energies

Article

Multiple Authorities Attribute-Based Verification Mechanism for Blockchain Mircogrid Transactions

Sarmadullah Khan [1,*] and Rafiullah Khan [2]

1 School of Computer Science and Informatics, De Montfort University, Leicester LE1 9BH, UK
2 School of Electronics, Electrical Engineering and Computer Science (EEECS), Queen's University Belfast, Belfast BT7 1NN, UK; rafiullah.khan@qub.ac.uk
* Correspondence: sarmadullah.khan@dmu.ac.uk

Received: 14 March 2018; Accepted: 1 May 2018; Published: 4 May 2018

Abstract: Recently, advancements in energy distribution models have fulfilled the needs of microgrids in finding a suitable energy distribution model between producer and consumer without the need of central controlling authority. Most of the energy distribution model deals with energy transactions and losses without considering the security aspects such as information tampering. The transaction data could be accessible online to keep track of the energy distribution between the consumer and producer (e.g., online payment records and supplier profiles). However this data is prone to modification and misuse if a consumer moves from one producer to other. Blockchain is considered to be one solution to allow users to exchange energy related data and keep track of it without exposing it to modification. In this paper, electrical transactions embedded in blockchain are validated using the signatures of multiple producers based on their assigned attributes. These signatures are verified and endorsed by the consumers satisfying those attributes without revealing any information. The public and private keys for these consumers are generated by the producers and endorsement procedure using these keys ensures that these consumers are authorized. This approach does not need any central authority. To resist against collision attacks, producers are given a secret pseudorandom function seed. The comparative analysis shows the efficiency of proposed approach over the existing ones.

Keywords: secure communication; microgrid; security

1. Introduction

Microgrids act as source of electricity to small geographical region such as healthcare centers, military units, homes etc. [1–3]. Micrograds can also be integrated with national power distribution centers and other renewable energy generation sources (e.g., solar, wind etc.). The basic architecture of mircogrid includes (1) electrical load, (2) energy storage unit and (3) a line to and from the main grid. The integration of micrograds with the main grid make them to operate in the connected mode along with their standalone operation capabilities. Hence microgrids first fulfill with the local energy requirements and then provide extra energy to the main grid to facilitate other consumers.

Various microgrid projects in USA use the blockchain technology for managing energy transactions and give an overview about new energy system. New energy system concept is based on distributed generation including renewable sources, energy transmission to consumers/main-grid, communication among various distribution and communication network elements and managing financial transactions. The main stakeholder also tries to reduce the time that spent on managing financial transactions. Consumer might gets energy from multiple distributors and billing server needs a reliable and authentic information. This is made possible by the use of blockchain that offers cryptocurrency for monetary transactions in energy field. Many companies have set up energy exchange platforms to bring the buyers and sellers on one page. For example, dutch company Vandebron [4] offers the possibility to buy energy directly from producers using a central entity that manages the network, prepares bills and checks the balance between production and consumption. In mircogrids, decentralized authorities make the transactions efficiently manageable using blockchain but this approach is very resource consuming considering all the stakeholders in the authentication and information processing. Also each stakeholder must have an access to desire transaction data instead of whole consumers transactions. Each stakeholder must have to verify its authenticity before making any changes to the consumer transactions. In authentication, multiple attribute-based signature is an efficient approach that meets with the requirement of distributed authentication procedure and also protects the consumers privacy [5].

The rest of the paper is organized as follows. A literature survey is provided in Section 2 while an overview of blockchain and attribute-based encryption are discussed in Section 3. Section 4 provides a brief mathematical details of attribute-based algorithms. The proposed security algorithm is discussed in Section 5. The performance evaluation of the proposed algorithm is presented in Section 6. Finally, Section 7 concludes the paper.

Main Contribution

The main contributions of this paper are:

1. develop a framework to keep the record of energy transactions for future use and verification purpose by new consumers
2. hide the actual transaction details while disclosing only the reputation and performance metrics of a mircogrid owners

2. Literature Survey

Cyber security in smart grids is analyzed in detail in [6]; however, this section only considers the relevant information of that analysis. Authentication in smart grids is one of the main critical security aspect that allows the users to access the its various elements. It is achieved using the digital signatures, username and password approach and hashing functions. In a digital signature, a user first generates the hash of a message using Secure Hash Algorithms (SHA) or Message-Digest algorithm (MD5) and then encrypts it with his private key using RSA. The encrypted hash can only be decrypted with the public key of same user who encrypted it. This ensures the authenticity of the message while the user authentication is achieved by the username and password approach.

Many other security approaches based on one time signature, message authentication code (MAC), RSA encryption are proposed in [7,8]. In one signature approach, each signature is used once to very a message. This helps to avoid the replay attacks as the message will be discarded if received after a threshold time value. Precomputed hashing approach is proposed in [9], however it suffers from a very large computational power to map the messages with the precomputed hashes.

In message authentication code, a single key is shared between the communicating parties to verify and authenticate the received messages. TESLA [7] used the same concept with slight changes. In TESLA, time is divided into slots and for each slot there is one secret key. A message for a particular time slot is encrypted with a key belonging to that slot. The message is then send to the receiver while the key belonging to the message is released after it expiry. Hence a receiver receives the messages, buffered them and wait for the corresponding keys. However this approach has a very high memory requirements as the receiver has to stored all the messages until it receives the keys. This approach is not feasible for the real time applications.

Most of the existing security solutions are proposed for the smart grids communications that are not suitable for the microgrids communications due to different architecture. A detailed security analysis based on the microgrid architecture is presented in [10]. However, this analysis did not address the communication security threats and solutions in mircogrid architecture.

Sahai and Waters [11] presented attribute-based framework to build a number of cryptographic primitives. The attribute-based signature scheme allows the user to attest the correctness of information while hiding its original contents from outside world. The signature is only a validation procedure that ensures the message is endorsed by the signer having valid attributes. An attribute-based signature ensures privacy to signer while it ensures unforgeability to the verifier. Khader [12] presented a group signature scheme based on attributes while the formal definition was presented in [5,13]. The security of these protocols were analyzed in only generic group model. A secure forward attribute-based signature schemes were presented in [14,15] however they did not consider the adaptive-predicate privacy and unforgeability. A fully secured attribute-based signature scheme for standard model and other models considering the non-monotone predicate is presented in [16]. However it is not suitable to apply in practice. To improve the efficiency, Chen [17] presented attribute-based short signature scheme but this scheme is based on single authority which does not fit in distributed applications. To improve the computational cost, an efficient attribute-based signature scheme with monotone predicate is presented by Gu [18]. To reduce the dependency on attribute authorities, escrow based attribute-based signature scheme is presented by Cui [19] where users could provide evidence to the verifier about their signature rights. However, these schemes are based on a single authority that is not suitable for distributed systems. In this paper, we propose multiple authorities attribute-based signature scheme for blockchain microgrid architecture that suits the distributed nature of system both in security as well as tamper proof energy transaction record.

3. Background

This section gives a breif overview of blockchain technology and attribute-based authentication mechanism. These two technologies in later sections are used to describe how user can benefit from them in making a secure and reliable energy transactions in microgrid architecture.

3.1. Blockchain

Blockchain technology concept is based on distributed database that keeps the records of all transactions in ordered list in which they are executed without the involvement of central authority (e.g., banks). Bitcoin (also known as crypto currency) is one of the main example that uses blockchain for all transactions without any central authority. Smart contracts are also established using blockchain and execute automatically when they fulfill the required conditions. Hence bloachchain is a distributed ledger that grows continuously with data/transaction record called block. Each individual block in the blockchain is time-stamped, connected with previous block, shared and not modifiable. In this paper, blockchain is used by the users to check and records all the transactions occurred in the network and selects an appropriate microgrid distributor to purchase and deal in energy. A user can verify a copy of blockchain or newly received block and add it into its chain. Once added into chain, block cannot be modified. Any attempt to modify the block in chain results in invalid chain. Blochchain technology has the following main elements:

1. verification mechanism
2. a network to share blocks (ledgers)

Including previous block hash into the new block connects them with each other and this enables the user to check the validity of blockchain by only verifying the authenticity of last block in a chain. The network allows each user to share the distributed ledger with other users. Figure 1 gives a pictorial representation of blockchain where each block contains one or more transactions. For example, who is purchasing and selling energy, amount of energy, duration and time-stamp. In this scenario, everyone in the network knows everyone else transaction details and may reveal private information if not secured. To attract the consumers, mircogrid owner also include his/her performance report as a block in chain. This performance report must be verified and attested by the his/her previous consumers.

In this paper, we are addressing the verification and attesting mechanism of such performance report block in blockchain using attribute-based authentication mechanism.

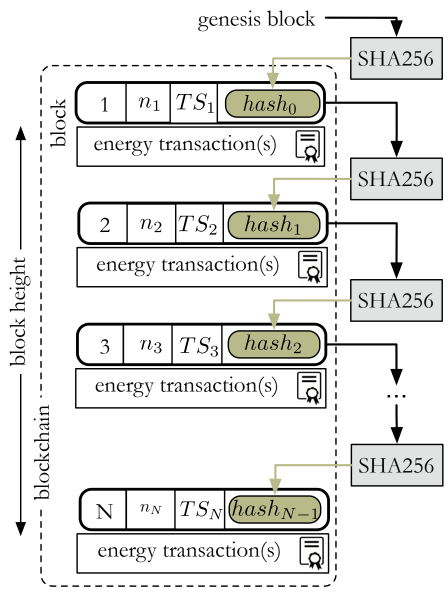

Figure 1. Blockchain model.

3.2. Attribute-Based Security

In attribute-based security, ciphertext is associated with some label of encryptors called attributes. Each private key is also associated with the access tree known as predicate. The predicate defines the policy how to decrypt the ciphertext with associated keys. Normally the predicate consists of AND, OR and threshold gates [11]. Goyal [20] showed how users can associate and include predicates into their private keys. It has two variants Key-Policy Attribute-Based Encryption (KP-ABE) and Ciphertext-Policy Attribute-Based Encryption (CP-ABE). In KP-ABE, user's secret keys are generated based on an predicate that defines the privileges scope of the concerned user, and data are encrypted over a set of attributes. However, CP-ABE uses predicate to encrypt data and user's secret keys are generated over a set of attributes. In CP-ABE, a user can decrypt the ciphertext if and only if his attribute set satisfies the predicate.

Attribute-based security algorithms consist of mainly four steps: (1) algorithm setup, (2) private key extraction, (3) signing and (4) verification. The universe of attributes is represented by S while predicate over the universe of attributes is a monotone boolean function takes inputs from U. We can say that an attribute set W satisfies a predicate β if $\beta(\mu) = 1$ (where an input is valid if the corresponding attributes are chosen from U).

4. Preliminaries

This section describes the notations used in this paper and some definitions as:

4.1. Bilinear Mapping

We consider two cyclic groups of prime order q i.e., $(G, +)$ and $(G_T, \times)$. $(G, +)$ is additive cyclic group while $(G_T, \times)$ is multiplicative cyclic group. Bilinear mapping $e : G \times G \rightarrow G_T$ have following properties:

1. ***Bilinearity***: For any $X, Y \in G$ and $p, q \in {Z_y}^*$, it has $e(pX, qY) = e(X, Y)^{pq}$
2. ***Non-degeneracy***: For any $X, Y \in G$ must satisfy $e(X, Y) \neq 1_{G_T}$
3. ***Computability***: For any $X, Y \in G$ it is easy to compute $e(X, Y)$

4.2. Computations

We have a finite cyclic group G of order y and $p, q, b, n \in Z_y{}^*$ are selected randomly. The security of this approach lies in discrete logarithmic problem and computational bilinear diffie hellman problem. These are defined as:

Discrete Logarithmic Problem: Given $X, Y \in G$, it is difficult to find the integer n such that $Y = nX$.
Computational Bilinear Diffie Hellman (CBDH) Problem: Given $A = pX, B = qX, C = bX \in G$ and bilinear mapping $e : G \times G \rightarrow G_T$, it is difficult to find p, q, b if given $e(X, X)^{pqb}$.

4.3. Predicate

Suppose we have a set of parties $X_p = \{X_{p_1}, X_{p_2}, X_{p_3}, ..., X_{p_n}\}$ and monotone access tree structure $\beta \in 2^{\{X_{p_1}, X_{p_2}, X_{p_3}, ..., X_{p_n}\}}$ such that for all $I, E \in \beta$ and $I \subseteq E$. Also access structure β is a collection of non empty subset of $\{X_{p_1}, X_{p_2}, X_{p_3}, ..., X_{p_n}\}$.

Suppose we have a universe of attributes B and monotone access tree structure over this universe is monotone Boolean function whose inputs are from B. There is another attributes set $W \in S$ that satisfies the predicate β if $\beta(W) = 1$. As β is a monotone in nature, for any set $W \in V$, $\beta(W) = 1$ implies $\beta(V) = 1$. In this paper, a microgrid is assigned a set of attributes and the authorized set is also included in monotone access tree structure β. Data verifier (consumer) would be able to verify the signature if and only if the attributes satisfy the access tree structure of the signature.

4.4. Multiple Authority Attribute-Based Signature

Multiple authority attribute-based signature scheme in microgrid architecture is split into five steps.

1. **Setup** $(1^\lambda \rightarrow Params)$**:** Security parameter (1^λ) is given as input to generate public parameters.
2. **Authority Setup** $(1^\lambda \rightarrow (K_k, l_k))$**:** Each authority (A_k) in the system generates a public key and a private key using this algorithm. Where $k = \{1, 2, 3, ..., N\}$ and N is the total number of authorities in the system.
3. **KeyGen** $(l_k, GID, S) \rightarrow (K_U, l_U)$**:** This algorithm generates the public and private key for microgrid (K_U, l_U) by taking as input the private key of the authority (l_k), global identifier of the micrgrid (GID) and a set of attributes B.
4. **Sign** $(K_k, l_U, M, \beta) \rightarrow \epsilon$**:** To sign the message M using the access tree β, this algorithm takes as input the public key of authority K_k, private key of microgrid l_U and access policy β and generates the signature ϵ of the message M.
5. **Verify** $(K_U, \epsilon, B, M, \beta) \rightarrow Accept/Reject$**:** Upon receiving the signature and message, this algorithm verifies the signature by taking inputs the public key of microgrid K_U, received signature ϵ, message M, attributes set B, access policy β and generates output in the form of *Accept or Reject*.

4.5. Security Definitions

Unforgeability is one of the main security feature that attribute-based signature scheme provides however it also suffers from the colluding authorities or users. To explain it in a better way, we consider a scenario between a challenger C and a forger F as follows.

Setup: During setup phase, the challenger C generates the public parameters using the using the secret parameter 1^λ and transmits it to F. F then sends the a predicate β^* and list of malicious users J_A to C.
Authority/User Setup: In this phase, the challenger generates the public and private keys (K_K, l_K) for the corrupted authorities and sends it to the forger F.
Queries: Now the challenger C initializes the integer $m = 0$ for the list $J = \{m, B, l_U\}$ and allows the forger F to execute the following steps.

Private key extraction oracle: Once the challenger C receives the m and set of attributes B, it returns the secret key l_U to the forger F otherwise it generates the l_U using the KeyGen algorithm and sends the generated l_U to the forger and adds this new entry (m, B, l_U) into list J.

Signing oracle: As the challenger receives the message M and predicate β, its generates the signature ϵ and sends it back to the forger

Forgery: The forger F makes the tuple (M^*, ϵ^*) and β^* public.

A forger F wins the above scenario if and only if (1) it has a valid signature ϵ^* of the message M^* with access policy β^* and (2) $\beta(S) \neq 1$. The winning probability of this scenario by the forger is given by $Adv^{EUF}_{MA-ABS}(\lambda)$.

A forger can break the MA-ABS scheme $(t, d_H, d_X, d_B, \epsilon)$ if it executes the scenario at least for t times and make d_H hash queries, d_X private key extraction queries, d_B signing queries and $Adv^{EUF}_{MA-ABS}(\lambda)$ is at least ϵ. The MA-ABS is unforgeable if there is no probabilistic polynomial time forger exists that breaks $(t, d_H, d_X, d_B, \epsilon)$.

MA-ABS scheme is perfectly private if all the parameters, messages, attributes sets, all private keys, predicates, distributed signature and actual signatures are equal. Also the signature should not reveal any private information of the signer.

5. Multi-Authority ABS Scheme

The proposed microgrid system model for transactions and losses records is presented in this section along with the ABS and blockchain.

5.1. System Model

The proposed ABS scheme is based on multiple authorities that is applicable to distributed microgrid architecture with blockchain technology. The proposed model consists of the following entities: (1) record server, (2) N authorities, (3) microgrids and (4) verifier (consumer). As shown in Figure 2, record server behaves like a storage server that keeps the copy of all transactions happening in the microgrid network. N authorities consists of various organizations (i.e., banks, consumer registration authority, comsumers). Microgrids normally manage and sign their own transactions records and create their own access policy. The verifier (consumer) accesses these information to ensure their authenticity.

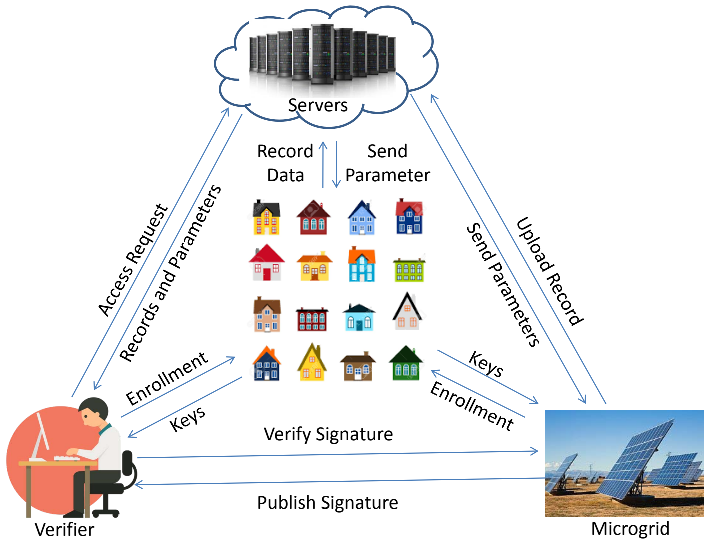

Figure 2. System model.

5.2. Proposed Approach

For any $m \in Z_y$, a set of attributes B whose elements also belongs to Z_y, the Lagrange coefficient is defined as

$$\triangle_{m,B}(v) = \prod_{n \in B, j \neq m} \frac{v-n}{m-n}$$

The proposed scheme associates each element of Z_y with each attribute. Detail description of the proposed scheme is as follows:

Setup: During the setup phase, microgrid server chooses two cyclic groups G and G_T of prime order d and bilinear mapping function such that $e : G \times G \rightarrow G_T$. Let X be a group generator of G and $H : \{0,1\}^* \rightarrow Z_y^*$ is a collision resistant hash function based on ECDH. Computing $r = H(GID)$ for microgrid global identifier GID. N authorities in the system are represented by $A_1, A_2, A_3, ..., A_N$ and each authority has a set of attributes $A_k = \{a_{k,1}, a_{k,2}, a_{k,3}, ..., a_{k,c_k}\}$. Also μ is randomly selected from Z_y^* and calculated Y as $Y = \mu X$. Now the overall public parameters for this system are $params = \langle e, d, X, Y, G, G_T, H \rangle$.

Authority Setup: Each authority randomly selects $\alpha_k \in Z_y^*$ and calculates $y_k = \alpha_k X$. Also each authority randomly selects $y_k \in Z_a^*$ for each attribute $p_{k,i} \in A_k$ and calculates $T_{k,n} = t_{k,m} X$. Two authorities (A_k, A_n) select randomly $s_{kn} \in Z_y^*$ and share it with each other as a seed for secret pseudorandom function (*PRF*) through a secure channel which then sets $s_{kn} = s_{nk}$. These authority also selects $v_m, v_n \in Z_y^*$ to define a common *PRF* as

$$PRF_{kn}(r) = (\frac{v_k v_n}{s_{kn} + r})Y$$

The authority A_k outputs the public key as

$$K_k = \langle y_k, \{T_{k,m}\}_{m \in \{1,2,3,...,c_k\}} \rangle$$

and private key as

$$l_k = \langle \alpha_k, v_k, \{s_{kn}\}_{n \in \{1,2,3,..,c_k\}}, \{t_{k,m}\}_{m \in \{1,2,3,...,c_k\}} \rangle.$$

KeyGen: Each microgrid is assigned a set of attributes A_U and each authority A_k picks $a_k \in Z_y^*$ to compute $B_{k,m} = \dfrac{a_k}{t_{k,m}}$ for $a_{k,m} \in A_U^k$ where $A_U^k = A_U \cap A_k$. Each mircogrid U communicates with each authority A_k for $N-1$ times to finalize and computes the key anonymously as

$$I_{kn} = \alpha_k X + a_k Y + PRF_{kn}(r) \text{ for } k > n$$

and

$$I_{kn} = \alpha_k X + a_k Y - PRF_{kn}(r) \text{ for } k \leq n$$

Finally

$$I_U = \textstyle\sum_{k,n \in \{1,2,...,N\} \times \{1,2,...,N\}} I_{kn}$$

$$= \textstyle\sum_{k \in \{1,2,...,N\}} (N-1)\alpha_k X + \sum_{k \in \{1,2,...,N\}} (N-1) a_k Y$$

The public key is declared as

$$K_U = \langle \{S_{k,n} Y\}_{k \in \{1,2,...,N\}, m \in \{1,2,...,c_k\}} \rangle$$

and the private key is declared as

$$l_U = \langle I_U, \{B_{k,n}\}_{k\in\{1,2,...,N\}, m\in\{1,2,...,c_k\}} \rangle$$

Sign: Every message is signed based on the access policy β. To do so, a polynomial d_v is selected for each leaf node/authority/party v. The degree of the polynomial is set as $k_v - 1$, where k_v is the threshold value of v. Starting from R (i.e., root node), set $d_R(0) = s$. Next another point on the predicate is selected and terminate the polynomial at that point. The microgrid selects randomly $f \in Z_y^*$ and calculates

$$\epsilon_1 = sI_U, \epsilon_2 = \frac{H(M)+f}{N-1}X, \epsilon_3 = \prod_{k\in\{1,2,..,N\}} e(sX, y_k),$$

$$\epsilon_4 = vsX, \epsilon_5 = sK_U, \epsilon_6 = f\epsilon_5, \epsilon_7 = \{d_v(0)T_{k,m}\}_{p_{k,m}\in A_\beta}$$

where $p_{k,m}$ is the value of attributes in access policy β. The final signature is

$$\epsilon = \{\epsilon_1, \epsilon_2, \epsilon_3, \epsilon_4, \epsilon_5, \epsilon_6, \epsilon_7\}$$

Verify: Each verifier (consumer) has a set of attributes denoted as $A_{IV} = \{q_1, q_2, q_3, ..., q_t\}$. If $\beta(A_{IV}) \neq 1$ then the output is *null*. Otherwise, the verifier gets the signature ϵ and performs the operations on this signature using the public key of microgrid K_U and node v from the access policy β as inputs to verify function $VerifyNode(\epsilon, K_U, v)$.

If $p_{k,m} \in A_U^k$, then

$$VerifyNode(\epsilon, K_U, v) = \prod_{k\in\{1,2,..,N\}} e(\epsilon_7, K_U)$$

$$= \prod_{k\in\{1,2,..,N\}} e(d_v(0)T_{k,m}, \frac{a_k}{t_{k,m}}Y)$$

$$= \prod_{k\in\{1,2,..,N\}} e(d_v(0)t_{k,m}X, \frac{a_k}{t_{k,m}}Y)$$

$$= \prod_{k\in\{1,2,..,N\}} e(X,Y)^{d_v(0)t_{k,m}\frac{a_k}{t_{k,m}}}$$

$$= \prod_{k\in\{1,2,..,N\}} e(X,Y)^{d_v(0) \sum_{k\in\{1,2,..,N\}} a_k}$$

If $p_{k,m} \notin A_U$ then output of the $VerifyNode(\epsilon, K_U, v)$ is *null*.

If node z is a child node of v, then $F_z = VerifyNode(\epsilon, K_U, z)$ is calculated and kept the output result. Suppose B_v is any arbitrary $k_v - sized$ set of child node z, makes the $F_z \neq null$. If there is no such set, then $F_z = null$. F_x is calculated as shown below where $B'_v = \{index(z) : z \in B_v\}$ and $i = index(z)$,

$$F_v = \prod_{z\in B_v} F_z^{\triangle_{i,B'_v}(0)}$$

$$= \prod_{z\in B_v} \left(e(X,Y)^{d_z(0)\left(\sum_{k\in\{1,2,..,N\}} a_k\right)} \right)^{\triangle_{i,B'_v}(0)}$$

$$= \prod_{z\in B_v} \left(e(X,Y)^{d_{parent(z)}(index(z))\left(\sum_{k\in\{1,2,..,N\}} a_k\right)} \right)^{\triangle_{i,B'_v}(0)}$$

$$= \prod_{z \in B_v} \left(e(X,Y)^{d_v(i)\left(\sum_{k \in \{1,2,..,N\}} a_k\right) \triangle_{i,B'_v}(0)} \right)$$

$$= e(X,Y)^{d_v(0)\left(\sum_{k \in \{1,2,..,N\}} a_k\right)}$$

For the access policy $\beta(A_U^k) = 1$, it is verified that

$$VerifyNode(\epsilon, K_U, v) = e(X,Y)^{s\left(\sum_{k \in 1,2,..,N} a_k\right)}.$$

The verifier (consumer) checks also

$$e(\epsilon_1, \epsilon_2) = \epsilon_3^{H(M)} \prod_{k \in 1,2,..,N} (e(\epsilon_4, y_k) e(H(M)\epsilon_5 + \epsilon_6, T_{k,m}))$$

Once all the above conditions are successfully validated then verifier (consumer) *Accepts* otherwise *Rejects*.

6. Performance and Evaluation

The security and performance analysis of the proposed protocol is performed using random oracle model. First security analysis is performed followed performance analysis.

6.1. Security Analysis

To evaluate the security of proposed protocol, we consider two authorities A_k and A_n in our system. These two authorities share secretly a PRF seed s_{kn}. This is important because if other $N-2$ authorities get corrupted, the *PRF* seed share between these two authorities remains un-corrupted. During the process of private key generation, all authorities private keys α_m are combined into mircogrid private key I_U using the **KeyGen** function. This approach protects I_U from disclosure even if there is only one single honest authority and rest get compromised by an attacker. In this way the protocol resists against collusion attack when there are $N-1$ corrupted authorities. In order to protect the privacy of microgrid, its GID is not revealed directly to authorities. Therefore corrupted authorities cannot trace the private record of microgrid.

The proposed MA-ABS security model for microgrid is unforgeable for selective access policy attacks using Computational Bilinear Diffie Hellman (CBDH). Suppose the forger F has some important information ϵ that can help the attacker using selective access policy attacks. In this case, the challenger C selects the security parameter 1^λ and runs the setup phase. The public parameters generated by setup phase are sent to the forger. Using the simulator ς that takes the F, public parameters and ϵ as inputs to solve the CBDH.

To launch an attack, forger makes d_X queries to extract the private keys, d_H queries to hash function and d_s queries to signing oracle. Now the simulator ς is given $\langle X, A = pX, B = qX, C = bX \rangle$ to compute $e(X,X)^{pqb}$ where $p, q, b \in Z_q^*$. The simulation is performed as:

- **Setup:** The forger F selects the challenger's access policy β^* and a set of attributes B^*. The forger gives β^* and B^* along with the list of corrupted authorities J_A to the simulator ς and sets $Y = (p + \mu)X$. The simulator returns A, B and C to the forger.
- **Authority Setup:** The simulator selects randomly $A_k^* \in \{A_1, A_2, .., A_N\} \setminus J_A$. If $A_k \in J_A$ then simulator selects $f_k, w_{k,m} \in Z_y^*$ randomly and calculates $T_{k,m} = w_{k,m}X$ for $p_{k,m} \in A_k$. Then the simulator selects $v_k \in Z_y^*$, a PRF seed $s_{k,n} \in Z_y^*$ for corrupted authorities A_k and A_n and returns the output $\langle f_k, w_{k,m}, v_k, s_{kn} \rangle$ and $\langle y_k, T_{k,m} \rangle$ to the forger where $y_k = f_k X$.

If $A_k \notin J_A$, then the simulator selects $f_k, w_{k,m} \in Z_y^*$ randomly and calculates $T_{k,m} = w_{k,m}X$ for $p_{k,m} \in \beta^*$ and $T_{k,m} = w_{k,m}A = w_{k,m}pX$ for $p_{k,m} \in \beta^*$. If $A_K \neq A_K^*$, the simulator sets $y_k = qf_kX$. Otherwise it sets

$$y_k = e(X,X)^{pq} \prod_{A_k \in J_A} e(X,X)^{-f_k} \prod_{A_k \in J_A, A_k \neq A_k^*} e(X,X)^{-qf_k}.$$

Then the simulator randomly selects a PRF seed $s_{kn} \in Z_y^*$ for the honest authorities and returns $\langle y_k, T_{k,m} \rangle$ to the forger.

- **Query:** Before starting the query process, simulator creates an empty list J and initialize an integer $m = 0$. The forger then sends out the query as follow:

Hashing-Query: The simulator maintains a list of hashing query J_H. This list contains the output of the hashing function oracle for queries. When a query M_m is received, where $m = \{1, 2, .., q_H\}$, first simulator checks the queries record list J_H. If the query already exists in J_H, the simulator outputs the entry of corresponding query. Otherwise it generates $H(M_m)$, adds it to the J_H and returns as $\langle M_m, H(M_m) \rangle$.

Private Key Generation Query: Once the attributes set S with $\beta(S) \neq 1$ is received, the simulator checks for the query $\langle m, B, l_U \rangle$ in the record list J. If query exists, it returns l_U otherwise the simulator executes the following steps:

1. For any $A_k \in J_A$ the simulator generates the secret key using

$$\langle f_k, w_{k,m}, v_k, s_{kn} \rangle$$

for the received set of attributes B.

2. If $A_k \notin J_A$, then the simulator randomly selects $a_k \in Z_y^*$ and calculates $\{s_{k,m} = \frac{a_k}{w_{k,m}}\}_{p_{k,m} \in \beta^*}$ and $\{s_{k,m} = \frac{a_k}{w_{k,m}p}\}_{p_{k,m} \in \beta^*}$. Now the simulator calculates I_{kn} as:

(a) If $A_k \neq A_k^*$ then for $k > n$,

$$I_{k,m} = f_kqX + a_kY + PRF_{kn}(U)$$

otherwise

$$I_{k,m} = f_kqX + a_kY - PRF_{kn}(U)$$

(b) If $A_k = A_k^*$, then for $k > n$,

$$I_{kn} = -\frac{q\mu}{s}X + \sum_{A_k \in J_A} ((-f_k)X) + \sum_{A_k \notin J_A, A_k \neq A_k^*} ((-f_k)qX) + a_kY + PRF_{kn}(U)$$

Otherwise

$$I_{kn} = -\frac{q\mu}{s}X + \sum_{A_k \in J_A} ((-f_k)X) + \sum_{A_k \notin J_A, A_k \neq A_k^*} ((-f_k)qX) + a_kY - PRF_{kn}(U).$$

Finally, the simulator adds $\langle m, B, l_U \rangle$ in J where $l_U = \langle I_U, \{B_{k,m}\} \rangle$ and also returns it to the forger.

Signing Query: Once the signing query $\langle M^*, \beta^*(B^*) \rangle$ is received, the simulator checks if $|B \cap B^*| < k$ then it generates the private key using the private key generation oracle.

Otherwise, it simulates the signature on M with $\beta^*(B)$ and calculates $Y^* = \mu(bX) = bY$ and signature output is as follows:

$$\epsilon_1^* = sI_U, \epsilon_2^* = \left(\frac{H(M)+f}{N-1}\right) bX$$

$$\epsilon_3^* = \prod_{k\in\{1,2,..,N\}} s(s(bX), y_k), \epsilon_4^* = fs(bX)$$

$$\epsilon_5^* = sK_U^*, \epsilon_6^* = f\epsilon_5^*, \epsilon_7^* = \{d_v(0)T_{k,m}\}_{p_{k,m}\in A_{\beta^*}}$$

where

$$K_U^* = s_{k,m}Y^*$$

Final signature returns by the simulator to forger is

$$\epsilon^* = \langle \epsilon_1^*, \epsilon_2^*, \epsilon_3^*, \epsilon_4^*, \epsilon_5^*, \epsilon_6^*, \epsilon_7^* \rangle$$

- **Forgery:** Once the forger generates the signature ϵ^* for the message M^* with $\beta^*(B^*)$, he/she makes it available to public. If this signature is verified successfully then it means that the forger successfully won the game. Let t_S and t_B denote the time that is consumed during the scalar multiplication over the elliptic curve group and bilinear pairing respectively. If attacker successfully breaks this algorithm (MA-ABS) in time t, then it is easy to calculate the time t' taken by the new algorithm to solve CBDH problem as $t' \approx t + q_H(t_S + t_B) + d_X(3+2N)N(N-1)t_S + d_S(6t_s + Nt_B)$.
- **Privacy:** To ensure and protect the privacy of the signer that has a set of attributes B for access policy β, a valid signature is created using another set of attributes B' that satisfies the same access policy β. Signature will not disclose the subset of attributes used to sign the message. This is because, any subset of k elements from a given set of attributes is used to sign the message and produce a valid signature. To ensure the privacy of signer, first the challenger runs the *Setup* and *Authority Setup* steps to generate the public parameters, public key K_k and the private key l_k of the authority for forger. The forger then outputs $\langle \beta, B_0, B_1, M^* \rangle$ after querying the private key oracle and signing oracle where $B_0 \supseteq B$ and $B_1 \supseteq B$. Forger also request to challenger to endorse the message M^* with respect to β using B_0 or B_1. The challenger now generates a challenge signature. As $B_0 \cap B = B$ and $B_1 \cap B = B$, the challenger selects randomly a bit $b \in \{0,1\}$ and outputs a signature ϵ^* with the private key lB_b over the set of attribute B_B. Using the Lagrange interpolation, it is observed that ϵ^* can be generated using l_{B_b} or $l_{B_{1-b}}$. Hence the forger is not able to steal the signer attributes.

6.2. Performance Analysis

In this section, the performance of the proposed algorithm is compared with the existing attribute-based signature approaches. To calculate the time consumption, we consider bilinear pairing operation, scalar multiplication operation, and exponentiation operation without considering the hash functions. T_X, T_S and T_e are the time consumed by these operations respectively. The results in Table 1 shows the effectiveness of the proposed algorithm in distributed environment with multiple authorities. The computational const in *SignVerify* operation increases linear with the number of authorities and attributes. More specifically, the computational cost in *sign* operation is $(6+t)T_S + NT_X$ while the computational cost of the *verify* operation is $T_S + T_e + (2tN+1)T_X$. The size of the signature depends on the number of attributes and defines the cost of communication overhead. The signature size in proposed algorithm is $(6+t)|G|$.

Table 1. Comparison of attribute-based signature schemes.

Properties	[5]	[16]	[18]	[19]	**[ours]**
Cost of signing	$(lt+t+3)T_e$	$(7l+15)T_e$	$(6+2l+lt)T_e$	$(l+t+16)T_e+3T_p$	$(6+t)T_s+NT_p$
Cost of Verifying	$(2lt+1)T_e+(l+2+(t-1)(l+1))T_p$	$(l+1)T_e+(L+2)T_p$	$(l+2)T_e+(l+4)T_p$	$(2lt+t+12)T_e+(l+7+(t-1)(l+1))T_p$	$T_s+T_e+(2tN+1)T_p$
Size of signature	$(l+t+2)\|G\|$	$(7l+11)\|G\|$	$(l+t+2)\|G\|$	$(l+t+11)\|G\|$	$(6+t)\|G\|$
Predicates	Monotone	Non-Monotone	Monotone	Monotone	Monotone
Multi Authority	Extensible	Extensible	No	No	Yes
Security Model	Generic Group	Standard	Standard	Generic Group	Random
Security Assumption	CR Hash	DLIN/CR Hash	CDH	CR Hash	CBDH
Privacy	Perfect	Perfect	Perfect	Imperfect	Perfect
Resisting Collusion Attack	No	No	No	No	Yes

Note: l shows the number of attributes, t shows the user attributes, CR Hash is Collision Resistance hash function, DLIN is decisional linear problem and CDH is Computational Diffie–Hellman.

7. Conclusions

To protect the privacy of microgrid transactions and losses using blockchain technology, the multiple authority attribute-based signature approach is introduced, which satisfies and meets the distributed requirement of microgrid as well as ensure the anonymity of information. The authorities agree on *PRF* seed and generates the private key for microgrid. If $N-1$ authorities collude, they cannot reveal the private key of microgrid. The security proof of the proposed protocol is discussed using CBDH assumption of unforgeability and privacy. Finally, the comparative analysis showed the effectiveness of the proposed protocol.

Author Contributions: All authors have equally contributed to this article.

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

References

1. Piagi, P.; Lasseter, R.H. Autonomous control of microgrids. In Proceedings of the 2006 IEEE Power Engineering Society General Meeting, Montreal, QC, Canada, 18–22 June 2006; p. 8.
2. Prodanovic, M.; Green, T.C. High-Quality Power Generation Through Distributed Control of a Power Park Microgrid. *IEEE Trans. Ind. Electron.* **2006**, *53*, 1471–1482. [CrossRef]
3. Anand, S.; Fernandes, B.G.; Guerrero, J. Distributed Control to Ensure Proportional Load Sharing and Improve Voltage Regulation in Low-Voltage DC Microgrids. *IEEE Trans. Power Electron.* **2013**, *28*, 1900–1913. [CrossRef]
4. Vandebron Energie B.V. Available online: https://vandebron.nl/ (accessed on 20 February 2018).
5. Maji, H.K.; Prabhakaran, M.; Rosulek, M. Attribute-Based Signatures: Achieving Attribute-Privacy and Collusion-Resistance. *IACR Cryptol. ePrint Arch.* **2008**, *2008*, 328.
6. Yan, Y.; Qian, Y.; Sharif, H.; Tipper, D. A Survey on Cyber Security for Smart Grid Communications. *IEEE Commun. Surv. Tutor.* **2012**, *14*, 998–1010. [CrossRef]
7. Perrig, A.; Canetti, R.; Tygar, J.D.; Song, D. Efficient authentication and signing of multicast streams over lossy channels. In Proceedings of the 2000 IEEE Symposium on Security and Privacy, S P 2000, Berkeley, CA, USA, 14–17 May 2000; pp. 56–73.
8. Cairns, K.; Hauser, C.; Gamage, T. Flexible data authentication evaluated for the smart grid. In Proceedings of the 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm), Vancouver, BC, Canada, 21–24 October 2013; pp. 492–497.
9. Wang, Q.; Khurana, H.; Huang, Y.; Nahrstedt, K. Time Valid One-Time Signature for Time-Critical Multicast Data Authentication. In Proceedings of the IEEE INFOCOM 2009, Rio de Janeiro, Brazil, 19–25 April 2009; pp. 1233–1241.
10. Veitch, C.K.; Henry, J.M.; Richardson, B.T.; Hart, D.H. *Microgrid Cyber Security Reference Architecture*; Sandia National Laboratories: Albuquerque, NM, USA; Livermore, CA, USA, 2013.

11. Sahai, A.; Waters, B. Fuzzy Identity-based Encryption. In Proceedings of the 24th Annual International Conference on Theory and Applications of Cryptographic Techniques (EUROCRYPT'05), Aarhus, Denmark, 22–26 May 2005; Springer: Berlin/Heidelberg, Germany, 2005; pp. 457–473.
12. Khader, D. Attribute Based Group Signature with Revocation. IACR Cryptology ePrint Archive, 15 April 2008.
13. Maji, H.K.; Prabhakaran, M.; Rosulek, M. Attribute-Based Signatures. *Topics in Cryptology–CT-RSA 2011*; Kiayias, A., Ed.; Springer: Berlin/Heidelberg, Germany, 2011; pp. 376–392.
14. Li, J.; Au, M.H.; Susilo, W.; Xie, D.; Ren, K. Attribute-based Signature and Its Applications. In Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS '10), Beijing, China, 13–16 April 2010; ACM: New York, NY, USA, 2010; pp. 60–69.
15. Herranz, J.; Laguillaumie, F.; Libert, B.; Ràfols, C. Short Attribute-Based Signatures for Threshold Predicates. In *Topics in Cryptology–CT-RSA 2012*; Dunkelman, O., Ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 51–67.
16. Okamoto, T.; Takashima, K. Efficient Attribute-Based Signatures for Non-Monotone Predicates in the Standard Model. *IEEE Trans. Cloud Comput.* **2014**, *2*, 409–421. [CrossRef]
17. Chen, C.; Chen, J.; Lim, H.W.; Zhang, Z.; Feng, D.; Ling, S.; Wang, H. Fully Secure Attribute-Based Systems with Short Ciphertexts/Signatures and Threshold Access Structures. In *Topics in Cryptology–CT-RSA 2013*; Dawson, E., Ed.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 50–67.
18. Gu, K.; Jia, W.; Wang, G.; Wen, S. Efficient and Secure Attribute-based Signature for Monotone Predicates. *Acta Inf.* **2017**, *54*, 521–541. [CrossRef]
19. Cui, H.; Wang, G.; Deng, R.H.; Qin, B. Escrow free attribute-based signature with self-revealability. *Inf. Sci.* **2016**, *367–368*, 660–672. [CrossRef]
20. Goyal, V.; Pandey, O.; Sahai, A.; Waters, B. Attribute-based Encryption for Fine-grained Access Control of Encrypted Data. In Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS '06), Alexandria, VA, USA, 30 October–3 November 2006; ACM: New York, NY, USA, 2006; pp. 89–98.