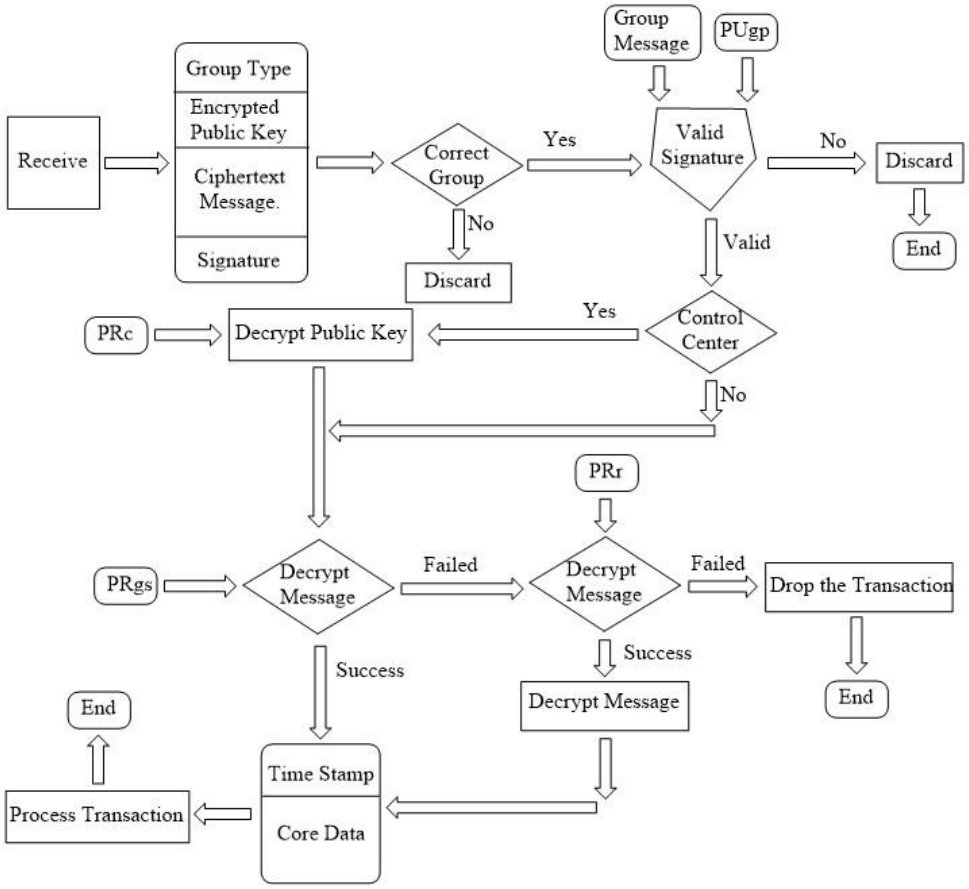

Figure 6. Subscriber Receive

if a transaction comes from a valid publisher and compare the core data without knowing it. The subscriber/receivers can decrypt the publisher's core data with its group private key and/or the receiver's private key. Only the control center can decrypt user's public key, which is the publisher's identity. This keeps the publisher's identity private and ensures data confidentiality and integrity. In a blockchain environment, each information group contains many nodes, which avoids single point failure and therefore ensures resource availability. Below are the common security attack analyses:

1. Denial of Services (DoS) Attack. In our design, all transactions are saved first, then processed by its information group nodes in a fixed interval. The service request rate is limited by its processing time interval [116] and almost all the information passed in the

energy network follows the predefined frequency. Any abnormal transaction frequency will alert the system.

2. Single Data Point Failure. In our design, the network nodes are divided into different information groups. Each group will have at least three nodes, this redundancy prevents single point failure.
3. Content Poisoning Attack. If any node's information has been changed, it will be discovered by other nodes during the consensus process.
4. Data manipulation in Data Input Layer. Since almost all the transactions, such as energy consumption, energy availability, are created by smart contracts automatically, no human interference is needed. Smart meters and PMUs usually are secured both physically and through software, it is very difficult to change the core data and forge transactions. A fake energy trading transaction can be made, but this will be discovered quickly by the users from the control center's confirmations or billing information.
5. Forged transaction. In order to forge a transaction, an adversary needs the control center's public key, the group subscriber's public key, the receiver's public key, and the group message. Unless all of these keys and messages are saved in the same place and stolen, the chance to obtain all of them is extremely low.
6. Null transaction block. Stopping an adversary from sending out blocks when there are no transactions in a time slot for some information groups. If in a time slot there are no transactions to process for any information groups, the selected master will send a null transaction block to report its group's status. More than one node broadcasting a block for the same information group in the same time period will alert the system.

3.4.2 Privacy

1. Identity privacy. All sender's public keys are encrypted by the control center's public key. Only the control center can decrypt a user's identity. This strategy protects the publisher's privacy and information sources.
2. Data privacy. All the publisher's data are encrypted, first by the receiver's public key if the transaction is for an individual receiver, then by its group public key. Only the intended receivers or subscribers will be able to decrypt the data with proper private keys.
3. Information origin privacy. Each transaction's signature will be created by its group message and the publisher's private key. The consensus nodes and the receivers or subscribers can verify that this transaction comes from a valid publisher by using the same group message and the group publisher's public key. All publishers from the same group use the same group message and the publisher's private key. By bundling the transactions, group message, and the publisher's private key together, the origin of the transaction is hidden from potential adversaries.

3.4.3 Scalability Analysis

Each blockchain's consensus takes some time to ensure that all the involved nodes receive responses from other nodes and finish the consensus process. This adaption negatively impacts the network throughput. Because our system classifies the blockchain's transactions into various categories and these different information groups execute their consensus in a simultaneous manner, in any time slot, more transactions will be processed than current blockchain systems. Additionally, because fewer nodes are involved in each group's consensus, the time required for each group's consensus will be reduced also, which enhances the network throughput.

3.4.4 Reduced Storage Redundancy

Suppose there are *n* nodes in the network. If there is only one information group, which means the information in the blockchain is not classified and not divided, all *n* nodes will save the same transactions and the storage redundancy is *n*; if the blockchain's information is divided into *k* groups, the redundancy will be reduced to *n/k*.

3.4.5 Enhanced Search Efficiency

Since we save different information in each blockchain based on its type, if the system needs to search for some transactions, the search will be faster because it only needs to go through the specific blockchain.

3.5 Implementation

Figure 7 shows transaction creation flow chart

3.5.1 Transaction Creation Contract

1. Transaction creation. The sender's account is encrypted by the control center's public key. The core data, time stamp, and energy consumption or energy ordering information are encrypted. If this transaction is for an individual receiver, this transaction will be encrypted by the receiver's public key first, then encrypted by the subscriber's public key. If this transaction is for group subscribers, it will be only encrypted by the group subscriber's public key. The transaction's signature will be created by using the group message and the publisher's private key. The transaction ID will be created by hashing the sender's account, core data, signature, and group number.
2. Sub Functions. We use Golang to implement consumer transaction forming and sending. Our implementation includes the following functions: create public-private key, create signature, verify signature, read consumer information, encrypt message,

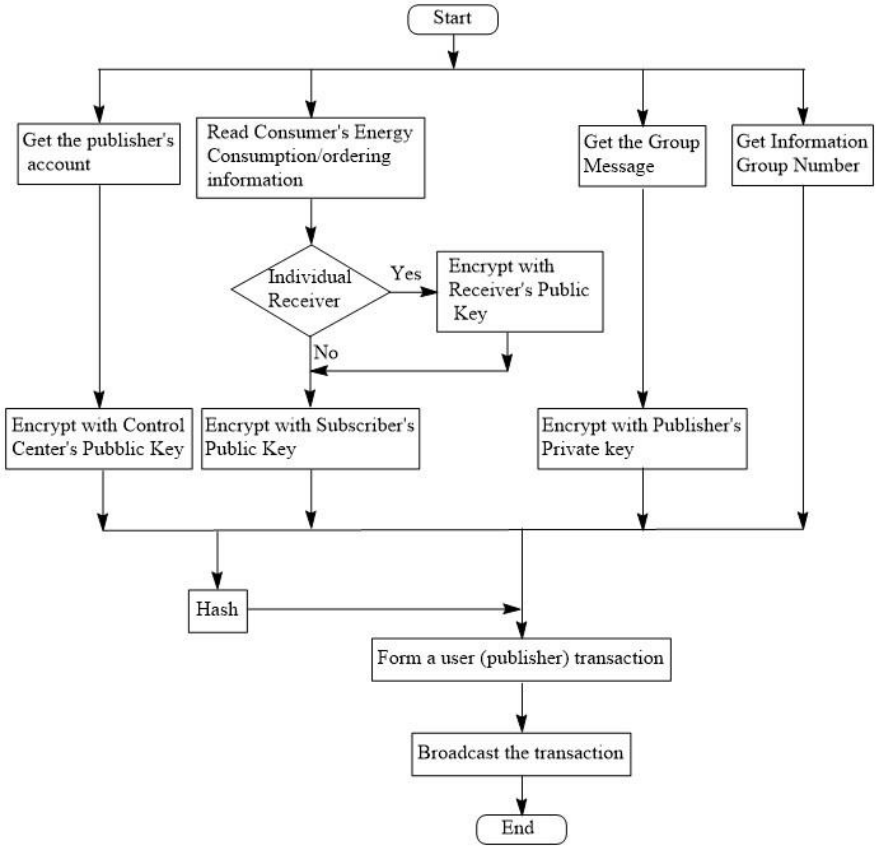

Figure 7. Transaction Creation Flow Chart

decrypt message, broadcast transaction, receive information, and append message.

3. Transaction components. A transaction includes information group number, transaction hash, encrypted core information, encrypted sender's account, and transaction signature.

3.5.2 Consensus Contract

Figure 8 shows the consensus flow chart.

The consensus nodes constantly check on transactions. When they receive a transaction, they check the transaction's group and verify the signature. If the group is correct and the signature is valid, this transaction will be saved locally. Otherwise, the transaction will be discarded. If it is consensus time, the nodes.

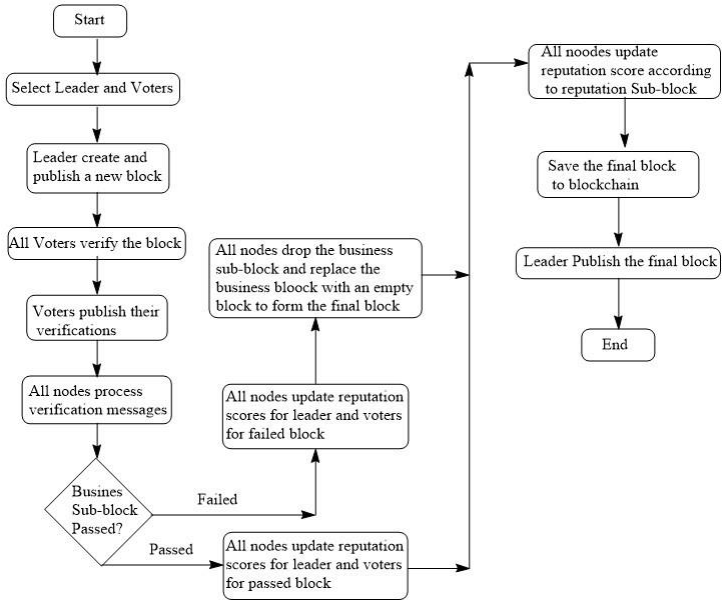

Figure 8. Consensus Flow Chart

start to elect a leader and broadcast their proposals. Each node calculates its votes. If it receives the most votes, it will know it is the leader and it will pack its local transactions and broadcast it. When the non-leader nodes receive the block, they start to compare the block's transactions with their local transactions one by one. If any transaction's group, signature, or the data fails during the comparison, the consensus will fail, and consensus procedure will be ended. If all the transactions in the block are the same as the local transactions, the consensus is passed. Each node broadcasts its consensus result and waits for other's results. If a node received two thirds passing notifications from the members, it will permanently save the block to its blockchain and delete its local transactions. When the leader receives two thirds passing notification, it will save the block to its blockchain and broadcast the final block to the subscribers/receivers.

3.5.3 Receive Contract

Figure 9 shows the receive flow chart.

When a subscriber receives the final block sent by the group leader, it will go through the block's transactions one by one. If the subscriber finds that a transaction belongs to the right

group, the signature is valid, and it decrypts the data with the group receiver's private key correctly, it will process this transaction according to its groups, such as energy consumption or energy order. If the first data decryption failed, the specific receiver's private key will be used.

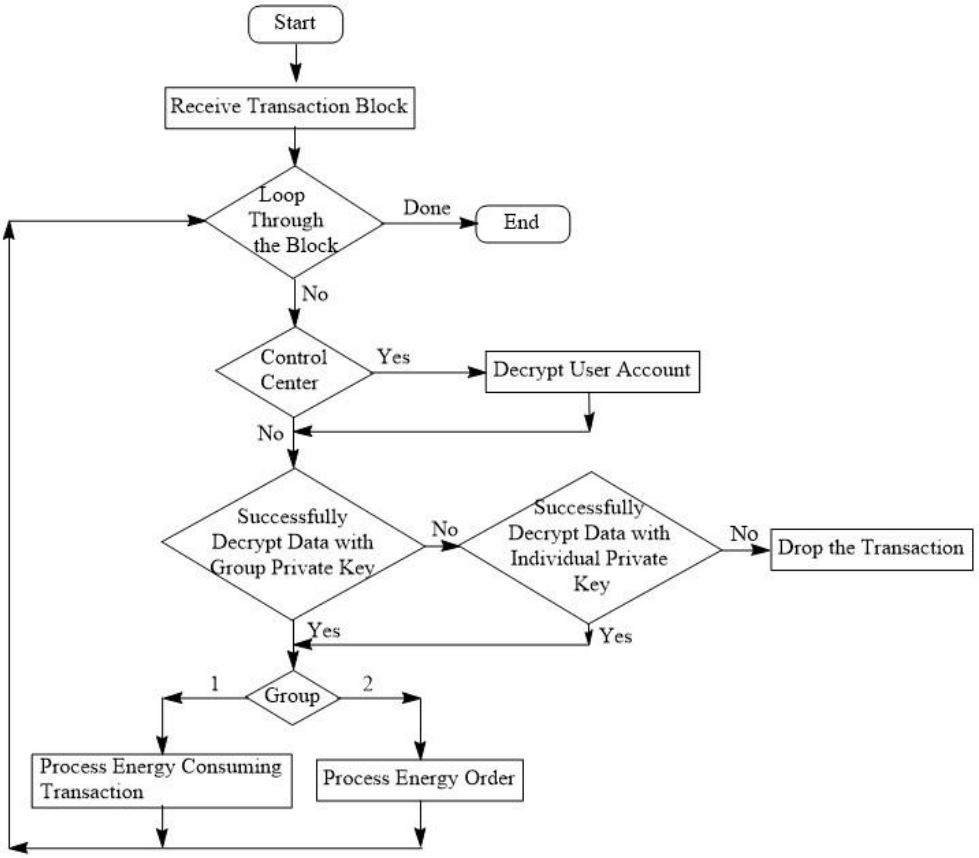

Figure 9. Receive Flow Chart

to decrypt this data. If the second decryption is successful, this transaction will be processed accordingly. If both decryptions failed, this transaction will be discarded. If the subscriber is the controller, it can decrypt the transaction with sender's public key also and will know who sends this transaction.

3.6 Conclusion

In this design, we have proposed an information-centric blockchain technology for smart grids to enhance security. Our implementation has demonstrated that our proposed design is

feasible, and it can protect sender's identity, camouflage transaction's sources, and protect data privacy. However, our implementation has been done with a simplified model with limited data. A comprehensive implementation needs to be carried out, and a more thorough testing needs to be performed.

CHAPTER 4

REPUTATION-BASED CONSENSUS MODEL

4.1 System Design Considerations

Consensus mechanisms are vulnerable to certain attacks. Double spending attacks, eclipse attacks, selfish attacks, and flash attacks are all examples [117]. By and large, all the existing contemporary consensus mechanisms, such as proof-of-work, proof-of-stack, and the variants based on them, rely on the assumption that most nodes are honest [118]. Although this assumption is reasonable, it is not based on facts, which brought out reputation-based consensus mechanism. Proof of Reputation (PoR) is an upgraded, stronger, and more secure form of Proof of Authority (PoA) [119]. PoR consensus model depends on the reputation of the participants to keep the network secure. A node must have a reputation that is important such that they would face significant financial and brand value loss if they were to attempt to subvert the system.

A reputation system must have the following three properties: (1) it must provide information that allows buyers to distinguish between trustworthy and non-trustworthy sellers; (2) it must encourage sellers to be trustworthy, and (3) it must discourage participation from those who are not trustworthy [120]. Below are the design principles for our reputation system.

1. The basic properties of reputation. The concept of trust is always related to behavior or context [121]. A high level of reputation and trust obtained in context or through behavior is not transferable. For example, being a good doctor should not imply the person would be a good manager. The reputation of a person acting as a doctor, or a manager needs to be treated separately. In the application to smart grid, a node's leadership reputation and voting reputation must be managed separately.

2. Fairness. A node's computation power, online time, etc., will be considered towards the node's reputation score. Time decay is also considered.
3. Reputation-based qualifications and privileges. A node's qualifications and privileges to create or vote for blocks depend on its reputation scores.
4. Multidimensionality. We introduce the R360 concept, which encapsulates the notion that a node's reputation will be measured in multiple dimensions. According to [122], a reputation system should include the following fundamental dimensions: history, context, collection, representation, aggregation, entities, presence, governance, fabric, interoperability, control, evaluation, data filtering, and data aging. Based on these dimensions, we elaborate the design of our reputation system.
5. Two-tier block leader and voter selection. We first screen potential block leaders and voters by their total reputation scores to make sure they are good nodes in general. Then the leader will be selected by its block creation score and the remaining members of the consensus group will be selected by their voting scores. This strategy is extensively used as common selection principles in society, such as human employment practice.

4.2 System Model

4.2.1 System Composition

Whether a node is trustworthy or not depends on several factors. The R360 reputation system includes the following components, shown in Figure 10.

1. Resources. CPU, memory, and storage are necessary resources of any nodes and will seriously affect a node's performance.
2. Defense history. A secure node should be safe and free from security breach. A node that frequently or recently experienced security breaches will be less trustworthy.

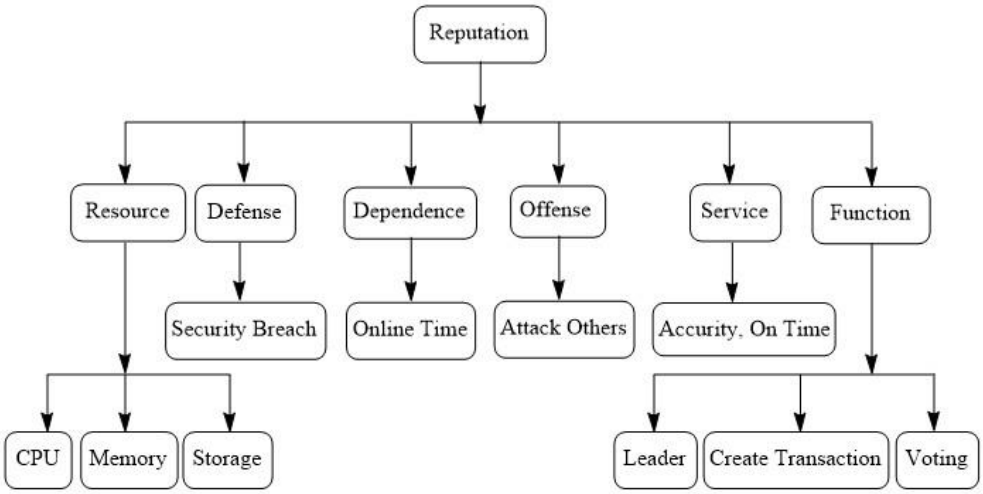

Figure 10. Reputation Structure

3. Offenses. This score is related to a node's serious offensive behavior, which is not tolerated and subject to harsh penalty.
4. Function. In blockchain context, the core functions of any nodes include creating transactions, creating a new block and publishing it, and participating in consensus. A reputable node should function as expected.
5. Service. The service score of a node is given by those who receive its services. For example, the receiver can create transactions to evaluate the solar energy quality it received from a producer. It can rate the accuracy and timeliness of the information or assistance it obtained from a node.
6. Dependence or Availability. This score measures a node's online time. A reliable node must be available to provide services when needed.

4.2.2 Assumption and Reputation Estimation

1. The resource scores are not cumulative and the resources a node has will seriously affect its performance, which affects its reputation. There are three kinds of resource, CPU, Memory, and Storage. Three of them form the resource total score. The point that a node gets for a

specific type of resource is decided by its value Vt related to the highest value Mt any node has in the network.

$$R_t = \begin{cases} 3 & : V_t \geq 80\% M_t \\ 2 & : V_t \geq 60\% M_t \\ 1 & : V_t \geq 30\% M_t \\ 0 & : V_t < 30 M_t\% \end{cases} \quad (13)$$

$$R = Rcpu + Rmem + Rstorage \quad (14)$$

2. The defense capability score is cumulative, and a node that can effectively defend itself will gain other's trust. Based on the assessment of the damage caused by a security breach, the score can be -1, -2, -3, with -3 being the value for the most serious. According to [17], the impact of data breach will likely diminish over time. In our design, we assume the impact of a breach will be completely erased after 5 years. Each year, 20% of defense score is reduced.

$$D_i = \begin{cases} -3 & : Serious \\ -2 & : Middle \\ -1 & : Minor \end{cases} \quad (15)$$

$$D = \sum_{i=1} D_i \quad (16)$$

$$D = D * 80\% \quad (17)$$

where *i* means the i^{th} security breach.

3. The service score is cumulative. A node that provides good service will get better service evaluation and win trust from other nodes.

$$S_i = \begin{cases} 3 & : Excellent \\ 2 & : Good \\ 1 & : Ok \end{cases} \quad (18)$$

$$S = \sum_{i=1} S_i \text{ Type equation here.} \quad (19)$$

*w*here *i* means the i^{th} service transaction. A node will give a service provider a possible score of 1, 2, or 3, depending on how well it values the service it received.

4. The function score is cumulative as a result of the work a node has performed in history. There are three kinds of function scores: transaction score for creating transactions, leading score for creating and publishing blocks, and voting score for participating consensus. All function scores are related to the number of transactions in the current blocks. If the number of transactions in a block T_b is at least 80% of the highest number of transactions in a block in history T_{mbh}, the voting score will be set to 3. If T_b is at least 50% of T_{mbh}, the voting score will be set to 2. If T_b is below 50% of T_{mbh}, the voting score will be set to 1. If there is no transaction in the block, the voting score will be 0.

$$Point = \begin{cases} 3 & : T_b \geq 80\% T_{mbh} \\ 2 & : T_b \geq 50\% T_{mbh} \\ 1 & : T_b \geq 1 \\ 0 & : T_b = 0 \end{cases} \tag{20}$$

A voter will receive the voting score, and the leader will earn twice as many points. If a block fail, the voters who has failed the block will receive the voting points, but the other voters and the leader of the block will receive the corresponding negative voting points. i means the i^{th} block.

$$L_i = \begin{cases} \pm Point & : Leading \\ \pm Point & : Voting \end{cases} \tag{21}$$

$$L = \sum_{i=1} L_i \tag{22}$$

$$V_i = \pm Point \tag{23}$$

$$V = \sum_{i=1} V_i \tag{24}$$

For the transaction creation score, if the number of transactions a node created T_c is at least 80% of the maximum number of transactions created by a node in the current block T_{mb}, the

node will be assigned 3 points; if the T_c of a node is at least 50% of T_{mb}, it will be assigned 2 points; the nodes that created at least one transaction will be assigned 1 point; if a node did not create any transactions, it will receive 0 points. If a block fails, the nodes that has created transactions in the failed block will receive the corresponding negative points. j means the j^{th} transaction.

$$T_j = \begin{cases} \pm 3 & : T_c \geq 80\% T_{mb} \\ \pm 2 & : T_c \geq 50\% T_{mb} \\ \pm 1 & : T_c \geq 1 \\ 0 & : T_c = 0 \end{cases} \tag{25}$$

$$T = \sum_{j=1} T_j \tag{26}$$

5. The availability score is not cumulative, and it is only effective for one year. A node that has longer uptime will more likely win other's trust.

$$A = \begin{cases} 3 & : Six - nines \\ 2 & : Five - nine \\ 1 & : Two - nines \end{cases} \tag{27}$$

Ninety-nine percent uptime, called "two-nines", will earn 1 point, "five nines" (99.999%) will earn 2 points, "Six-nines" (99.9999%) will earn 3 points. The node whose uptime is below 99% will earn 0 points. If a node's uptime changes, the node will broadcast a transaction to inform others about its uptime or availability status.

6. The offense score is cumulative, and a high offense score means less trusted by others. The node will get O_i points for each offensive behavior. In this design, O_i equals to -3.

$$O = \sum_{i=1} O_i \tag{28}$$

where i means the i^{th} offense.

7. The total reputation score of a node will be the sum of its six scores obtained in individual attributes:

$$Reputation = Resource + Defense + Availability + Offense + Service + Function \quad (29)$$

4.2.3 Consensus Mechanism

Figure 11 is the block structure and Figure 12 is the reputation consensus flow chart.

1. Each node selects nodes with at least 90% of the highest total reputation score among all nodes as potential leaders.
2. Select the node with the highest block creation score from the potential leaders as the leader.
3. Each node selects nodes with at least 10% of the highest total reputation score as potential voters.
4. From the potential voter pool, select nodes with at least 10% of the highest voting score as voters.
5. The leader creates a new block and publishes it.
6. The voters receive the block, verify the normal transactions, the reputation transactions, and publish their verification results.
7. When the consensus time expires, late responses will be ignored.
8. All the nodes process the received verification results, record the nodes that have validated or failed the block or failed to send out their verifications.
9. If the block is validated, all nodes that have voted for validation will receive a positive voting score. Those voters that have failed the block or failed to send out their verification responses will receive negative voting points.
10. If the block is validated, all nodes will update any node's reputation according to the reputation transactions in the reputation sub-block.

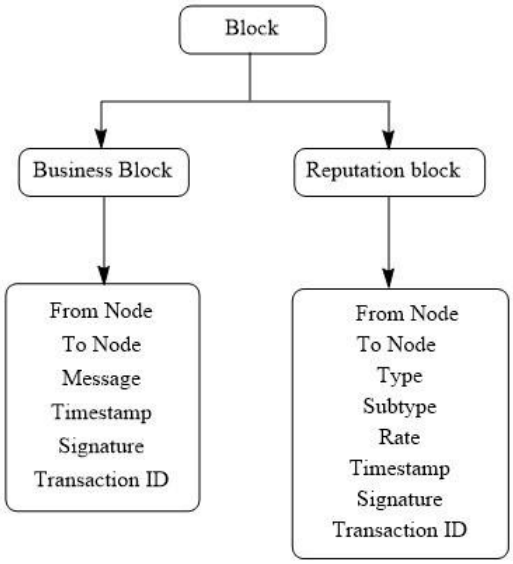

Figure 11. Block Structure

11. The validated block will be added to the blockchain, including business and reputation sub-blocks, voters, and the maximum transactions in the blockchain so far.
12. If the block failed to be validated, all nodes that have voted for validation or failed to send out their verification responses will receive a negative voting score. Those voters that have voted to fail the block will receive positive voting points. The failed block will be dropped.
13. The leader publishes the validated block or a failed block message.

4.3 Implementation and Evaluation

4.3.1 Block Composition and Block Voting

Each block contains two sub-blocks as shown in Figure 11. One is normal transaction block, named business block, and the other is reputation transaction block, named reputation block, which contains reputation transactions.

1. Business transaction contains information of from node, to node, message, timestamp, signature, and transaction ID, which is the hash of all the components in the transaction except the transaction ID itself.

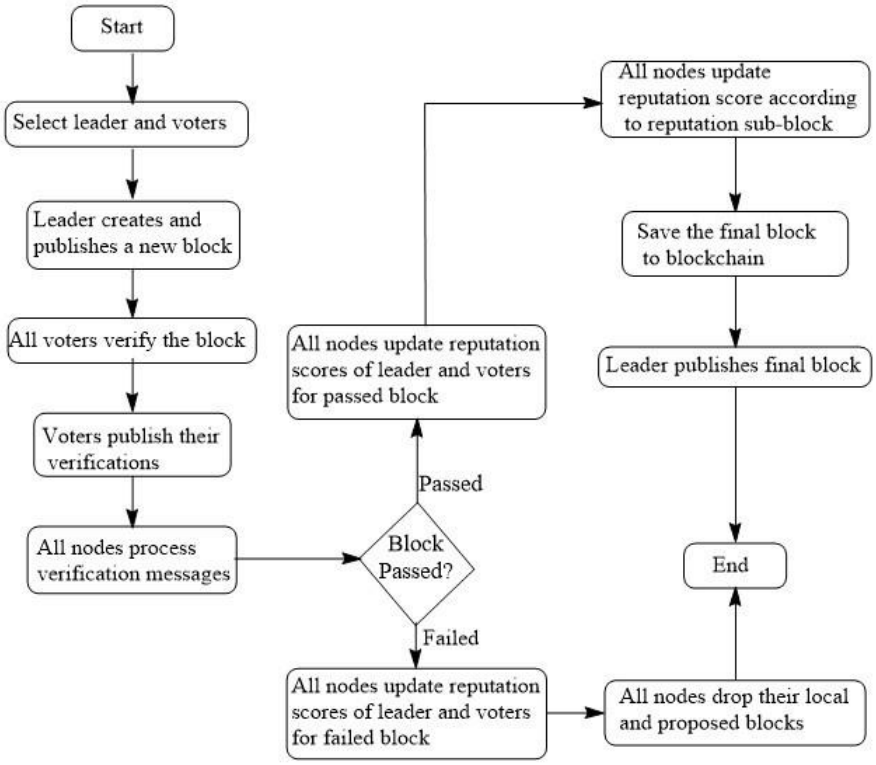

Figure 12. The Reputation Consensus Process

2. Reputation transaction contains information of from node, to node, evaluation type, subtype, rate, timestamp, signature, and transaction ID, which is the hash of all the components in the reputation transaction except transaction ID itself. The evaluation type will be resource, defense, service, offense, and availability. If the evaluation type is resource, it will have three subtypes: CPU, memory, and storage. If the evaluation type is service, more detailed information about this service can be put into the subtype field.
3. A node will validate a block only when all business and reputation transactions are the same as its local business and reputation transactions.
4. A node's voting response contains voter, verification status, signature, and timestamp. The verification status has two possible values: 0 and 1, with 0 meaning fail, and 1 denoting validation.

5. A block will be validated only when both sub-blocks are validated by at least 2/3 of the voters.

4.3.2 Consensus Algorithms

Table 2 lists the nomenclature for the consensus algorithms.

Algorithm 1 is run by all nodes to select the block leader and the group of voters. Algorithm 2 updates the block leader's and voter's reputation scores after a block is validated. Algorithm 3 updates all node's transaction creation scores after a block is validated. Algorithm 4 updates the block leader's and voter's reputation scores when a block fails to be validated. Algorithm 5 will run after a block is validated and will update all node's reputation according to the reputation transactions in the reputation block. The transactions in the reputation block are sorted by timestamps before the update to ensure the correct chronological order of transaction creation.

4.3.3 Experimental Environment

1. The design was implemented on a MacBook Pro with MacOS Monterey, 2.7GHz Dual-Core Intel Core i5 processor, and 8GB 1867 MHZ DDR3 memory.
2. The programming language was Golang, Version Go1.17.6 Darwin/AMD64.
3. We created ten potential nodes to participate the implementation, eight of which, including the leader, were selected to vote.
4. Each node had its own directory. All the related information, *e.g.*, blockchain, node name, starting balance, private key, was saved in that location.

Table 2. Nomenclature

Variable	Meaning
Leader	Block leader
MaxNumBT	The highest number of transactions
MaxTR	The maximum total reputation score among all nodes
MaxTrans	The maximum transactions created by any node in a block
MaxVR	The maximum voting reputation score
NodeAR	Node's availability score
NodeCPU	Node's CPU core
NodeDFR	Node's defense reputation score
NodeMEM	Node's memory score
NodeRR	Node's resource reputation score
NodeSR	Node's service reputation score
NodeSTOR	Node's storage score
NodeTCR	Node's transaction creation score
NodeTR	Node's total reputation score
NodeTrans	The number of transactions created by a node
NodeVR	Node's voting score
NumBT	Number of transactions in a block, including transactions in business and reputation blocks
perTrans	The percentage of transactions created by a node relative to MaxTrans
PLeaders	Potential block leaders
PVoter	Potential voters
Rate	The value given to a reputation transaction
Voters	The group of voters
ir	Node r is evaluated in transaction i, *e.g.*, $NodeRR_{ir}$ represents one of node r's subtype resources being evaluated in transaction i; $NodeTR_{ir}$ - the total reputation score of node r being evaluated in transaction i; $NodeSR_{ir}$ - the service score of node r being evaluated in transaction i; $NodeDFR_{ir}$ - the defense score of node r being evaluated in transaction i

Algorithm 1. Select leader and voters

Find the potential leaders and potential voters:
for all nodes **do**
  if $NodeTR_i \geq$ 90% of MaxTR **then**
    $Node_i \Rightarrow PLeaders$
  else if $NodeTR_i \geq$ 10% of MaxTR **then**
    $Node_i \Rightarrow PVoters$
  end if
end for

Find the voters:
for all PVoters **do**
  if $NodeV\ R_i \geq$ 10% of MaxVR **then**
    $Node_i \Rightarrow voters$
  end if
end for

Find the leader:
for all PLeaders **do**
  if $Node_i$ has MaxLR **then**
    $Node_i$ is the leader
  end if
end for
The block leader packs all its transactions in a defined period to create a new block and publishes it

Algorithm 2. Update leader's and voter's reputation scores after a block's validation.

Find the voting point:
if $NumBT \geq$ 90% of MaxNumBT **then**
  $point \leftarrow 3$
else if $NumBT \geq$ 50% of MaxNumBT **then**
  $point \leftarrow 2$
else if $NumBT \geq 1$ **then**
  $point \leftarrow 1$
else
  $point \leftarrow 0$
end if

Update the leader's and all voter's voting and total reputation scores:
for all nodes in voters **do**
  if $Node_i$ is the block leader **then** $NodeV\ R_i \leftarrow NodeV\ R_i + point$
    $NodeLR_i \leftarrow NodeLR_i + point$

$NodeTR_i \leftarrow NodeTR_i + 2\times$point
else if $Node_i$ validated the block **then**
$NodeV\ R_i \leftarrow NodeV\ R_i + point$
$NodeTR_i \leftarrow NodeTR_i + point$
else if $Node_i$ failed the block **then**
$NodeV\ R_i \leftarrow NodeV\ R_i - point$
$NodeTR_i \leftarrow NodeTR_i - point$
else if $Node_i$ failed to send verification response **then**
$NodeV\ R_i \leftarrow NodeV\ R_i - point$
$NodeTR_i \leftarrow NodeTR_i - point$
end if
end for

Algorithm 3. Update all node's transaction creation scores

Find MaxTrans created by a node in the block
for all Nodes in the network **do**
$PerTrans \leftarrow (NodeTrans_i/MaxTrans) * 100$
if $PerTrans \geq 80$ **then**
$NodeTCR_i \leftarrow NodeTCR_i + 3$
$NodeTR_i \leftarrow NodeTR_i + 3$
else if $PerTrans \geq 50$ **then**
$NodeTCR_i \leftarrow NodeTCR_i + 2$
$NodeTR_i \leftarrow NodeTR_i + 2$
else if $PerTrans \geq 1$ **then**
$NodeTCR_i \leftarrow NodeTCR_i + 1$
$NodeTR_i \leftarrow NodeTR_i + 1$
end if
end for

Algorithm 4. Update reputation scores after a block fails to validate

Find the voting point:
if $NumBT \geq 90\%$ of MaxNumBT **then**
$point \leftarrow 3$
else if $NumBT \geq 50\%$ of MaxNumBT **then**
$point \leftarrow 2$
else if $NumBT \geq 1$ **then**
$point \leftarrow 1$
else
$point \leftarrow 0$
end if

Update all the voter's voting and total scores:

for all Node in voters **do**
if $Node_i$ is the block leader **then** $NodeV\ R_i \leftarrow NodeV\ R_i - point$
$NodeLR_i \leftarrow NodeLR_i - point$
$NodeTR_i \leftarrow NodeTR_i - 2 * point$
else if $Node_i$ failed the block **then**
$NodeV\ R_i \leftarrow NodeV\ R_i + point$
$NodeTR_i \leftarrow NodeTR_i + point$
else if $Node_i$ validated the block **then**
$NodeV\ R_i \leftarrow NodeV\ R_i - point$
$NodeTR_i \leftarrow NodeTR_i - point$
else if $Node_i$ failed to send verification response **then**
$NodeV\ R_i \leftarrow NodeV\ R_i - point$
$NodeTR_i \leftarrow NodeTR_i - point$
end if
end for

Algorithm 5. Update reputation scores according to reputation transactions

for all Transactions in the reputation-block **do**
if Resource Transaction **then**
if Subtype Is CPU **then**
$NodeCPU_{ir} = Rate$
else if Subtype Is MEM **then**
$NodeMEM_{ir} = Rate$
else if Subtype Is STOR **then**
$NodeSTOR_{ir} = Rate$
end if
$NodeTR_{ir} \leftarrow NodeTR_{ir} - NodeRR_{ir}$
$NodeRR_{ir} = NodeCPU_{ir} + NodeMEM_{ir} + NodeSTOR_{ir}$
$NodeTR_{ir} = NodeTR_{ir} + NodeRR_{ir}$
else if Service Transaction **then**
$NodeSR_{ir} \leftarrow NodeSR_{ir} + Rate$
$NodeTR_{ir} \leftarrow NodeTR_{ir} + Rate$
else if Available Transaction **then**
$NodeTR_{ir} \leftarrow NodeTR_{ir} - NodeAR_{ir}$
$NodeAR_{ir} \leftarrow$ Rate
$NodeTR_{ir} \leftarrow NodeTR_{ir} + Rate$ **else if** Defense Transaction **then**
$NodeDFR_{ir} \leftarrow NodeDFR_{ir} + Rate$
$NodeTR_{ir} \leftarrow NodeTR_{ir} + Rate$ **else if** Offense Transaction **then**
$NodeLR \leftarrow 0$
$NodeV\ R \leftarrow 0$
$NodeTCR_{ir} \leftarrow 0$
$NodeRR_{ir} \leftarrow 0$
Keep $NodeDFR_{ir}$ unchanged
$NodeAR_{ir} \leftarrow 0$

$NodeSR_{ir} \leftarrow 0$
$NodeOFR_{ir} \leftarrow NodeOFR_{ir} - 3$
$NodeTCR_{ir} \leftarrow 0$
$NodeCPUR_{ir} \leftarrow 0$
$NodeMEMR_{ir} \leftarrow 0$
$NodeSTOR_{ir} \leftarrow 0$
$NodeTR_{ir} \leftarrow NodeDFR_{ir} + NodeOFR_{ir}$
end if
end for

5. Two nodes actively created transactions and performed the consensus procedure.
6. The nodes were connected to each other through their private keys.
7. When a new node joined the network, it would receive an initial base balance - the initial scores, node name, and a private key.
8. In our experiment, we limited our block size to 1.1 MB[123], which contains a maximum of 2000 transactions.
9. Both business and reputation transactions were created in the same loop. In each loop, at

Table 3. Function Performance

Trial	Number of Transactions	Consensus Time (Seconds)	Block Size	Block Status
1	4	6	3 KB	validated
2	50	6	29 KB	validated
3	100	6	56 KB	validated
4	200	6	111 KB	validated
5	400	7	222 KB	validated
6	600	9	333 KB	validated
7	800	12	443 BB	validated
8	1000	19	554 KB	validated
9	2000	52	1.1 MB	validated

most only one transaction was created for each type.

4.3.4 Consensus Performance Evaluation

We evaluated the performance of R360 relative to other reputation systems in three scenarios. The first one was business transactions only, with no reputation transactions, which was named Function. A node's trustworthiness is measured by its leading score and voting score. The second scenario included business and service reputation transactions. In this model, each run contained half business transactions and half service reputation transactions. The third was R360, which measures a node's trustworthiness from six dimensions, including function, resource, availability, defense, offense, and service. In R360, half of the total transactions were business transactions, and half were reputation transactions.

As we can see from Figure 13, the time needed to reach consensus increased as the number of transactions went up. This is understandable because the complexities of our design are $O(N^2)$ for finding leader and voters, O(NT) for updating reputation scores, and (T^2) for consensus, where N is the number of nodes in the network system, and T is the number of transactions in a block.

Table 3, 4, and 5 display Function, Service, and R360 performances, respectively. Table 6 lists the sub-reputation systems and their corresponding transaction types.

With the same number of transactions, R360 took the shortest consensus time, followed by service reputation, and function reputation model took the longest. This is because the complexity for the consensus algorithm is $O(T^2)$. For R360, the total transactions are divided equally between business transactions and reputation transactions, and its complexity is half of $O(T^2)$. The service reputation transactions contained more information in its subtype than other

reputation transactions, which was why the service model took a little more time than R360. However, the time difference is insignificant. As the number of transactions went

Table 4. Service Performance

Trial	Number of Transactions	Consensus Time (Seconds)	Block Size	Block Status
1	4	6	3 KB	validated
2	52	6	30 KB	validated
3	100	6	58 KB	validated
4	200	6	114 KB	validated
5	400	6	228 KB	validated
6	800	9	454 KB	validated
7	1200	13	681 KB	validated
8	1600	19	908 KB	validated
9	2000	29	1.1 MB	validated

Table 5. R360 Performance

Trial	Number of Transactions	Consensus Time (Seconds)	Block Size	Block Status
1	4	6	3 KB	validated
2	52	6	30 KB	validated
3	100	6	57 KB	validated
4	200	6	113 KB	validated
5	400	7	226 KB	validated
6	800	8	451 KB	validated
7	1200	14	675 KB	validated
8	1600	18	901 KB	validated
9	2000	22	1.1 MB	validated

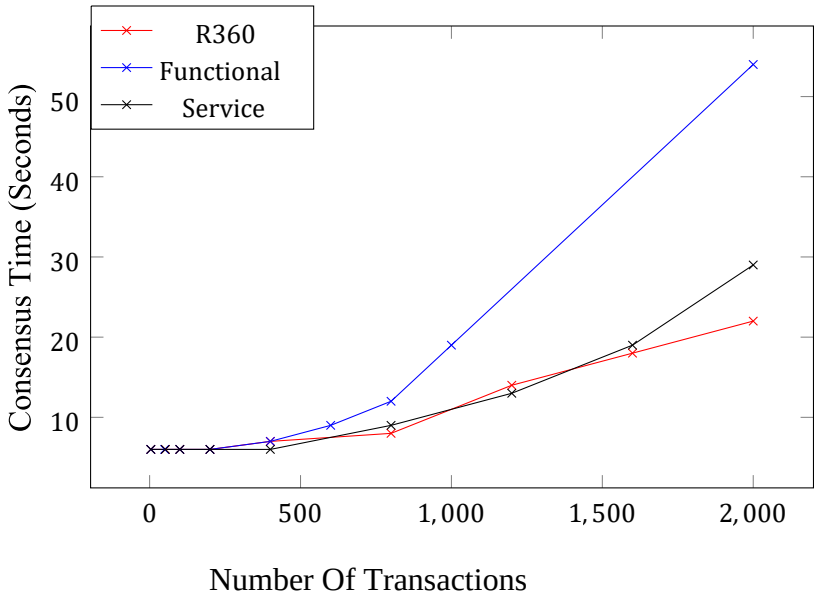

Figure 13. Consensus Performance

Table 6. Reputation System Transactions

No.	Sub-reputation System	Having Business Transactions	Types of Reputation Transactions	Note
1	Function	Yes	0	Business transactions only
2	Resource	Yes	3	Resource Transaction: CPU, Memory, and Storage
3	Availability	Yes	1	Online time rate per year
4	Defense	Yes	1	Security breach transactions
5	Offense	Yes	1	Transactions attacking other nodes or network
6	Services	Yes	1	Service transactions to other nodes or system
7	R360	Yes	7	All the transactions listed above

Table 7. Selfish Attack

Case	Original Transaction	Changes	Number of Transaction in Block	Block Status
1	Not Existing	Add a Transaction	40	Failed
2	Existing	Change Rate Value	40	Failed
3	Existing	Remove a Transaction	40	Failed
4	Existing	Change Receiver	40	Failed

up, the difference in time among the three models increased. R360 increased more slowly than the other two models. If R360 system can greatly increase a node's defensibility and reduce its offense ability, the reputation transactions can be reduced to contain only resource, availability, and service transactions, the block can have more business transactions while maintaining the security of the network. However, because R360 collects more information to measure a node's trustworthiness, more bandwidths will be required. More extensive experiments are needed to evaluate how R360 affects the overall network traffic.

4.3.5 Security Testing

We also carried out a security test with selfish attack in four scenarios: the leading node added an extra transaction to the to-be-proposed block, removed a transaction from the block, changed a transaction's rate value, and changed the evaluation's receiver. From Table 7, we can see that the consensus failed when the block leader modified a transaction. This is because the voters uncovered their local blocks were not the same as the altered blocks, so they failed the proposed blocks.

4.4 Security Analysis

We tested selfish attack with R360 system. In this section, we analyze the threats of other common security attacks on R360.

1. Bad-mouthing attack [94], [95]. Bad-mouthing attack provides dishonest recommendations to defame good nodes, which is the most straightforward attack [124]. In our protocol, a reputation transaction must obtain the consensus of two thirds of voters in order to be effective, so malicious nodes cannot continuously malign a specific node or any other nodes and hope to improve its own trustworthiness ranking in return.
2. Replay attack. If the same transaction was used multiple times, other nodes that were involved in this transaction would take notice. If such malicious behavior is uncovered, the offensive node's positive reputation scores will be all wiped out.
3. On-off attack. A node's malicious behavior will result in a bad reputation score, which will negatively impact its future reputation scores by reducing this node's chance to become a leader or a voter. Losing chance means losing the opportunities to improve its reputation.
4. Sybil attack. In R360, each node has a public key that is tied to a personal identification when a node is registered, no node can "legally" create multiple IDs. Therefore, Sybil attack cannot occur.
5. Flash attack. Our reputation system is resilient to flash attacks. Even an attacker with high computing power, depending on when that attacker joins, needs to accumulate good scores over a very long period of time before being able to gain enough reputation to harm the system. For example, a node has to accumulate enough high leader score in order to be a leader, enough high voting score in order to be a voter.
6. Blockchain consistency and system liveness. R360 is built on blockchain network concept,

the voters must either validate or fail a block.

7. Double spending attacks. Since digital currency is not used in R360 system, double spending attacks will not occur in our system.
8. Eclipse attacks and isolated leaders. Since all voters broadcast their results to the entire network, the leader will be able to receive the messages too. If a voter refuses to send its response, it will be penalized with a negative reputation score.

4.5 Conclusion

A new system, R360, has been designed to thoroughly measure and maintain the reputation of nodes in an untrusted network. The design has been implemented to test the consensus performance and security. The results showed that R360 took less consensus time and enhanced system security. Our experiments showed R360 can prevent selfish attack. Further analysis illustrated that it could prevent common attacks on blockchain reputation system. By putting on strict reputation evaluation rules, R360 can greatly enhance the security in untrusted blockchain smart grid. However, it also increases the network traffic. Our design was simulated on a single computer. Future work should implement R360 on real network systems to test its network performance and security. Additionally, the scoring system may need to be optimized. Further study on how such reputation scores affect individual node’s behavior is also desirable.

CHAPTER 5

A COUNTERMEASURE FOR DOUBLE SPENDING

5.1 System Design

5.1.1 The key points of the proposed design

1. The detection results of double spending attacks will be included in the block consensus.
2. Only one of the most reputable nodes is selected as the detector, which frees up other nodes to perform other duties. This detector checks the transactions during the entire transaction receiving period.
3. The detector changes frequently and its communication style is not much different than other nodes. These two features reduce the chance that the detection node becomes the target of DDoS.
4. The detector and other nodes work in a parallel fashion. The detector checks double spending during the whole transaction collecting time interval and sends out a warning message upon finding any conflicting transactions.
5. Upon discovering a conflicting transaction, as a penalty to the attacker, the detector will create two penalizing transactions, one requiring the offending node to pay the detector the same payment as in the conflicting transaction, the other one reducing the offending node's reputation scores. Lower reputation scores will reduce the node's chance to participate in important tasks in the future, including creating transactions.
6. Upon receiving double spending warning message, all nodes remove the conflicting transaction from their memory pools and add the two penalizing transactions to their memory pools.
7. Our final plan is to check all transaction's correctness before consensus.

5.1.2 Double Spending Occurrence Scenarios

To implement a double spending attack, the attacker first creates two transactions. The first transaction TV, transaction to vendor, lists the vendor as the recipient of the payment, and the second transaction TA, transaction to attacker, lists the attacker as the recipient of the payment. The attacker's goal is to have the vendor accept TV long enough to deliver the goods or services and have the rest of the network accept TA so that the attacker keeps the money. The attacker sends out both transactions. TV is transmitted directly to the vendor, while TA is broadcasted to the rest of the network. In order for a double-spending attack to be successful, 1) The attacker must know the vendor's IP address so it can connect to the vendor directly and send TV to the vendor; 2) The vendor must receive TV before TA arrives [104] to ensure that TA will be automatically dropped when the vendor eventually receives it; 3) TA must be confirmed in the block chain first or else TV will actually be confirmed and that block will become the accepted block in the network; 4) Given an equal propagation of both transactions, there is a 50 percent chance for either transaction to be confirmed. More nodes are required to work on TA than on TV to increase TA's likelihood of being accepted into the blockchain, and it requires that the vendor only sees TV. Because the neighbors of the vendor will likely get TV first (directly from the vendor) and thus drop TA rather than propagate it to the vendor. This kind of double-spending can succeed in fast-paying transactions in which the vendor does not wait for confirmation.

Another form of double-spending attack is the block withholding attack [125], [126] in which the attacker pools resources to create a block BV, which contains TV. The attacker blocks all other connections to the vendor and prevents the vendor from ever receiving all other blocks confirming TA while sending BV the moment it is calculated. BV represents the block

containing TV. The attacker essentially creates a fork in the block chain containing BV that will eventually be disregarded since no other mining pools work to extend this side of the fork [108]. This method of double-spending can succeed in slow-pay transactions in which the vendor awaits confirmation.

5.1.3 Design Assumptions

1. Our design is based on energy trading in smart grid and the payment methods can be tokens, money orders, checks, or any other payments that can be defined as unique and reusable.
2. Our double spending countermeasure is for slow-payment situations, such as paying for electricity bill or buying renewable energy by consumers. We assume the attacker will try to use the same payment in at most two consecutive blocks, one is a previously validated block and the other is currently collecting transactions. If there are two conflicting transactions in one validated block, the transaction with the later timestamp will decide the block's final status.
3. Our design is based on our previous paper [127] and adds another reputation score, detection, to the reputation formula. The score of detection DT is cumulative and DT is calculated the same way as voter's score.

$$DT_i = \pm Point \tag{30}$$

$$DT = \sum_{i=1} DT_i \tag{31}$$

The offense score is calculated with all offense behaviors except double spending. The total reputation score is:

$$Reputation = Resource + Defense + Availability + Offense + Service + Function + Detection + DoubleSpend \tag{32}$$

4. We also use a similar consensus algorithm as in [127] by adding detection steps to the consensus in [127].
5. All the fields in the business transactions, except these two fields: the timestamp and the payment's receiver [127], are compared to decide if two transactions are conflicting or not.
6. The experimental environment is the same as in [127].

5.1.4 Double Spending Attack Models

1. Case 1: suppose the attacker only sends a vendor TV, and only the vendor's block BV (block created by vender) contains TV. When the vendor receives the leader's block BL (block created by leader) and finds out:

$$BV \neq BL$$

The vendor will fail the BL block. If BL is passed, the vendor will drop its local block BV and take BL. The attacker will not get the goods or service.

2. Case 2: if both TV and TA are put into the current block, the transaction with the later timestamp will be discovered by the detector. If BL is validated, the attacker either gets the service or keeps its money. It also gets two penalty transactions at the same time.
3. Case 3: TV is in a validated block and TA is added to the current block. Upon discovering TA conflicts with a transaction in the previously validated block, the detector asks all nodes to drop TA and add two penalty transactions in their memory pools. The attacker will not get its money back.
4. Case 4: The victim is the block leader and TA was only sent to the leader, a special case for Case 1. BL is dropped and the attacker failed its purpose. The detector is unaware of the attack so there will be no penalty for the attacker.
5. Case 5: The detector is the victim, another special case for Case 1. Any double spending

will be discovered.

5.2 Double Spending Detection Procedure

Figure 14 shows the double-spending detection flow chart.

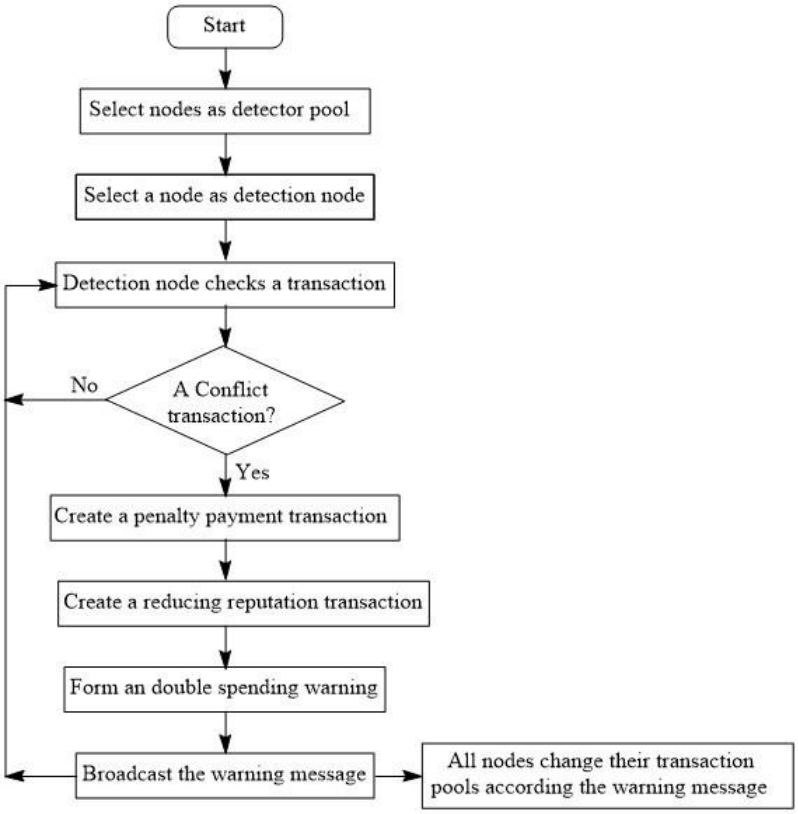

Figure 14. Double-Spending Detection Flow Chart

1. At the beginning of each time interval, each node selects nodes with at least 90% of the highest total reputation score among all nodes as potential detector pool.
2. Select the node with the highest detection score from the potential detector pool as the detector.
3. The detector continuously checks and broadcasts conflicting transactions against the previously validated block and the transactions in current time interval. If an offending transaction is discovered, as a reward to the detector and a penalty to the attacker, the detector will create two transactions, one transaction will pay the same amount as in the

conflicting transaction to the detector, the other one will be an offending reputation transaction to the attacker. The offending transaction and the two penalty transactions will form a warning message and be broadcasted to all nodes in the network.

Table 8. Detection and Consensus Performance

Testing	Number of Transaction	Time of Consensus	Time of Detection	Status of Block	Double Spending Detected	Size of Block
1	7	5	5	Passed	Yes	4KB
2	53	5	5	Passed	Yes	32KB
3	103	5	5	Passed	Yes	58KB
4	203	5	5	Passed	Yes	115KB
5	403	6	5	Passed	Yes	227KB
6	803	8	5	Passed	Yes	452KB
7	1203	12	5	Passed	Yes	677KB
8	1603	18	5	Passed	Yes	902KB
9	2003	19	5	Passed	Yes	1.1MB

4. Upon receiving the detector's warning message, all nodes replace the offending transaction with the detector's reward transaction and add the attacker's offending reputation transaction in their memory pools.

Upon finishing the consensus, all nodes update the offending node's scores accordingly, and update the detector's reputation scores the same way as updating a node's voting scores.

5.3 Experimental Results and Analysis

5.3.1 Detection and Consensus Performance

Table 8 and Figure 15 display the experimental data and graph. The normal detection time was five seconds for one double-spending detection, which was not affected by the number

of transactions in the block, the time was just necessary to run the detection program, which performed much better than all the three cases in [107], All ENHOBS, 1% Skinny and 2% Skinny. The consensus time did not change until the number of transactions reached 400. The pattern and values of the consensus are similar to the results we found previously [127]. This is expected because the consensus was conducted in a similar way.

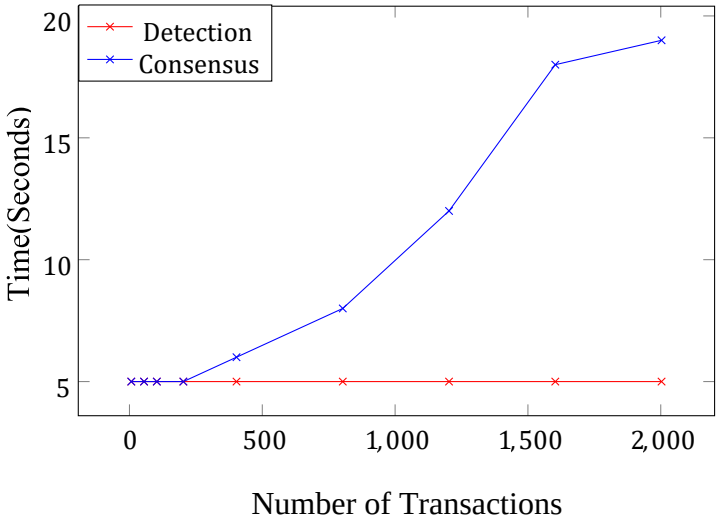

Figure 15. Double-Spending Consensus and Detection Performance

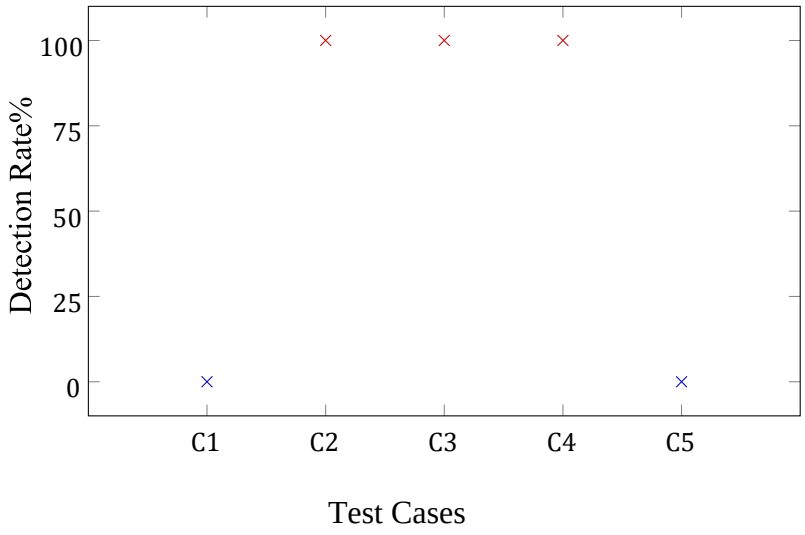

Figure 16. Double-Spending Detection Effectiveness

Our approach used much less CPU time than [107]. Our detection node used 0.1% CPU, while with every node acting as observers in [107], the CPU utilization jumped from 31% to 50.6%, with the maximum CPU utilization reaching as high as 96%. This is understandable because one detection node will definitely use much less resources than many nodes as detection nodes at the same time.

We tested all five double spending attack cases, the detection rate was 100% for C2, C3, C4, and 0 for C1 and C5, as shown in Figure 16. We tested the detection rates with blockchain standard maximum block size 1.1 MB, which is equivalent to 2000 transactions. Worth noticing is that the detection rate is 0 when the double spending victim is the vendor or the block leader. This is because the conflicting transaction was only sent to the vendor or leader and the detector didn't receive TA. In both situations, the local block was not the same as the proposed block. More system resources were wasted when the leader was the victim than when non-leader node was attacked. This is because non-leader node only needs to drop its local block while the leader's block was dropped after all the nodes in the network had verified it.

5.3.2 Detection Complexity

The complexity of the double spending detection is O(N), where N is the number of transactions in a block. The conflicting transaction will be searched throughout the previously validated block and current block, so the detection time is 2O(N). The complexity of consensus algorithm is $O(N^2)$ because two block transactions are compared.

5.3.3 Security Analysis

1. TV was only sent to the vendor, whereas other nodes received TA. The block BL, which contained TA, was validated, and the vendor dropped its local block BV, which contained

TV. The attack failed to get the service and kept its money. No penalty was given to the attacker either because the attack was not uncovered.

2. When TV was included in the previous block and TA was in current block. The detector found TA was a conflicting transaction and sent a rewarding transaction to itself and a penalty reputation transaction to the attacker.

The block BL without TA was validated. The attacker only got the service.

3. When TV was in current block already and TA was broadcasted again, the detector found TA and sent a rewarding transaction to itself and a penalty reputation transaction to the attacker. The block containing TV not TA was validated the attacker only received the service.
4. When the leader was attacked, the detection failed. The attacker either got the service if TV was in the previous block or got nothing if both TV and TA in current block. No damage to any nodes but the system resources were wasted.
5. When the detector was attacked, the double spending detection rate is 100%. Other nodes did not need to change their memory pools. The attacker got the service and two penalty transactions.

5.4 Conclusion

In this design, we proposed a double spending countermeasure which can effectively detect double spending in two consecutive blocks. Our design puts detection results into the consensus mechanism, handles the offending transaction, and has a mechanism to prevent the perpetrator to double spend again. Comparing to other countermeasures, such as time-period monitoring and inserting observers, our method constantly monitors transactions, uses less computing resources, and reduces network traffic and the management overhead on observers.

Another advantage of our design is our detection node is not fixated and it does not have a specific communication pattern, which will less likely attract DDoS attack. Future research will, 1) handle the situation when the block leader is attacked; 2) check all kinds of transactions, such as sybil attack, selfish mining attack, business and service transactions; 3) expand the detection to the entire blockchain rather than just two consecutive blocks. 2) and 3) are parts of the reasons we chose to have a single node to perform the detection duty. 4) select a set of nodes as the detectors to share the checking responsibilities and avoid malicious detectors.

CHAPTER 6

SUMMARY AND FUTURE WORK

6.1 Dissertation Summary

The rapid advancement of modern technology demands that the power system be transformed from traditional centralized system to a distributed system dubbed smart grid, which should connect existing and emerging energy producers and consumers efficiently and seamlessly through a highly effective two-way communication network. However, such an open network will create unprecedented challenges to its security and management, as it must be secure, fault-tolerant, and available to the users.

Blockchain technology is a transparent, secure, decentralized, and trusted cyber infrastructure that shares, replicates, and synchronizes data across the entire network to achieve security through consensus and redundancy. This technology has found applications in various fields, e.g., cryptocurrency, medical data sharing, and personal identity security. However, because of transparency, blockchain also has its inherent disadvantages, e.g., violating user privacy, low network throughput, and trustless network environment.

The research in this dissertation aims to mitigate some of the challenges that blockchain faces when it is applied to smart grid. Specifically, three systems have been designed to tackle some of these problems.

In the first study, the information-centric blockchain model, we have focused on data privacy. Because blockchain is transparent, and all data can be seen by every node in the network, this may expose user's IP address, violating privacy and making the user vulnerable to attacks. We have built the information centric blockchain model to hide the sources and destinations of data. In this model, the transactions created by nodes in the network are

categorized into different groups and encrypted by asymmetric keys, which guarantees that only the intended receivers can see the data so that data confidentiality is preserved. All transactions are sent on behalf of their groups, rather than individual users, which hides the data sources to preserve the privacy. Preliminary implementation has verified the feasibility of the model, and our analysis has demonstrated its effectiveness in securing data source privacy, increasing network throughput, and reducing storage usage in a simulated environment.

In the second study, we aim to increase the trustworthiness in an untrusted network environment. Blockchain network is composed of trustless nodes. To enhance the trustworthiness of such network and encourage honest behavior among the participating nodes, an R360 reputation system has been designed to evaluate the behaviors of all nodes. In the R360 system, the computing power, online time, defense ability, function, offense, and service quality of the nodes are assigned scores and constantly monitored and evaluated. All nodes can create and be involved in validating blocks. The performance of a node in the measured dimensions will affect its reputation scores, which in turn will affect its future qualification, privileges, and job assignments either positively or negatively. Preliminary implementation showed that our thorough, self-operated, and closed-loop design enhanced network security by preventing both internal and external attacks. Further security analysis showed that the reputation model is feasible and enhances blockchain system's security.

Double spending is one of the most concerned security attacks for blockchain networks where payments are involved. In the third study, a countermeasure for double spending has been designed, where one of the most reputable nodes was selected as the detector. The detector continuously monitors transactions in two consecutive blocks to uncover any conflicting transactions. When such transactions are uncovered, the perpetrating node will be identified and

penalized for the current attack. Additionally, the perpetrating node will also damage its reputation score. This mechanism can be used to prevent potential attacks from the same node in the future. Our experiment has shown our design can detect double spending effectively while using much less detection time and resources than other designs reported previously.

6.2 Summary of Research Contributions and Technical Insights

This dissertation provides three novel approaches that address the challenges that arise when blockchain technology is applied in smart grid.

1. Protect user's privacy and mitigate some blockchain-associated problems in information-centric model.
   (a) A user's public key is encrypted and only the authority can decrypt it. This protects user's identity.
   (b) All transactions are encrypted and only the intended users can decrypt them. This keeps the information confidential.
   (c) All transactions are sent based on information groups, not individual nodes, which hides the true sources of transactions.
   (d) All nodes and transactions are divided into groups. The nodes will only save transactions that belong to the same group, avoiding the requirement that every node stores all transactions in the entire network. This reduces storage redundancy, lowers energy consumption with increased efficiency.
   (e) All group's consensuses are executed at the same time, which increases the network throughput and the network's scalability.
   (f) Transactions are saved in different blockchains based on the types, which simplifies transaction searching overload and enhances search efficiency.

2. Enhance network security and mitigate security attacks.
    (a) Our design is the first research to measure a node's trustworthiness in a more thorough manner.
    (b) All node's reputation scores are based on their the computing power, online time, defense ability, function, offense, and service quality, which are designed to control node's behavior, privileges, and job assignments.
    (c) The experiment showed that the reputation-based consensus model performs better than models with no reputation considerations.
    (d) The experiment showed the reputation-based model can prevent selfish attack.
3. A countermeasure for double spending.
    (a) This model provides a systematic method against double spending, which not only detects, warning, and punishing the attacker but also uses this information to set up a strategy to prevent future attacks.
    (b) Because only one node acts as the detector, system resources will be saved and management overhead will be reduced.
    (c) The perpetrating node will be penalized for its misbehaving and be prevented from double spending again.
    (d) A validated transaction does not necessarily mean it is an accurate transaction. It only means that every node has received the same transaction. Our design can be used to check if a transaction is truly accurate.

6.3 Future Work

In this section, we briefly discuss some limitations of the three studies and potential future work.

1. In the information-centric blockchain model for smart grids, we have implemented the model and demonstrated the feasibility for its application to enhance blockchain security. However, our implementation is on a simplified model with limited testing data. A more comprehensive study needs to be carried out, and a more thorough testing with a large data set needs to be performed.
2. In the reputation-based model, our design was simulated on a single computer. Future work should implement the R360 system on real network systems to test its network performance and security. Additionally, the scoring system may need to be fine-tuned and optimized. Further study on how such reputation scores affect individual node's behavior in real world is also desirable.
3. Double spending countermeasure may be expanded to detect other types of attacks, e.g., sybil attack, and check all transactions; the current design only check against the two consecutive blocks, this checking can be applied on the whole blockchain; in current model, only one node was used as the detection node. In future research, a set of nodes can be used as the detectors; this design also needs to be tested in real network for its effectiveness and impact on network's overall performance.

REFERENCES

[1] Muhammad Baqer Mollah, Jun Zhao, Dusit Niyato, Kwok-Yan Lam, Xin Zhang, Amer M.Y.M. Ghias, Leong Hai Koh, Lei Yang. Blockchain for future smart grid: A comprehensive survey. *IEEE Internet of Things Journal*, pages 1–1, 2020.

[2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2009, http://www.bitcoin.org/bitcoin.pdf.

[3] Brandon T. McDaniel, Samreen Mahmood, Mehmood Chadhar, Selena Firmin. Undecided. *Human Behavior and Emerging Technologies*, 2022:7384000, 2022.

[4] Hao Liang Peng Zhuang, Talha Zamir. Blockchain for cybersecurity in smart grid: A comprehensive survey. *IEEE Transactions on Industrial Informatics*, 17(1):3–19, 2021.

[5] Arshdeep Bahga, Vijay K. Madisetti. Blockchain platform for industrial internet of things. *Journal of Software Engineering and Applications*, 9(10):[36]533–546, 2016.

[6] Li, Yun and Teng, Yun and Cao, Rongrong and Li, Ningfeng. Research on coordination control system of virtual power plant based on blockchain. *2019 IEEE 8th International Conference on Advanced Power System Automation and Protection (APAP)*, pages 1377–1383, 2019.

[7] Ralph C. Merkle. A digital signature based on a conventional encryption function. *Conference on the theory and application of cryptographic techniques*, 293:369–378, 1987.

[8] Gavin Wood. Ethereum: A secure decentralised generalised transaction ledger. *Ethereum project yellow paper*, 151:1–32, 2014.

[9] Muhammad Salek Ali, Massimo Vecchio, Miguel Pincheira, Koustabh Dolui, Fabio Antonelli, Mubashir Husain Rehmani. Applications of blockchains in the internet of things:

A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 21(2):1676–1717, 2019.

[10] Julija Golosova, Andrejs Romanovs. The advantages and disadvantages of the blockchain technology. *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, pages 1–6, 2018.

[11] Zhaoyang Dong, Fengji Luo, Gaoqi Liang. Blockchain: a secure, decentralized, trusted cyber infrastructure solution for future energy systems. *Journal of Modern Power Systems and Clean Energy*, 6:958—-967, Jul. 2018.

[12] Types of blockchains–decide which one is better for your investment needs. *Data Flair*, Accessed: November, 2022, https://dataflair.training/blogs/types-of-blockchain/.

[13] Yohannes T. Aklilu, Jianguo Ding. Survey on blockchain for smart grid management, control, and operation. *Energies*, 15(1), 2022.

[14] Blockchain for hospitality. Accessed: November 15, 2022, https://www.hospitalitynet.org/file/152008497.pdf.

[15] Omar Dib, Kei-Leo Brousmiche, Antoine Durand, Eric Thea, Elyes Ben Hamida. Consortium blockchains: Overview, applications, and challenges. *Int. J. Adv. Telecommun*, 11:51–64, 2018.

[16] Seyed Mojtaba Hosseini Bamakan, Amirhossein Motavali, Alireza Babaei Bondarti. A survey of blockchain consensus algorithms performance evaluation criteria. *Expert Systems with Applications*, 154:113385, 2020.

[17] Hamid Malik, Ahsan Manzoor, Mika Ylianttila, Madhusanka Liyanage. Performance analysis of blockchain based smart grids with ethereum and hyperledger implementations.

2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS), pages 1– 5, 2019.

[18] Xiang Fu, Huaimin Wang, Peichang Shi. A survey of blockchain consensus algorithms: mechanism, design, and applications. *Science China Information Sciences*, 64(2):121101, 2020.

[19] Florian Tschorsch, Bjorn Scheuermann. Bitcoin and beyond: A technical survey on decentralized digital currencies. *IEEE Communications Surveys & Tutorials*, 18(3):2084–2123, 2016.

[20] Nick Szabo. Smart contracts : Building blocks for digital markets. 2018.

[21] Smart Contracts. Accessed: October, 2022, https://www.fon.hum.uva.nl/rob/courses/informationinspeech/cdrom/literature/lotwinterschool2006/szabo.best.vwh.net/smart.contracts.html.

[22] Vitalik Buterin. A next-generation smart contract and decentralized application platform. *Ethereum White Paper*, 3:37, 2014.

[23] Jamal Hayat Mosakheil. Security threats classification in blockchains. *Culminating Projects in Information Assurance*, (48), May 2018.

[24] Dataflair Team. Advantages and disadvantages of blockchain technology. *online*, Accessed: October, 2022, https://dataflair.training/blogs/advantages-and-disadvantages-of-blockchain/.

[25] Apriorit. Blockchain attack vectors: Vulnerabilities of the most secure technology. Accessed: October, 2022, https://www.apriorit.com/devblog/578-blockchain-attack-vectors.

[26] Shravan Garlapati. Blockchain for iot-based nans and hans in smart grid. *Electrical Engineering and Systems Science*, Jan. 2020.

[27] BUS 237-E103: Woochul Song, Stone Shi, Victoria Xu, Gursahib Gill. Advantages & disadvantages of blockchain technology. *Blockchain Technology*, November 2016.

[28] Warren Fauvel. Blockchain advantage and disadvantages. *Nudjed*, August 2017.

[29] Abigael Okikijesu Bada, Amalia Damianou, Constantinos Marios Angelopoulos, Vasilios Katos. Towards a green blockchain: A review of consensus mechanisms and their energy consumption. *2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 503–511, 2021.

[30] Deepak Puthal, Nisha Malik, Saraju P. Mohanty, Elias Kougianos, Das, Gautam. Everything you wanted to know about the blockchain: Its promise, components, processes, and problems. *IEEE Consumer Electronics Magazine*, 7(4):6–14, 2018.

[31] Markus Jakobsson, Ari Juels. *Proofs of Work and Bread Pudding Protocols(Extended Abstract)*, pages 258–272. Springer US, Boston, MA, 1999.

[32] Jason Spasovski, Peter Eklund. Proof of stake blockchain: Performance and scalability for groupware communications. *Conference paper*, November 2017.

[33] Proof of stack. Accessed: November 2022, https://en. bitcoin.it/wiki/Proof of Stake.

[34] Cong T. Nguyen, Dinh Thai Hoang, Diep N. Nguyen, Dusit Niyato, Huynh Tuong Nguyen, Eryk Dutkiewicz. Proof-of-stake consensus mechanisms for future blockchain networks: Fundamentals, applications, and opportunities. *IEEE Access*, 7:85727–85745, 2019.

[35] DPoS. Accessed: October 2022. https://en.bitcoinwiki.org/wiki/DPoS.

[36] LPoS. Leasing proof of stake (pos/lpos). Accessed: October 2022, https://tokens-economy.gitbook.io/consensus/chain-based-proof-ofstake/leasing-proof-of-stake-pos-lpos.

[37] Iddo Bentov, Char-Tung Lee, Alex Mizrahi, Meni Rosenfeld. Proof of activity: Extending bitcoin's proof of work via proof of stake [extended abstract]y. *SIGMETRICS Perform. Evaluation Rev.*, 42:34–37, 2014.

[38] Tuyet Duong, Lei Fan, Jonathan Katz, Phuc Thai, Hong-Sheng Zhou. 2-hop blockchain: Combining proof-of-work and proof-of-stake securely. *European Symposium on Research in Computer Security*, 2020.

[39] Alexander Chepurnoy, Tuyet Duong, Lei Fan, Hong-Sheng Zhou. Twinscoin: A cryptocurrency via proof-of-work and proof-of-stake. *Cryptology ePrint Archive*, page 232, 2017.

[40] Proof of Burn. Accessed: October 2022, https: //en.bitcoin.it/wiki/Proof of burn.

[41] Slimcoin a peer-to-peer crypto-currency with proof-of-burn. Accessed: October 2022, https://slimcoin.info/whitepaperSLM.pdf.

[42] Lin Chen, Lei Xu, Nolan Shah, Zhimin Gao, Yang Lu, Weidong Shi. On security analysis of proof-of-elapsed-time (poet). *International Symposium on Stabilization, Safety, and Security of Distributed Systems*, pages 282– 297, 2017.

[43] Sawtooth hyperledger. Accessed: November 2023, https://sawtooth.hyperledger.org/.

[44] Stefano De Angelis, Leonardo Aniello, Roberto Baldoni, Federico Lombardi, Andrea Margheri, Vladimiro Sassone. Pbft vs proof-of-authority: Applying the cap theorem to permissioned blockchain. *University of Southampton Institutional Repository*, 2018.

[45] Parity Ethereum. Accessed: November 2022, https://wiki.parity.io/parity-ethereum.

[46] Proof of reputation (por). Accessed: November 2022, https://tokenseconomy.gitbook.io/consensus/chain-based-proof-of-capacityspace/proof-of-reputation-por.

[47] Consensus. Accessed: November 2022, https://tokenseconomy.gitbook.io/consensus/chain-based-proof-of-capacityspace/proof-of-reputation-por.

[48] Sidharth Shyamsukha, Pronaya Bhattacharya, Farnazbanu Patel, Sudeep Tanwar, Rajesh Gupta, Emil Pricop. Porf: Proof-of-reputation-based consensus scheme for fair transaction ordering. pages 1–6, 2021.

[49] Miguel Castro, Barbara Liskov. Practical byzantine fault tolerance. *OSDI*, 99(1999):173–186, 1999.

[50] Leslie Lamport, Robert Shostak, Marshall Pease. The byzantine generals problem. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 4:382—-401, 1982.

[51] Sang, L., Hexmoor, H. (2021). Information-Centric Blockchain Technology for the Smart Grid. International Journal of Network Security & Its Applications (IJNSA), Vol.13, No.3, May 2021.

[52] Barbara Vieira, Erik Poll. A security protocol for information-centric networking in smart grids. *SEGS '13 Proceedings of the first ACM workshop on Smart energy grid security*, pages 1–10, 2013.

[53] Xin Chen, Jiachen Shen, Zhenfu Cao, Xiaolei Dong. A blockchain-based privacy-preserving scheme for smart grids. *Association for Computing machinery*, pages 120–124, Mar. 2020.

[54] Proof of Importance (PoI) in Blockchain. Accessed: November 2022, https://www.naukri.com/learning/articles/proof-of-importance-poi-inblockchain.

[55] Proof of Capacity. Accessed: November 2022, https://learn.bybit.com/glossary/definition-proof-of-capacity-poc.

[56] What Is Proof of Weight? Accessed: November 2022, https://coincodex.com/article/2617/what-is-proof-of-weight.

[57] William Stallings. *Fundamental Principles of Optical Lithography*. pages 6-14, Wiley 2007.

[58] Application security definition. *Vmware*, Accessed: November 2022, https://www.vmware.com/topics/glossary/content/applicationsecurity.html.

[59] Oxford learner's dictionary. Accessed: July 2022, https://www.oxfordlearnersdictionaries.com/us/definition.

[60] Trust. Accessed: July 2022, https://dictionary.cambridge.org/dictionary/english/trust.

[61] Trust. Accessed: July 2022, https://www.ldoceonline.com/dictionary/trust.

[62] Trust. Accessed: July 2022, https://www.merriamwebster.com/dictionary/trust.

[63] Lik Mui, Mojdeh Mohtashemi,Ari Halberstadt. A computational model of trust and reputation. *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pages 2431–2439, 2002.

[64] I. Safak Bayram, Muhammad Z. Shakir, Mohamed Abdallah, Khalid Qaraqe. A survey on energy trading in smart grid. *In Proceedings of the 2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada*, pages 258–262, November 2014.

[65] Zahraoui Younes, Ibrahim Alhamrouni, Saad Mekhilef, M. Reyasudin Basir Khan. Blockchain applications and challenges in smart grid. *2021 IEEE Conference on Energy Conversion (CENCON)*, pages 208–213, 2021.

[66] Pardeep Kumar, Yun Lin, Guangdong Bai, Andrew Paverd, Jin Song Don, Andrew Martin. Smart grid metering networks: A survey on security, privacy, and open research issues. *IEEE Communications Surveys Tutorials*, 21(3):2886–2927, 2019.

[67] Reza Soltani, Marzia Zaman, Rohit Joshi, Srinivas Sampalli. Distributed ledger technologies and their applications: A review. *Applied Sciences*, 12(15), 2022.

[68] Adedayo Aderibole, Aamna Aljarwan, Muhammad Habib Ur Rehman, Hatem H. Zeineldin, Toufic Mezher, Khaled Salah, Ernesto Damiani, Davor Svetinovic. Blockchain technology for smart grids: Decentralized nist conceptual model. *IEEE Access*, 8:43177–43190, 2020.

[69] Michael Mylrea, Sri Nikhil Gupta Gourisetti. Blockchain for smart grid resilience: Exchanging distributed energy at speed, scale, and security. *2017 Resilience Week (RWS)*, pages 18–23, 2017.

[70] Mina Bergeroy Ryssdal. Blockchain technology implementation for electric vehicle charging within the smart grid architecture model. *Master's thesis in Energy and Environmental Engineering*, pages 103–107, 2019.

[71] Sarmistha De Dutta, Ramjee Prasad. Security for smart grid in 5g and beyond networks. *Wireless Personal Communications*, 106(1):261–273, 2019.

[72] Amrita Ghosal, Mauro Conti. Key management systems for smart grid advanced metering infrastructure: A survey. *IEEE Communications Surveys Tutorials*, 21(3):2831–2848, 2019.

[73] Muhammed Zekeriya Gunduza, Resul Das. Cyber-security on smart grid: Threats and potential solutions. *Computer Networks*, page 107094, 2020.

[74] Shama Naz Islam, Zubair Baig, Sherali Zeadally. Physical layer security for the smart grid: Vulnerabilities, threats, and countermeasures. *IEEE Transactions on Industrial Informatics*, 15(12):6522–6530, 2019.

[75] Sagar K. Rastogi, Arun Sankar, Kushagra Manglik, Santanu K. Mishra, Saraju P. Mohanty. Toward the vision of all-electric vehicles in a decade [energy and security. *IEEE Consumer Electronics Magazine*, 8(2):103–107, 2019.

[76] Zhitao Guan, Guanlin Si, Xiaosong Zhang, Longfei Wu, Nadra Guizani, Xiaojiang Du, Yinglong Ma. Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities. *IEEE Communications Magazine*, 56(7):82–88, Jul. 2018.

[77] Zijian Zhang, Zhan Qin, Liehuang Zhu, Jian Weng, Kui Ren. Cost-friendly differential privacy for smart meters: Exploiting the dual roles of the noise. *IEEE Transactions on Smart Grid*, 8(2):619–626, 2017.

[78] Kaiping Xue, Shaohua Li, Jianan Hong, Yingjie Xue, Nenghai Yu, Peilin Hong. Two-cloud secure database for numeric-related sql range queries with privacy preserving. *IEEE Transactions on Information Forensics and Security*, 12(7):1596–1608, 2017.

[79] Jun Wu, Mianxiong Dong, Kaoru Ota, Lin Liang, Zhenyu Zhou. Securing distributed storage for social internet of things using regenerating code and blom key agreement. *Peer-to-Peer Networking and Applications*, 8:1133– 1142, 2015.

[80] Keke Gai, Yulu Wu, Liehuang Zhu, Meikang Qiu, Meng Shen. Privacypreserving energy trading using consortium blockchain in smart grid. *IEEE Transactions on Industrial Informatics*, 15(6):3548–3558, 2019.

[81] Zakaria About EI Houda, Abdelhakim Hafid, Lyes Khoukhi. Blockchain meets ami: Towards secure advanced metering infrastructures. *IEEE ICC*, February 2020.

[82] Quang Nhat Tran, Benjamin P. Turnbull, Hao-Tian Wu, A. J. S. de Silva, Katerina Kormusheva, Jiankun Hu. A survey on privacy-preserving blockchain systems (ppbs) and a novel ppbs-based framework for smart agriculture. *IEEE Open Journal of the Computer Society*, 2:72–84, 2021.

[83] Feng Gao, Liehuang Zhu, Meng Shen, Kashif Sharif, Zhiguo Wan, Kui Ren. A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks. *IEEE Network*, 32(6):184–192, 2018.

[84] David Gabay, Kemal Akkaya, Mumin Cebe. Privacy-preserving authentication scheme for connected electric vehicles using blockchain and zero knowledge proofs. *IEEE Transactions on Vehicular Technology*, 69(6):5760–5772, 2020.

[85] Keke Gai, Yulu Wu, Liehuang Zhu, Lei Xu, Yan Zhang. Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks. *IEEE Internet of Things Journal*, 6(5):7992–8004, Oct. 2019.

[86] Mohamed Baza, and Mahmoud Nabil, and Muhammad Ismail, and Mohamed Mahmoud, and Erchin Serpedin, and Mohammad Ashiqur Rahman. Blockchain-based charging coordination mechanism for smart grid energy storage units. *2019 IEEE International Conference on Blockchain (Blockchain)*, pages 504–509, 2019.

[87] Zhitao Guan, Yue Zhang, Liehuang Zhu, Longfei Wu, Shui Yu. Effect: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. *Science China Information Sciences volume*, 62(32103), 2019.

[88] Kun Wang, Yun Shao, Lei Shu, Chunsheng Zhu, Yan Zhang. Mobile big data fault-tolerant processing for ehealth networks. *IEEE Network*, 30(1):36–42, 2016.

[89] Xuanxia Yao, Xiaoguang Han, Xiaojiang Du, Xianwei Zhou. A lightweight multicast authentication mechanism for small scale iot applications. *IEEE Sensors Journal*, 13(10):3693–3701, 2013.

[90] Shengmin Tan, Xu Wang, Chuanwen Jiang. Privacy-preserving energy scheduling for escos based on energy blockchain network. *Energies*, 12(8), 2019.

[91] Marwa Keshk, Benjamin Turnbull, Nour Moustafa, Dinusha Vatsalan, Kim-Kwang Raymond Choo. A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks. *IEEE Transactions on Industrial Informatics*, 16(8):5110–5118, Aug. 2020.

[92] Mahdi Daghmehchi Firoozjaei, Ali Ghorbani, Hyoungshick Kim, JaeSeung Song. Hy-bridge: A hybrid blockchain for privacy-preserving and trustful energy transactions in internet-of-things platforms. *Sensors*, 20(3), 2020.

[93] Jiangshan Yu, David Kozhaya, Jeremie Decouchant, Paulo EstevesVerissimo. Repucoin: Your reputation is your power. *IEEE Transactions on Computers*, 68(8):1225–1237, 2019.

[94] Qianwei Zhuang, Yuan Liu, Lisi Chen, Zhengpeng Ai. Proof of reputation: A reputation-based consensus protocol for blockchain based systems. *IECC ' 19: 2019 International Electronics Communication Conference*, pages 131–138, July 2019.

[95] Fangyu. Gai, Baosheng. Wang, Wenping. Deng, Wei. Peng. Proof of reputation: A reputation-based consensus protocol for peer-to-peer network. *International Conference on Database Systems for Advanced Applications*, pages 666–681, 2018.

[96] Jie Duan, Mo-Yuen Chow. A resilient consensus-based distributed energy management algorithm against data integrity attacks. *IEEE Transactions on Smart Grid*, 10(5):4729–4740, 2019.

[97] Wenjun Cai, Wei Jiang, Ke Xie, Yan Zhu, Yingli Liu, Tao Shen. Dynamic reputation–based consensus mechanism: Real-time transactions for energy blockchain. *International Journal of Distributed Sensor Networks*, 16(3):1550147720907335, 2020.

[98] Dodo Khan, Low Tang Jung, Manzoor Ahmed Hashmani. Proof-of-review: A review based consensus protocol for blockchain application. *International Journal of Advanced Computer Science and Applications(IJACSA)*, 12(3), 2021.

[99] Jose E. Fadul, Kenneth M. Hopkinson, Todd R. Andel, Christopher A. Sheffield. A trust-management toolkit for smart-grid protection systems. *IEEE Transactions on Power Delivery*, 29(4):1768–1779, 2014.

[100] Zheyuan Cheng, Mo-Yuen Chow. Reputation-based collaborative distributed energy management system framework for cyber-physical microgrids: Resilience against profit-driven attacks. *2020 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*, pages 1–5, 2020.

[101] Tonghe Wang, Jian Guo, Songpu Ai, Junwei Cao. Rbt: A distributed reputation system for blockchain-based peer-to-peer energy trading with fairness consideration. *Applied Energy*, 295:117056, 2021.

[102] Logan O.Mailloux, Michael R.Grimaila, John M.Colombi, Douglas D.Hodson, Gerald Baumgartner. Emerging trends in ict security. *ScienceDirect*, pages 5–23, 2014.

[103] Double-spending. *Wikipedia*, Accessed: November 2022, https://en.wikipedia.org/wiki/Double-spending.

[104] Tobias Bamert, Christian Decker, Lennart Elsen, Roger Wattenhofer, Samuel Welten. Have a snack, pay with bitcoins. *IEEE P2P 2013 Proceedings*, pages 1–5, 2013.

[105] Rosenfeld, Meni. Analysis of hashrate-based double spending. *arXiv eprints*, page arXiv:1402.2009, Feberary 2014.

[106] Matthias Grundmann, Till Neudecker, Hannes Hartenstein. Exploiting transaction accumulation and double spends for topology inference in bitcoin. *Financial Cryptography and Data Security*, pages 113–126, 2019.

[107] John P. Podolanko, Jiang Ming. Countering double-spend attacks on bitcoin fast-pay transactions. 2017. https://www.ieeesecurity.org/TC/SPW2017/ConPro/papers/podolanko-conpro17.pdf.

[108] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Capkun. Misbehavior in bitcoin: A study of double-spending and accountability. *ACM Transactions on Information and System Security (TISSEC)*, 18, June 2015.

[109] Ghassan O. Karame, Elli Androulaki, Srdjan Capkun. Double-spending attacks on fast payments in bitcoin. *ACM Conference on Computer and Communications Security (CCS'12)*, page 906–917, 2012.

[110] Micha Ober, Stefan Katzenbeisser, Kay Hamacher. Structure and anonymity of the bitcoin transaction graph. *Future Internet*, 5:237–250, 2013.

[111] Cristina Perez-Sola, Sergi Delgado-Segura, Guillermo Navarro-Arribas, Jordi Herrera-Joancomarti. Double-spending prevention for bitcoin zero-confirmation transactions. *Cryptology ePrint Archive*, 2017, https://eprint.iacr.org/2017/394.

[112] Ivan Osipkov, Eugene Y. Vasserman, Nicholas Hopper, Yongdae Kim. Combating double-spending using cooperative p2p systems. *27th International Conference on Distributed Computing Systems (ICDCS '07)*, pages 41–41, 2007.

[113] George Xylomenos, Christopher N. Ververidis, Vasilios A. Siris, Nikos Fotiou, Christos Tsilopoulos, Xenofon Vasilakos, Konstantinos V. Katsaros, and George C. Polyzos. A survey of information-centric networking research. *IEEE Communications Surveys and Tutorials*, July 2013.

[114] Konstantinos V. Katsaros, Wei Koong Chai, Ning Wang, George Pavlou, Herman Bontius, Mario Paolone. Information-centric networking for machine-to-machine data delivery: A case study in smart grid applications. *IEEE Network*, 28(3):58–64, Jun. 2014.

[115] Muhammad Fateh Khan Sial. Undecided. *EEE Blockchain Technical Briefs*, (Blockchain Technology – Prospects, Challenges and Opportunities), June 2019.

[116] R. Tourani, S. Misra, T. Mick, and G. Panwar. Security, privacy, and access control in information-centric networking: A survey. *EEE Communications Surveys and Tutorials*, 20(1):566–600, 2018.

[117] N Anita., M Vijayalakshmi. Blockchain security attack: A brief survey. *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, pages 1–6, 2019.

[118] Yunhua He, Hong Li, Xiuzhen Cheng, Yan Liu, Chao Yang, Limin Sun. A blockchain based truthful incentive mechanism for distributed p2p applications. *IEEE Access*, 6:27324–27335, 2018.

[119] Blockchain consensus encyclopedia infographic. https://tokenseconomy.gitbook.io/consensus/chain-based-proof-of-capacityspace/proof-of-reputation-por.

[120] Paul Resnick, Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of eBay' s reputation system. *The Economics of the Internet and E-commerce (Advances in Applied Microeconomics*, 11:127–157, 2002.

[121] Emanuele Bellini, Youssef Iraqi, Ernesto Damiani. Blockchain-based distributed trust and reputation management systems: A survey. *IEEE Access*, 8:21127–21151, 2020.

[122] Ferry Hendrikx, Kris Bubendorfer, Ryan Chard. Reputation systems: A survey and taxonomy. *Journal of Parallel and Distributed Computing*, 75:184–197, 2015.

[123] J. Göbel, A.E. Krzesinski. *Increased block size and Bitcoin blockchain dynamics*. 2017.

[124] Chrysanthos Dellarocas. Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems. *ICIS: International Conference on Computers and Information Systems*, pages 520–525, 2000.

[125] Arthur Gervais, Hubert Ritzdorf, Ghassan O. Karame, Srdjan Capkun. Tampering with the delivery of blocks and transactions in bitcoin. *in Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS'15)*, page 692–705, October 2015.

[126] Samiran Bag, Sushmita Ruj, Kouichi Sakurai. Bitcoin block withholding attack: Analysis and mitigation. *IEEE Transactions on Information Forensics and Security*, 12(8):1967–1978, 2017.

[127] Sang, L., Hexmoor, H. (2022). Reputation-Based Consensus for Blockchain Technology in Smart Grid. International Journal of Network Security & Its Applications (IJNSA), Vol. 14, No. 4, July 2022.

VITA

Graduate School
Southern Illinois University Carbondale

Lanqin Sang

lanqinsang@msn.com

National University of Defense Technology
Bachelor of Engineering, Electronic Communications, 1986

Southern Illinois University Carbondale
Master of Science, Rehabilitation, 1998

Southern Illinois University Carbondale
Master of Science, Computer Science, 2000

Dissertation Paper Title:
Security research for blockchain in smart grid

Major Professor: Dr. Henry Hexmoor

Publications:

Journal papers

1. Sang, L., Hexmoor, H. (2021). Information-Centric Blockchain Technology for the Smart Grid. International Journal of Network Security & Its Applications (IJNSA), Vol.13, No.3, May 2021.

2. Sang, L., Hexmoor, H. (2022). Reputation-Based Consensus for Blockchain Technology in Smart Grid. International Journal of Network Security & Its Applications (IJNSA), Vol.14, No.4, July 2022.

3. Sang, L., Hexmoor, H. (2023). A Countermeasure for Double-Spending on Blockchain Technology in Smart Grid. International Journal of Network Security & Its Applications (IJNSA), Vol.15, No.2, March 2023.