# Blockchain Meets AMI: Towards Secure Advanced Metering Infrastructures

Zakaria Abou El Houda [1,2], Abdelhakim Hafid [1] and Lyes Khoukhi [2]

[1] Department of Computer Science and Operational Research, University of Montreal, Canada
[2] ICD/ERA, University of Technology of Troyes, France
zakaria.abou.el.houda@umontreal.ca, ahafid@iro.umontreal.ca
{zakaria.abou_el_houda, lyes.khoukhi}@utt.fr

*Abstract*— **Smart grids (SGs) and advanced metering infrastructures (AMIs) are considered as the new evolution of classical electrical grids. The recent emergence of smart meters is paving the way for the proliferation of smart grids, where billions of smart meters are interconnected to provide novel pervasive services (e.g., real time pricing application and real time energy consumption), and automate diagnostic and daily energy metering (i.e., gas, electric) tasks (e.g., billing, monitoring, planning and predicting of energy usage). The recent explosion in the number of insecure smart meters is changing the view towards SG from enabler of smart homes into a powerful amplifying tool that creates new vectors for cyberattacks (i.e., smart-homes DDoS attacks) at large scale. This motivated us to design a new flexible, secure, efficient and trustworthy access control scheme based on blockchain and smart contract. Although access control exists in AMI, it is based on a centralized model (i.e., router/gateway, firewall) which introduces a bottleneck (i.e., single point of failure) and causes the collapse of the system. In this paper, we propose a new decentralized-based access control architecture for SG based on blockchain; it uses smart contracts (i.e., Ethereum's smart contracts) in order to manage permissions in a fully distributed and trustworthy manner. The architecture is implemented, tested and deployed on the Ethereum official test network Ropsten [1]. The results confirm that the proposed blockchain based access control scheme achieves security, flexibility, efficiency, and cost effectiveness making it a promising solution to mitigate DDoS attacks in SGs.**

**Keywords— SG; AMI; Smart contract; Blockchain;**

## I.INTRODUCTION

SG is playing a vital role in sophisticated society to improve reliability and energy efficiency since it incorporates the strength of the information communication technology (ICT) in the energy power distribution, and the automation of billing, monitoring, planning and prediction of energy usage. The main benefit of using SG is the ability to control the two-way flow of information between power providers (i.e., power plants) and end users. For example, in classical electrical grids, power providers generate electricity and, through a transmission network, transmit the required energy to end users. However, in SG by using smart meters, end users can provide electricity to the power grid (e.g., using home solar panels) and, thus, create a distributed and automated way to provide new functionalities (e.g., grid resilience, integration of renewable energy and real time control). SGs address critical problems of classical electrical grids; however, they face numerous security threats (e.g., real-time energy consumption data can be tampered with for electricity theft or false data injection) [2]). While SG brings enormous benefits to end-users, it is accompanied by numerous risks related to security and privacy [3]. Illegitimate users can easily exploit the vulnerabilities of smart meters, take control of them remotely, and then create amplifying tools (e.g., Botnet as-a-service) to earn money. A critical problem of SG is the cascading risk; indeed, if SG is damaged, other critical infrastructures (e.g., transport and healthcare) will be impacted leading to a large-scale blackout. A recent study [4] did show that approximately 80% of electrical enterprises were victims of large-scale DDoS attacks. Thus, it becomes critical to design schemes to secure SG especially against DDoS attacks. This motivated us to develop a secure scheme that provides a fully distributed and trustworthy access control to smart meters; the objective is to protect these meters from any type of unauthorized access that may compromise them to generate large-scale DDoS attacks. In this type of attacks, attackers (i.e., bot-masters) exploit the vulnerabilities in the underlying ICT infrastructure of smart grid AMI network (i.e., smart meters) using scanning techniques in commonly used industrial protocols (i.e., Modbus [5] and DNP3 [6]). These vulnerabilities allow attackers to get access to smart meters and conduct large-scale DDoS attacks. Fig. 1 shows the steps involved in attacking smart meters in the context of smart homes. In step (1), the bot-master remotely triggers the malware inside the home network (e.g., using mobile phone applications. In 2016, a study did show that 90% of mobile applications had unsecured binary code [7]). Then, the malware scouts for smart meters using some standards protocols (e.g., Universal Plug and Play (UPnP) based on Simple Service Discovery Protocol (SSDP)). Once done, the malware reports those unsecured smart meters to the remote bot-master. In step (2), the bot-master instructs the malware to update port-mapping in the home gateway so it can gain access to the smart meter. As depicted in step (3), once the access rules update has taken effect, the bot-master has direct access to the smart meters and can now attack those meters to exploit their vulnerabilities and use them as amplifying platforms. Routers/gateways used in smart homes today, by virtue of their in-built NAT and firewall capability, prevent outside entities from launching attacks on smart meters inside the home network. However, as shown in steps described above, this centralized model (i.e., router/gateway, firewall) introduces a bottleneck (i.e., single point of failure) and does not prevent outside attackers from penetrating smart

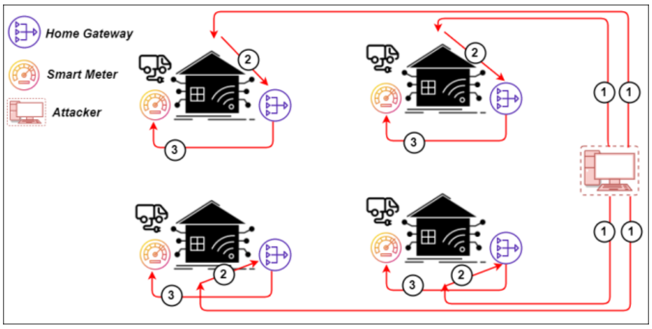


Fig. 1. DDoS in AMI network

home devices and using them as tools to conduct large scale DDoS. For example, in October 2016, Mirai botnet controlled a huge number of smart home devices (i.e., closed-circuit television cameras (CCTV)) to launch the largest ever DDoS attack, exceeding a rate of 1 Tbit/s [8]. This motivates the design of a new decentralized access control scheme that aims to prevent smart home network from any unauthorized access even if attackers remotely trigger the malware inside the home network. The features of immutability, security and transparency of blockchain did convince us to implement our proposed decentralized access control model using this new technology. Indeed, Blockchain has proven its success, in multiple domains (e.g., IoT [9], DDoS collaboration [10], and financial field [11]), in achieving high level of transparency and security. One such domain's application is the smart grid, due to its decentralized architecture and its resource-constrained smart meters. We believe that Blockchain can play a vital role to secure the underlying infrastructure of SG.

Existing access control schemes are mainly based on well-known access control schemes (i.e., role-based access control scheme (RBAC), attribute-based access control scheme (ABAC), and Capability-based access control scheme (CapBAC)). These schemes suffer from low flexibility and are centralized. The centralized approach, by its nature, introduces a bottleneck and causes single point of failure (see Fig. 1). The new emerging technologies, such as blockchain and smart contract, open new opportunities for efficient, flexible, transparent, trustworthy and immutable access control. Indeed, blockchain has been investigated to provide a decentralized access control in trustless network environments. This is the first work to examine access control, using blockchain, in AMI architecture to provide secure SG.

This paper designs and implements a blockchain-based access control contract (ACC) scheme; it provides efficient, flexible, transparent, trustworthy and immutable access control. The implementation of our scheme is deployed on Ethereum official test network Ropsten, an open blockchain platform. The main contributions of our paper can be summarized as follows:

- We design a decentralized, efficient and trustworthy access control scheme for SG based on blockchain technology using smart contract.
- We implement a smart contract-based scheme that makes use of Ethereum's smart contract technology, to realize a secure, decentralized, trustworthy, flexible and low-cost access control in SG.
- We evaluate the performance of our scheme in terms of security, flexibility, efficiency, and cost effectiveness. The experiments results show that our blockchain-based ACC scheme can effectively be used to ensure a secure, trustworthy, flexible, efficient and low-cost access control management scheme.

The reminder of this paper is organized as follows. Section II presents related works. Section III describes our system model. Section IV presents the implementation of our ACC scheme. Section V evaluates the proposed ACC. Finally, Section VI concludes the paper.

## II.Related work

Recently, the security and privacy of SG have caught the attention of both industrials and academia. Several schemes have been proposed to deal with the critical access control problem. In the following, we discuss some of the most prominent schemes as well as their security issues. Several contributions [12-13] have demonstrated that smart home devices lack security safeguards by exploiting their vulnerabilities to conduct large scale and devastating DDoS attacks [14]. Traditional access control schemes are mainly based on well-known access control schemes (i.e., RBAC [15], ABAC [16], CapBAC [17]). In RBAC, roles (e.g., administrator) of a subject (i.e., entity that requests the access to the object's resources) are used to restrict access to the object's resources. Each role is associated with several access control rights (e.g., execute, read, write) and is assigned to a subject. The roles refer to the levels of access that subjects have to the object's resources. In highly dynamic environment (i.e., Smart Grid environment), role lacks flexibility and efficiency in access control management due to the huge number of smart devices as well as their dynamic behavior. ABAC model [16] aims to solve this problem by associating access rules with the subject or/and object attributes; however, the number of access rules rapidly increases with the number of smart devices; furthermore, the validation of access rules is mainly performed by a centralized entity causing a single point of failure. To alleviate this issue, CapBAC access control models [17] have been proposed recently where the validation of access rules is performed by the resource owners themselves (i.e., requested object) rather than a centralized entity. However, due to the constrained nature of smart home devices, the validation of access rules can be easily compromised. In [18], Sankar et al. proposed a centralized access control scheme for SG; the access rules are validated through a regional transmission organization (RTO). This scheme requires RTO to be online to validate the access rules; this may cause the collapse of the system since RTO becomes a single point of failure. In [19], Bezawada integrated ABAC scheme into smart home network in order to enforce the subject (i.e., requestor) and the objects (i.e., requested IoT devices) attributes as access rules. However, this scheme [19] faces numerous problems related to scalability and flexibility; more specifically, the number of access rules grows dramatically with the number of end devices, due to the highly dynamic SG environment. Moreover, the centralized validation entity introduces a bottleneck (i.e., single point of failure) and may cause the collapse of the system. In [20], Sun et al. proposed an identity-based encryption (ABE) access control scheme in SG to alleviate the computational overhead of intelligent terminals. However, this scheme is vulnerable to the man in the middle attacks. To alleviate this, a secure public-key infrastructure (PKI) can be used; nonetheless, it is costly to maintain and setup. In [21], Sciancalepore et al. proposed the use of OAuth protocol to implement a CapBAC scheme to remotely grant tokens to end users in IoT environment. In this scheme [21], a centralized entity (i.e., IoT gateway) manages access to devices based on the requestor's access token. Further the single point failure problem, this scheme does not scale with the number of devices making it unsuitable for highly dynamic environment. To overcome these issues, Dorri et al. [22] proposed to use blockchain for access control in the context of IoT. They require that each smart home miner stores access control rules related to the home's access request, i.e., incoming, and outgoing access requests. However, they have eliminated the critical proof-of-work consensus protocol in their proposed blockchain scheme, resulting in an untrustworthy access control scheme. In [23], Zhang et al. proposed a smart contract-based access control for IoT environment using blockchain; however, the authenticity of requestors, flexibility, scalability and efficiency remain challenging issues in this blockchain based scheme [23].

To address the weaknesses of these existing solutions [15-23], we propose an efficient, flexible, transparent, trustworthy and secure access control scheme based on blockchain using smart contracts. Our scheme allows for an efficient and trustworthy access control model; in addition, it ensures the authenticity, the flexibility, removes the need for a central entity to maintain the access and enforces permissions. Our scheme provides the resources owner (i.e., contract owner) with the flexibility for (adding/removing) access control policies; moreover, this will be run and verified by all participants (i.e., miners) in the blockchain system (i.e., Ethereum blockchain), which ensures the trustworthiness of the access control.

## III.System Model

### A. Sytem Overview

In this section, we present an overview of our proposed access control scheme. More specifically, we investigate how this scheme can effectively ensures fully decentralized and trustworthy access control for SG.

As illustrated in Fig. 2, this paper considers access control in AMI environment where a large number of smart meters, smart home devices, electric vehicles (EVs), and smart home gateways are interconnected through a peer-to-peer (P2P) network and accessible to outside's home network via smart home gateways. In this paper, we consider different use cases for access control management, i.e., Utility server (i.e., power provider) can read smart meters' data, a user device can use services related to the smart meters (e.g., billing, monitoring, planning and predicting of energy usage).

We propose an access control contract scheme based on Ethereum's smart contract to achieve fully decentralized and trustworthy access control for SG. Each ACC is created and deployed by the resources owners to manage their resources without relying on any third party/central entity. Each ACC provides a mapping of one-to-many access control for a subject-object pair (see Fig. 3), and only authenticated subject can be authorized to get access to the object's resources during a specific time and prevent unauthorized access to the users resources and/or data generated by the smart meters, that consists of sensitive data (e.g., daily usage electricity, this can be used by attackers to know the period of inactivity of the home and get access to it). A smart meter must be able to accept request from authorized users (e.g., smart meter owners, authenticated power grid provide) and deny unauthorized users from accessing/retrieving its data. To realize this, we adopt a security model based on access control matrix as illustrated in Fig. 3. In our scheme, we define a set of subjects $S$, which are power providers or/and smart home users that wish to access the resources of smart meters (i.e., smart meters data), and a set of objects $O$, which are smart meters that collect sensitive data. Each Subject $s \in S$ has an Externally Owned Account (EOA) address (i.e., public key) and some other information (e.g., subject notes and the timestamp of when the subject was added) and can get access to a set of Object's resources $Ro$; each $ro \in \mathrm{Ro}$ is associated with a set of policies $p \in \mathrm{P}$. For each subject $s$, Object $o$ and resource $ro$, mapping $M(s, o, ro) \subseteq \mathrm{p}$ is defined to specify the access rights. Each entry in P specifies a subject, an object, an object resource, the action performed by the subject on the object resources and the permission (e.g., allow and deny) on this action. Each row in the policy items, P, corresponds to 4-tuple (subject, object, object resource, action) and is defined as follows:

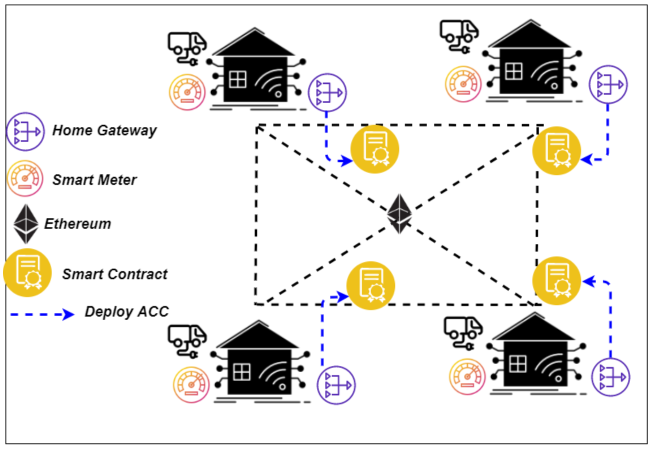
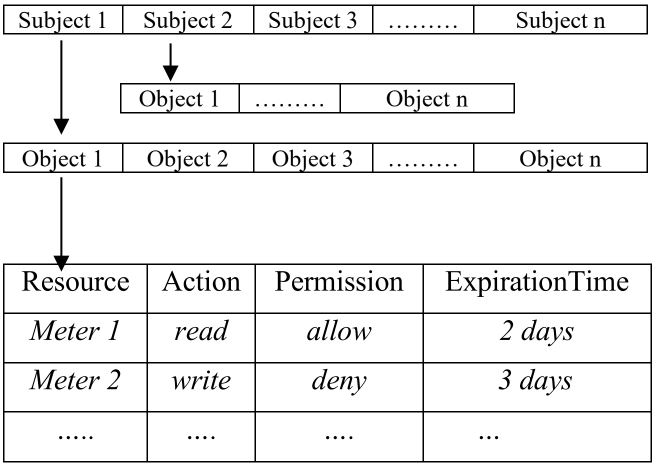


Fig. 2. The considered decentralized access control system

- Subject: the smart meter.
- Resource: The resources of the smart meter, such as the measurement of energy consumption data, etc.
- Action: The specified action performed by the subject (e.g., Utility server) on the object resource, such as write, read and execute.
- Permission: The permission on the action performed by the subject on an object resource, such as allow, deny.
- Expiration Time: The validity period of each permission.

### B. System interaction

The use case illustration of the access control in SG is shown in Fig. 4, where the utility server (i.e., subject) wants to access the measurement of energy consumption data of a smart meter (i.e., object). To realize this, in step (1), the utility server (subject) sends an access request transaction to blockchain through the Application Binary Interface (ABI) of the deployed ACC. The transaction includes the required information for access control (e.g., resource, action to perform on the specified resource). Once the transaction is



| Resource | Action | Permission | ExpirationTime |
|---|---|---|---|
| *Meter 1* | *read* | *allow* | *2 days* |
| *Meter 2* | *write* | *deny* | *3 days* |
| ….. | …. | …. | … |

Fig. 3. Smart contract system.

confirmed, the subject and the object get the access result from the smart contract deployed in the blockchain. Then, in step (3) and step (4), the subject, if authorized, can access the specific object resource.

### *C. System's smart contract*

We consider a resource owner (i.e., contract owner) that would like to manage the access control process for his resources. First, it creates the access control smart contract and deploys it in the blockchain (i.e., Ethereum's blockchain [24]). The use of blockchain in the access control process allows for transparency, trustworthiness and security while maintaining the process of access control "pseudonymity". First, to initialize the smart contract, the resource owner (r) generates a keypair of private key and the corresponding EOA. This keypair will be used to create the smart contract (SC) and execute its functions. Then, the resource owner adds, via SC, the subjects. It includes the subject's address and some other information (e.g., subject notes, the timestamp of when the subject was added). SC allows the resource owner to: (1) add subjects to the access control smart contract; (2) easily manage the access control process in a transparent, flexible and trustworthy manner; (3) easily remove subjects from the access control contract if needed; (4) manage access rules to its resources in a secure and efficient manner without relying on any third party/central entity; and (5) update/remove access policies from the access control smart contract when needed. The ACC provides following functions, where s denotes an instance of the Subject and r an instance of resource:

.

AddSubject (s.EOA, s.Infos): Only the resource owner of the ACC can invoke this function to add subjects; it takes as input the Externally Owned Account (s.EOA) of the subject and his information (s.Infos) and adds the Subject to the ACC. This happens if the ACC is activated and the subject's identity is authenticated.

RemoveSubject (s.EOA): Only the owner of the ACC can invoke this function to remove subjects; it takes as input the s.EOA and removes the subject from the ACC. This happens if the ACC is activated and the subject's identity has been registred before.

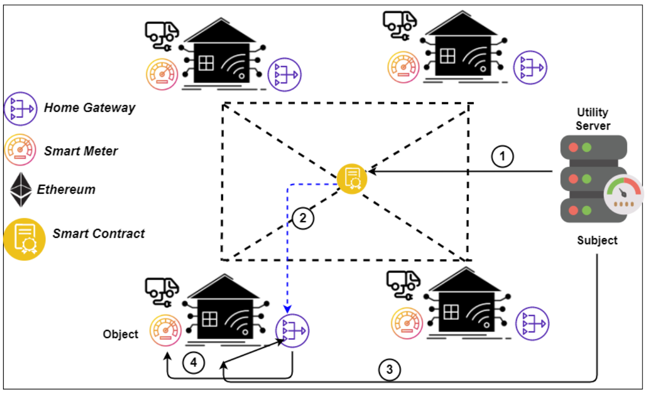


Fig. 4. Illustration of access control in SG environment.

ChangeStatus(bool status): Only the owner of the ACC can invoke this function to either activate/deactivate the ACC.

AddPolicy (s.EOA, r, action, permission, expiration_time): Only the owner of the ACC can invoke this function to add a new access policy to the policy list. This happens if the ACC is activated and the subject's identity is authenticated.

UpdatePolicy (s.EOA, r, action, permission, expiration_time): Only the owner of the ACC can invoke this function to update the policy. This happens if the ACC is activated, the access policy has been added to the ACC, and the subject's identity is authenticated.

DeletePolicy (s.EOA, r, action, permission, expiration_time): Only the owner of the ACC can invoke this function to remove a specific policy between object and subject pair. This happens if the ACC is activated, the access exists in the ACC, and the subject's identity is authenticated.

CheckAccess (s.EOA, r, action, permission, expiration_time): either the owner of the ACC or the subject that has already been added in the SC can invoke this function to check the access control and returns the access result. This happens if the ACC is activated and the subject's identity is authenticated

## IV. IMPLEMENTATION

We implemented and tested ACC using a private (i.e., Ganache simulator [24]) and a public blockchain (i.e., Ethereum official test network Ropsten). Once ACC is deployed by the owner, it can be self-executed without any third party/central entity intervention. To manage the deployment process of ACC, we have used a decentralized development framework (i.e., truffle framework [25]). First, we implemented ACC using the high-level language programming solidity [26]. Then, we compiled ACC into EVM byte code; once ACC gets compiled, it generates the EVM byte code and ABI. Then, we deployed ACC to the blockchain network (i.e., Ethereum's blockchain). Initially, we deployed ACC on the Ganache's private blockchain; then, we used the Ethereum official test network Ropsten. Once ACC is deployed, it can be invoked using its address and ABI.

TABLE. 1. Transaction details of the ACC

| Details of ACC Creation Transaction in Ropsten test network | |
|---|---|
| TxHash | 0x0c941b91338c299a472156cb0271bf03fb8a831fc7979a9e208253c2a9588eb2 |
| Block Height | 6658517(5775 Block Confirmations) |
| Timestamp | Oct-27-2019 09:52:24 PM +UTC |
| From | 0xa70836a9a115f774cb848134d0f8b2473e27d181 |
| To | 0x8240b68ad6fdd90cba3aaa8563bb3a266314049c |
| Gas Used by Tx | 2400519 |

Table. 1 shows the transaction details of ACC. The details of such a transaction can be found using Ropsten Etherscan [27].

## V. Evaluation

In this section, we evaluate our ACC scheme in terms of flexibility, security and cost effectiveness; then, we compare our scheme to other prominent works.

### 1. Flexibility

ACC scheme provides two levels of flexibility: (1) it provides the resources owner (i.e., ACC owner) with flexibility for (adding/deleting) subjects (to/from) the access control system using (addSubject ()/removeSubject ()) functions and to easily manage (add/update /remove) access policies (to/from) using (Addpolicy()/Updatepolicy()/Deletepolicy()) functions; (2) ACC provides resources owner (i.e., ACC owner) with the flexibility for joining or/and leaving the ACC. To join the system, the resources owner deploys ACC. To leave the ACC scheme, the resources owner easily deactivate ACC using ChangeStatus() function. Since all these updates are verified by anyone in the blockchain network (i.e., Ethereum's blockchain), it ensures the trustworthiness of the access control system.

### 2. Security

Only authorized subjects that are authenticated can access to specific owner resources. ACC achieves this by using modifiers. For example, the modifier "OnlyOwner" allows only the resources owner of ACC to execute addSubject(), removeSubject(), Addpolicy (), Deletepolicy() and changeStatus() functions. If a malicious user tries to execute these functions to grant himself acces to a specifc resource, the execution will fail and no action will be recorded on the blockchain. The same restriction rule applies for the "OnlySubjects" modifier for the execution of CheckaccessControl () function; only subjects (and also the contract owner) can check if they have access to objects resources.

### 3. Low Cost

In this section, we estimate the cost of ACC creation as well as the execution of its functions. When conducted the experiment, the gasPrice was set to $1Gwei$, where $1Gwei = 10^9 wei = 10^{-9} ether$, and , and 1 ether was equal to $184,66USD$. Table. 2 shows the cost of the execution of different functions implement by ACC. The highest cost corresponds to the creation ACC at 0.443 USD. But, the deployment of ACC is only performed once to setup the ACC scheme. All functions, provided by ACC, have low costs. Thus, we can consider that the ACC deployment in SG is cost effective.

TABLE. 2. ACC creation and functions costs

| Function | Gas Used | Actual Cost(ether) | USD |
|---|---|---|---|
| Create ACC | 2400519 | 0.002400519 | 0.443 |
| AddSubject () | 37735 | 0.000037735 | 0.0069 |
| RemoveSubject () | 145038 | 0.000145038 | 0.026 |
| AddPolicy () | 96907 | 0.000096907 | 0.017 |
| UpdatePolicy () | 40400 | 0.0000404 | 0.0074 |
| DeletePolicy () | 27157 | 0.000027157 | 0.005 |
| CheckAccess() | 35854 | 0.000035854 | 0.006 |

### 4. Analysis

First, our ACC scheme preserves "pseudonymity" and does not suffer from single point of failure problem since it runs on Ethereum's blockchain; it allows for an efficient and trustworthy access control scheme; in addition, it ensures the authenticity, the flexibility, removes the need for a central entity to maintain the access and enforces permissions. Our scheme provides the resources owner with the flexibility for (adding/removing) access control policies; moreover, all executions are verified by all participants (i.e., miners) in the blockchain system (i.e., Ethereum blockchain), which ensures the trustworthiness of the proposed access control.

## VI. Conclusion

In this paper, we proposed an efficient, flexible, transparent, trustworthy and secure access control scheme based on blockchain using Ethereum's smart contracts. The ACC scheme has been implemented, tested, evaluated and deployed on Ethereum official test network Ropsten; the appendix shows the ACC address in Ropsten. The results confirm that our blockchain based access control scheme achieves security, flexibility, efficiency, and cost effectiveness making it a promising solution to mitigate DDoS attacks in SG.

## *APPENDIX*

ACC was deployed on the Ropsten Testnet of Ethereum with the following address:

Organization Owner of account address: 0xa70836a9a115f774cb848134d0f8b2473e27d1811.
ACC address: 0x8240b68ad6fdd90cba3aaa8563bb3a266314049c
Using this address, the transactions can be seen at: https://ropsten.etherscan.io/.

## *REFERENCES*

[1] Etherscan. The Ethereum Block Explorer: ROPSTEN TESTNET. Accessed: October. 1, 2019. [Online]. Available: https://ropsten.etherscan.io.

[2] R. Deng, and H. Liang, "False Data Injection Attacks With Limited Susceptance Information and New Countermeasures in Smart Grid," in *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1619-1628, March 2019.

[3] M. I. Oozeer, and S. Haykin, "Cognitive Risk Control for Mitigating Cyber-Attack in Smart Grid," in *IEEE Access*, vol. 7, pp. 125806-125826, 2019.

[4] Stewart Baker, Steptoe Johnson, and CSIS, "Critical Industries Confront Cyberattacks", Accessed: October. 1, 2019. [Online]. Available: https://csrc.nist.gov/CSRC/media/Events/ISPAB-JULY-2011-MEETING/documents/Jul14_CIP-CSIS-2011-ISPAB.pdf.

[5] Modbus. Accessed: October. 1, 2019. [Online]. Available: http://www.modbus.org/.

[6] DNP3. Accessed: October. 1, 2019. [Online]. Available: https://www.ixiacom.com/company/blog/scada-distributed-network-protocol-dnp3.

[7] Accessed: October. 1, 2019. [Online]. Available: https://www.techrepublic.com/article/report-90-of-mobile-health-and-finance-apps-vulnerable-to-critical-security-risks/.

[8] B. Schneier. Lessons From the Dyn DDoS Attack. Accessed: October. 1, 2019. [Online]. Available: https://www.schneier.com/blog/archives/2016/11/lessons_from_th_5.html.

[9] Z. A. El Houda, A. Hafid, and L. Khoukhi, "Co-IoT—A collaborative DDoS mitigation scheme in IoT environment based on blockchain using SDN, " in Proc. IEEE Global Commun. Conf. (GLOBECOM), Dec. 2019.

[10] Z. Abou El Houda, A. S. Hafid, and L. Khoukhi, "Cochain-SC: An Intra- and Inter-Domain Ddos Mitigation Scheme Based on Blockchain Using SDN and Smart Contract," in IEEE Access, vol. 7, pp. 98893-98907, 2019.

[11] H. Yao, T. Mai, J. Wang, Z. Ji, C. Jiang, and Y. Qian, "Resource Trading in Blockchain-Based Industrial Internet of Things," in IEEE Transactions on Industrial Informatics, vol. 15, no. 6, pp. 3602-3609, June 2019.

[12] S. Notra, M. Siddiqi, H. H. Gharakheili, V. Sivaraman, and R. Boreli, "An experimental study of security and privacy risks with emerging household appliances," in Communications and Network Security (CNS), pp. 79–84, 2014.

[13] V. Sivaraman, D. Chan, D. Earl, and R. Boreli, "Smart-phones attacking smart-homes," in Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks. ACM, pp. 195–200, 2016.

[14] Z. A. El Houda, L. Khoukhi, and A. Hafid, "ChainSecure - A Scalable and Proactive Solution for Protecting Blockchain Applications Using SDN," in 2018 IEEE Global Communications Conference (GLOBECOM), Abu Dhabi, United Arab Emirates, pp. 1-6, 2018.

[15] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role based access control models, " Computer, vol. 29, no. 2, pp. 38–47, Feb. 1996.

[16] V. C. Hu, D. R. Kuhn, and D. F. Ferraiolo, "Attribute-based access control, " Computer, vol. 48, no. 2, pp. 85–88, Feb. 2015.

[17] R. S. Sandhu, and P. Samarati, "Access control: Principle and practice, " in IEEE Commun. Mag., vol. 32, no. 9, pp. 40–48, Sep. 1994.

[18] L. Sankar, S. Kar, R. Tandon, and H. V. Poor, "Competitive privacy in the smart grid: An information-theoretic approach," in 2011 IEEE International Conference on Smart Grid Communications (SmartGridComm), Brussels, pp. 220-225, 2011.

[19] Bezawada. B., Haefner. K., and Ray. Indrakshi "Securing Home IoT Environments with Attribute-Based Access Control," in proceedings of the Third ACM Workshop on Attribute-Based Access Control. ACM, p. 43-53, 2018.

[20] Z.W. Sun, and R.G. Zhang: "Access Control for communication Network of Smart Distribution Grid," in Power System Protection and Control, vol. 21, no. 38, pp. 118–121, 2010.

[21] Sciancalepore. S., Piro. G., Caldarola. D., Boggia. G., and Bianchi. G. "OAuth-IoT: An access control framework for the Internet of Things based on open standards," in 2017 IEEE Symposium on Computers and Communications (ISCC). IEEE, p. 676-681, 2017.

[22] A. Dorri, S. S. Kanhere, R. Jurdak, and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home, " In Proc. IEEE Int. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops), pp. 618–623, Mar. 2017.

[23] Zhang, S. Kasahara, Y. Shen, X. Jiang and J. Wan, "Smart Contract-Based Access Control for the Internet of Things," in IEEE Internet of Things Journal, vol. 6, no. 2, pp. 1594-1605, April 2019.

[24] G. Wood, ‘‘Ethereum: A secure decentralised generalised transaction ledger,’’ Yellow Paper. Accessed: October. 1, 2019. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf.

[25] Ganache. Accessed: October. 1, 2019. [Online]. Available: https://truffleframework.com/docs/ganache/overview.

[26] Truffle. Accessed: October. 1, 2019. [Online]. Available: https://truffleframework.com/.

[27] “Solidity”, Accessed: October. 1, 2019. [Online]. Available: https://solidity.readthedocs.io/en/develop/.

[28] Ropsten. Accessed: October. 1, 2019. [Online]. Available: https://ropsten.etherscan.io/.