Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

An Ethereum-based solution for energy trading in smart grids

Francesco Buccafurri [*], Gianluca Lax, Lorenzo Musarella, Antonia Russo

Mediterranea University of Reggio Calabria, DIIES, Reggio Calabria, Italy

A R T I C L E I N F O

Keywords:
Blockchain
Smart contract
Accountability
Smart energy
Smart grids

A B S T R A C T

The need for a flexible, dynamic, and decentralized energy market has rapidly grown in recent years. As a matter of fact, Industry 4.0 and Smart Grids are pursuing a path of automation of operations to insure all the steps among consumers and producers are getting closer. This leads towards solutions that exploit the paradigm of public blockchain, which represents the best platform to design *flat* and *liquid* markets for which providing trust and accountability to mutual interactions becomes crucial. On the other hand, one of the risks arising in this situation is that personal information is exposed to the network, with intolerable threats to privacy. In this paper, we propose a solution for energy trading, based on the blockchain Ethereum and Smart Contracts.The solution aims to be a concrete proposal to satisfy the needs of energy trading in smart grids, including the important feature that no information about the identity of the peers of the network is disclosed in advance.

1. Introduction

Due to continued growth in energy demand, how to increase its production, on the one hand, and how to limit environmental pollution, on the other hand, are becoming global challenges.

Of course, it is necessary yet not sufficient to extend the usage of renewable energy. Only in 2018, the usage of renewable energy rose by 4%, accounting for almost one-quarter of global energy demand growth [1].

Moreover, there is a need to improve and update the classic electric grid infrastructure, to make it more efficient, flexible, and dynamic. In recent years the new concept of Smart Grids (SGs) is emerging. A smart grid can be considered as the evolution of classic grids with the main target to be eco-friendly, faster and more innovative. SGs are born also to improve the overall reliability of the whole energy cycle and to guarantee a better ratio demand/response to arise the interest of the financial field as well by applying a new energy market pattern [2]. Moreover, by increasing the energy demand and the number of entities involved in the energy market, SGs have to face the problem of guaranteeing a certain level of data and message availability in transmissions among peers of the network [3].

Energy systems in SGs are taking the direction of decentralized architectures in which a device, known as smart meter, can manage requests and responses through the whole network. Since it would be not appropriate to implement centralized protocols over SGs, it is fundamental to accommodate this decentralized and distributed direction by using technologies that are decentralized and distributed as well. In this way, blockchain technology appears to be the best solution, because of its proven properties, such as immutability, transparency and decentralization [4]. Indeed, thanks to the evolution of the blockchain paradigm originally produced with Bitcoin blockchain [5] (mainly devoted to the cryptocurrency Bitcoin), blockchains that support Smart contracts (SCs) like Ethereum [6] can be viewed as platforms for secure, interoperable, and decentralized applications, in which conflicting parties may enter into agreements and exchange value with no need to build trust between each other. Energy trading in SGs perfectly fits with these features. Therefore, one of the interesting research directions is to explore how to fully exploit the power of blockchain and SCs to envisage innovative applications and to increase the effectiveness of the notion of smart grid. In addition, the usage of blockchain may introduce flexibility among operations carried out by stakeholders inside the energy trading market. In particular, if we check these features with the energy industry we can deduce that the sector that can benefit most from them is energy trading among applicants and bidders. Indeed, a blockchain-based solution for energy trading is able to improve accountability, reliability, fairness and to reduce time and costs.

However, there are still open challenges and limitations in the implementation of blockchain-based applications to energy market trading, such as the scalability, security and efficiency [7].

This paper is just placed in this research track, by proposing an Ethereum-based solution for energy trading and aiming also to enhance scalability compared with other approaches presented in other research

[*] Corresponding author.
E-mail addresses: bucca@unirc.it (F. Buccafurri), lax@unirc.it (G. Lax), lorenzo.musarella@unirc.it (L. Musarella), antonia.russo@unirc.it (A. Russo).

https://doi.org/10.1016/j.dcan.2021.12.004
Received 5 December 2020; Received in revised form 28 November 2021; Accepted 23 December 2021
Available online 30 December 2021

literature. The blockchain enables parties to transfer assets without the participation of a trusted third-party and all the transactions are stored and validated by the network, with no centralized unit control. To the best of our knowledge, our approach presents an innovative aspect compared with existing related solutions. Indeed, the power of SCs is also exploited to manage the offers in a blind fashion, so we can actually talk about an auction, in which identities are disclosed only when the agreement is established. Interestingly, the entire auction is managed with no intervention from any external referee.

The paper is structured as follows. In Section 2 we introduce SGs, energy trading, and the Ethereum blockchain. In Section 3, we present the context in which we apply our solution and we give some motivations that draw us to face this problem. In Section 4, we present the model of our solution. Section 5 describes the details of the Ethereum SC that implements our solution. In Section 6, we discuss the most important security aspects and properties of our solution. The related work is discussed in Section 7. Finally, in Section 8, we draw our conclusions.

2. Background

In this section, we provide the technical background that makes the paper self-contained, and gives the basis underlying the motivations of our proposal. We start by describing the notion of smart grid, which is the application domain focused by our paper.

SGs are designed be mainly used in renewable energies. The direction taken by SGs is switching from a centralized to a distributed energy market model, in which customers have more decision-making power, according to their role of producers or suppliers of energy as well. Therefore, there is a new figure of the energy user in SGs: the prosumer, who acts, in the smart grid environment, as both the consumer and the producer of energy. Indeed, the smart grid protocol is quite close to a Peer-to-Peer (*P*2*P*) solution in which there is not anymore a hierarchical relationship among nodes.

The smart meters play a fundamental role in the whole communication process in SGs. Indeed, a smart meter is a hardware component that can run software that makes it capable to manage (also by sending) electricity generated by a prosumer and to respond to external requests [8].

Energy trading is one of the most important components in the SGs' energy management as well as in the more classical energy market, in which it represents the last phase of the cycle. In particular, a very high-level description of how the energy market works is the following:

1. Energy is produced by generators;
2. Energy is transmitted to the distribution network;
3. Now, retailers are in charge of connecting the distributors to consumers by buying and selling energy;
4. Consumers can obtain the energy needed by paying for it to retailers.

Currently, in classical electric grids these steps are almost detached from each other, resulting in disadvantages to the efficiency and effectiveness of this solution. SGs aim to make these steps closer to each other, to improve the overall system efficiency, as well as to reduce costs and time.

If we just think of the new figure of the prosumer, it is clear that this cycle is inherently faster in a smart grid scenario compared with a classic grid, since there are not only central generators but also that energy can be created and transmitted to the distribution network via prosumers themselves too. Anyway, step 4 is quite a bottleneck in SGs because it still needs to follow some traditional criteria that are not fully compliant with the smart grid proposal so new approaches need to be developed.

The second macro-component is the Ethereum blockchain, which represents the key factor of our solution.

In last years, after the incredible and exponential success of Bitcoin [5] and others cryptocurrencies blockchains, the second era of blockchain, known as Blockchain 2.0, is emerging.

Ethereum is the progenitor and the most relevant technology of this new generation. It is not a blockchain for cryptocurrencies anymore but a platform that can be defined as a programmable public and permissionless blockchain, whose main function is to provide an alternative protocol for building, in a fast, secure and interoperable way, Decentralized Applications (DApps) [9]. The Ethereum decentralized and distributed Virtual Machine (EVM) [10] can execute programs called smart contracts (SCs). They are written in the Turing-complete language Solidity [11] that implements the EVM bytecode. It is worth noting that the definition of a SC depicts it as a self-executing contract that has inside the terms of the agreement between two parties without the need for a central authority. The ether is the cryptocurrency generated by the Ethereum platform and used also to pay transaction fees. In Ethereum, differently from the Bitcoin blockchain [5], there are two different types of accounts: (1) Externally Owned Accounts (EOAs), and (2) Contract Accounts (CAs). The former are accessed by private keys and controlled by people who own these private keys, while the latter are controlled by the contract code. In detail, every account has a 20-byte address and has an ether-balance. Indeed ethers can be transferred among accounts. An EOA can send transactions that are stored inside the blockchain to create a SC or invoke a function inside it, or again, simply to transfer ether to another account. Instead, the CA can be activated only by an EOA. These families of accounts open new horizons regarding transactions. Indeed, in Ethereum, there are the so-called External Transactions (ETs) and the Internal Transactions (ITs) (known also as Contract Transactions). The former are generated by the users and they are publicly and transparently recorded inside the blockchain [9] while the latter are sent from a contract to other contracts and these are not recorded on the blockchain and do not affect the states of other accounts [9]. An Ethereum environment [12] is used to create and publish SCs and DApps.

Tokens represent probably the killer feature of the Ethereum environment. Indeed, they are virtual assets that can be created by every peer of the blockchain [13]. There is no a well-defined associated economic value, because it depends on the context in which tokens are used. We can distinguish two main families of Ethereum tokens: fungibles and non-fungibles. The former are defined as tokens that are fully interchangeable (i.e., all tokens are alike) and they could be used, among others, as sub-cryptocurrency for payments, while the latter are tokens that have an identifier or a label, so that they are mostly used as virtual collectables [13]. The community has developed some standards to facilitate the creation and the exchanging of tokens that can be seen as interfaces to be implemented in such a way users can use them. In particular, the standard Ethereum Request for Comments, number 20 (ERC-20) is one of the most popular fungible tokens. It is composed of six mandatory functions to be implemented, plus three optional ones. However, since ERC-20 has some limitations in terms of costs and functions, the new ERC-223 standard has been recently proposed. It is fully backward compatible with ERC-20 and it solves the above limitations. On the other side, the most popular standard for non-fungible tokens is ERC-721, which has been recently improved by the new ERC-1155. These kinds of tokens are associated with some metadata, in which it is possible to save information that characterized uniquely the token itself.

Ethereum, as a blockchain, requires that transactions and blocks have to be validated by miners. In addition, every computational step carried out requires also some extra-charges to be paid by users. This kind of fuel is called gas, which is a unit of measuring the computational work of running transactions or SCs in the Ethereum network. In particular, gas is expressed through gwei, a subunit of ether.

Furthermore, every user can specify, through the field GAS_PRICE, how much she/he is willing to pay for each computational step. Obviously, the higher this field is, the earlier the transaction will be chosen by miners. Another interesting field is STARTGAS, representing the maximum number of computational steps the transaction execution is allowed to take and that helps to avoid loops.

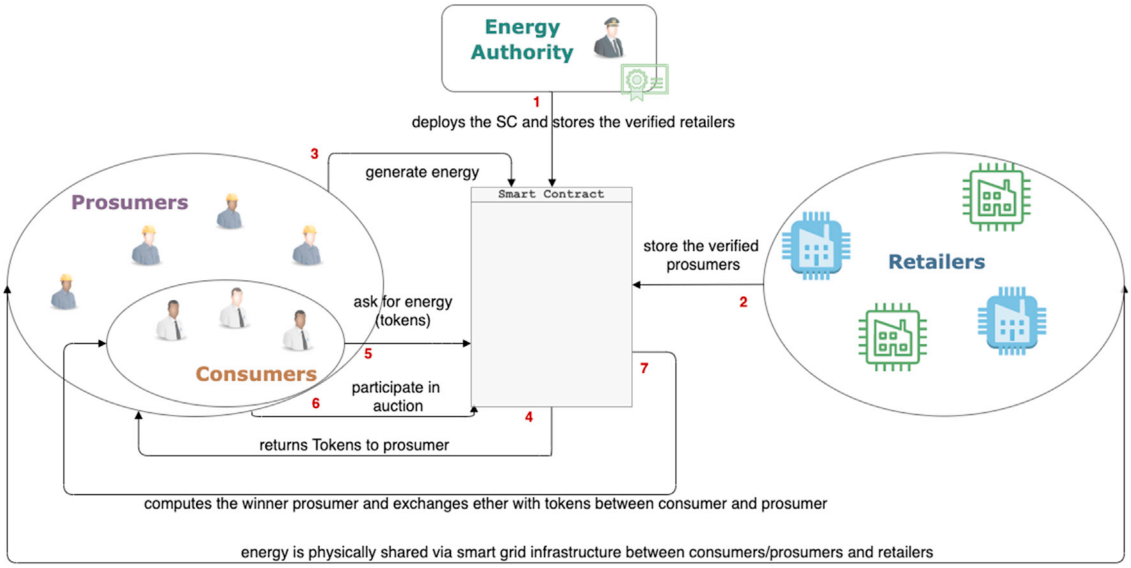
Energy Authority
1
deploys the SC and stores the verified retailers
3
generate energy
Smart Contract
Prosumers
Consumers
ask for energy (tokens)
5
participate in auction
6
4
returns Tokens to prosumer
store the verified prosumers
2
Retailers
7
computes the winner prosumer and exchanges ether with tokens between consumer and prosumer
energy is physically shared via smart grid infrastructure between consumers/prosumers and retailers

Fig. 1. The architecture of our solution.

3. Scenario and motivations

From the beginning, the electricity grid was conceived as a centralized system in which energy is produced in huge power plants. It is clear that this kind of system has limits in reliability, availability, and, as a consequence, in business terms. Moreover, the growing world population generates a rising demand for energy and, due to the increasing level of pollution, the request for sustainable and renewable energy is necessary. Indeed, investing in renewable energy is becoming central in most of the world governments for environmental protection. Energy is a raw material and for this reason it can be exchanged; energy trading term means buying, selling, and moving energy from where it is produced to where it is needed. The concept of Energy Internet [14] stands for the open, collaborative, and interactive process of energy production and consumption. Through the years, the energy systems have been developed into four different stages: (*i*) decentralized energy systems, (*ii*) centralized energy systems, (*iii*) distributed energy systems, and (*iv*) smart and connected energy systems.

The decentralized approach can be used to evaluate energy exchange [15]. Indeed, the decreasing price of distributed energy resources in the ten past years allows known energy consumers become prosumers, that is they can both consume and generate energy. The consumers, instead, only purchase energy. There are several business initiatives to improve the energy use and consumption all over a (smart) grid. Many of these initiatives are characterized by similar actors and operations. Indeed, the actors in an energy trading scenario could be represented by

- *Consumer*, a physical person who needs to buy electricity.
- *Prosumer*, an entity that acts as an energy supplier (such as farmers with wind turbines or an individual who produces additional energy) and at the same time uses and buys electricity. In detail, we can consider a prosumer as a consumer with the ability to produce energy as well. So, every prosumer is a consumer while the contrary is not always true.
- *Retailer*, an entity that buys electricity from prosumers and sells it to customers (both prosumers and consumers). The retailer is also responsible for getting customers connected to the network and for customers' billing and service.

The operations carried out by the actors could be divided into three phases implemented through different approaches, as the study [16] suggests. The first step is to determine their (the actors) own amount of energy supply and demand in the network. This step requires adequate controls to ensure the privacy and security of the actors. The second step is to match consumers and prosumers. Specifically, the consumer chooses the most suitable prosumer able to fulfill the request. This phase may run many rounds through an auction process. The transaction settlement is the last phase, which is to establish the rules, among the parties, to guarantee the transfer of energy.

During the various processes in which the prosumers are currently involved performing an energy trading protocol, their identities may be disclosed, leading to some privacy problems. The aim of this paper is to provide a protocol that takes into account the security and privacy requirements in an energy trading scenario.

When a consumer demands for energy, an auction starts, the winner prosumer stipulates a contract with the consumer. During these phases, being aware of the actors' identity could cause a possible impairment. Furthermore, dynamicity is required because prosumers are not known first. For this reason, an important issue (addressed in this paper) is to implement a privacy-preserving approach in the auction phase.

Exploiting the blockchain technology in an energy trading scenario can include the well-known advantages of distributed ledger, such as the elimination of a central governing institution, a distributed consensus, and the immutability and accountability of transactions. Observe that several auctions can be executed at the same time by suitably designing SCs. The execution of more auctions does not bring more possibilities of malfunction in the consensus agreement thanks to the properties of the Ethereum blockchain that can prevent from latency and propagation problems by implementing a modified Greedy Heaviest Observed Subtree (GHOST) protocol. Furthermore, the blockchain protocol prevents from double-spending attempts by design.

At the same time, designing a fully automated smart grid can be advantageous and helpful in the cost reduction of transactions and electricity. Although the other proposed solution in the integration of blockchain for energy trading seems to solve the problem, there are still open issues such as the spreading of energy trading in a public blockchain, or the responsibility in the transactions derived from the anonymity ensured by blockchain. For these reasons, in this paper, we propose an approach that includes the management of the actors' identity to make transactions accountable.

This way, the final agreement will be achieved among nonanonymous entities.

4. Our solution

In this section, we describe our proposal. First, we present the involved entities and, then, we present the main steps of the entire process of energy trading.

Fig. 1 illustrates the overall architecture of our solution.

As described in Section 3, the actors we consider are consumers, prosumers, and retailers. We exploit the Ethereum blockchain to store the information in a distributed and immutable way and also to guarantee the security properties. Consequently, in our proposal, we include a new actor, the Energy Authority, which is the entity that deploys the SC needed to drive our solution.

In our solution, the following steps can be identified:

4.1. Setup

In the initialization phase, a suitable SC is deployed on Ethereum by the Energy Authority. It implements the functions that are described and used in the following steps. Moreover, both prosumers and retailers register an Ethereum address.

4.2. System registration

In this phase, the entities join the system. First, the owner of the SC identifies each retailer and verifies its Ethereum address by a challenge-response scheme. In particular, the retailer must sign a challenge sent by the owner by using the private key of its Ethereum account. For each verified retailer, the SC owner invokes a function of SC and gives the retailer's address A_R as an input parameter. This function adds A_R to the list of the verified retailers L_R managed by the SC. A verified retailer R can register one or more prosumers P in the system. This operation is done by calling another function of SC and giving the prosumer's address A_P as an input. Again, the retailer verifies the prosumer's address by a challenge-response procedure. The result of the function call is the inclusion of this address to the list of the verified prosumers L_P, which is also managed by the SC. These procedures are repeated every time the SC owner wants to add a new retailer or a retailer wants to add a new prosumer to the system. At the end of this phase, it is possible to verify whether an Ethereum address, that we call Main Ethereum Address (MEA), is associated with a verified retailer or prosumer.

4.3. Energy production

The actor involved in this step is the prosumer, who generates energy and trades it with the retailers. In particular, given the prosumer P_i, she/he can transfer a given amount of energy, say E, to the retailer R_j thanks to the smart grid infrastructure. Indeed, in the smart grid environment, there exists an IoT (Internet of Things) device, the smart meter, that is fundamental to link the consumer to the whole energy infrastructure. We propose an easy extension of such a smart meter that will include also the possibility of connecting to the Ethereum blockchain. This can be reached by adding a new feature on this device that will have associated an Ethereum address and, through the Internet, it will be able to interact with the blockchain network. In particular, this device acknowledges an input and output energy transfer in terms of tokens via *SC*.

The SC sends a certain amount Tk of tokens to the prosumer. The value Tk is computed as $Tk = E \cdot c_{i,j}$, where $c_{i,j}$ is the exchange rate between the prosumer P_i and the retailer R_j. Moreover, *SC* generates, at this point, an event Transfer to log the operation carried out.

4.4. Energy request

In this step, a consumer (or a prosumer acting as a consumer) asks for energy (i.e., tokens) by building a request containing the amount of energy needed. In particular, this task is carried out by calling the function newAuction() of the SC and giving as input parameters the amount of requested energy and two timestamps d_1 and d_2, that are used as deadlines of the auction. Observe that the consumer does generate a new address for the function call, which we call Temporary Ethereum Address (TEA). This address is generated by the same consumer or prosumer and is a disposable address since it is never reused for another auction in order to avoid to link different auctions done by the same actor. Indeed, a public blockchain allows everyone to rebuild the graph of interactions and transactions among users. Thus, if an actor uses her/his MEA in every step of the solution, a competitor retrieves her/his bids to obtain commercial and economic advantages for the future. For this reason, our model requires that only the winner prosumer and the energy applicant disclose their MEAs only in the Agreement phase.

At this point, the SC starts a blind auction with a fixed deadline d_1.

4.5. Auction

Any prosumer can participate in the auction by bidding a price p for this supply. In particular, the prosumer has to call the function SendBlindBid() of the SC by giving it the blind offer of the price $H(p\|r)$, where r is a random value and H stands for a cryptographic hash function. In this way, the real bid is hidden to the other competitors.

We remind that to prevent identity disclosure the prosumer uses a new TEA to participate in this auction.

4.6. Awarding

At the auction deadline d_1, each prosumer that participated in the auction calls the function sendBid() and passes as parameters the values p and r in plain-text to disclose its offer.

After all participants reveal their offers or after the deadline d_2 established previously by the energy applicant, the auction is awarded to the best bidder. In fact, the energy applicant retrieves the best offer related to its auction by calling the function endAuction(), which computes the best offer and returns the winner bidder.

Before establishing the winner, this function calculates $H(p\|r)$ and verifies that the result is equal to the value submitted in the previous step, thus validating the offer.

4.7. Agreement

Now, both the consumer and the awarded prosumer must disclose their identities. For this purpose, the prosumer has to link its TEA used in the previous phase to its MEA (which is publicly available) by generating a transaction from MEA to TEA and another from TEA to MEA.

This way, the prosumer proves to be the owner of both the Ethereum addresses. It is now necessary to check that the MEA associated with the awarded prosumer has, at least, p token available in its wallet. This means that the prosumer can fulfill the consumer request. If this check fails, the SC discards the awarded prosumer and, by shifting the list of prosumers participants in the auction, it repeats the operations with the newly awarded prosumer. This cycle is repeated until all the requirements are fully satisfied.

At this point, the consumer has to prove to be the owner of the address A used during the auction. To do this, the prosumer generates a random value *r* (challenge), which is sent to the consumer. The consumer generates a new transaction from *A* to MEA of the prosumer having as payload *r*, thus proving to be able to win the challenge. Moreover, the consumer uses an identity-based authentication scheme to disclose her/his identity: for example, schemes such as OpenId-Connect and SAMLv2 [17] can be used (this aspect is out of the scope of the paper). If these operations succeed, the consumer and prosumer complete the auction by exchanging tokens and ethers as resulting from the energy request and auction.

4.8. Redeem tokens for energy

This step can be carried out by everyone with tokens in their Ethereum wallets, including both prosumers and consumers, which want to redeem tokens for energy. In this step, the energy applicant has to send tokens towards the retailer by using a given function of the SC. This function will check that the sender has the amount of token in the wallet and that the recipient of this amount is a registered retailer. If these controls succeed, then tokens are transferred from the applicant wallet to the retailer's. At this point, the retailer sends to the applicant electricity via the smart grid's infrastructure and generates, at the same time, an Ethereum transaction with the information about the amount of energy sent.

However, since the retailer is not fully trusted (as it happens in real-life architectures as well), it is necessary to adopt some countermeasures to contrast a hypothetical malicious behavior of the retailer. At this point, the applicant's smart meter plays a fundamental role.

There are, potentially, four options: (*i*) the energy received is equal to the agreed amount, (*ii*) the energy received is less than the agreed amount, (*iii*) the energy received is more than the agreed amount, (*iv*) the energy is not received. Based on these situations, the smart meter will generate, as an answer, an Ethereum transaction by calling a function of success or failure. In this last case, a dispute arises between the energy applicant and the retailer. The Energy Authority is involved as a super party to effectively mitigate and solve the problem.

5. Implementation

In this section, we present a possible implementation of our proposal and describe the Ethereum SC that provides the needed functionalities. First, it is necessary to set up the environment useful for the development of such a SC. In particular, we use *Remix* as Integrated Development Environment (IDE) and *MetaMask* that is a browser extension that allows us to run Decentralized Applications (dApps) directly on the browser without running a full Ethereum node. The programming language is Solidity [11] and the SC has been deployed on the Ropsten TestNet. For the sake of presentation, we only focus the attention on the most relevant aspects of the implementation.

In Table 1, we show costs associated with our implementation. In particular, we focus on the most common and used functions of the SC and also the entire (and unique) deployment of the SC, reporting both the values in milliether and in US dollars (in July 2020).

First, we had to declare the token *ERC*20 interface in such a way our SC can inherit it by implementing its functions. We gave the token the name of *SET*, which stands for both Smart Energy Token and Smart Energy Transfer. Because of the aim of such a token, the *ICO* period is not necessary so that the initial total supply was given totally to the developer of the SC by means of the constructor function. Indeed, in Solidity, the constructor method is called and executed only when the SC is deployed. We use also this property for storing the information about the actual developer of the SC in the owner variable. We remind that, in our case, the developer of the SC is the Energy Authority.

Another fundamental Solidity property we exploited is the modifier, which is used to limit the access to functions. In particular, in Listing 1, we implemented a modifier that, if declared in a given function, limits the access only to the developer of the contract. For example, we used this modifier in function _add_retailer(), which can be called only by the Energy Authority to add the addresses of verified retailers to this particular list. An analog pattern has been used also in the case of the insertion of verified prosumers into the list by declaring the function_add_prosumer() with the corresponding modifier onlyRetailer. Generally speaking, this pattern is used every time a function needs this kind of restriction.

```
 modifier onlyOwner() {
      if (msg.sender != owner) {
         revert();
      }
      _;
   }
function _add_retailer(address
   _new_retailer)onlyOwner public{
   retailers_list[_new_retailer]=true;
   }
```

Listing 1: Application of the Solidity modifier in our smart contract.

In the whole demand-response cycle, the first operation that is carried out in the Ethereum environment is the Energy Request. In particular, in Listing 2, we implemented the function newAuction() that generates a new auction on the system. The applicant has to declare how many kWhs are needed and the periods of time she/he wants to wait for the completion of the whole process. In detail, the applicant has to give two timeouts to the function. The first timeout denotes the period of time in which the auction is active while the second one denotes the period of time until the prosumer can send the plaintext bid.

When the new auction is created the SC adds it into the mapping all_auctions and the function emits also an event to log this operation.

```
function newAuction(uint kWh, uint
   timeout1, uint timeout2)public {
   uint id_auction = getID();
   all_auctions[id_auction].consumer =
      msg.sender;
   all_auctions[id_auction].active = true
      ;
   all_auctions[id_auction].
      end_of_auction = now+timeout1;
   all_auctions[id_auction].
      end_of_disclosurement = now+
      timeout1+timeout2;
   emit newAuctionGenerated(msg.sender,
      id_auction, kWh, now+timeout1, now
      +timeout1+timeout2, now+timeout1+
      timeout2);}
```

Listing 2: Creation of a new auction.

Table 1
Costs of the deployment of the Smart Contract.

Function	Milliether	US Dollars
newAuction()	0,149	0,035
sendBlindBid()	0,023	0,005
sendBid()	0,027	0,006
endAuction()	3	0,71
Whole SC	4,684	1,12

At this point, prosumers can send their blind bids to answer the token request by calling the function sendBlindBid and, before the second timeout expires, they call the function sendBid(), in which they reveal the real offer made (Listing 3).

```
function sendBlindBid(uint idAuction,
    bytes32 blind, bytes32 hashRandom)
    public returns (bool) {
    require(all_auctions[idAuction].active
        ==true && now<all_auctions[
        idAuction].end_of_auction, "The
        auction is now closed");
    blindBid[msg.sender].idAuction=
        idAuction;
    blindBid[msg.sender].blind=blind;
    blindBid[msg.sender].hashRandom=
        hashRandom;
    blindBids[idAuction].push(blindBid[msg
        .sender]);
    return true;}

function sendBid( uint idAuction, uint
    cost , uint _random ) public returns (
    bool){
    require(all_auctions[idAuction].active
        ==true  && now>all_auctions[
        idAuction].end_of_disclosurement ,
         "It's too late");
    if(blindBid[msg.sender].blind ==
        keccak256(abi.encodePacked(toBytes
        (cost),toBytes(_random)))){
        bid[msg.sender].cost=cost;
        bid[msg.sender].idAuction=
            idAuction;
        bid[msg.sender].bidderAddress=msg.
            sender;
        bid[msg.sender].random=_random;
        bids[idAuction].push(bid[msg.
            sender]);}
        return true;}
```

Listing 3: Functions sendBlindBid() and sendBlind().

The next step is to compute the winner prosumer after the end of the auction. So, the tokens applicant calls the function endAuction() that first checks whether the auction is closed and, if this operation successes, it computes the winner prosumer. The code of these steps is shown in Listing 4.

```
function getBestValue(uint idAuction)
    public returns(offer memory){
    require(all_auctions[idAuction].
        consumer==msg.sender && now >
        all_auctions[idAuction].
        end_of_disclosurement);
    offer memory _o = bids[idAuction][0];
    uint best_cost= bids[idAuction][0].
        cost;
    uint n=bids[idAuction].length;
    uint pos = 0;
    for(uint j=1;j<(n);j++) {
        if (bids[idAuction][j].cost<
            best_cost && bids[idAuction][j
            ].unvalid == false ){
            best_cost = bids[idAuction][j
                ].cost;
            _o = bids[idAuction][j];
            pos = j;}}
        bids[idAuction][pos].unvalid=true;
        return _o;
        }

function endAuction(uint idAuction) public
     returns ( address, uint) {
    require(all_auctions[idAuction].
        consumer==msg.sender && now>
        all_auctions[idAuction].
        end_of_auction, "The auction is
        still active");
    offer memory best_offer=getBestValue(
        idAuction);
    address winnerAddress= best_offer.
        bidderAddress;
    uint winnerBid = best_offer.cost;
    all_auctions[idAuction].winner=
        winnerAddress;
    emit eventEndAuction(winnerAddress,
        winnerBid);
    return (winnerAddress, winnerBid);
    }
```

Listing 4: Ending of the auction and computation of the winner prosumer.

Now, the winner prosumer and the energy applicant have to send, respectively, tokens and ethers to the SC, which will collect and exchange them with each other. Since the prosumer participating in the auction is with a TEA, now it has to use the MEA to send tokens and receive ether. In particular, the function putToken() is called by the MEA of the prosumer, to demonstrate it is the real owner also of the address that won the auction. To achieve this goal, the prosumer has to carry on the following steps. First, it has to sign the hashed MEA with the private key corresponding to the TEA that has been used to participate in the auction. At this point, the prosumer uses its MEA to send this signed hashed information together with tokens in such a way to demonstrate it is the actual possessor of both the MEA and the TEA.

Finally, the energy applicant, which can be both a prosumer or a consumer, has to exchange its tokens with the retailer to obtain physically the energy needed. This operation is carried out by calling another function that is used to receive tokens and triggering the dispatch of the electricity thanks to the smart grid infrastructure.

6. Security aspects

In this section, we discuss the security properties and the adversary model of the solution described above. We show that the following security properties are guaranteed:

Data confidentiality refers to protecting information from unauthorized users. In our case, the real values of the bids should be hidden and protected from the other auction competitors until the auction deadline.

Data integrity refers to the completeness, consistency, and accuracy of data. Data used for energy trading should not be tampered with: in particular, once declared, the price of bids during the auction phase should not be modified by anyone.

Privacy requires that no identifying or sensitive information is disclosed if not necessary. In our case, both prosumers' and consumers' identity information should be preserved during an auction to assure fairness.

Authentication guarantees the verification of the identity of the entities accessing a protected system or a resource. We require that, after the auction, the involved actors are aware of their reciprocal identity.

Accountability assures that the operations carried out in a collaborative system occur in an open and accountable way. In our solution, we refer to the accountability of every transaction among actors.

Reliability is the probability that a system can perform a predetermined function under given conditions for a given time. In our scenario, reliability means that the actors can exploit system functionalities, such as the request for energy, the auction, or the agreement between prosumers and consumers ensuring the continuity of correct services.

After describing the security properties to guarantee, we define the adversary model. In our analysis, the energy authority is a trusted party and behaves responsibly and correctly in the system. In contrast, a retailer, a prosumer, or a consumer can be malicious and act as an adversary in the system. Clearly, the adversary can also be an external entity of the system. In our attack model, the adversary cannot compromise the behavior of the energy authority and cannot guess randomly generated values, secret information, blockchain private keys, passwords of the other entities. Furthermore, the adversary cannot execute transactions from the Ethereum accounts of the other entities. The adversary cannot break the cryptography primitives (e.g., it cannot revert cryptographic hash values or decrypt ciphered messages) and cannot perform physical attacks on the infrastructure (e.g., tampering with smart meters). The goal of the adversary is to violate at least one of the security properties listed above.

Let start by describing how these properties are guaranteed.

Data confidentiality is reached during the auction. Indeed, the prosumer does not send to the SC the price p of the supply in plain text but sends the value $H(p\|r)$, where r is a random value. To violate the confidentiality of the price p, the adversary should either (1) break the one-wayness property of H or (2) guess the random r and use a brute-force approach. Both of these possibilities are unfeasible.

Concerning data integrity, the price p of the supply offered in the auction as $h = H(p\|r)$ cannot be modified. Suppose that, in the awarding phase, the adversary sends the values p_1 and r_1, with $p_1 \neq p$, thus trying to change the offered price. As the SC calculates $h_1 = H(p_1\|r_1)$, if $h_1 \neq h$, this attack is detected. Note that $h_1 = h$ with $p_1 \neq p$ is impossible because this would violate the second pre-image resistance property of cryptographic hash function [18]. Moreover, the integrity of the values sent to the SC cannot be tampered with, thanks to the immutability of blockchain transactions: when transactions are mined by the network, data contained into the transactions are stored and not modifiable any more.

The privacy of the users is obtained because the identity of the auction winner and the consumer is disclosed only after the end of the auction, in the last phase of the energy request/supply. Indeed, the auction participants do not use their MEA, which is linked to their identity, but a TEA that is randomly generated and used only for this auction. In effect, the reuse of blockchain addresses is strongly discouraged since the initial adoption of the blockchain technology [19]: Ethereum addresses are pseudo-anonymous and their reuse can favor the break of pseudo-anonymity of the owners. It is worth noting that the main address is never reused to contrast attacks based on behavior. Indeed, an attacker could track and link the activities of prosumers and consumers to gather useful information for predictive analysis based on energy consumption or the price offered for supply.

The authentication is achieved by using a challenge-response protocol, a protocol widely used for authentication [20], which is robust provided that the random number used as a challenge is generated from a sufficiently large domain and is never reused. The awarded prosumer has to link its TEA to its MEA. To do this, the prosumer signs by the TEA private key the value MEA, thus declaring its MEA. This association is guaranteed by the secretness of the TEA private key. Consumers have also to disclose their identity when a request of energy is supplied by the winner prosumer. The robustness of this authentication depends on the corresponding robustness of the digital identity chosen. Indeed, our solution is orthogonal to the identification scheme. We suggest the use of a digital identity compliant with the eIDAS Regulation [21], which is recognized to be robust and provides a normative basis for secure electronic interactions among citizens and companies all over Europe.

Accountability is reached because all the operations of energy production, energy request, energy provision, and payment are logged and stored in a public blockchain. By looking at the Blockchain transactions, it is possible to verify the behavior of any entity. The accountability of the operations carried out in the entire environment avoids the arising of disputes among the actors: no one can claim something different from what has been reported on the blockchain.

The reliability of the solution is based on the features of blockchain. The robust Ethereum network counts a large number of nodes that work for keeping alive the network, ensuring the reliability of the blockchain-based solutions. Observe that every actor is encouraged to behave well: indeed, participating in the auction requires a fee to be paid by every participant and this fee is not refunded in case of protocol violations. For example, the prosumer winner is discouraged from not providing the offered token because, in this case, the participation fee is not refunded by the SC (thus, protecting against attacks aiming at the denial of service).

7. Related work

In this section, we relate our proposal with the state of the art. The survey [22] provides an overview of solutions exploiting the blockchain technology in energy sector. The authors classify the proposals into different categories based on the field of activity (e.g., e-mobility, grid management, decentralized energy trading), the platform used, and the relative consensus algorithm. The authors of [22] introduce security and identity management as a possible outcome of the blockchain technology in energy applications.

They conclude that SCs simplify and make faster the cooperation and competition among energy suppliers. According to this result, our solution aims at protecting consumers' and prosumers' privacy by creating TEA exploited for the auction phase. In this way, no information related to the real identity is exchanged before the agreement phase. The authors of [23] focus on the security and privacy challenges of energy trading in SGs. The proposed system, PriWatt, relies on Bitcoin and Bitmessage: the former technology guarantees security and privacy without the need of a third party, and the latter assures anonymity through encrypted messages in messaging streams. A system limitation regards the message redundancy in the communication necessary to guarantee high levels of privacy and security.

The system provided in [24] is based on an Ethereum private blockchain that allows the participation of only authorized users. No identity management mechanisms are implemented but the access control and authentication are guaranteed through the blockchain's SC feature of restrict modifiers. In our SC we assure that only authorized users can run the functions through the modifiers but we assume that this is not enough. Indeed, during the system registration (see Section 4) we propose a challenge-response protocol to verify the MEA, which has been used to confirm the agreement between the parties.

The authors of [25] present a secure private blockchain-based platform assuring the privacy of producers and consumers. While the producer can exploit different energy accounts, the consumers' privacy is preserved by changeable public keys of their smart meters. Nevertheless, to reduce the computation, the negotiation between the producer and consumer is conducted off-the-chain. This choice limits the security properties of energy bids that are not evaluated by the SC as our solution contemplates. Indeed, one of our strengths results in the creation of a blind auction managed by the SC, in a trusted way and avoiding unfairness among prosumers.

In [26], the authors propose a solution to implement traceable energy governance in smart grid networks. The schema provides a transparent and traceable tracking of energy usage and consumption via the blockchain transactions. This proposal uses permissioned blockchain and super-nodes in charge of validating users' identities and activities. In contrast, our approach uses a public blockchain (Ethereum), which allows us to implement an auction without referees. The authors of [27] deal with Energy Storage Units (ESU) in SGs. In their proposal, they use certified pseudonyms and SCs with no centralized authority. Despite the similarity of the above choices with those of our proposal, the focus of their paper is quite different. Indeed, it does not deal with energy trading but only with the problem of charging coordination to avoid blackout. The authors of [28] solve the problem of privacy in an energy trading scenario with a consortium blockchain-oriented approach. During the energy trading phases, the authors introduce a privacy-preserving module named Black Box Module (BBM), whose main principle is to create a mapping accounts for energy sellers. Again, the focus of the paper is different from this paper, as this solution concerns the protection of data stored in blocks against linking attacks and malicious data mining algorithms.

In [29], the authors face the problem of privacy in the blockchain-based solutions for energy trading in SGs. Their proposal is based on the function-hiding inner product encryption to match every bid with its bidder. However, this solution requires a central trusted entity, the distributed system operator, that acts as a mediator between the user and the network. In our solution, no centralized entity is required.

A smart and scalable distributed ledger system for SGs is proposed in [30]. The authors analyze the properties of this new protocol and instantiate it in an electrical vehicles scenario. Ecash is the energy cryptocurrency of the system, used as a digital asset for energy transactions. These transactions are added in form of a directed acyclic graph. The validation of transaction is done by checking the balance amount of Ecash spent or used in the transaction and through the proof of time instead of the proof of work of Bitcoin. If the transaction is validated by more than half of the total smart chains, then the transaction is considered valid. Two chains are proposed: the seller and the buyer chains where the respective transactions are stored. This proposal is contrary to the current solution that relies on the already existing Blockchain technologies, as our schema does. Indeed, the authors of [30] design a new system inspired by the blockchain paradigm and aiming at meeting the limited computational resources of electric vehicles.

In [31], an implementation of a blockchain-enabled IoT approach for microgrids is presented. The study underlines the need for a system that considers the security and privacy of microgrid operations. The authors demand these security requirements to the IoT network, made of different energy devices, and to the microgrid central control which securely collects the data and transfers it to the blockchain. Then, the blockchain enables the IoT to provide the requested power. Even if the solution aims at providing the network with security properties, it is not clear how the blockchain enables these operations. The IoT network needs to be resilient and reliable to assure data security and privacy. Although, as declared, the data is only accessible for the respective area microgrids, the problem of privacy still exists, because the energy request and supply are shared inside the same area. In our solution, a SC certificates the validity of energy transactions and distributes them to the blockchain network. Furthermore, the privacy of stakeholders is guaranteed from the first phase of an energy request. Only when the prosumer wins the auction and the consumer is willing to buy the energy, they reveal their real identities.

Table 2
Comparison with existing solutions.

Techniques	[23]	[24]	[25], 26	[28]	[29]	This paper
Blind Auction	Yes	No	No	No	Yes	Yes
Data Confidentiality	Yes	Yes	No	No	Yes	Yes
Identity Management	Yes	No	Yes	Yes	No	Yes
Privacy-Preserving	Yes	No	Yes	Yes	Yes	Yes
Scalability	Medium	Low	Low	Medium	Low	High

We conclude this section with the comparison between our proposal and the solutions of the state of the art carried out considering four aspects:

1. If a solution contemplates a *Blind Auction.*
2. If *Data Confidentiality* of bids is preserved.
3. If *Identity Management* mechanisms or schema are considered.
4. If a solution preserves users' privacy (*Privacy-Preserving*).
5. How much the solution is scalable (*Scalability*).

In our comparison, Blind Auction, Data Confidentiality, Identity Management and Privacy-Preserving are boolean measures, whereas we use the values *low*, *medium*, and *high* for Scalability. Specifically, solutions based on private blockchain are labeled as low scalability; solutions exploiting consortium blockchain have medium scalability, whereas scalability is high when public blockchains are used. Table 2 summarizes the results obtained from our comparison.

This comparison allows us to claim that our solution outperforms the state of the art. Among others, our solution includes innovative features, as it does not require a centralized unit control, it manages the energy trading in a blind fashion until the agreement, and it does not exploit any external referee.

8. Conclusions

In this paper, we propose a solution for energy trading in SGs based on Ethereum blockchain and SCs. SGs are a domain in which the power of blockchain can be profitably exploited to achieve the aimed goals. This paper testifies the above statement, by showing that SCs can enable a robust solution allowing energy trading as an auction with no referee and without requiring that different parties trust each other. An important aspect we remark here is that the implementation issues regarding SCs, including efficiency, scalability and costs, have been fully addressed, to provide a concrete proposal. Also the security analysis does not identify drawbacks of the solution. As a future work we plan to experiment it in a real-life setting we are defining in the context of an industrial research project.

References

[1] M. for primary industries, global energy & CO2 status report 2019 (IEA), https://www.iea.org/reports/global-energy-co2-status-report-2019, 2019.

[2] P. Siano, Demand response and smart grids—a survey, Renew. Sustain. Energy Rev. 30 (2014) 461–478.

[3] F. Buccafurri, L. Musarella, R. Nardone, A routing algorithm increasing the transmission availability in smart grids, in: Proceedings of the 2019 Summer Simulation Conference, Society for Computer Simulation International, 2019, p. 47.

[4] L. Peng, W. Feng, Z. Yan, Y. Li, X. Zhou, S. Shimizu, Privacy preservation in permissionless blockchain: A survey, Digital Communications and Networks 7 (3) (2021) 295-307.

[5] S. Nakamoto, Bitcoin: A peer-to-peer electronic cash system, Decentralized Business Review (2008) 21260.

[6] Ethereum. https://www.ethereum.org/, 2019.

[7] Z. Guan, X. Lu, N. Wang, J. Wu, X. Du, M. Guizani, Towards secure and efficient energy trading in iiot-enabled energy internet: a blockchain approach, Future Generat. Comput. Syst. 110 (2020) 686–695.

[8] F. Lombardi, L. Aniello, S. De Angelis, A. Margheri, V. Sassone, A blockchain-based infrastructure for reliable and cost-effective IoT-aided smart grids, in: Living in the Internet of Things: Cybersecurity of the IoT, 2018, pp. 1–6.

[9] A next-generation smart contract and decentralized application platform. https://github.com/ethereum/wiki/wiki/White-Paper, 2019.

[10] G. Wood, et al., Ethereum: a secure decentralised generalised transaction ledger, Ethereum project yellow paper 151 (2014) 1–32, 2014.

[11] Solidity, Documentation. https://solidity.readthedocs.io/en/v0.5.3/, 2019 (accessed 19 January 2023).

[12] E. Developers, Resources. 2019. https://www.ethereum.org/developers/, (accessed 19 January 2023).

[13] F. Victor, B.K. Lüders, Measuring ethereum-based erc20 token networks, in: International Conference on Financial Cryptography and Data Security, Springer, 2019, pp. 113–129.

[14] K. Zhou, S. Yang, Z. Shao, Energy internet: the business perspective, Appl. Energy 178 (2016) 212–222.

[15] C. Zhang, J. Wu, Y. Zhou, M. Cheng, C. Long, Peer-to-peer energy trading in a microgrid, Appl. Energy 220 (2018) 1–12.

[16] N. Wang, X. Zhou, X. Lu, Z. Guan, L. Wu, X. Du, M. Guizani, When energy trading meets blockchain in electrical power system: the state of the art, Appl. Sci. 9 (8) (2019) 1561.

[17] N. Naik, P. Jenkins, Securing digital identities in the cloud by selecting an apposite federated identity management from saml, oauth and openid connect, in: 2017 11th International Conference on Research Challenges in Information Science (RCIS), IEEE, 2017, pp. 163–174.

[18] P. Rogaway, T. Shrimpton, Cryptographic hash-function basics: definitions, implications, and separations for preimage resistance, second-preimage resistance, and collision resistance, in: International Workshop on Fast Software Encryption, Springer, 2004, pp. 371–388.

[19] J. Barcelo, User privacy in the public bitcoin blockchain, URL: http://www. dtic. upf. edu/jbarcelo/papers/20140704 User Privacy in the Public Bitcoin Blockc hain/ paper.pdf).

[20] C.A. Meadows, Analyzing the needham-schroeder public key protocol: a comparison of two approaches, in: European Symposium on Research in Computer Security, Springer, 1996, pp. 351–364.

[21] J. Dumortier, Regulation (eu) no 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eidas regulation), in: EU Regulation of E-Commerce, Edward Elgar Publishing, 2017.

[22] M. Andoni, V. Robu, D. Flynn, S. Abram, D. Geach, D. Jenkins, P. McCallum, A. Peacock, Blockchain technology in the energy sector: a systematic review of challenges and opportunities, Renew. Sustain. Energy Rev. 100 (2019) 143–174.

[23] N.Z. Aitzhan, D. Svetinovic, Security and privacy in decentralized energy trading through multi-signatures, blockchain and anonymous messaging streams, IEEE Trans. Dependable Secure Comput. 15 (5) (2016) 840–852.

[24] A.S. Yahaya, N. Javaid, F.A. Alzahrani, A. Rehman, I. Ullah, A. Shahid, M. Shafiq, Blockchain based sustainable local energy trading considering home energy management and demurrage mechanism, Sustainability 12 (8) (2020) 3385.

[25] A. Dorri, F. Luo, S.S. Kanhere, R. Jurdak, Z.Y. Dong, Spb: a secure private blockchain-based solution for distributed energy trading, IEEE Commun. Mag. 57 (7) (2019) 120–126.

[26] K. Gai, Y. Wu, L. Zhu, L. Xu, Y. Zhang, Permissioned blockchain and edge computing empowered privacy-preserving smart grid networks, IEEE Internet Things J. 6 (5) (2019) 7992–8004.

[27] M. Baza, M. Nabil, M. Ismail, M. Mahmoud, E. Serpedin, M.A. Rahman, Blockchain-based charging coordination mechanism for smart grid energy storage units, in: 2019 IEEE International Conference on Blockchain (Blockchain), IEEE, 2019, pp. 504–509.

[28] K. Gai, Y. Wu, L. Zhu, M. Qiu, M. Shen, Privacy-preserving energy trading using consortium blockchain in smart grid, IEEE Trans. Ind. Inf. 15 (6) (2019) 3548–3558.

[29] Y.-B. Son, J.-H. Im, H.-Y. Kwon, S.-Y. Jeon, M.-K. Lee, Privacy-preserving peer-to-peer energy trading in blockchain-enabled smart grids using functional encryption, Energies 13 (6) (2020) 1321.

[30] G. Bansal, A. Dua, G.S. Aujla, M. Singh, N. Kumar, Smartchain: a smart and scalable blockchain consortium for smart grid systems, in: 2019 IEEE International Conference on Communications Workshops (ICC Workshops), IEEE, 2019, pp. 1–6.

[31] M. Dabbaghjamanesh, B. Wang, S. Mehraeen, J. Zhang, A. Kavousi-Fard, Networked microgrid security and privacy enhancement by the blockchain-enabled internet of things approach, in: 2019 IEEE Green Technologies Conference (GreenTech), IEEE, 2019, pp. 1–5.