INFORMATION-CENTRIC BLOCKCHAIN TECHNOLOGY FOR THE SMART GRID

Lanqin Sang and Henry Hexmoor

Department of Computer Engineering, Southern Illinois University Carbondale, Carbondale, Illinois, USA

ABSTRACT

This paper proposes an application of blockchain technology for securing the infrastructure of the modern power grid - an Information-Centric design for the blockchain network. In this design, all the transactions in the blockchain network are classified into different groups, and each group has a group number. A sender's identity is encrypted by the control centre's public key; energy data is encrypted by the subscriber's public key, and by a receiver's public key if this transaction is for a specific receiver; a valid signature is created via a group message and the group publisher's private key. Our implementation of the design demonstrated the proposal is applicable, publisher's identities are protected, data sources are hidden, data privacy is maintained, and data consistency is preserved.

KEYWORDS

Information-Centric, Blockchain, Smart Grid, Network Security, Distributed System.

1. INTRODUCTION

The power supply system is an integral part of the modern society, affecting every aspect of our daily lives. Traditionally, the power supply system is a centralized distribution network, where the power generators produce and deliver electricity to the consumers at the terminals of the power grid. Technological advancement has brought increasingly power-thirsty appliances, ranging from electric vehicles [1] to smart phones and wearable devices to our society. Recently, the development of renewable energy, such as solar and wind energy, makes it possible for consumers to produce energy for themselves as well as energy consumption (i.e., becoming prosumers). They may also redistribute excess power back onto the power grid, just like wind and solar farms, so that the distributors can re-route it to those who need it. However, doing so creates stress on the current power system, as the excess renewable energy will inevitably fluctuate from time to time, and the current power system was not designed to meet such challenges presented by the rapid technological evolution.

It has been proposed that the current centralized power system must be transformed into a distributed power system to allow the renewable energy producers of various sizes to connect to the power grid. To manage the distributed power system effectively and use the power more efficiently, it is necessary for the new power system to have two-way communication and power delivery capabilities. Such a power system is dubbed smart grid, which integrates the traditional power grid with nearly real-time communication system and intelligent control system [2].

According to [3], the future smart grid must incorporate centralized as well as distributed energy sources, balance energy supply and demand through energy sharing, and ensure flexible energy generation during marketing and consumption of energy. This will ensure that all participants and

components could independently interact with peers, exchange both energy and associated information in multiple ways, and access large scale different types of distributed energy resources efficiently. The future smart grid must integrate and coordinate numerous connections, such as growing distributed energy producers, their consumers, electric vehicles, smart devices, and cyber-physical systems. Managing such a complicated and ever-growing network will require sophisticated information and communication infrastructures. The future smart grid also faces security, privacy, and trust concerns and various innovative technologies that embody them [4–9]. For smart grids to meet such challenges, blockchain technology offers novel solutions and possibilities.

Blockchain technology is a secure, decentralized, trusted cyber-infrastructure solution for future energy systems [10]. Multiple entities in the network can create, maintain, and store a chain of blocks. Blockchain technology creates redundant systems, which are resilient to single-point failure, and cyber-attacks. Although bockchain heralds huge potential for a secure, distributed cyber infrastructure solution for future energy systems, it does have limitations and practical challenges, which include performance scalability, privacy [3], and redundancy [10]. Blockchain's voting and consensus mechanisms take time. As the number of entities increase, much longer time will be needed to reach consensus, which may render it impractical because real-time smart grid requires high throughput and low latency. Since the blocks are transparent to all the nodes in the network, user identity and behaviour can be inferred from the information collected from the users, threatening user privacy [11] [12]. Adversaries can use this information to attack the users, such as stealing their power or changing their smart meters. [13] In a blockchain, individual nodes must save every transaction of the network. Thus, blockchain creates multiple data copies, which takes extra storage space and consumes more power [10].

In this paper, we propose a blockchain infrastructure to preserve privacy and hiding data sources, increase network throughput, and reduce information storage redundancy. The structure of the paper is: First, we introduce the related privacy-preserving techniques. Second, we outline the fundamental principles of our architecture. Third, we offer details of the proposed design. Fourth, we conduct system analysis, including security analysis, scalability analysis, and redundancy analysis. Fifth, we describe our design implementation, and finally, we conclude our paper.

This document describes, and is written to conform to, author guidelines for the journals of AIRCC series. It is prepared in Microsoft Word as a .doc document. Although other means of preparation are acceptable, final, camera-ready versions must conform to this layout. Microsoft Word terminology is used where appropriate in this document. Although formatting instructions may often appear daunting, the simplest approach is to use this template and insert headings and text into it as appropriate.

2. RELATED RESEARCH

A customer's privacy includes user identity and user data. The proposed techniques to address the problem of preserving privacy can be classified into two categories, protecting user identity and data.

2.1. Protecting user identity

Preserving identity with pseudonyms. Guan et al. [14] used pseudonyms [15] to hide user identities based on blockchain. Each user is allowed to create multiple pseudonyms and associate his or her data with different pseudonyms. The bloom filter has been adopted to validate the pseudonyms. However, an attacker can use cluster analysis and time analysis to estimate the

relationship between the input and output addresses, mapping pseudonyms to uncover the user's real identity.

Preserving privacy by data aggregation [14]. Guan's paper has divided users into different groups according to their electricity consumption types. In each time-period, a node is chosen as the mining node according to its group's average electricity consumption data. The mining node is responsible for aggregating the data and recording these data into the blockchain. It is not entirely clear in Guan's paper how electricity consumption types are determined. When data are aggregated, not only individual user's data are hard to separate, but also the user's privacy is preserved because the final data is an aggregate from many users.

Preserving identity with anonymity. In our design, the sender's identity is encrypted with the control center's public key. Only the control center will know who has sent the transactions.

2.2. Protecting user data

Preserving data 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 encrypting the message.

Preserving data with data obfuscation. Data obfuscation adds noise into the user data to obfuscate (i.e., hide) the original data. The noise can be random [16] or user-specific [17]. The noise will be managed by the control center so it can be cancelled out properly later.

Preserving data with group signature [18]. 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 the user's 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 informal groups, not user groups.

3. PRELIMINARIES

3.1. Blockchain Technology

Blockchain technology is a distributed ledger framework defined as a system to produce a consensus of replicated, shared, and synchronized digital data. The data are shared across the network and can be accessed by each node of the network. As a chain of blocks, blockchain maintains a collection of blocks that registers different records of data or transactions. These blocks are latched together, and each block references the cryptographic hash of the previous block's data. Newly generated blocks are continuously appended to the chain at regular intervals and this chain is replicated among the members of the network. Each block may also include time stamp, nonce, a hash tree named Merkle tree [19], smart contract scripts [20], node state, and so on. The hash and Merkle tree allow verification that the content inside the block is not modified, i.e., ensuring integrity. The chain structure makes it very hard to modify any data block because in order to alter any block's content, it is required to change all the blocks. Since the hash of a

block becomes different almost surely if any of its content changes, and each block has the previous block's hash, it is practically impossible to modify the chain maliciously, therefore, ensuring data integrity. Also due to the replication of the chain of blocks, the single point failure is avoided, and the data availability is ensured. The hash, replication, encryption, and signatures are the key mechanisms for blockchain security.

3.2. Verifiable Random Function

A Verifiable Random Function (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 [21]. The VRF algorithm is:

1. A VRF's key generation algorithm generates a pair of public-private keys *(pk, sk)*
2. A VRF hashes an input message using the private VRF key *sk* to obtain a VRF hash output hash:
   $hash = VRF\_hash(sk, message)$ (1)
3. The VRF hash algorithm uses the private *sk* to construct a proof that the hash is the correct hash output:
   $proof = VRF\_prove(sk, message)$ (2)
4. The prover sends out the *(hash, proof)* to the verifier.
5. The verifier gets *hash, proof* and calculates the hash directly from the proof as:
   $hash = VRF\_proof\ 2hash(proof)$ (3)
6. The verifier will determine if there is a unique correspondence between the message and the hash:
   $True/False = VRF\_verify(pk, message, proof)$ (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, the VRF is often used in blockchain consensus mechanism to select the master node and transactions verification [18].

3.3. Consensus Mechanism

The Practical Byzantine Fault Tolerance (PBFT) is used in our consensus mechanism. PBFT is an algorithm that optimizes aspects of Byzantine Fault Tolerance. Byzantine Fault Tolerance is the ability of a distributed computer network to correctly reach a sufficient consensus despite malicious nodes in the system failing or sending out incorrect information. For the PBFT system to function, at least two-thirds of the total number of nodes must be honest.

3.4. Smart Contract

The smart contract is a computerized script stored and deployed in the blockchain. A smart contract records specific conditions and actions. When the conditions are met, the corresponding actions will be taken automatically and the results of executing the smart contract will be recorded in the blockchain. There are three types of smart contracts in our design: transaction creation, consensus mechanism, and block receiving.

3.5. 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 (INC) [22]. INC has placed data at the center of the networking landscape where information is published, resolved, delivered, and stored [23]. INC 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 [21]. PMUs are used for recording synchronized measurements across a wide area. This is a part of the effort to develop a new communications network in order to provide real-time monitoring of the PMUs. When there are many parties exchanging and sharing information in the blockchain environment, the INC provides more flexibility than traditional host-centric solutions. In our paper, we apply this INC concept and divide the transactions into different information groups to hide data sources and preserve user privacy.

3.6. Group Signature

The group signature [18] is used to protect privacy and traceability. In the group signature scheme, each member of the group can send a transaction on behalf the group without disclosing his own identities. The nodes with the group public key can verify whether the signature is from someone in the group and decide if the received transaction is valid.

3.7. Information Classification

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

- Energy consumption. This information will be created by smart meters.
- Energy generation. This information will be produced by PMUs.
- Energy storage. This information will be sent out by IoT.
- Energy trading is buying and dispatching. This information will be put into computers by human.
- Control. This information will be sent out by the control center.
- 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 different 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, the publishers which generate the transactions, such as the smart meters, PMU (Phasor Measurement Unit), and computers for information input in the data input layer; the relays that consent to verify, store and broadcast the transactions, such as the nodes in the blockchain layer; the information consumers, also called sub- scribers, such as the billing center, control centers in the receiver layer, and the users who receive other's data, such as available energy data and energy trading related information.

4. PRIVACY-PRESERVING INFORMATION-CENTRIC SCHEME

4.1. System Model

There are four layers in our design - information input, blockchain for data collection, verification, and storage, communication, receiving layer management, with the Public Key Infrastructure (PKI) spanning across all four layers (Figure 1). In Figure 1, coloured circles in the blockchain layer represent different information groups.

Information 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 broadcast to the entire network.

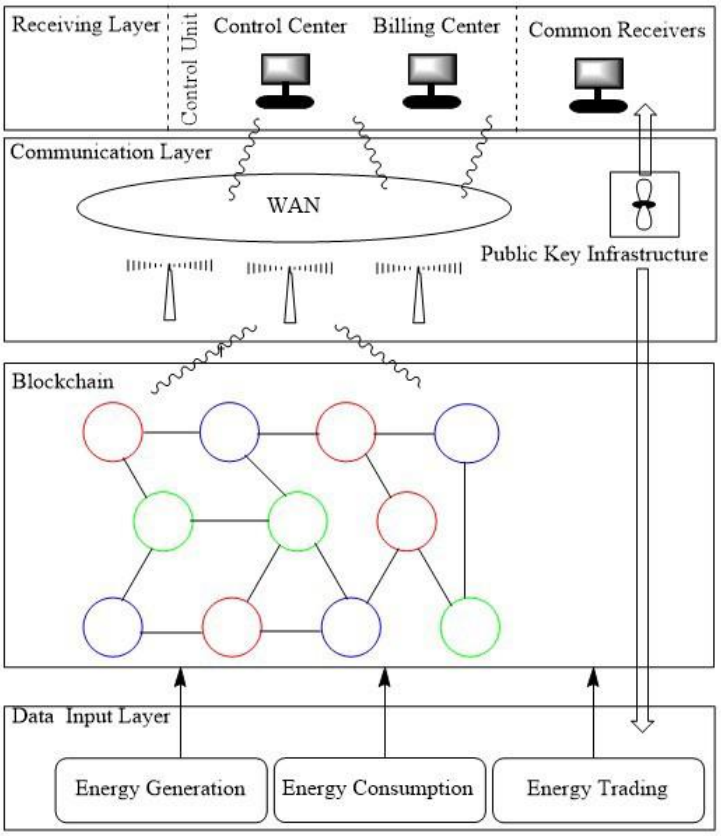


Figure 1. System Model

Blockchain layer. This layer is for transaction auditing, validating, and blockchain storage. A valid block will be broadcast to the network for subscribers/receiver to process.

Communication layer. In this layer, all needed information passes through the entire network.

Information 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.

Public Key Infrastructure creates public-private keys for all the participants, including each information group, which will get a group message and two pairs of public-private keys PUgp-PRgp, one for the publishers and one for the subscribers PUgs-PRgs. The PKI layer covers all the four layers in our system.

4.2. System Initialization

Key Generation and Group Membership

All the participants in the system need to acquire public-private key pairs. For receivers, we use PUr-PRr.

All publishers will get the group message, the group publisher's private key, PRgp, receiver's public key PUr, and subscriber's public key, PUgs.

All receivers will get the group message and the group publisher's public key, PUgp, and the group subscriber's private key PRgs.

Control Center's public key PUc will be given to all publishers and the publisher will encrypt its public key with the control center's public key. The control center can decrypt the user's public key if needed. This way we can maintain the privacy of individual sender's identity (public key PUs).

4.3. System Smart Contract

Transaction Creation

This smart contract will be installed in 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, 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 the power consumption data, available power, appliance states, or energy order information. The sender's public key PUs are encrypted by the control center's public key PUc; the transaction signature created by the group message and the group publisher's private key PRgp. After the transaction is created, it will be broadcast to the network. See Figure 2.

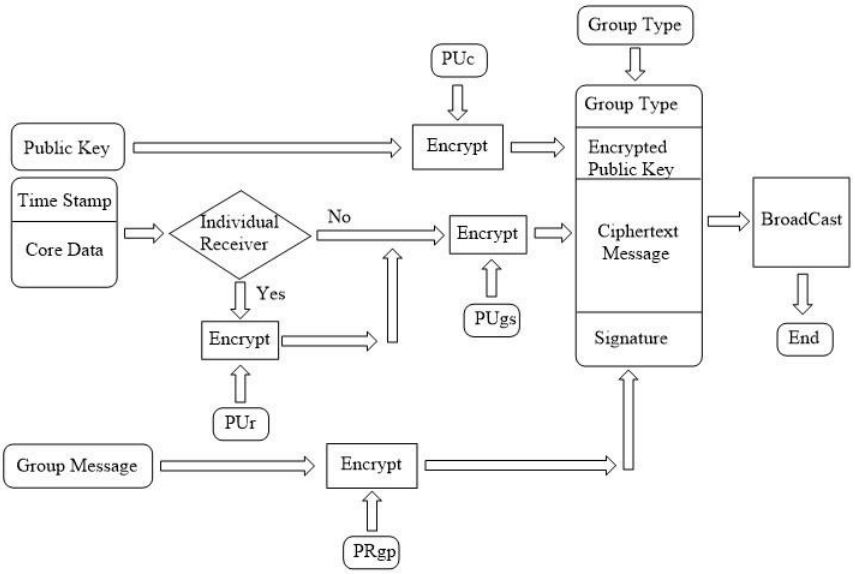


Figure 2. Transaction Creation

Consensus Mechanism

The consensus smart contract will be installed in 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 3.

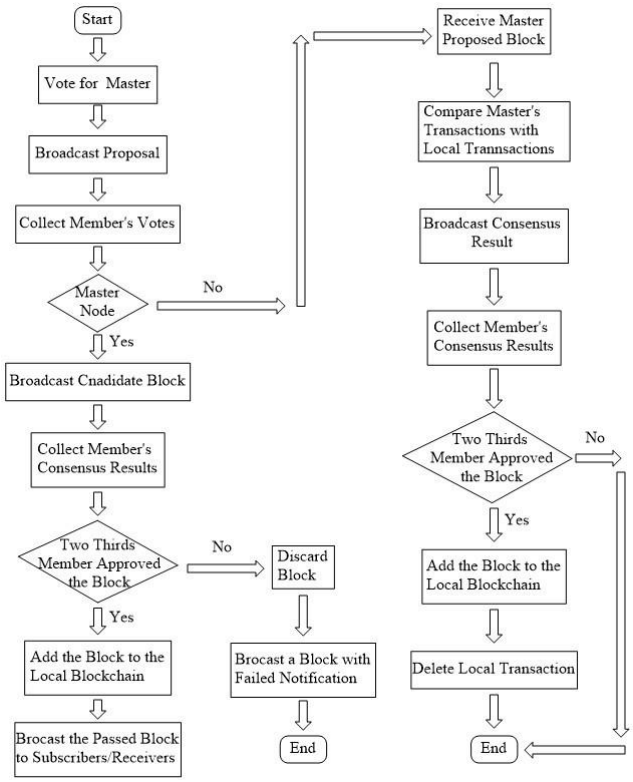


Figure 3. Consensus Mechanism

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, for example at 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 during 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 broadcast.
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.

Transaction Receiving

When the subscribers/receivers receive a block, it will go through the block. If a transaction is not in the right information group or a valid publisher, the transaction will be discarded. Otherwise, the transaction will be first decrypted by 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 4.

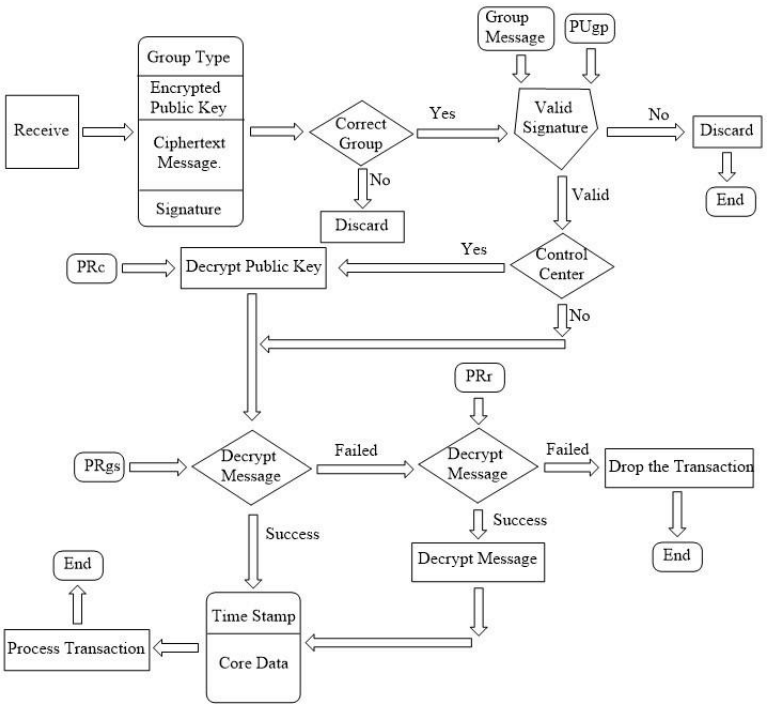


Figure 4. Subscriber Reception

4.4. System Analysis

4.4.1. Security Analysis

The three basic security requirements are: Confidentiality, Integrity and Availability. Confidentiality means only authorized entities can access the information; Integrity means the information is identical in its source and destination; Availability means the network resources are available to legal users or devices. 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, the information group's nodes can verify 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 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, this avoids single point failure and therefore ensures resource availability. Below are the common security attack analyses:

Denial of Services (DoS) Attack [24]. DoS attacks aim to overwhelm the network services by inundating them with requests. In blockchain, all transactions are saved first, then processed by its information group nodes in a fixed interval, therefore the service request rate is limited by its processing time interval [25]. Also, almost all the information passed in the energy network follows the predefined frequency. Any abnormal transaction frequency will alert the system.

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. Content Poisoning Attack. If any node's information has been changed, it will be discovered by other nodes during the consensus process.

Change the Data in Data Input Layer. Since almost all the transactions, such as energy consumption, energy availability, are created by smart con- tracts automatically, no human interference is needed. Also, smart meters and PMUs usually are secured both physically and through software, it is hard 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.

Content Poisoning Attack. If any node's information has been changed, it will be discovered by other nodes during the consensus process.

Change the Data 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. Also, smart meters and PMUs usually are secured both physically and through software, it is hard 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.

Forged transaction. To forge a transaction, the 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 these the keys and message are saved in the same place and are stolen, the chance to get all of them is extremely low.

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

group, 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 for the same time period will alert the system.

4.4.2. Privacy

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.

Data privacy. All the publishers' 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 receiver or subscribers will be able to decrypt the data with proper private keys.

Information origin privacy. Each transaction's signature will be created by group message and the publisher's private key. The consensus nodes and the receiver 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 the same group's publishers use the same group message and the publisher's private key. This bundling together method hides where a transaction comes from.

4.4.3. Scalability Analysis

Each blockchain's consensus needs to take some time to ensure that all the involved nodes get other nodes' responses and finish the consensus process. This adaption negatively impacts the network throughput. Because our system classifies the blockchain's transactions into different categories and these different information groups execute their consensus in a simultaneously, 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 spent in each group's consensus will be reduced also, which enhances the network throughput.

4.4.4. 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 the 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.

4.4.5. 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 searching will be fast because the information is saved separately according to the types, and the search only needs to go through the specific blockchain.

5. IMPLEMENTATION

5.1. Transaction creation contract

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 the 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 (Figure 5).

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

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

5.2. Consensus Contract

The consensus nodes are constantly checking. 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; or the transaction will be discarded. If it is consensus time, the nodes start to elect a leader and broadcast their proposals. Each node calculates its ballots. 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,

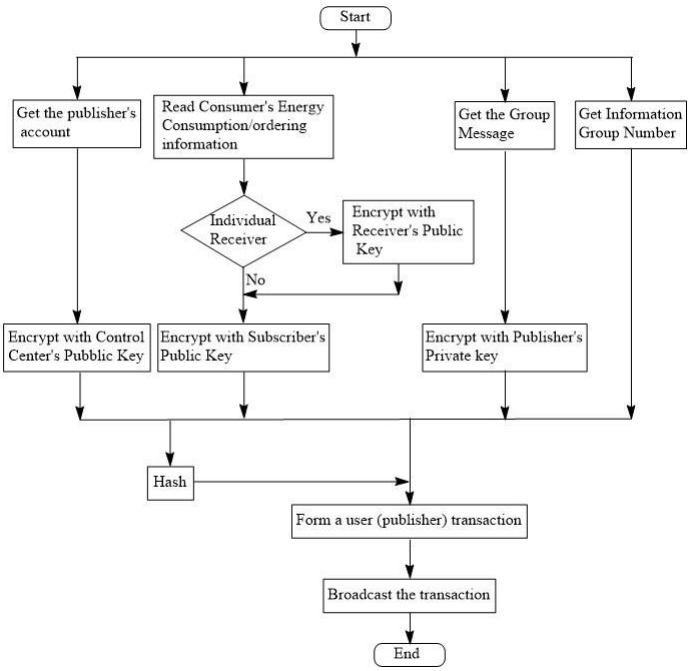


Figure 5. Transaction Creation Flow Chart

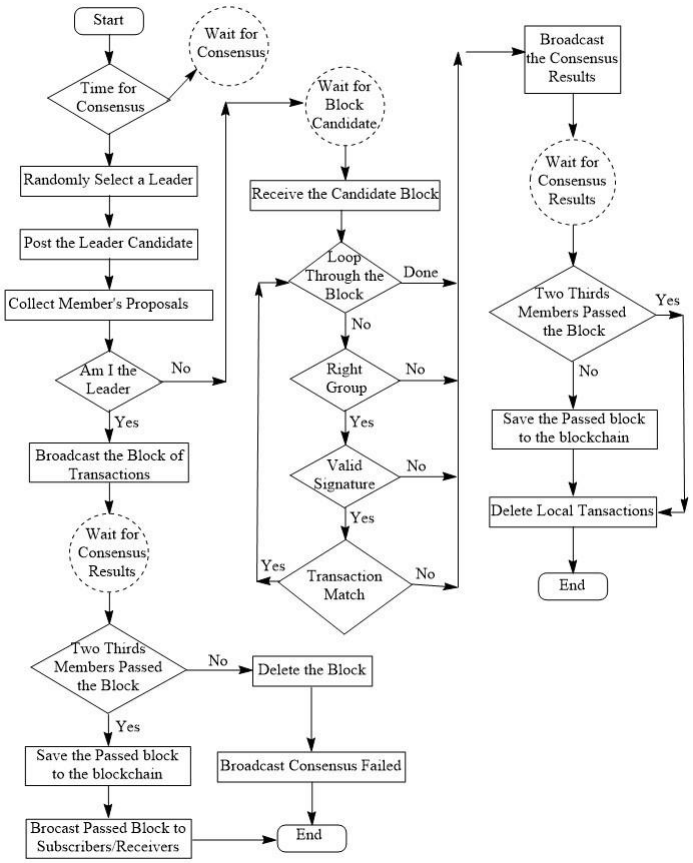


Figure 6. Consensus Flow Chart

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 its 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 (Figure 6).

5.3. Receive Contract

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 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 to decrypt this data. If the second decryption is successful, this transaction will be processed according to its group. If both decryptions failed, this transaction will be discarded. If the subscriber is the controller, it can decrypt the transaction sender's public key also and will know who sends this transaction (Figure 7).

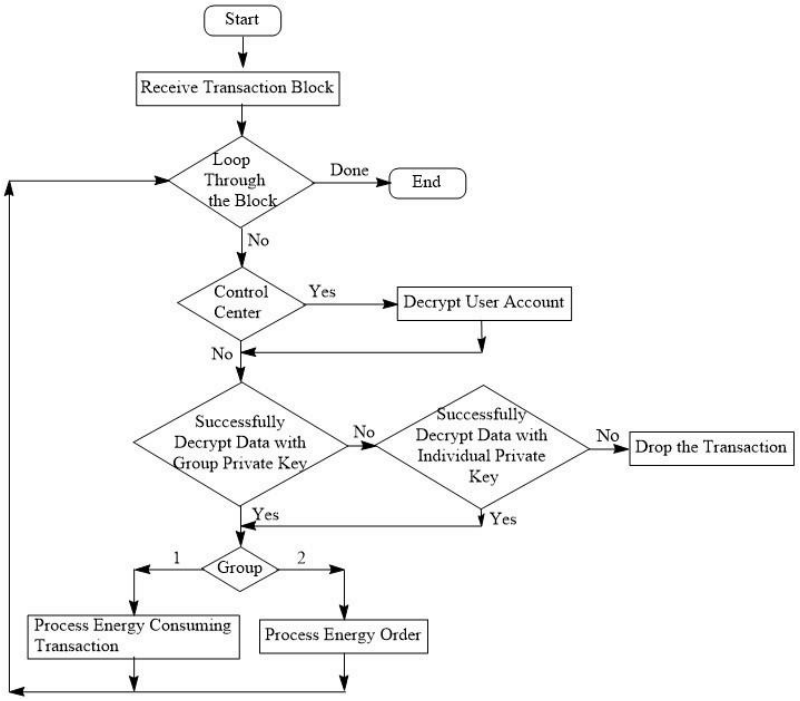


Figure 7. Receive Flow Chart

5.4. Implementation Analysis

As we have predicted in Section 4.4, our implementation has demonstrated that our proposed design is feasible, proved that our design can protect the sender's identity, camouflage transaction sources, protect data privacy, and preserve data consistency.

6. CONCLUSION

In this paper, we have proposed an information-centric blockchain technology for the smart grid to enhance security. By implementing the whole process, we can see this design is applicable and efficient to enhance blockchain security. However, our implementation is a simplified model with little data to test. A comprehensive implementation needs to be carried out, and thorough testing needs to be performed.

REFERENCES

[1] Mina Bergeroy Ryssdal. Blockchain technology implementation for electric vehicle charging within the smart grid architecture model, 2019.
[2] Bing Wang, Weiyang Liu, Min Wang, Wangping Shen. Research on bidding mechanism for power grid with electric vehicles based on smart contract technology. Energies, 13(390), 2020.
[3] 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.
[4] Sarmistha De Dutta, Ramjee Prasad. Security for smart grid in 5g and be- yond networks. Wireless Personal Communications, 106(1):261–273, 2019.
[5] Amrita Ghosal, Mauro Conti. Key management systems for smart grid advanced metering infrastructure: A survey. IEEE Communications Surveys Tutorials, 21(3):2831–2848, 2019.
[6] Muhammed Zekeriya Gunduza, Resul Das. Cyber-security on smart grid: Threats and potential solutions. Computer Networks, page 107094, 2020.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

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

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

[14] 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.

[15] Xiaobin Tan, Jiangsu Zheng, Cliff Zou, Yukun Niu. Pseudonym-based privacy-preserving scheme for data collection in smart grid. International Journal of Ad Hoc and Ubiquitous Computing, 22(2):120–127, Feb. 2016.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

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

[22] 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, Jul. 2013.

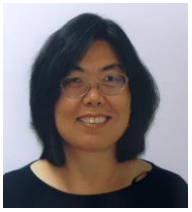
[23] Konstantinos V. Katsaros, Wei Koong Chai, Ning Wang, George Pavlou, Herman Bontius, Mario Paolone. Information-centric networking for machine-to-machine data delivery: A case study in smart grid applications. IEEE Network, 28(3):58–64, Jun. 2014.

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

[25] 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.

AUTHORS

Henry Hexmoor received the M.S. degree from Georgia Tech, Atlanta, and the Ph.D. in computer science from the State University of New York, Buffalo, in 1996. He has been an IEEE senior member since 2002. He has taught at the University of North Dakota before a stint at the University of Arkansas. Currently, he is a professor at the School of Computing, Southern Illinois University, Carbondale, IL. He has published widely in artificial intelligence and multiagent systems. His research interests include multiagent systems, artificial intelligence, cognitive science, mobile robotics, complex networks, digital forensics, and blockchain. His work has sought to take a principled approach to the development of use-inspired intelligent systems.



Lanqin Sang obtained her MS degree in computer science from Southern Illinois University Carbondale, where she is currently a Ph.D. student in computer science working with Professor Henry Hexmoor on computer security.