Authentication and Authorization Scheme for Various User-Roles and Devices in Smart Grid

Neetesh Saxena, *Member, IEEE,* Bong Jun Choi, *Member, IEEE,* and Rongxing Lu, *Senior Member, IEEE*

Abstract—The smart grid, as the next generation of the power grid, is characterized by employing many different types of intelligent devices, such as intelligent electronic devices located at substations, smart meters positioned in the home area network, and outdoor field equipment deployed in the fields. Also, there are various users in the smart grid network, including customers, operators, maintenance personnel, and etc., who use these devices for various purposes. Therefore, a secure and efficient mutual authentication and authorization scheme is needed in the smart grid to prevent various insider and outsider attacks on many different devices. In this paper, we propose an authentication and authorization scheme for mitigating outsider and insider threats in the smart grid by verifying the user authorization and performing the user authentication together whenever a user accesses the devices. The proposed scheme computes each user-role dynamically using an attribute-based access control and verifies the identity of user together with the device. Security and performance analysis show that the proposed scheme resists various insider as well as outsider attacks, and is more efficient in terms of communication and computation costs in comparison with the existing schemes. The correctness of the proposed scheme is also proved using BAN-Logic and Proverif.

Index Terms—Smart Grid, Authentication, Authorization, Insider Threat, Security.

1 Introduction

The smart grid (*SG*) is a future opportunistic platform for ensuring electrical power transmission and distribution in a reliable, secure, and efficient manner. However, there are many evolving challenges in the smart grid security. Many *SG* security challenges have focused on protecting the system against various forms of external (outsider) cyber-attacks, including man-in-the-middle (*MITM*) attacks, intrusion-based attacks, malware-based attacks, denial of service (*DoS*) attacks, isolated attacks, and coordinated attacks [1]. Although known external attacks have been protected by well-known practices, a severe threat that modern critical infrastructures are newly facing is an insider threat or an insider attack. An insider threat is a user who has appropriate permissions to access required resources of the system and misuses its privileges. For smart grid having many integrated components and user groups, insider threats can influence the system massively. Insider threats can compromise many of the security goals of the system. They can compromise integrity by modifying data without authorization, availability by creating delays where low latency is required (4 milliseconds for protective relaying, few seconds for feeding data to supervisory control and data acquisition (*SCADA*), data transmission to substations and wide area monitoring messages, few minutes for monitoring equipment and market pricing information, and few hours for smart meter (*SM*) reading), confidentiality by exposing privacy of customer information and some part of electric market information, and accountability by avoiding liability and responsibility [2]. In fact, such insider threats can influence the security level of the *SG* system from low to high [3]. Hence, the countermeasures must address outsider as well as insider attacks.

In the advance metering infrastructure (*AMI*) system of the smart grid, the insider attacks can be performed at the customer endpoint as well as at the opposite end of the *AMI* system. Smart meters generally have some connectivity to the *AMI* head end, but this connectivity may be as slow as 1200 baud, or lower [3]. Consequently, the insider attacker can get access to modify meter readings and can view private information of the customer at the customer endpoint. Similarly, insider attacker may be able to access the electricity price information, network infrastructure information, and other information communicated by protocols. Some of these systems and protocols are energy management system (*EMS*), distributed network protocol (*DNP3*), inter-control center communications protocol (*ICCP*), and open smart grid protocol (*OSGP*). The *EMS* enables transmission of real-time information, such as grid's status, remote automation of grid functionalities, and etc. The *OSGP* provides reliable and efficient delivery of command and control information among various smart grid devices, including smart meters, control modules, and gateways. The *DNP3* is used by *SCADA* master stations (control centers (*CC*)), and the *ICCP* is used for inter-master station communications.

To provide authentication and authorization of users, the intelligent electronic devices (*IED*), smart meters, and outdoor field equipment (*OFE*) (*e.g.*, pole-top devices, such as recloser) use various local passwords. Several passwords allow different user-roles to access the device for various

- *N. Saxena and B. J. Choi are with the Computer Science Department, State University of New York Korea, S. Korea and also with the Department of Computer Science, Stony Brook University, USA.*
E-mail: mr.neetesh.saxena@ieee.org, bjchoi@sunykorea.ac.kr
- *R. Lu is with the School of Electrical and Electronic Engineering, Nangyang Technological University, Singapore.*
E-mail: rxlu@ntu.edu.sg

Manuscript received June XX, 20XX; revised XXX XX, 20XX.

purposes. These role-based passwords are generally shared among various users accessing devices, and due to a large number of devices, these passwords are often the same across all devices in the utility and seldom change [3]. User authentication and authorization are challenging due to the fact that these devices may be accessed physically on-site as well as remotely through wired (optical)/wireless (from different locations). There may also be various types of users, such as employees (*EMP*), vendor engineers (*VE*), maintenance personnel (*MP*), security officer (*SO*) etc., and various types of roles like auditor (read), employee (read-write), administrator (add-modify-delete), maintenance personnel (read), and etc. Since each role has a different password, the system can recognize the role of a user, but, it cannot identify which user (name or identity) is trying to access the system. Table 1 and Table 2 outline the access behavior of different devices and access permissions for different user-roles, respectively. In this paper, in order to better explain our problem, we consider a specific user-role, *i.e.*, maintenance personnel. Note that the maintenance personnel role can be extended to any other user-role, such as customer, vendor engineer, contractor, and etc.

According to the *NIST* report [4], one of the crucial challenges in the future smart grid is to authenticate and authorize users (such as maintenance personnel) whenever they access *IED*/*SM*/*OFE* located at substations/homes/fields in such a way that the resource access is specific to a user, the user specific authentication information is not shared among users, *e.g.*, identity and password, and the control of authentication and authorization is hierarchically managed by substations (*SS*) and the utility's central station (*C*). This ensures that only authenticated users can perform the assigned authorized actions onto the intended devices in a controlled and scalable manner. Therefore, mutual authentications between the user and the substation's server to access different devices with a specific user-role authorization is needed to mitigate insider attacks in the *SG* network.

Different from traditional networks, availability, integrity of information, and performance efficiency are critical requirements for infrastructure networks, such as smart grid that must operate continuously and satisfy system requirements under diverse operating conditions [5]. Furthermore, unlike the traditional power grid system, where a centralized and radial topology is used to generate and deliver power from one end to the other ends, the smart grid provides intelligent transmission and distribution automation in a decentralized environment [6]. Considering these differences, authentication and authorization schemes in the smart grid need to be specifically designed to achieve critical requirements, such as availability and integrity of information. Therefore, authenticating different entities and verifying their authorization are required simultaneously [7]. There are many schemes for the traditional networks [8], such as remote authentication dial-in user service (*RADIUS*) and *Diameter* protocols. However, These protocols do not fit well in the smart grid network. It provides centralized services and maintains a central database. However, the smart grid requires decentralized solutions, as a single-point-of-failure can massively affect the whole system. Furthermore, *RADIUS* has poor scalability and uses the user datagram protocol (*UDP*), which does not provide reliable data transfer. This is not suitable for the smart grid where the availability of information is critical for its operation. Furthermore, *Diameter* is an authentication, authorization, and accounting protocol that instead supports transmission control protocol (*TCP*). However, *Diameter* implements peer authentication between communication endpoints using pre-shared keys, which raises key management issues, and therefore not suitable for large systems like the smart grid.

TABLE 1: Access Behavior of Different Devices

No.	Device Name	Access Behavior
1	Intelligent electronic device	Physically/remotely by wired/wireless
2	Smart meter	Physically through optical port of *SM*, remotely through *AMI* infrastructure or *HAN* gateway
3	Outdoor field equipment	Remotely through wired/short-range radio (Bluetooth/802.11)

TABLE 2: Access Permissions for Different User Roles

No.	User Role	Department	Access Permission
1	Employee	Internal	Read-write (*RW*)
2	Auditor	Internal	Read-only (*R*)
3	Vendor engineer	External	Read-write (*RW*)
4	Customer	Customer	Read-only (*R*)
5	Contractor	External	Read-only (*R*)
6	Maintenance personnel	Maintenance	Read-only (*R*)
7	Administrator	Internal	Add-modify-delete (*AMD*)
8	Security officer	Internal	Read-write-modify (*RWM*)

Numerous challenges arise with the integration of cyber and physical systems along with human behavior and regulatory policy. Some challenges are quite similar to those of traditional networks, but involves more complex interactions [9]. The smart grid system has various user-roles, such as operator, vendor, engineer, administrator, etc., accessing many different types of devices in its network, such as smart meter, intelligent electronic device, etc., simultaneously. It also has more strict delay and execution time requirements. Whereas authentication and authorization are executed as two separate processes in the traditional network , executing them as one process is needed in the smart grid to handle frequent authentications among billions of devices and dynamic user-role authorizations for a large number of users. It can also reduce the total execution time, which can help to make the system more efficient to achieve its performance requirements.

In this paper, we propose a scheme that provides a mutual authentication between the user and the server, and a dynamic authorization for each user-role by computing the attribute-based hash value. The authorization is maintained so that each user can perform only those actions that are allowed under the access permissions granted to it. Our scheme provides a two-factor authentication. First, the authentication is performed by verifying the identity of each user as well as the device in a batch with the signature verification of each device at the server of the substation. Then, a one-time password (*OTP*) is sent to the user's mobile phone in order to verify the actual user who is accessing the device. A shared secret key is also generated between the user and the device for secure communication using the bilinear pairing technique. Specifically, the contributions of our scheme are identified as follows.

1. Simultaneously provides user authentication and authorization for different devices. Hence, it reduces the need of having separate systems for each type of device. It also works for different types of users interacting with many different types of devices in the smart grid.

2. Deals with both, the physical and the remote access of the devices by dynamically computing the role of a user. Any user (with a defined role) can interact with any device anywhere with authentication and authorization within the premises of the central station.

3. Does not use any shared passwords. If a shared password is compromised, it can reveal the confidential information to the adversary. Further, an adversary cannot retrieve any information based on linkability among different devices, as these devices store hash values corresponding to each user-role. For each device, these values are different for each user-role.

4. Improves the efficiency of the system by verifying the signatures of the devices in a batch at the authentication server of the substation. Also, computational overhead is reduced, as each user needs to generate only a public key pair $(Y_{1_{mp}}, Y_{2_{mp}})$, and hash H_1 each time it accesses a different device. Other parameters remain the same. This scheme provides two-factor authentication (one when substation's authentication server (AS_{ss}) verifies user identity from the message received by the *IED* and other when an *OTP* is sent to user's mobile phone) that ensures valid user identification even when temporary identity is compromised.

5. Defeats various outsider attacks as well as insider attacks, including man-in-the-middle attacks, replay attacks, impersonation attacks, integrity violations, attacks by customer, known key attacks, and repudiation attacks. It also prevents insider attacks where (i) a user accesses the device with the credential of his/her friend or family member without notifying him/her, and (ii) a rogue device is installed by a legitimate engineer in the network.

The rest of the paper is organized as follows: Section 2 discusses existing work on user authentication and authorization in the smart grid. Section 3 outlines the communication, system, and attack models. Our proposed scheme is described in Section 4. The security and performance analysis of our scheme are discussed in Section 5. Finally, Section 6 concludes the paper. In addition, Table 3 describes symbols used in this paper along with their sizes.

TABLE 3: Symbols And Abbreviations

Symbol	Description	Size (bits)
AS/AS′/SSC	Authentication server	–
MP/UA	Maintenance personnel/user agent	–
SS	Name of substation	128
Name	Name of user	128
s/S_i	Signature of user	128
P	Generator of group	128
ID	Identity of user/device/substation	128
x	Private key of user/device/server	128
Y_1/Y_2	Public key pair of user/device/server	128
K	Secret shared key	128
SDP	Secret device parameter	64
H()	One-way hash function	–
h/H	Hash value	64
Role/cRole	Role of user	64
mode	Mode of access	16
location	User/device location	32
department	Department of user	16
T	Timestamp	64

2 Related Work

Recently, a lot of research has been done for securing the *SG* network, including the device and network authentication [10], [11], privacy preservation in *AMI* [12], vehicular-to-grid (*V2G*) networks [13] and dynamic price management [14], and attribute revocation in data aggregation [15], [16] with attribute-based encryption scheme. Some literature focuses on outsider attacks, such as cyber-physical attacks [17], load altering attacks [18] and distribution attacks [19], cyber security of substations [20], data attacks [21], false data injection attacks [22], data integrity attacks [23], traffic analysis attacks [24], man-in-the-middle attacks [25], *DoS* attacks [26], and etc. However, insider attacks in the *SG* network, such as attacks by the customer, attacks by the operators/maintenance personnel, and etc., have not been well investigated. These possibilities exist, if the user/device authentication and access control are weak or not provided.

Access control in the distributed system is more challenging, as the management of activities by a single central authority might not be possible or could be more resource demanding [27]. In a role-based access control (*RBAC*), there is a specific role for each user or a set of users created by the administrator for accessing the resources with the specified permissions. There are some role-based access control models for the *SG* network existing in the literature that focus on the user-role-based authentication [28], [29]. However, role-based schemes are generally expensive to implement and do not provide a real-time access control in many situations, specifically when the user has dynamic attributes, such as shift/job timing, location, time of the day, and etc. In addition, device authentication mechanisms for smart home area network [30], [31], *SG* network [32], *SG* electric vehicle system [33], and *AMI* network [34] have also been proposed by researchers. Authentication scheme in [30] is based on a public-key cryptography using elliptic curves over finite fields, every device shares a pair-wise key with the center of trust in [31], and the homomorphic keyed hash values are used in [32]. A contextual factor based on physical connectivity in the grid with conventional authentication factor in the challenge-response are combined and implemented in [33] on *NXP-ATOP* with *ARM* processor. Further, the scheme in [34] uses an *ID*-based authentication and a *PKI*, which generates a huge overhead and is expensive to implement. However, these schemes do not deal with the user authentication in the *SG* network. Recently, a multi-factor authentication for fragile communications is proposed that provides authentication service in a slow connection situation and when central server is down. But the scheme needs additional smart cards and biometric devices [35].

Furthermore, the key management among various users and devices, and the communication protocol level security are also necessary factors in order to provide stronger secu-

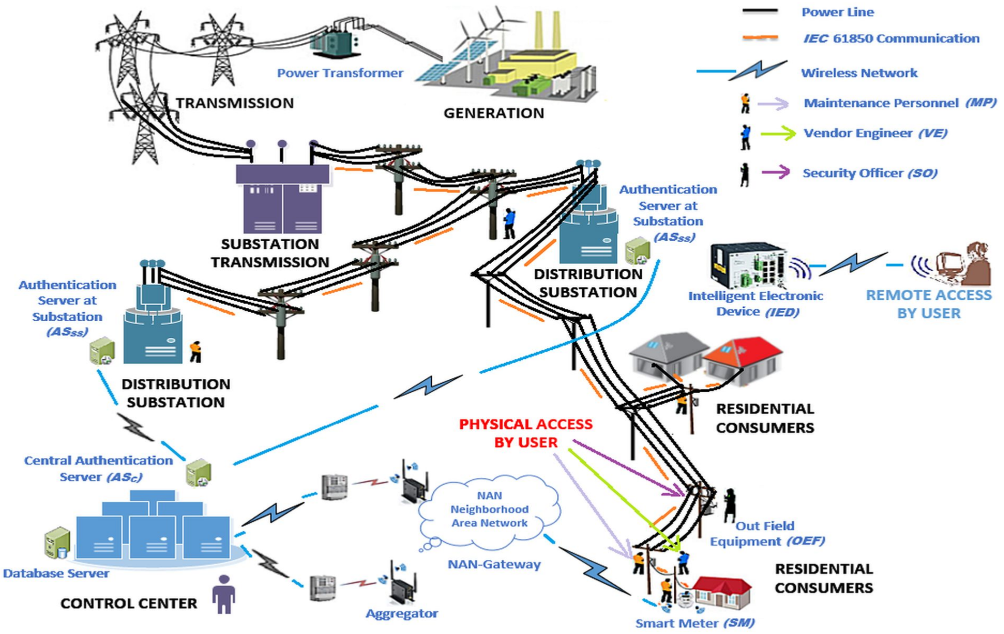


Fig. 1: Smart grid system model consisting of various devices and users connected by communication networks.

rity to the system. In this direction, a scheme was proposed that deals with key generation, revocation management and share keys between devices, and secure transmission of meter-reading data [36]. An encryption key management technique based on certificateless public key cryptography was proposed that provides end-to-end security in *AMI*, but the approach generates a huge overhead [37]. An extension of distributed network protocol (*DNP3*) to the *DNP3* secure authentication (*SA*) considers multiple users at the master site [38]. This scheme presumes that both, the master station and the substation, share a common secret key, which is used to generate a session key. Furthermore, there exists a substation-level authentication scheme in the literature where *IED*s and other resource-constrained devices can be authenticated by any remote users with the help of the substation controller (*SSC*) [39]. However, they considered remote access of the *IED*s using passwords shared among users, lacking message integrity check, batch verification, and prevention against replay attacks. Using the same password for each user-role to access all devices cannot support verification of user identities. Instead of using public keys for a huge number of handheld devices (*HD*), symmetric keys using physical unclonable functions were proposed to ensure a key agreement between the *HD* and the telemetric devices under a scalable password changing protocol [41].

To the best of our knowledge, there is no such scheme in the literature providing authentication and authorization for various users accessing different devices in the *SG* system. Our scheme tackles this challenging problem in this paper.

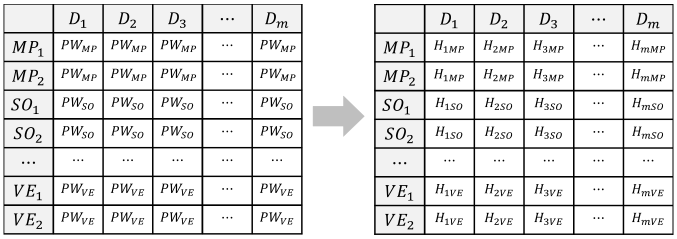
3 Communication, System, and Attack Models

This section presents our communication and system model, as well as attack model.

3.1 Communication and System Model

Consider a *SG* system model, including *AMI* infrastructure network, as shown in Figure 1, where *IED*s, *SM*s, and *OFE*s are placed at different geographical locations under different substations. These devices can be accessed by different users, such as maintenance personnel (*MP*), vendor engineer (*VE*), and security officer (*SO*) physically as well as remotely. If a user accesses a device physically, then it is assumed that the device provides an interface for the integrity of information. Additionally, if a user accesses a device remotely via wireless network, then it requires a mechanism of integrity protection. There is a central authentication server (AS_c) stored at the control center (*CC*). This AS_c is connected with a number of substation servers (AS_{ss}), each using a different pre-shared key. The communication in our *SG* system is governed by the *DNP3* or *IEC 61850*, represented by Orange lines in Figure 1. A *WAN*/cellular technology can be used in the wireless network.

As illustrated in Figure 2, in the existing problem, the maintenance personnel MP_1 and MP_2 share a common password PW_{MP}, security officers SO_1 and SO_2 share a secret password PW_{SO}, vendor engineers VE_1 and VE_2 share a password PW_{VE}, and so on. We propose a new



	D_1	D_2	D_3	$\cdots$	D_m
MP_1	PW_{MP}	PW_{MP}	PW_{MP}	$\cdots$	PW_{MP}
MP_2	PW_{MP}	PW_{MP}	PW_{MP}	$\cdots$	PW_{MP}
SO_1	PW_{SO}	PW_{SO}	PW_{SO}	$\cdots$	PW_{SO}
SO_2	PW_{SO}	PW_{SO}	PW_{SO}	$\cdots$	PW_{SO}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
VE_1	PW_{VE}	PW_{VE}	PW_{VE}	$\cdots$	PW_{VE}
VE_2	PW_{VE}	PW_{VE}	PW_{VE}	$\cdots$	PW_{VE}

Existing Problem

	D_1	D_2	D_3	$\cdots$	D_m
MP_1	H_{1MP}	H_{2MP}	H_{3MP}	$\cdots$	H_{mMP}
MP_2	H_{1MP}	H_{2MP}	H_{3MP}	$\cdots$	H_{mMP}
SO_1	H_{1SO}	H_{2SO}	H_{3SO}	$\cdots$	H_{mSO}
SO_2	H_{1SO}	H_{2SO}	H_{3SO}	$\cdots$	H_{mSO}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
VE_1	H_{1VE}	H_{2VE}	H_{3VE}	$\cdots$	H_{mVE}
VE_2	H_{1VE}	H_{2VE}	H_{3VE}	$\cdots$	H_{mVE}

New Formulation

Fig. 2: Comparison of password matrix.

password matrix where each password is replaced by a hash value for a specific group of users. However, this hash value is different for different devices placed at different locations.

There can be a number of central stations in a larger *SG* network. However, in this paper, we consider one central station directly connected with n-substations as shown in Figure 3. The authentication server of each substation (AS_{ss}) and its corresponding central station (AS_c) can securely communicate with each other using a pre-shared symmetric key, and this is true for all substations. Substation-to-substation communication is done only through the central station. In this paper, for better understanding, we explain our scheme considering a particular user-role named *MP* and the *IED*s as devices. Therefore, in our system, there are n-*MP*s (MP_1, MP_2, ..., MP_n) and m-devices (D_1, D_2, ..., D_m). Each device must first compute a common secret key (say a password) as computed by the user (say *MP*) in order to provide a role-based user authorization. Each device must verify the role of a user, and must perform authorization and user verification. In our scheme, for each user-role, a hash value is dynamically computed. For a specific group of users, the hash value is different for different devices. In addition, a session key is used between the user and the device in order to maintain confidentiality of information.

3.2 Attack Model

Various security attacks (outsider and/or insider) are possible on *SG* network. An attacker may perform a *man-in-the-middle attack* (outsider attack) by creating an active connection between a user and a server, and makes them believe that they are directly communicating with each other by a secure connection over the *SG* network. The attacker may delay or repeat the transmitted message to the user or the server over the network resulting in a *replay attack*. Also, an *integrity violation* can be performed by an external attacker, if the attacker is successful at modifying the transmitted messages over the *SG* network. An attacker may also perform an *impersonation attack* where it tries to impersonate the users involved in the *SG* system. *Change of security parameters* can be done by both, insider and outsider attackers. An unauthorized user may change the security parameters of the *SG* system or the device in order to gain access to it. Furthermore, the customer can tamper with the smart meter and the *AMI* network to reduce the cost of electricity usage. In addition, prevention against a *repudiation attack* is one of the most important requirements, as even the higher authorities of the *SG* and *AMI* networks can alter data and then later deny it. This attack can be intentionally performed by an insider as well as an outsider. There are many more attacks possible in the *SG*, including *attacks by consumers/operators/maintenance personnel*. We also address the insider attacks where an insider, accessing the devices, can harm the system in the absence of proper authentication and authorization.

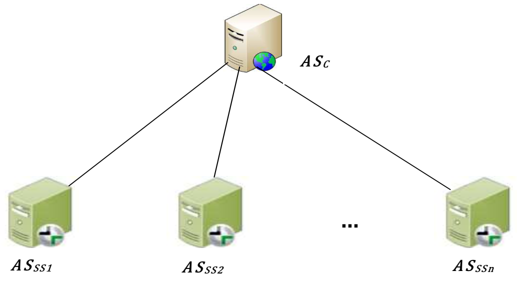


Fig. 3: Communication among various substations.

4 Proposed Scheme

This section presents a preliminary discussion on access control, and then proposes a solution for preventing various attacks in the *SG*. Our scheme (a) first, derives and identifies the role of a user, and then verifies the identities of each user and the device, (b) allows to have a symmetric secret key at both ends (user and device) for secure communication without transmitting the key to another end over the network, (c) supports two-factor authentication to defeat *MITM* and other attacks, (d) is applicable in dynamic environment varying the number of users and devices in the *SG* network, and (e) is verified with formal security proofs.

4.1 Preliminary Discussion on Access Control

An access control restricts a user to have a limited access to resources according to its domain of interest and permissions. In *RBAC*, it is easy to audit users' permissions and the permissions granted to a user, whereas in attribute-based access control (*ABAC*), it is more difficult to audit users having resource access to the given permission and the permissions granted to a given user. The reason is that the *ABAC* uses a large number of attributes that requires substantial understanding and manageability, and these attributes do not have any meaning until they are associated with an entity [42]. It would be advantageous to combine both schemes together to provide flexibility, auditability, scalability, understandability, and manageability. A role-centric attribute-based access control (*RABAC*) scheme proposed by Xin *et al.* [43] combines roles and attributes to provide an access control in a reliable manner. Moreover, role-centric and dynamic-role capabilities with *ABAC*'s fine-grained access control are also being developed and implemented for commercial purposes [44].

4.2 Basic Description of Our System and Assumptions

The proposed scheme provides a dynamic-role attribute-based access control, where a role is formed based on various attributes of an entity. Each user has a defined role as *RBAC* registered at corresponding AS_{ss}, whereas dynamic role of each user is computed using *ABAC* at each device. A part of our proposed scheme is based on bilinear map and pair-based cryptography [45], where the secret keys generated at both ends have been proved equal. The following are some basic assumptions made in this paper:

1. In case of physical access of the devices, a user interface provides input/output to/from each device and is capable of performing light computations on it.

2. The presented scenario is similar for *IED*s, *SM*s, and *OFE*s. The *IED*s scenario can be easily extended to explain scenarios with other types of devices.

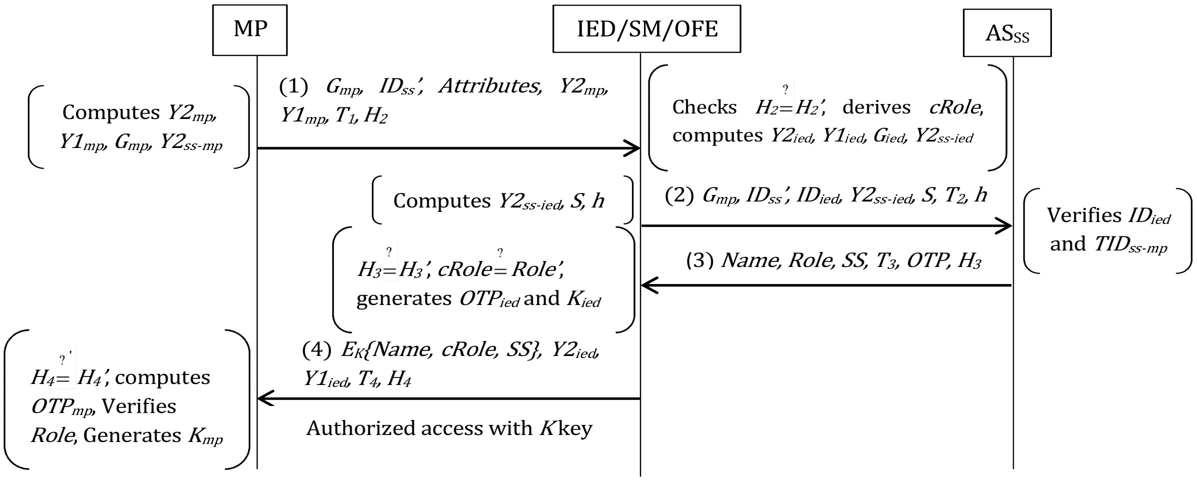

MP
IED/SM/OFE
ASSS
Computes Y2mp, Y1mp, Gmp, Y2ss-mp
(1) Gmp, IDss', Attributes, Y2mp, Y1mp, T1, H2
Checks H2=?H2', derives cRole, computes Y2ied, Y1ied, Gied, Y2ss-ied
Computes Y2ss-ied, S, h
(2) Gmp, IDss', IDied, Y2ss-ied, S, T2, h
Verifies IDied and TIDss-mp
H3=?H3', cRole=?Role', generates OTPied and Kied
(3) Name, Role, SS, T3, OTP, H3
H4=?' H4', computes OTPmp, Verifies Role, Generates Kmp
(4) EK{Name, cRole, SS}, Y2ied, Y1ied, T4, H4
Authorized access with K key

Fig. 4: Proposed scheme for remote access of the device.

3. Each user and the corresponding substation share the user credential, *i.e.*, user identity and password.

4. The communication channel between a device and a server is secure, as the devices are always connected with the respective substations.

Our proposed scheme is based on the bilinear pairing technique. Let $\mathbb{G}_1$ be an additive group and $\mathbb{G}_2$ be a multiplicative group on a symmetric pairing function e. Both groups are of order q, where q is a large prime. Let P be an arbitrary generator of $\mathbb{G}_1$. Assume that the discrete logarithm problem (*DLP*) is hard in both $\mathbb{G}_1$ and $\mathbb{G}_2$.
Definition: A bilinear pairing on ($\mathbb{G}_1$, $\mathbb{G}_2$) is a map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$ that satisfies the following properties of the cryptographic bilinear map:
Properties: (1) Bilinearity: $e(aP, bQ) = e(bP, aQ) = e(abP, Q) = e(P, abQ) = e(P, Q)^{ab}$; $\forall\ a, b \in \mathbb{Z}_q^*$, and $\forall\ P, Q \in \mathbb{G}_1$
(2) Non-degeneracy: $e(P, P) \neq 1$
(3) Computability: There exists an efficient algorithm to compute $e(P, Q)$ for $\forall P, Q \in \mathbb{G}_1$.

Here, given $P, aP, bP, cP \in \mathbb{G}_1$, and $a, b, c \in \mathbb{Z}_q^*$, it is easy to verify whether $c = ab \mod q$, however, it is difficult to compute abP. The group $\mathbb{G}_1$ is called a gap Diffie-Hellman (DH) group [46].

4.3 A New Generic Design of Proposed Scheme

Our proposed scheme has four subsections: (i) initial setup, (ii) identity creation, (iii) accessing device, and (iv) verification of the identities.

i) Initial Setup: Let $\mathbb{G}$, $\mathbb{G}_T$ be two cyclic groups of the same prime order q, and P be a generator of group $\mathbb{G}$. Suppose $\mathbb{G}$ and $\mathbb{G}_T$ are equipped with a pairing, *i.e.*, a non-degenerated and efficiently computable bilinear map $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ such that $e(P, P) \neq 1_{\mathbb{G}_T}$ and $e(aP_1, bQ_1) = e(bP_1, aQ_1) = e(P_1, Q_1)^{ab} \in \mathbb{G}_T$ for all $a, b \in \mathbb{Z}_q^*$ and any $P_1, Q_1 \in \mathbb{G}$. Central station and all substations agree on an elliptic curve over a finite field $\mathbb{E}(\mathbb{F}_q)$. Note that the elliptic curves are more efficient than finite fields and make pairings even more useful in terms of space and time [47]. Further, $ID_{ss} \in \mathbb{Z}_q^*$ is an identity of the substation (*SS*) while public key of the substation is $Y_{2_{ss}} = ID_{ss}P$. We refer to [48] for a more comprehensive description of pairing assumptions.

ii) Identity Creation: We assume that the identity of each substation ID_{ss} is publicly known. Each day, each substation's authentication server AS_{ss} chooses a random private key $x_{ss} \in \mathbb{Z}_q^*$. Similarly, each *MP* selects a new random private key $x_{mp} \in \mathbb{Z}_q^*$ and generates its public identity in a group (substation) as G_{mp}=$x_{mp}P$. The *MP* requests to the $AS_{ss}^{'}$ (under which it is registered) for the verification of its public identity. It submits G_{mp} to the $AS_{ss}^{'}$, and the $AS_{ss}^{'}$ verifies whether any other registered *MP* has the same identity. If yes, the requested *MP* needs to change its public identity. The chances of selecting the same x_{mp} by two *MP*s are very rare because q is a very large prime. The $AS_{ss}^{'}$ returns a temporary identity as TID_{ss-mp}=$x_{ss}G_{mp}$ to the *MP*. The $AS_{ss}^{'}$ also stores TID_{ss-mp} in its database in addition to the details of the *MP*, such as name of the person (*Name*), role of the person (*Role*), identity of the person (ID_{mp}), and contact number (*Mobno*) in order to send *OTP* for verification (two-factor authentication). The *MP* can use this TID_{ss-mp} for accessing devices under different substations for certain duration per session and per substation. Note that the secret key and the temporary identity of the user are only valid for a short period and are removed from the $AS_{ss}^{'}$'s database after its use. However, each $AS_{ss}^{'}$ maintains a daily report containing the details of the accessed devices (ID_{ied}, $ID_{ss}^{'}$, type of device) by the *MP*s. The request for generating a temporary identity can be completed either by physically at substations or via remotely using its login credentials to the $AS_{ss}^{'}$.

iii) Accessing Device: This subsection describes the steps for accessing devices remotely as well as physically.

a) Remote Access of Device: Each step of remote access scheme, as illustrated in Figure 4, is explained as follows:

Step-1: First, the *MP* generates a public key pair $Y_{1_{mp}}{=}x_{mp}H_1$, $Y_{2_{mp}}{=}ID_{ss}Y_{1_{mp}}$, and a $SS{-}MP$ compatible public key as $Y_{2_{ss-mp}}{=}x_{mp}Y_{2_{ss}}$. Note that Y_1 and Y_2 both are public parameters. Here, $H_1{=}H(mode, department, location, SDP)$ and ID_{ss} is the identity of the substation to which the device is registered, and *SDP* is secret device parameter. Then, the *MP* sends message-1 $\{G_{mp}, ID^{'}_{ss}, Attributes, Y_{2_{mp}}, Y_{1_{mp}}, T_1\}$ to the *IED*/*SM*/*OFE* where *Attributes* are *mode of access*, *department*, *location*, and *SDP*, T_1 is a timestamp value at the time of message creation, and $ID^{'}_{ss}$ is the identity of substation to which the *MP* is registered. Location information is unique for each device. Note that the *MP* needs to provide location information of the device for the remote access of the device. Otherwise, the location of the device will be used as the location of the *MP* in case of physical access. For the remote access, the *MP* also includes a one-way hash value H_2 along with message-1, where $H_2{=}H(G_{mp}, ID^{'}_{ss}, Attributes, Y_{2_{mp}}, Y_{1_{mp}}, T_1)$ for ensuring the integrity of message over the network.

Step-2: On receiving message-1, the *IED* computes $H^{'}_2{=}H(G_{mp}, ID^{'}_{ss}, Attributes, Y_{2_{mp}}, Y_{1_{mp}}, T_1)$ and verifies $H_2 \overset{?}{=} H^{'}_2$. If both are equal, the integrity of message-1 is successfully verified. Otherwise, the connection is terminated. Then, the *IED* computes $H^{'}_1{=}H(mode, department, location, SDP)$, its public identity in the group (substation) as $G_{ied}{=}x_{ied}P$ using its private key $x_{ied} \in \mathbb{Z}^*_q$, a public key pair $Y_{1_{ied}}{=}x_{ied}H_1$ and $Y_{2_{ied}}{=}ID_{ss}Y_{1_{ied}}$, and a *SS-IED* compatible public key $Y_{2_{ss-ied}}{=}x_{ied}Y_{2_{ss}}$. The computation of $H^{'}_1$ always requires the $location$ information supplied by the device itself, not by the *MP* during physical or remote access of the device. Then, the role of a user is computed by the *IED* by matching the value of $H^{'}_1$ with the stored role values, *i.e.*, $cRole \overset{?}{=} H^{'}_1$. Further, the *IED* also verifies the identity of the *MP* by sending message-2 $\{G_{mp}, ID^{'}_{ss}, ID_{ied}, Y_{2_{ss-ied}}, S, T_2, h\}$ using a secure channel to the respective AS_{ss} to which the device belongs. Here, $S{=}ID_{ied}h{+}ID_{ied}G_{ied}$, and $h{=}H(G_{mp}, ID^{'}_{ss}, ID_{ied}, Y_{2_{ss-ied}}, S, T_2)$.

Step-3: On receiving message-2, the AS_{ss} first verifies the identity of the device, *i.e.*, ID_{ied}. The verification of ID_{ied} is done in a batch mode. Once, ID_{ied} verification is over, the AS_{ss} checks whether its ID_{ss} is same as the received $ID^{'}_{ss}$. If they are different, the AS_{ss} passes $\{ID_{ss}, ID^{'}_{ss}, G_{mp}, ID_{ied}\}$ to the AS_c in order to verify the identity of the user by its registered server of the substation. This is done by encrypting the message with a pre-shared key between the AS_{ss} and the AS_c. The AS_c transmits the message $\{ID_{ss}, ID^{'}_{ss}, G_{mp}, ID_{ied}\}$ to the corresponding $AS^{'}_{ss}$ having identity $ID^{'}_{ss}$ encrypted with pre-shared key between the $AS^{'}_{ss}$ and the AS_c. Thereafter, the $AS^{'}_{ss}$ verifies TID_{ss-mp} using the received G_{mp} and its x_{ss}, and sends response $\{Name, Role, SS\}$ back to the AS_{ss} via AS_c by a secure channel, where *SS* is the name of the $AS^{'}_{ss}$ to which *MP* belongs. If $ID_{ss} \overset{?}{=} ID^{'}_{ss}$, it means the *MP* belongs to the ID_{ss}, so the AS_{ss} itself handles TID_{ss-mp} verification and report generation. We perform a simple verification of TID_{ss-mp} to improve the system efficiency. Furthermore, an *OTP* is sent to the *MP* for verifying its identity. This *OTP* can be sent using *EasySMS* that provides end-to-end security to the *SMS* over the network [49]. If the two-factor verification is successful, the AS_{ss} sends message-3 $\{Name, Role, SS, T_3, OTP, H_3\}$ to the *IED*, where $H_3{=}H(Name, Role, SS, T_3, OTP)$. Otherwise, the connection is discarded.

Step-4: After receiving message-3, the *IED* computes and verifies $H_3 \overset{?}{=} H^{'}_3$. If it is true, the *IED* also checks $cRole \overset{?}{=} Role$. If it is also true, the *IED* transmits message-4 $\{E_K\{Name, Role, SS\}, T_4, Y_{1_{ied}}, Y_{2_{ied}}, H_4\}$ to the *MP*, where $H_4{=}H(E_K\{Name, Role, SS\}, T_4, Y_{1_{ied}}, Y_{2_{ied}})$. Here, E denotes the encryption that can be performed using the *MAES* algorithm [49], which generates a ciphertext of 158 characters (1111 bits) from a plaintext of 160 characters (1120 bits). Hence, this encryption will maintain the system efficiency almost as it is. The secret key K_{ied}, used in *MAES*, is generated at *IED* and is expanded from 128 to 256 bits using an expand function and an *OTP*. The *MP* computes and compares $H_4 \overset{?}{=} H^{'}_4$. If it is true, the *MP* computes a secret key K_{mp}, decrypts the message, verifies its role, and performs actions based on an authorized role. Note that the secret keys generated at *IED* and *MP* are same, *i.e.*, $K_{ied}{=}K_{mp}$ (Theorem-1 in sec. 5.1). Hence, this secret shared key (say K) is used for encrypting message-4. Now, the *MP* can directly access the device using the shared key until its expiry time. Here, the expiry time of the key is a session time, and the access time per role per attempt is a sub-session time within a session. For example, if the working shift is about 8 hours, then the key expiry time (session time) may configure to be 4 hours, the user can access the device multiple times (per sub-session) depending on the role specified for each type of user. For read-only access users, the sub-session time may configure to be 5 *sec.* (s).

b) Physical Access of Device: Note that in case of the physical access, as shown in Figure 5, the device uses its own default location for computing hash value. The timestamp values T_1 and T_4 are not required for the physical access, as input (set of parameters) is acquired on-site. Also, H_2 and H_4 do not need to be computed.

iv) Verification of the Identities of IED and MP: Since the maintenance work is generally scheduled in advance and is done at regular intervals, the verification of user and device identities by the authentication server in a batch would be more efficient during that period. In an emergency situation, a one-to-one maintenance service can be provided to any device. Here, we focus on an efficient batch verification of identities of the devices as shown in Figure 6.

All IED_i (i = 1, 2, 3,..., m; m = number of *IED*s requested for verification to the server at one time) send message-2 $\{G_{mp_i}, ID^{'}_{ss_i}, ID_{ied_i}, Y_{2_{ss-ied_i}}, S_i, T_i, h_i\}$ to the AS_{ss_i}. Here, $S_i{=}ID_{ied_i}h_i{+}ID_{ied_i}G_{ied_i}$ and $h_i{=}H(G_{mp_i}, ID^{'}_{ss_i}, ID_{ied_i}, Y_{2_{ss-ied_i}}, S_i, T_i)$. On receiving message-2, the AS_{ss_i} computes $s{=}\sum_i^m S_i$ and $X{=}sID_{ss_i}$, and compares $\sum_{i=1}^m [ID_{ied_i}(Y_{2_{ss-ied_i}} + h_iID_{ss_i})] \overset{?}{=} X$, where ID_{ss_i} is the identity of the AS_{ss_i}. If it is false, indicating that one or more IED_i are malicious, the AS_{ss_i} terminates the connection for each invalid request. Then, it

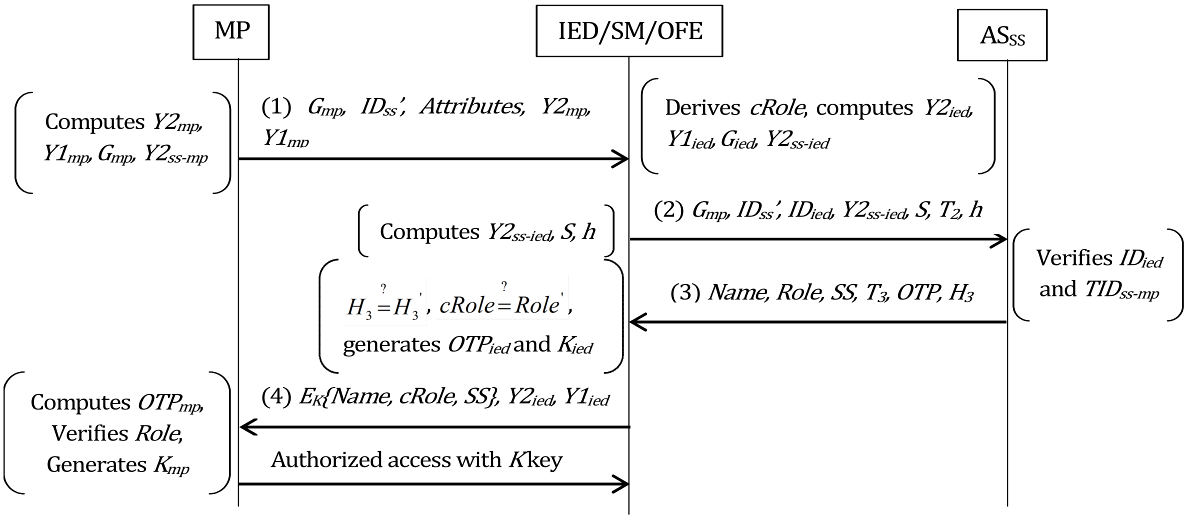

MP
IED/SM/OFE
AS_{SS}
Computes $Y2_{mp}$, $Y1_{mp}$, G_{mp}, $Y2_{ss\text{-}mp}$
(1) G_{mp}, ID_{ss}', $Attributes$, $Y2_{mp}$, $Y1_{mp}$
Derives $cRole$, computes $Y2_{ied}$, $Y1_{ied}$, G_{ied}, $Y2_{ss\text{-}ied}$
Computes $Y2_{ss\text{-}ied}$, S, h
(2) G_{mp}, ID_{ss}', ID_{ied}, $Y2_{ss\text{-}ied}$, S, T_2, h
Verifies ID_{ied} and $TID_{ss\text{-}mp}$
$H_3 \overset{?}{=} H_3'$, $cRole \overset{?}{=} Role'$, generates OTP_{ied} and K_{ied}
(3) $Name$, $Role$, SS, T_3, OTP, H_3
Computes OTP_{mp}, Verifies $Role$, Generates K_{mp}
(4) $E_K\{Name, cRole, SS\}$, $Y2_{ied}$, $Y1_{ied}$
Authorized access with K key

Fig. 5: Proposed scheme for physical access of the device.

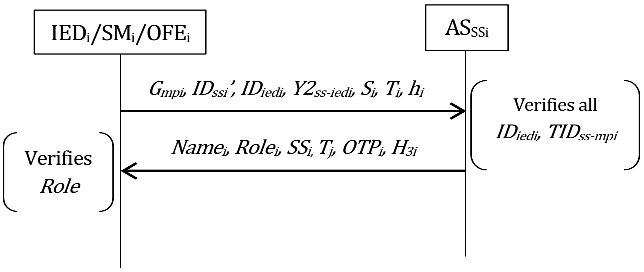

$IED_i/SM_i/OFE_i$
AS_{SSi}
G_{mpi}, ID_{ssi}', ID_{iedi}, $Y2_{ss\text{-}iedi}$, S_i, T_i, h_i
Verifies all ID_{iedi}, $TID_{ss\text{-}mpi}$
$Name_i$, $Role_i$, SS_i, T_j, OTP_i, H_{3i}
Verifies $Role$

Fig. 6: Verification of device and user identities.

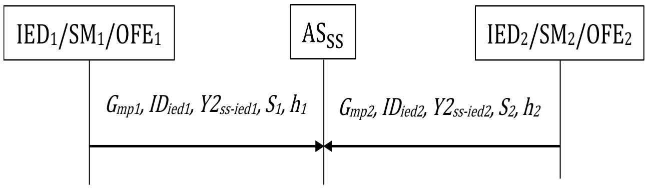

$IED_1/SM_1/OFE_1$
AS_{SS}
$IED_2/SM_2/OFE_2$
G_{mp1}, ID_{ied1}, $Y2_{ss\text{-}ied1}$, S_1, h_1
G_{mp2}, ID_{ied2}, $Y2_{ss\text{-}ied2}$, S_2, h_2

Fig. 7: An example for identities and roles verification.

identifies and removes the malicious IED_i from the batch using an algorithm presented in [50]. Thereafter, it again performs a batch verification for all valid IED_i and checks $\sum_{i=1}^{m}[ID_{ied_i}(Y_{2_{ss-ied_i}} + h_i ID_{ss_i})] \overset{?}{=} X$. The AS_{ss_i} checks $ID_{ss_i^{'}}$ from the message received by each verified IED_i. If $ID_{ss_i^{'}} \overset{?}{=} ID_{ss_i}$, the MP belongs to this particular AS_{ss_i}. Otherwise, the encrypted $\{G_{mp_i}, ID_{ss_i}, ID_{ss_i^{'}}, ID_{ied_i}\}$ is forwarded to the corresponding $AS_{ss_i^{'}}$ via AS_c. The AS_c keeps the information regarding the identities and names of all substations. Thereafter, the $AS_{ss_i^{'}}$ verifies TID_{ss-mp_i} and sends response $\{Name_i, Role_i, SS_i^{'}\}$ back to the AS_{ss_i} via AS_c. Upon verification, the AS_{ss_i} sends message-3 $\{Name_i, Role_i, SS_i, T_j, OTP_i, H_{3_i}\}$ to all valid IED_i. The process of verifying $\sum_{i=1}^{m}[ID_{ied_i}(Y_{2_{ss-ied_i}} + h_i ID_{ss_i})] \overset{?}{=} X$ is illustrated in Figure 7, where for simplicity, we have omitted $ID_{ss_i^{'}}$ and T_i. We assume that $ID_{ss_i}{=}ID_{ss_i^{'}}$, hence, $AS_{ss_i}{=}AS_{ss_i^{'}}$ and $SS_i{=}SS_i^{'}$. Consider two IEDs, *i.e.*, IED_1 and IED_2 that send message-2 to the AS_{ss} at the same time. The message received by the AS_{ss} from the IED_1 is $\{G_{mp_1}, ID_{ied_1}, Y_{2_{ss-ied_1}}, S_1, h_1\}$ while message from the IED_2 is $\{G_{mp_2}, ID_{ied_2}, Y_{2_{ss-ied_2}}, S_2, h_2\}$. On receiving, the AS_{ss} computes and compares $\sum_{i=1}^{m}[ID_{ied_i}(Y_{2_{ss-ied_i}} + h_i ID_{ss_i})] \overset{?}{=} X$. In order to verify the identity of each MP, the AS_{ss} first computes $TID_{ss-mp_i}^{'}{=}x_{ss}G_{mp_i}$ from the received G_{mp_i} and compares it with the stored TID_{ss-mp_i}. If it is true, the identity of the user is verified successfully. The proposed scheme can be applied with any other user-role, such as customer, vendor engineer, contractor, and etc.

5 SECURITY PROOFS, ANALYSIS, AND PERFORMANCE EVALUATION

This section presents computation proofs in the scheme, and security and performance analysis of the proposed scheme.

5.1 Computation Proofs

This subsection derives the computational proofs for the statements used in the proposed scheme.

Theorem 1. The proposed scheme generates a shared secret key at *MP* and *IED*.
Proof: *Generation of a Shared Secret Key at MP and IED:* In order to generate same key at both ends, the *MP* computes $OTP_{mp} = Y_{2_{ss-mp}}OTP$ and generates K_{mp} whereas the *IED* computes $OTP_{ied} = G_{ied}OTP$ and generates K_{ied}.

$$\begin{aligned}
K_{mp} &= e(Y_{2_{ied}}, G_{mp})e(Y_{1_{ied}}, OTP_{mp}) \\
&= e(Y_{2_{ied}}, G_{mp})e(Y_{1_{ied}}, Y_{2_{ss-mp}}OTP) \\
&= e(ID_{ss}Y_{1_{ied}}, x_{mp}P)e(x_{ied}H_1, x_{mp}Y_{2_{ss}}OTP) \\
&= e(x_{ied}ID_{ss}H_1, x_{mp}P)e(x_{ied}H_1, x_{mp}ID_{ss}P.OTP) \\
&= e(x_{mp}H_1, x_{ied}ID_{ss}P)e(x_{mp}ID_{ss}H_1, x_{ied}P.OTP) \\
&\quad since\{e(aP, bQ) = e(bP, aQ)\} \\
&= e(Y_{1_{mp}}, Y_{2_{ss-ied}})e(Y_{1_{mp}}ID_{ss}, G_{ied}OTP) \\
&= e(Y_{1_{mp}}, Y_{2_{ss-ied}})e(Y_{2_{mp}}, OTP_{ied}) \\
&= K_{ied}
\end{aligned}$$

In a similar way, each *IED* can generate a shared secret key with other users.

Theorem 2. If all the requests are made by the legitimate *MP*s to various *IED*s, the AS_{ss} verifies all the requests correctly.
Proof: *Batch Verification at* AS_{ss}*:*

$$\begin{aligned}
R.H.S. &= X \\
&= sID_{ss_i} \\
&= ID_{ss}[(ID_{ied_1}h_1 + ID_{ied_1}G_{ied_1}) \\
&\quad + (ID_{ied_2}h_2 + ID_{ied_2}G_{ied_2})] \\
&= ID_{ss}[(ID_{ied_1}G_{ied_1} + ID_{ied_2}G_{ied_2}) \\
&\quad + (ID_{ied_1}h_1 + ID_{ied_2}h_2)]
\end{aligned}$$

$$\begin{aligned}
L.H.S. &= \sum_{i=1}^{m}[ID_{ied_i}(Y_{2_{ss-ied_i}} + h_iID_{ss_i})] \\
&= ID_{ied_1}Y_{2_{ss-ied_1}} + ID_{ied_2}Y_{2_{ss-ied_2}} \\
&\quad + ID_{ss}[ID_{ied_1}h_1 + ID_{ied_2}h_2] \\
&= ID_{ied_1}x_{ied_1}ID_{ss}P + ID_{ied_2}x_{ied_2}ID_{ss}P \\
&\quad + ID_{ss}[ID_{ied_1}h_1 + ID_{ied_2}h_2] \\
&= ID_{ss}[ID_{ied_1}x_{ied_1}P + ID_{ied_2}x_{ied_2}P] \\
&\quad + ID_{ss}[ID_{ied_1}h_1 + ID_{ied_2}h_2] \\
&= ID_{ss}[(ID_{ied_1}G_{ied_1} + ID_{ied_2}G_{ied_2}) \\
&\quad + (ID_{ied_1}h_1 + ID_{ied_2}h_2)]
\end{aligned}$$

Hence, $\sum_{i=1}^{m}[ID_{ied_i}(Y_{2_{ss-ied_i}} + h_iID_{ss_i})] \overset{?}{=} X$ is true.

5.2 Security Analysis

In this subsection, authentication, session key establishment, and privacy preservation are discussed along with prevention against different attacks.

i) Mutual Authentication: A mutual authentication is provided between the user, *i.e.*, *MP*, and the server, *i.e.*, AS_{ss}. The AS_{ss} authenticates the *MP* by verifying $TID_{ss-mp} \overset{?}{=} x_{ss}G_{mp}$, and each *MP* authenticates the *AS* by comparing its name, role, and substation name with the received information, *i.e.*, *Name*, *Role*, and *SS*. Note the G_{mp} is used only once. Further, we used two-factor authentication where an *OTP* is sent to the user. Adversary $\mathcal{A}$ cannot verify *OTP*.

ii) Session Key Establishment: Each K_{mp}/K_{ied} key is used as a session shared secret key for each authentication between the user *MP* and the device *IED*. The same key is used for a session within the expiry time.

iii) Privacy Preservation: The privacy of each *MP* is well protected during the authentication over the network. The TID_{ss-mp} is computed by the AS_{ss} and is assigned to the *MP*. The intermediate operators and operator at AS_{ss} cannot identify the actual *MP* by viewing G_{mp}, as a random number x_{ss} is selected by the server as private key, which is securely stored in the database of AS_{ss} considering the fact that operator at server cannot access master table of the database. The *IED*s also cannot identify the *MP*. Hence, an attacker cannot retrieve actual *MP*'s identity by forging the *IED* or by traffic analysis over the network. Since, TID_{ss-mp} of the *MP* changes for each session, the adversary cannot gain useful information from a long term analysis.

iv) Integrity Protection: The proposed scheme provides integrity protection by using hash functions on each transmitted message over the network. If an adversary $\mathcal{A}$ intentionally changes any transmitted parameter (including public key), the received and computed hash values will not match at receiver and the connection will be terminated.

v) Prevention Against Various Attacks: The proposed scheme defeats the following security attacks:

a. Impersonation Attack: $\mathcal{A}$ needs to know the temporary identity and secret key of the victim user assigned by the server. However, $\mathcal{A}$ cannot obtain secret key without knowing its parameters to generate. A different key pair is used at each device to prevent the use of old parameter values in other devices. There are two possible cases for an impersonation attack as follows:

- *Case-1:* $\mathcal{A}$ *impersonates the MP:* (1) $\mathcal{A}$ changes G_{mp} as $\mathcal{A}G_{mp}$. On receiving message-1, the *IED* finds $H_2 \neq H_2'$, and hence terminates the connection.
  (2) $\mathcal{A}$ sends a fake G_{mp} as $\mathcal{A}G_{mp}$ with a new hash $\mathcal{A}H_2$. On receiving the message, the AS_{ss} verifies received $\mathcal{A}G_{mp}$ with the stored G_{mp}'s in the database. As, $\mathcal{A}G_{mp} \neq G_{mp}$, the *AS* rejects the request.
- *Case-2:* $\mathcal{A}$ *impersonates the IED:* If $\mathcal{A}$ tries to impersonate the *IED*, the identity and signature of the device are verified by the AS_{ss}. In this case, the signature would be different and identity of the device will not match. Hence, the connection will be terminated.

b. MITM Attack: A packet or message sniffing allows $\mathcal{A}$ to capture the message information over the network. Once the information is captured, $\mathcal{A}$ can gain access to the system. For a *MITM*, $\mathcal{A}$ tries to build a connection between both the involved parties. There are two possible cases as follows:

- *Case-1: Key-exchange by adversary* $\mathcal{A}$*:* Consider $\mathcal{A}$ is located between the *MP* and the *IED*. Since ($Y_{1_{mp}}$, $Y_{2_{mp}}$) and ($Y_{1_{ied}}$, $Y_{2_{ied}}$) are being sent in plaintext over the network, $\mathcal{A}$ may try to learn secret key.

However, OTP_{mp} and OTP_{ied} are not sent over the network, and $Y_{2_{ss-ied}}$ is sent over a secure network. $\mathcal{A}$ does not know G_{ied} and the actual identity of the *MP*. Therefore, $\mathcal{A}$ cannot generate secret key and also cannot trace *MP*'s identity.

- *Case-2: Adversary $\mathcal{A}$ as a rogue device:* $\mathcal{A}$ may install a fake device in place of a legitimate device. In such a case, $\mathcal{A}$ can extract the information provided by the user to the device and can later use that information to access the system from a valid device's interface. In order to protect such access, after receiving the message from the device, the AS_{ss} sends an *OTP* to the user in order to verify its identity. Hence, two-factor authentication takes place, one by matching TID_{ss-mp} and other by sending an *OTP*.
- *Case-3: Adversary $\mathcal{A}$ tries to extract information from the message:* $\mathcal{A}$ may try to extract some information from message-4 that is being sent over the network. $\mathcal{A}$ cannot decrypt the message as it is not able to generate the secret key.

c. Replay and Injection Attacks: $\mathcal{A}$ can intercept a message in order to perform a replay attack. It can also inject message information during communication over the network. The proposed scheme can resist replay attacks by using timestamp values T_i in all transmitted messages, and also x_{mp}, x_{ied}, x_{ss}, *OTP* are chosen randomly in each session. There can be three different cases under this scenario as follows:

- *Case-1: MP replay and injection attacks:* (1) $\mathcal{A}$ captures and later sends message-1 to the *IED*. On receiving the message, the *IED* detects that message-1 was resent, as the received timestamp T_1 is outdated. The message is considered valid only when $T_1 + T_{threshold} \leq T_{current}$, where $T_{threshold}$ is a threshold timestamp that is a maximum time considered for reaching a message from one entity to another. Hence, the connection is terminated.
  (2) $\mathcal{A}$ sends message-1 to the *IED* with a new timestamp $\mathcal{A}T_1$ and a new hash $\mathcal{A}H_2$. The *IED* forwards the message to the AS_{ss} for identity verification of the *MP*. The AS_{ss} checks the identity of the *MP* and finds a mismatch between the received *MP*'s identity and the stored identities of all the *MP*s. Hence, the connection is terminated.
- *Case-2: AS_{ss} replay and injection attacks:* (1) When $\mathcal{A}$ replays message-3 to the *IED*, it checks timestamp validity. If it is not valid, the *IED* refuses the message.
  (2) Message-3 is sent with a new $\mathcal{A}T_3$ and new $\mathcal{A}H_3$, then the *IED* computes user-role and compares the role received from the AS_{ss}. As $\mathcal{A}$'s role will be different, the *IED* terminates the connection.
- *Case-3: IED replay and injection attacks:* (1) $\mathcal{A}IED$ sends message-2 as a replay to the AS_{ss}, the AS_{ss} verifies timestamp validity. As it is outdated, the AS_{ss} discards the request. The same case applies when the *IED* sends a replay message-4 to the *MP*.
  (2) $\mathcal{A}$ modifies T_2 of the message-2 to a new $\mathcal{A}T_2$ and sends to the AS_{ss}. On receiving, the AS_{ss} finds $h \neq h'$, and terminates the connection. The same case applies when $\mathcal{A}$ sends message-4 to the *MP*.
  (3) $\mathcal{A}$ injects a new $\mathcal{A}T_2$ as well as a new $\mathcal{A}h$ in message-2 and sends it to the AS_{ss}. The AS_{ss} compares G_{mp} with the stored G_{mp}'s. Since it is an invalid G_{mp}, the AS_{ss} discards the request. Even if $\mathcal{A}$ modifies ID_{ied}, the connection is terminated, as AS_{ss} fails to verify received signature of the device.
  (4) $\mathcal{A}$ injects $\mathcal{A}T_4$ and $\mathcal{A}H_4$ to the message-4. However, $\mathcal{A}$ fails to generate the secret key shared between the *MP* and the *IED*. As a result, the message cannot not be decrypted correctly by *MP*'s secret key.

d. Redirection Attack: Each time when a new user tries to access a device, it has to provide location information to the device. Thereafter, the device verifies its location by computing and comparing $H_2 \overset{?}{=} H_2'$. This helps to detect redirection attacks in the *SG* system. We illustrate this scenario by the following two cases:

- *Case-1: MP redirection attack:* If $\mathcal{A}$ provides wrong location in message-1, on receiving the message, the *IED* verifies $H_2 \neq H_2'$ and refuses the message. The *IED* finds a mismatch between location information received from $\mathcal{A}$ and its stored information. Hence, the *IED* terminates the connection.
- *Case-2: IED redirection attack:* $\mathcal{A}$ can also send message-2 with a fake $\mathcal{A}ID_{ied}$ and ID_{ss}'. If the ID_{ss}' in message-2 is same as the communicated ID_{ss}, the AS_{ss} verifies $\mathcal{A}ID_{ied}$, otherwise it sends the message to the AS_c for verification. Since it is a fake $\mathcal{A}ID_{ied}$, the AS_c finds it invalid. As a result, the AS_c terminates the connection.

e. Attacks by the Consumer/Operator/Maintenance Personnel: The customer and maintenance personnel have read-only user-roles belonging to different departments, hence they cannot extract/modify other information. An operator (insider) cannot extract the actual identity of the user (ID_{mp}) from the received identity (in a substation group), *i.e.*, G_{mp}, over the network, as the identity changes after each session.

f. Other Attacks: Our scheme prevents the *SG* system against *Known Key Attack*. The server of the substation cannot generate the next session user's public identity G_{mp} from the previous one. Further, for each session key generation, private key x_{mp} is different and the public identity of each user is newly generated for each session at each device. The identity and signature verification used in the scheme prevent *Repudiation Attack*. A user can modify (or even access) the system only after the user authentication and authorization verification. Hence, a malicious user or attacker cannot *Change the Security Parameters* of the device.

5.3 Performance Analysis

Consider a *SG* network scenario with an authentication server remotely connected with various devices. The specification of the *PC* system is 1.70 *GHz* Core i3-4005U *CPU* with 4*GB RAM* and 500*GB* drive. We performed *IEC61850* Client/Server *IED* simulation on a *PC* with *The SmartGridware Java IEC61850* Client/Server *SDK* [40]. Furthermore, the performance of the proposed scheme is evaluated in terms of communication and computation overheads. We compare our scheme with the schemes presented in [39] and [35] because these are only comparable work that consider user/device-server authentication while accessing a device. However, these schemes use a password for each user-role.

i) Communication Overhead: In order to reduce communication overhead, the *MP* is assigned for accessing the devices from the same substation whenever is possible. Communication overhead is the total number of bits transmitted over the network during protocol/scheme/approach execution. Total communication overhead of our scheme is as follows:
Message-1: G_{mp}(128) + *Attributes*(64) + $ID^{'}_{ss}$(128) + $Y_{2_{mp}}$(128) + $Y_{1_{mp}}$(128) + T_1(64) + H_2(64) = 704 bits,
Message-2: G_{mp}(128) + $ID^{'}_{ss}$(128) + ID_{ied}(128) + $Y_{2_{ss-ied}}$(128) + S(128) + T_2(64) + h(64) = 768 bits,
Message-3: *Name*(128) + *Role*(128) + *SS*(64) + T_3(64) + *OTP*(3) + H_3(64) = 451 bits,
Message-4: *Name*(128) + *cRole*(128) + *SS*(64) + T_3(64) + $Y_{2_{ied}}$(128) + $Y_{1_{ied}}$(128) + H_4(64) = 704 bits,
Total overhead = 2627 bits (328.375 bytes).

As shown in Table 4, the overheads of our scheme between the *MP-IED* and the *IED-AS* are 1408 bits and 1219 bits, respectively. The total communication overhead of our scheme is lower than the scheme [39] adding integrity check and timestamp values, as well as scheme [35]. We assume that there are m-number of *MP*s that are accessing different devices within a substation simultaneously. In this case, the total communication overhead (for the first attempt) of our scheme, scheme [39], and scheme [35] would be 2627×m, 2752×m, and 2944×m, respectively. Further, we assume that r-number of attempts are allowed for accessing the same device within a session. For any subsequent authentication, our scheme generates 256×r communication overhead (since we use *MAES* with 256 bits of block size sending mode, location, department, *SDP* in encrypted form), while scheme [39] and scheme [35] produce 2752×r and 2944×r, respectively. As shown in Figure 8, our scheme is able to provide authorization and authentication to different *MP*s accessing different devices without increasing the overhead.

ii) Computation Overhead: We assume the operation under multiplication group as M, pairing function as P, hash function as H, operation under addition group (addition) as A, subtraction as S, encryption as E, decryption as D, probability generation function as *Gen*, signature as *Sig*, key generation *KG*, message authentication code as *MAC*, verification function as *Ver*, reproduction algorithm as *Rep*, user credential generation as C, device credential generation as *DE*, and authentication server credential generation as *AG*. We compute the overhead for a single *MP* scheme run at *MP*, *IED*, and *AS* as presented in Table 5.

The total computation overhead of scheme [39] with integrity and timestamp values, and scheme [35] are 14M, 21H, 12A, 1S and 2*Sig*, 4E, 4D, 2*Ver*, 2*KG*, 4H, 2*Gen*, 2*MAC*, 1S, 1*Rep*, 3*AG*, 5C, 5*DE*, respectively, while for our scheme it is 1P, 14M, 9H, 2A, 1E, 1D. If we assume a unit value for each operation, then we can say that our scheme is efficient than schemes in [39] and [35], as the total number of computations (operations) performed by our scheme is 29, while it is 48 and 37 for the schemes in [39] and [35], respectively. The actual computation time by each scheme depends upon the actual time taken by each operation. Furthermore, for multiple authentications scenario, scheme [39], scheme [35], and our scheme generate 48×m, 37×m, and 30×m-1 computation overhead, respectively, as shown in Figure 9 assuming a unit value for each function.

iii) Discussion: In order to measure time required for computing and checking the user-role, we implemented H_1 as *SHA256* function on Intel i3, Window7 in *Java*, which took 20 milliseconds (ms). In order to implement pairing function, we converted a hash string to an octet string and then an elliptic curve point [54]. A pairing function using J-pairing is performed in 197 ms, while it took 246 ms for all multiplications (scalar and elliptic curve). Further, hashing, addition, subtraction, *AES* encryption and decryption took 20 ms, 0.03 ms, 0.03 ms, 0.23 ms, and 0.13 ms, respectively. The average mobile broadband download speed on *4G* (15.1 *Mbit/s*) is more than twice as fast as *3G* (6.1 *Mbit/s*) [51]. Our scheme takes 0.45 ms on *3G* network while 0.18 ms on *4G* network to transmit all 4-messages over the network. Overall, the execution time for our scheme is 0.62 s while it is 0.67 s for the scheme in [39]. The structures of various functions are not defined in scheme [35]. Therefore, it is not possible to compute total execution time of the scheme.

To better demonstrate the advantage of our scheme, we also conduct simulations in *Java* in comparison with two separate authentication and authorization schemes instead of our authentication and authorization scheme. Considering two schemes instead of one require two separate connections between *MP-IED* and *IED-AS* by each user. Here, the user needs to generate two temporary identities, respectively, for during authentication and during authorization. Also, since the computed roles H_1 at *MP* and *IED* are used in $Y_{1_{mp}}$ and $Y_{1_{ied}}$, respectively, and are later used in computing the shared secret key for authentication, H_1

TABLE 4: Communication Overhead (in bits)

Schemes	User Registration	User-Device	Device-Server	Total
Proposed Scheme	–	1408 (*MP*−*IED*)	1219 (*IED*−*AS*)	2627
Scheme [39] with integrity & timestamp	832 (*UA*−*TA*)	960 (*UA*−*SSC*)	960 (*SSC*−*IED*)	2752
Scheme [35]	1664 (*C*−*AS*)	–	1280 (*D*−*AS*)	2944

TABLE 5: Computation Overhead

Schemes	*TA*	*MP/UA*	*IED*	*AS/SSC*
Proposed Scheme	–	4*M*, 2*H*, 1*D*, 1*P*	6*M*, 5*H*, 1*A*, 1*E*	4*M*, 2*H*, 1*A*
Scheme [39] with integrity & timestamp	7*M*, 4*H*, 4*A*	4*M*, 8*H*, 5*A*, 1*S*	1*M*, 4*H*, 2*A*	2*M*, 5*H*, 1*A*
Scheme [35]	–	2*Gen*, 1*E*, 1*S*, 1*MAC*, 1*D*, 1*Rep*, 5*C*, 2*H*	2*D*, 2*Ver*, 5*DE*	2*Sig*, 3*E*, 2*KG*, 1*D*, 2*H*, 3*AG*, 1*MAC*

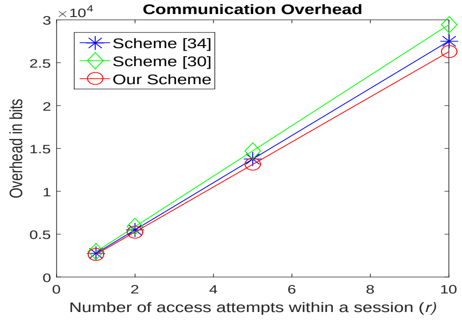


Fig. 8: Communication overhead for session authentications.

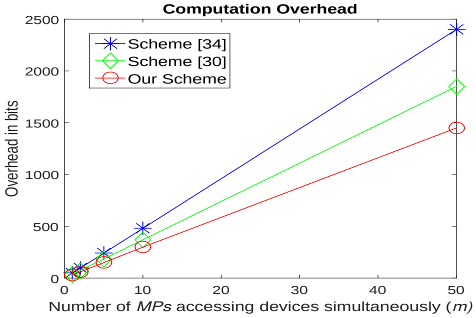


Fig. 9: Computation overhead for multiple *MP*s.

needs to be computed twice. We simulated these schemes for 50 users belonging to the same authentication server simultaneously accessing different devices. The connection establishment time is 3216 ms for our scheme, and 3212 ms and 2042 ms, respectively, for the separate authentication and authorization schemes. On average, the execution time per user is 3.83 s for our scheme, while 3.81 s and 2.08 s, respectively, for the authentication scheme and authorization scheme. Hence, since the two separate schemes take 5.89 s in total, they are 35% slower than our scheme that executes authentication and authorization simultaneously together.

6 Formal Proof

This section presents the formal proof of the proposed scheme using a well-known *BAN-Logic* [52] as well as automatic security verification tool named *Proverif* [53].

1) Security Proof using BAN-Logic: In order to justify our analysis, we use *BAN-Logic* symbols to formally proof our scheme. Notations of *BAN-Logic* can be followed in [52].

1) The formal messages in the proposed scheme:
(1) $MP_i \rightarrow IED_i$: $ID^{'}_{ss_i}$, $Attributes$, $Y_{2_{mp_i}}$, $Y_{1_{mp_i}}$, G_{mp_i}, $T_{1_i}, H_{2_i}; TID_{ss-mp_i} = x_{ss}G_{mp_i}$;
(2) $IED_i \rightarrow AS_{ss_i}$: $ID^{'}_{ss_i}, ID_{ied_i}, G_{mp_i}, Y_{2_{ss-ied_i}}, S_i, T_{2_i}, h_i$;
(3) $AS_{ss_i} \rightarrow IED_i$: $Name_i, Role_i, SS_i, OTP_i, T_{3_i}, H_{3_i}$;
(4) $IED_i \rightarrow MP_i$: $E_K\{Name_i, cRole_i, SS_i\}$, T_{4_i}, H_{4_i}; $cRole_i = Role_i$;

2) Security Assumption: The *MP* and the *IED* share a secret key, *i.e.*, $K_{mp_i} = K_{ied_i}$.
(1) *MP* has a secure key K_{mp_i} and $MP_i|\equiv MP_i \overset{K_{mp_i}}{\leftrightarrow} IED_i$.
(2) *IED* has a secure key K_{ied_i} and $IED_i|\equiv IED_i \overset{K_{ied_i}}{\leftrightarrow} MP_i$.

3) Message Meaning Rule:
(1) $\frac{MP_i|\equiv(MP_i \overset{K_{mp_i}}{\leftrightarrow} IED_i), MP_i \triangleleft TID_{ss-mp_i}}{MP_i|\equiv AS_i|\sim G_{mp_i}}$
(2) $\frac{IED_i|\equiv(IED_i \overset{K_{ied_i}}{\leftrightarrow} MP_i), AS_i \triangleleft TID_{ss-mp_i}}{AS_i|\equiv MP_i|\sim G_{mp_i}}$

4) Timestamp Verification Rule:
(1) $\frac{MP_i|\equiv\#(T_i), MP_i|\equiv AS_i|\sim G_{mp_i}}{MP_i|\equiv AS_i|\equiv G_{mp_i}}$
(2) $\frac{AS_i|\equiv\#(T_j), AS_i|\equiv MP_i|\sim G_{mp_i}}{AS_i|\equiv MP_i|\equiv G_{mp_i}}$

5) Jurisdiction Rule:
(1) $\frac{MP_i|\equiv AS_i \Rightarrow TID_{ss-mp_i}, MP_i \triangleleft MP_i|\sim TID_{ss-mp_i}}{MP_i|\equiv AS_i}$
(2) $\frac{AS_i|\equiv MP_i \Rightarrow TID_{ss-mp_i}, AS_i \triangleleft AS_i|\sim TID_{ss-mp_i}}{AS_i|\equiv MP_i}$

6) Protocol Goals:
a. Mutual Authentication: $MP_i|\equiv IED_i|\equiv AS_i \wedge AS_i|\equiv IED_i|\equiv MP_i \rightarrow MP_i|\equiv IED_i \wedge AS_i$. Thus, mutual authentication holds.
b. Session Key Agreement: Each K_i key between each MP_i and the IED_i provides session key agreement.
c. Freshness of messages: $AS_i|\equiv\#(T_j) \wedge MP_i|\equiv\#(T_j)$, Thus, freshness of messages between MP_i, IED_i, and AS_i hold.
d. Integrity and Privacy between each MP_i and the IED_i:
(1) $\frac{MP_i|\equiv(MP_i \overset{K_{mp_i}}{\leftrightarrow} IED_i), MP_i \triangleleft H(Msg)}{MP_i|\equiv IED_i|\sim Msg}$
(2) $\frac{IED_i|\equiv(IED_i \overset{K_{ied_i}}{\leftrightarrow} MP_i), MP_i \triangleleft TID_{ss-mp_i}}{MP_i|\equiv IED_i|\sim G_{mp_i}}$

2) Security Proof using Proverif: The following are the input and output observed from the *Proverif* tool:
(* Public channel between the *MP* and the *IED* *)
free *pubChannel* : channel.
(* Secure channel between the *IED* and the *SS* *)
free *secureChannel* : channel [private].
(* types *)
type key. type ident. type nonce. type msgHdr.
type resp. type sessKey. type hash.
(* constant message headers *)
const MSG_1,MSG_2,MSG_3,MSG_4,CMC,MSG: msgHdr.
(* Functions *)
fun sha256 (nonce): hash.
fun sha3842 (ident,nonce,nonce,nonce,nonce,nonce): hash.
fun sha3844 (bitstring,nonce,nonce,nonce): hash.
fun sha3843 (nonce,hash,nonce,nonce): hash.
fun sha384h (ident,ident,nonce,nonce,nonce,nonce): hash.
fun tempid (nonce,key): ident.
fun findrole (hash,hash): hash.
fun e(nonce, nonce,nonce,nonce): sessKey.
fun cipherfun(nonce,hash,nonce): bitstring.
fun sencrypt (bitstring,sessKey): bitstring.
reduc forall m: bitstring, k: sessKey;
sdecrypt(sencrypt(m,k),k) = m.
(* Key table consists of pairs (ident,key) shared between *MP* and *SS*. Table is not accessible by the attacker *)
free s: bitstring [private].
query attacker(s).
(* K_i is secret if and only if all K_i are secret *)
free K_i: sessKey [private].
query attacker(K_i).
not attacker(new x_{mp}).
(* Authentication queries *)
event begIED(nonce,sessKey).
event endIED(nonce,sessKey).
event begMP(nonce,sessKey).
event endMP(nonce,sessKey).
event begIED(msgHdr). event endIED(msgHdr).

event begMP(msgHdr). event endMP(msgHdr).
query x_1: nonce, x_2: sessKey;
event(endIED(x_1,x_2)) ==> event(begIED(x_1,x_2)).
event(endMP(x_1,x_2)) ==> event(begMP(x_1,x_2)).
event(endIED(MSG)) ==> event(begIED(MSG)).
event(endMP(MSG)) ==> event(begMP(MSG)).
event enableEnc .
(* When the attacker knows s, the event enbleEnc has been executed by attacker. *)
query attacker(s) ==> event(enableEnc).

let processMP =
(* The ident and pre-shared key of the *MP* *)
new ID_{mp}: ident, x_{mp}: key, G_{mp}: nonce, $Y_{1_{mp}}$: nonce, $Y_{2_{mp}}$: nonce, $Y_{2_{ss-mp}}$: nonce, *attr*: nonce, T_1: nonce;
new x_{ss}: key, $K_{i_{mp}}$: sessKey, ID_{ss}: ident, OTP_{mp}: nonce;
let H_2:hash=sha3842(ID_{ss},*attr*,$Y_{1_{mp}}$,$Y_{2_{mp}}$,G_{mp},T_1) in
let H_1: hash = sha256(*attr*) in
(** Note: compute G_{mp}, $Y_{1_{mp}}$, $Y_{2_{mp}}$, and $Y_{2_{ss-mp}}$ **)
let TID_{mp}: ident = tempid(G_{mp},x_{ss}) in
out ($pubChannel$,(MSG_1,ID_{ss},H_2,*attr*,$Y_{1_{mp}}$,$Y_{2_{mp}}$,G_{mp},T_1));
event begIED(MSG_4);
in($pubChannel$,(=MSG_4,*cipher*: bitstring,H_{44}: hash, $Y_{2_{ied}}$: nonce,$Y_{1_{ied}}$: nonce,T_4: nonce));
let H_4: hash = sha3844(*cipher*,T_4,$Y_{2_{ied}}$,$Y_{1_{ied}}$) in
if H_{44} = H_4 then
event endIED (MSG_4);
(* Note: compute OTP_{mp} after receiving OTP *)
let $K_{i_{mp}}$: sessKey = e($Y_{1_{ied}}$,$Y_{2_{ied}}$,G_{mp},OTP_{mp}) in
in($pubChannel$,(=CMC,$enableEnc_{mp}$: bool));
(** Note: verify *cRole* **)
event begMP(G_{mp},$K_{i_{mp}}$);
(*Receive message from *IED* *)
in($pubChannel$,(=MSG,*cipher*: bitstring));
let *msgcontent*: bitstring=sdecrypt(*cipher*,$K_{i_{mp}}$) in 0.
out($pubChannel$,sencrypt(msg_1: bitstring,$K_{i_{mp}}$));
if $enableEnc_{mp}$ = true then
in($pubChannel$,(=MSG,msg_2: bitstring));
let $msgcontent_2$: bitstring = sdecrypt(msg_2,$K_{i_{mp}}$) in 0.
event endMP(G_{mp},$K_{i_{mp}}$);

let processIED =
new ID_{ied}: ident, x_{ied}: key, G_{ied}: nonce, $Y_{1_{ied}}$: nonce, OTP_{ied}: nonce, new $Y_{2_{ied}}$: nonce, $Y_{2_{ss-ied}}$: nonce;
new *attr*: nonce, T_2: nonce, new T_4: nonce, $K_{i_{ied}}$: sessKey, H_{111}: hash, S: nonce, $cipher_3$: bitstring;
event begMP(MSG_1);
in($pubChannel$,(=MSG_1,ID_{ss}:ident,H_{21}:hash,$attr_1$:nonce, $Y_{1_{mp1}}$: nonce,$Y_{2_{mp1}}$: nonce,G_{mp1}: nonce,T_{11}: nonce));
let H_{211}: hash=sha3842(ID_{ss1},$attr_1$,$Y_{1_{mp1}}$,$Y_{2_{mp1}}$,G_{mp1},T_{11}) in
if H_{21} = H_{211} then
event endMP(MSG_1);
let H_{11}: hash = sha256($attr_1$) in
let *crole*: hash = findrole(H_{11},H_{111}) in
if H_{11} = H_{111} then
(** Note: compute G_{ied}, $Y_{1_{ied}}$, $Y_{2_{ied}}$, $Y_{2_{ss-ied}}$, and S **)
let h: hash = sha384h(ID_{ss1},ID_{ied},G_{mp1},$Y_{2_{ss-ied}}$,S,T_2) in
out($secureChannel$,(MSG_2,ID_{ss1},ID_{ied},G_{mp1},$Y_{2_{ss-ied}}$,S, T_2,h));
in($secureChannel$,(=MSG_3,$name_1$: nonce,$role_1$: hash, SS_1: nonce,OTP: nonce,T_{31}: nonce,H_{31}: hash));
let H_{311}: hash = sha3843($name_1$,$role_1$,SS_1,T_{31}, OTP) in
if H_{31} = H_{311} then
if *crole* = $role_1$ then
(** Note: compute OTP_{ied} **)
let $K_{i_{ied}}$: sessKey = e($Y_{1_{mp1}}$,$Y_{2_{mp1}}$,OTP_{ied},$Y_{2_{ss-ied}}$) in
let $cipher_3$ = cipherfun($name_1$,$role_1$,SS_1) in
let H_4: hash = sha3844($cipher_3$,T_4,$Y_{2_{ied}}$,$Y_{1_{ied}}$) in
out($pubChannel$, (MSG_4,$cipher_3$,H_4,$Y_{2_{ied}}$,$Y_{1_{ied}}$,T_4));
(**IED* encrypts messages and send to the *MP**)
new $enableEnc_{ied}$: bool, *msg*: bitstring;
event begIED(G_{ied},$K_{i_{ied}}$);
if $enableEnc_{ied}$ = false then
event enableEnc;
out($pubChannel$,(MSG,s))
else
out($pubChannel$,(MSG,sencrypt(s,$K_{i_{ied}}$)))
(*Send out cipher mode command *)
out($pubChannel$,(CMC,$enableEnc_{ied}$));
out($pubChannel$,sencrypt($cipher_3$,$K_{i_{ied}}$));
if $enableEnc_{ied}$ = true then
in($pubChannel$,(=MSG,msg_1: bitstring));
let $msgcontent_3$: bitstring = sdecrypt(msg_1,$K_{i_{ied}}$) in 0.
out($pubChannel$,sencrypt(msg_2: bitstring,$K_{i_{ied}}$));
event endIED(G_{ied},$K_{i_{ied}}$);

let processSS =
new $name_{22}$: nonce, $role_{22}$: hash, SS_{22}: nonce;
new x_{ss1}: key, T_3: nonce;
let H_3: hash = sha3843 ($name_{22}$,$role_{22}$,SS_{22},T_3) in
in($secureChannel$,(=MSG_2, ID_{ss11}: ident, ID_{ied1}: ident, G_{mp11}: nonce, $Y_{2_{ss-ied1}}$: nonce, S_1: nonce, T_{21}: nonce, h_1: hash));
let h_{11}: hash = sha384h(ID_{ss11},ID_{ied1},G_{mp11},$Y_{2_{ss-ied1}}$, S_1, T_{21}) in
if h_1 = h_{11} then
(** Note: verify ID_{ied} and TID_{ss-mp} **)
let TID_{mp2} = tempid(G_{mp11},x_{ss1}) in
let H_3: hash = sha3843($name_{22}$,$role_{22}$,SS_{22},T_3,OTP) in
out($secureChannel$,(MSG_3,$name_{22}$,$role_{22}$,SS_{22},OTP,T_3, H_3));
0.
process
((! processMP) | processIED | processSS)

Run & Output:

```
Neetesh@Neetesh-PC /proverif1.88
$ ./proverif examples/insiderattack.pv
```

–*Query attacker(s[])==>event(enableEnc)*
Completing...ok, secrecy assumption verified:
fact unreachable attacker (x_{mp}[!1 = v_1037])
Starting query attacker(s[])==>event(enableEnc)
RESULT attacker(s[])==>event(enableEnc) is true.
–*Query event(endMP(MSG_1))==>event(begMP(MSG_1))*
Completing...ok, secrecy assumption verified:
fact unreachable attacker (x_{mp}[!1=v_2164])
Starting query event(endMP(MSG_1))==>
event(begMP(MSG_1))
goal reachable: end(endMP(G_{mp}[!1=@sid_2344]))
Abbreviations: G_{mp}_2365=G_{mp}[!1=@sid_2363]
RESULT event(endMP(MSG_1))==>

event(begMP(MSG_1)) is true.
*–Query event(endIED(x_1,x_2_2488)) $==>$
event(begIED(x_1,x_2_2488))*
Completing...ok, secrecy assumption verified:
fact unreachable attacker (x_{mp}[!1=v_3377])
Starting query event(endIED(x_1,x_2_2488))
$==>$event(begIED(x_1,x_2_2488)) goal reachable:
attacker($Y_{2_{ied}}$_3600) && attacker($Y_{1_{ied}}$_3601) &&
begin(begIED(G_{mp}[!1=@sid_3602],
$K_{i_{mp}}$[!1=@sid_3602]))$->$ end(endIED(G_{mp}[!1=@sid_3602],
e($Y_{1_{ied}}$_3601, $Y_{2_{ied}}$_3600,
G_{mp}[!1=@sid_3602], $Y_{2_{ss-mp}}$[!1=@sid_3602])))
Abbreviations: G_{mp}_3622=G_{mp}[!1=@sid_3619]
$Y_{2_{ss-mp}}$_3623=$Y_{2_{ss-mp}}$[!1=@sid_3619]
$K_{i_{mp}}$_3624=$K_{i_{mp}}$[!1=@sid_3619]
RESULT event(endIED(x_1,x_2_2488))$==>$
event(begIED(x_1,x_2_2488)) is true.
–Query not attacker(K_i[])
Completing...ok, secrecy assumption verified:
fact unreachable attacker (x_{mp}[!1=v_4614])
Starting query not attacker(K_i[])
RESULT not attacker(K_i[]) is true.
–Query not attacker(s[])
Completing...ok, secrecy assumption verified:
fact unreachable attacker (x_{mp}[!1=v_5650])
Starting query not attacker(s[])
RESULT not attacker(s[]) is true.

7 Conclusion

In this paper, we proposed a user authentication and authorization scheme for accessing many different types devices in the *SG*. Our scheme can be easily applied to different user-roles, such as auditors, operators, and etc., who access different devices in the *SG* system, as each user-role is computed dynamically based on attribute-based access control using a *SHA256* hash function with (*mode of access*, *department*, *location*, *SDP*) attributes provided by each user. Our scheme enables two-factor authentication so that a rogue device could not re-use the previous captured information of a legitimate user. A bilinear pairing cryptography-based shared secret key is generated between the user and the device for further secure communications within a session. The proposed scheme is efficient in terms of both, communication and computation overheads in comparison with the existing schemes, and is able to defeat many well-known outsider attacks as well as insider attacks. The correctness of the proposed scheme is confirmed by the formal proof with *BAN-Logic* as well as by *Proverif*.

Acknowledgments

This research was supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ICT Consilience Creative Program (IITP-2015-R0346-15-1007) supervised by the IITP (Institute for Information & communications Technology Promotion) and under the Basic Science Research Program (2013010489) through the NRF (National Research Foundation).

References

[1] White paper, "Cyber-physical systems security for smart grid," Power Systems Engineering Research Center, Feb. 2012, 29 pages.

[2] N. M. Pindoriya, D. Dasgupta, D. Srinivasan, and M. Carvalho, "Infrastructure security for smart electric grids: a survey," V. Pappu et al. (eds.) *Optimization and Security Challenges in Smart Power Grids, Energy Systems*, 2013, pp. 161-172.

[3] "Smart grid cyber security potential threats, vulnerabilities and risks," Public Interest Energy Research (PIER) Program Interim Report, California State University Sacramento, May 2012, 83 pages.

[4] "Guidelines for smart grid cyber security: vol. 3, supportive analyses and references," NISTIR 7628, The Smart Grid Interoperability Panel - Cyber Security Working Group, Aug. 2010.

[5] "A secure distribution area network architecture for smart grids," Tropos Wireless Communication Systems, ABB, 2012. [Online]. https://library.e.abb.com/public/e61e21b95fd4482585257bcb006d71d6/GridCom-secure-distribution-network-architecture.pdf.

[6] C. H. Lo and N. Ansari, "Decentralized controls and communications for autonomous distribution networks in smart grid," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 66-77, Mar. 2013.

[7] "Smart grid information assurance and security technology assessment," Energy Research and Development Division, Final Project Report, 2010. [Online]. http://www.energy.ca.gov/2013publications/CEC-500-2013-056/CEC-500-2013-056.pdf.

[8] N. Saxena and B. J. Choi, "State of the art authentication, access control, and secure integration in smart grid," *Energies*, vol. 8, pp. 11883-11915, Oct. 2015.

[9] H. Khurana and M. Hadley, "Smart-grid security issues," *IEEE Security & Privacy Magazine*, vol. 8, no. 1, pp. 81-85, Feb. 2010.

[10] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Systems Journal*, vol. 8, no. 2, pp. 655-662, May 2014.

[11] H. Nicanfar, P. Jokar, K. Beznosov, and V. C. M. Leung, "Efficient authentication and key management mechanisms for smart grid communication," *IEEE Systems Journal*, vol. 8, no. 2, pp. 629-640, Jun. 2014.

[12] P. Deng and L. Yang, "A secure and privacy-preserving communication scheme for Advanced Metering Infrastructure," in *Proc. PES Innovative Smart Grid Technolo.*, Columbia, USA, Jan. 2012, pp. 1-5.

[13] H. Liu, H. Ning, Y. Zhang, and L. Yang, "Aggregated-proofs based privacy-preserving authentication for V2G networks in the smart grid," *IEEE Tr. on Smart Grid*, vol. 3, no. 4, pp. 1722-1733, Dec. 2012.

[14] Z. Baharlouei and M. Hashemi, "Efficiency-fairness trade-off in privacy-preserving autonomous demand side management," *IEEE Transactions on Smart Grid*, vol. 5, no. 2, pp. 799-808, Mar. 2014.

[15] D. Liu, H. Li, Y. Yang, and H. Yang, "Achieving multi-authority access control with efficient attribute revocation in smart grid," in *Proc. IEEE Communication and Information Systems Security Symposium*, Sydney, Australia, Jun. 2014, pp. 634-639.

[16] S. Ruj and A. Nayak, "A decentralized security framework for data aggregation and access control in smart grids," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 160-169, Mar. 2013.

[17] T. M. Chen, J. C. Sanchez-Aarnoutse, and J. Buford, "Petri net modeling of cyber-physical attacks on smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 741-749, Dec. 2011.

[18] A. H. Mohsenian-Rad and A. Leon-Garcia, "Distributed internet-based load altering attacks against smart power grids," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 667-674, Dec. 2011.

[19] X. Liu and Z. Li, "Local load redistribution attacks in power systems with incomplete network information," *IEEE Transactions on Smart Grid*, vol. 5, no. 4, pp. 1665-1676, Jul. 2014.

[20] C. W. Ten, J. Hong, and C. C. Liu, "Anomaly detection for cybersecurity of the substations," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 865-873, Dec. 2011.

[21] O. Kosut, L. Jia, R. J. Thomas, and L. Tong, "Malicious data attacks on the smart grid," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 645-658, Dec. 2011.

[22] M. Esmalifalak, G. Shi, Z. Han, and L. Song, "Bad data injection attack and defense in electricity market using game theory study," *IEEE Transactions on Smart Grid*, vol. 4, no. 1, pp. 160-169, Mar. 2013.

[23] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks," *IEEE Transactions on Smart Grid*, vol. 4, no. 3, pp. 1244-1253, Sep. 2013.

[24] B. Sikdar and J. H. Chow, "Defending synchrophasor data networks against traffic analysis attacks," *IEEE Transactions on Smart Grid*, vol. 2, no. 4, pp. 819-826, Dec. 2011.

[25] Y. Yang, K. McLaughlin, T. Littler, S. Sezer, G. Eul, Z.Q. Yao, and H. F. Wang, "Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems," in *Proc. SUPERGEN*, Hangzhou, China, Sep. 2012, pp. 1-8.

[26] L. Shichao, X. P. Liu, and A. E. Saddik, "Denial-of-service (DoS) attacks on load frequency control in smart grids," in *Proc. PES Innovative Smart Grid Tech.*, Washington, USA, Feb. 2013, pp. 1-6.

[27] I. Stojmenovic, "Access control in distributed systems: merging theory with practice," in *Proc. IEEE TrustCom*, Changsha, China, Nov. 2011, pp. 1-2.

[28] H. Cheung, A. Hamlyn, T. Mander, C. Yang, and R. Cheung, "Role based model security access control for smart power-grids computer networks," in *Proc. PES General Meeting*, Pennsylvania, USA, Jul. 2008, pp. 1-7.

[29] D. Rosic, U. Novak, and S. Vukmirovic, "Role-based access control model supporting regional division in smart grid System," in *Proc. ICCICSN*, Tainan, Taiwan, Jun. 2013, pp. 197-201.

[30] B. Vaidya, D. Makrakis, and H. T. Mouftah, "Device authentication mechanism for smart energy home area networks," in *Proc. IEEE ICCE*, Las Vegas, USA, Jan. 2011, pp. 787-788.

[31] E. Ayday and S. Rajagopal, "Secure, intuitive, and low-cost device," in *Proc. IEEE CCNC*, Las Vegas, USA, Jan. 2011, pp. 1161-1165.

[32] Y. S. Kim and J. Heo, "Device authentication protocol for smart grid systems using homomorphic hash," *Journal of Communications and Networks*, vol. 14, no. 6, pp. 606-613, Dec. 2012.

[33] A. C. F. Chan and J. Zhou, "Cyberphysical device authentication for the smart grid electric vehicle ecosystem," *IEEE J. on Selected Areas in Communications*, vol. 32, no. 7, pp. 1509-1517, Jul. 2014.

[34] S. Lee, J. Bong, S. Shin, and Y. Shin, "A security mechanism of smart grid AMI network through smart device mutual authentication," in *Proc. ICOIN*, Phuket, Thailand, Jan. 2014, pp. 592-595.

[35] X. Huang, Y. Xiang, E. Bertino, J. Zhou, and L. Xu, "Robust multi-factor authentication for fragile communications," *IEEE Tr. on Dependable & Secure Comp.*, vol. 11, no. 6, pp. 568-581, Dec. 2014.

[36] S. Kim, E. Y. Kwon, M. Kim, J. H. Cheon, S. H. Ju, Y. H. Lim, and M. S. Choi, "A secure smart-metering protocol over power-line communication," *IEEE Transactions on Power Delivery*, vol. 26, no. 4, pp. 2370-2379, Oct. 2011.

[37] S. H. Seo, X. Ding, and E. Bertino, "Encryption key management for secure communication in smart advanced metering infrastructures," in *Proc. SmartGridComm*, Vancouver, Oct. 2013, pp. 498-503.

[38] G. Gilchrist, "Secure authentication for DNP3," in *Proc. PES General Meeting*, Pennsylvania, USA, July 2008, pp. 1-3.

[39] B. Vaidya, D. Makrakis, and H. Mouftah, "Provisioning Substation-level Authentication in the Smart Grid Networks," in *Proc. Military Communi.*, Baltimore, USA, Nov. 2011, pp. 1189-1194.

[40] SmartGridware IEC61850 Java Software Solutions. [Online]. Available: http://www.smartgridware.com.

[41] R. Tabassum, K. Nahrstedt, E. Rogers, and K. S. Lui, "SCAPACH: scalable password-changing protocol for smart grid device authentication," in *Proc. ICCCN*, Nassau, Bahamas, Aug. 2013, pp. 1-5.

[42] E. Coyne and T. R. Weil, "ABAC and RBAC: scalable, flexible, and auditable access management," *Insecure IT, IT Pro*, Published by the IEEE Computer Society, pp. 14-16, May/Jun. 2013.

[43] J. Xin, R. Krishnan, and R. Sandhu, "A role-based administration model for attributes," in *Proc. WSRAS*, Minneapolis, USA, Sep. 2012, pp. 7-12.

[44] White Paper, "Best Practices in Enterprise Authorization: The RBAC/ABAC Hybrid Approach," *EmpowerID*, 2013. [Online]. http://blog.empowerid.com/Portals/174819/docs/EmpowerID-WhitePaper-RBACABAC-Hybrid-Model.pdf.

[45] D. M. Freeman, "Converting pairing-based cryptosystems from composite-order groups to prime-order groups," in *Proc. 29th EUROCRYPT*, French Riviera, May/Jun. 2010, pp. 44-61.

[46] F. Zhang and K. Kim, "ID-based blind signature and ring signatures from pairings," in *Proc. ASIACRYPT*, Dunedin, New Zealand, Dec. 2002, pp. 533-547.

[47] B. Lynn, "On the implementation of pairing-based cryptosystems," PhD Dissertation, Stanford University, 2007. [Online] Available: https://crypto.stanford.edu/pbc/thesis.pdf.

[48] D. Boneh and M. K. Franklin, "Identity-based encryption from the weil pairing," in *Proc. CRYPTO*, S.B., USA, Aug. 2001, pp. 213-229.

[49] N. Saxena and N. S. Chaudhari, "EasySMS: a protocol for end-to-end secure transmission of SMS," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 7, pp. 1157-1168, Jul. 2014.

[50] J. L. Huang, L. Y. Yeh, and H. Y. Chien, "ABAKA: an anonymous batch authenticated and key agreement scheme for value-added services in vehicular ad hoc networks," *IEEE Transaction on Vehicular Technology*, vol. 60, no. 1, pp. 248-262, Jan. 2011.

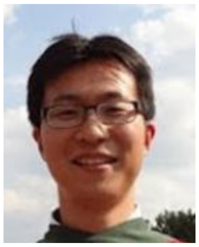
[51] "Measuring mobile broadband performance in the UK," 13 Nov. 2014. [Online]. http://stakeholders.ofcom.org.uk/binaries/research/broadband-research/mbb-nov14.pdf.

[52] J. Wessels, "Applications of BAN-logic," CMG Finance B.V., 2001. [Online]. win.tue.nl/ipa/archive/springdays2001/banwessels.pdf.

[53] ProVerif: cryptographic protocol verifier. [Online]. http://prosecco.gforge.inria.fr/personal/bblanche/proverif.

[54] "SEC1: Elliptic Curve Cryptography," Standards for Efficient Cryptography, Certicom Research, 2000. [Online]. www.secg.org/SEC1-Ver- 1.0.pdf.

Netesh Saxena (S'09-M'14) received his PhD in Computer Science & Engineering from the Indian Institute of Technology, Indore, India. He is currently a Post-Doctoral Researcher at the Department of Computer Science, State University of New York Korea, South Korea, and a Visiting Researcher at the Department of Computer Science, Stony Brook University, USA. In 2013-14, he was a Visiting Research Student and a DAAD Scholar at Bonn-Aachen International Center for Information Technology (B-IT), Rheinische-Friedrich-Wilhelms Universitt, Bonn, Germany. He was also a TCS Research Scholar during Jan. 2012 - Apr. 2014. He works in the area of security and privacy. His current research interests include smart grid security, vehicle-to-grid security and privacy, cryptography, security and privacy in the cellular networks, and secure mobile applications. He has published several papers in various international peer-reviewed journals and conferences. He is a member of IEEE, ACM, and CSI.



Bong Jun Choi (S'09-M'11) received his B.Sc. and M.Sc. degrees from Yonsei University, Korea, both in electrical and electronics engineering, and the Ph.D. degree from University of Waterloo, Canada, in electrical and computer engineering. He is currently an assistant professor at the Department of Computer Science, State University of New York Korea, Korea, and jointly a research assistant professor at the Department of Computer Science, Stony Brook University, USA. His current research focuses on energy efficient networks, distributed mobile wireless networks, smart grid communications, and network security. He serves as an editor of KSII Transactions on Internet and Information Systems and a member of the Smart Grid Core Security Technology Development Steering Committee, Korea. He also serves on the technical program committees for many international conferences such as IEEE PECON, IFIP NTMS, and IEEE CMC. He is a member of the IEEE and the ACM.

Rongxing Lu (S'09-M'11-SM'15) received the Ph.D degree in computer science from Shanghai Jiao Tong University, Shanghai, China in 2006 and the Ph.D. degree (awarded Canada Governor General Gold Medal) in electrical and computer engineering from the University of Waterloo, Waterloo, Ontario, Canada, in 2012. Since May 2013, he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, as an Assistant Professor. His research interests include computer, network and communication security, applied cryptography, security and privacy analysis for vehicular network, eHealthcare system, and smart grid communications. He won the IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award in 2013.