Securing Restricted Publisher-Subscriber Communications in Smart Grid Substations

Neetesh Saxena[1], Santiago Grijalva[2], and Bong Jun Choi[3]
[1]Department of Computing & Informatics, Bournemouth University, UK
[2]School of Electrical & Computer Engineering, Georgia Institute of Technology, USA
[3]Department of Computer Science, The State University of New York, Korea & Stony Brook University, USA
Email: nsaxena@ieee.org, sgrijalva@ece.gatech.edu, bjchoi@sunykorea.ac.kr

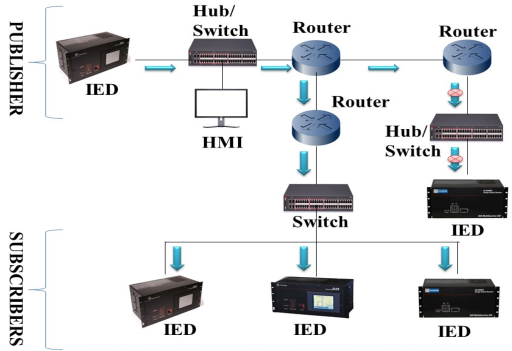
***Abstract*—Smart Grid applications require accurate and correct data transmission from publisher to subscribers with critical communication latency requirements. Since the smart grid is being supported by distributed communication networks, deployed using various wired and wireless technologies, including IP-based networks, securing the communication infrastructure is both critically important and challenging. In this paper, we propose a secure and efficient data delivery scheme, based on a restricted yet dynamic publisher-subscriber architecture, for the published messages from a publisher to the subscribers distributed in the smart grid network. The scheme ensures that the published message is delivered from an authentic publisher to only those authorized subscribers by verifying publisher's signature and access structure of all subscribers. Operation overheads are reduced by performing only one encryption and decryption or hashing per subscriber location using a proxy node as a remote terminal unit. Our analysis shows that the scheme is resistant against replay, man-in-the-middle, and impersonation attacks. Performance evaluation shows that the scheme can support 600 subscribers given the communication latency requirement of 3 ms. We provide the performance of the scheme under different scenarios, and observe that the efficiency of our scheme increases as the ratio of the geographical locations within a substation to the number of subscribers increases.**

***Index Terms*—Smart grid security, publisher-subscriber model, communication latency, measurement data.**

I. Introduction

The Smart Grid (SG), a next-generation power system, has received massive attention by the industry, government, and academic research organizations. The smart grid provides a two-way communication network to support advanced applications that ensure supplying electricity to the consumers in the efficient, reliable, and cost effective manner. In recent years, the number of cyber-attack attempts on the power industry has increased significantly. Cyber-attacks, such as the power grid attack in Ukraine [1], can cause blackouts, damage power equipment and cost significant financial loses. The smart grid communications are governed by unicast, multicast, and broadcast messages for different purposes. For example, in the case of substation automation applications, multicast communications within a power substation are governed by a publisher-subscriber communication service that supports asynchronous many-to-many communications among different control components, such as Intelligent Electronic Devices (IED). The publisher in the smart grid system is a Phasor Gateway (PG), which is connect to the network through routers, whereas the subscribers are destinations asking for data, which are generally the power usage vendors. Multicast communications in the smart grid allow the publisher to transmit a single copy of the message directed by a series of routers over the network. The message is then replicated and forwarded by the intermediate router to all subscribers that have previously subscribed the service.

A publisher-subscriber system model in the smart grid is very different from the synchronous request-response model, client-server model, and master-slave model. In order to provide insight into the need for a new publisher-subscriber model, let us consider the scenario shown in Figure 1. A synchronous request-response model is based on timely synchronization of the entities and is initiated by the requester followed by the responder. The client-server model involves the communications between one or many clients and a server exhibiting real-time messages in which any further communication by the client is blocked once it makes the request and until the server replies to it. Also, there are client-server models that support process multi-threading in order to overcome such blocking issues of the client. However, the client-server model is not useful for multicasting data within smart grid applications as it does not support many-to-many communications simultaneously. The requirements for the multicast communication in the smart grid are asynchronous, loosely coupled, and one/many-to-many information flow in nature. Therefore, we argue that a publisher-subscriber model has advantages in the smart grid over the master-slave



IED: Intelligent Electronic Device, **HMI:** Human Machine Interface

Fig. 1: A scenario of publishing IED's data to the subscribers.

model due to the following features: decoupled synchronous communication, distributed, peer-to-peer, and high scalability. It relies on the interest expressed by the subscribers to receive messages from the publisher. However, in order to implement a publisher-subscriber model in the smart grid, specific issues of platform compatibility and delay between Local Area Network (LAN) and Wide Area Network (WAN) must be taken into consideration [2].

The secure delivery of these multicast messages is critical for the smart grid substations. In many cases, messages from IEDs to IEDs and/or IEDs to the Human Machine Interface (HMI) application are delivered with inaccurate measurement data due to malicious activities carried out by adversary over the weakly secure or open network. If an adversary gets access to the network, he or she could compromise these messages or can target the publisher in order to affect its operations and services. Because different multicast messages at the substation have different real-time transmission requirements, as shown in Table I, communication latency requirements for specific types of messages that do not support traditional encryption become even more critical. For example, according to IEC 61850-5, type 1A, 4, and 7B messages have a stringent latency requirement of delivery within 3 milliseconds (ms) [3], [4]. The security solution must be flexible enough to accommodate such critical requirements. Therefore, a secure and efficient scheme that could support publisher-subscriber model in the smart grid system is needed.

A. *Our Contribution*

In this paper, we propose a novel secure and efficient scheme for restricted publisher-subscriber architecture in the smart grid based on access structure of the subscribers for the secure delivery of published messages. The idea is to perform minimal decryption operations at each geographic destination by allowing a primary power component, such as Remote Terminal Unit (RTU), to generate an aggregated key from all subscribers for decrypting the received messages. Our contributions are as follow:

1) The proposed architecture not only allows the secure and efficient multicasting of messages, but also performs specific delivery operations at reduced overhead using Virtual-LAN (VLAN).
2) The proposed scheme is lightweight and provides a secure delivery of data from the publisher to all its subscribers using only one encryption and one decryption per location, thus significantly reducing the overhead incurred by each subscriber. The scheme can support up to 600 subscribers to receive published messages from a publisher when the communication latency requirement is 3 ms. Except for the time synchronization, our scheme is flexible enough to support a large number of subscribers for other delay requirements.
3) The scheme is also applicable and efficient in the scenario where publisher and subscribers use different data formats, and there is a converter just before each publisher to convert data format. In such case, a converter can be attached to the proxy node that will later forward the message to all its authorized subscribers. This will reduce the usage and overhead involved in conversion.

The rest of the paper is organized as follows. Section 2 presents related work in the context of multicast messages delivering in publisher-subscriber smart grid system. Section 3 presents the communication system and an adversary models. Section 4 proposes and presents a new restricted publisher-subscriber architecture and a scheme that ensure the secure and efficient delivery of published messages. Section 5 and 6 present security analysis and performance evaluation of the scheme, respectively. Finally, Section 7 concludes the work.

II. Related Work

In this section, we present and discuss the related work on the multicasting messages in the publisher-subscriber communication system. Security in the smart grid system is different from other traditional communication systems due to the fact that the performance and reliability requirements of the power system operations are much more critical.

In this direction, [3] and [4] discussed and performed conformance test on IEC 61850 standard. However, they do not discuss any security solution for the multicast messages to be delivered over the insecure network. Ozansoy *et al.* in [7] proposed a publisher-subscriber communication model for satisfying the communications need of the IEC 61850 protocol. Falk [8] proposed securing the multicast authentication mechanisms of Generic Object Oriented Substation Events (GOOSE) and Sampled Measured Values (SMV) messages, but without any implementation or practical details. Hong *et al.* [9] performed an analysis of AES encryption, and SEED, MD5, and HMAC integrity. However, since every connected device has to perform encryption and message integrity, their experiment generates a large overhead and delay. Fateri *et al.* [10] presented a publisher-subscriber model with simulation-based traffic analysis. However, the critical latency and security concerns in the smart grid network are not discussed. Kumar *et al.* [11] analyzed suitable network architectures that can meet all the requirements of North American Syncro-Phasor Initiative Network (NASPInet). The schemes [10] and [11] did not clearly describe the functions used by these schemes. Recently, Heimgaertner *et al.* [12] proposed a cyber-secure publish/subscribe middle-ware for control plane and data plane communications in the smart grid. However, they do not consider critical latency requirements of the smart grid communication messages and do not fit well in real scenarios.

III. Communication System and Adversary Models

In this section, we discuss the publisher-subscriber communication model and the adversary model.

A. *Communication Model*

IEC 61850 standards are defined with Abstract Communication Service Interface (ACSI) to provide flexibility to bound them with any middleware technology. A general publisher-subscriber model is shown in Figure 2 where different multicast messages, such as GOOSE and SMV are used to transmit

TABLE I: Transmission Time Requirements for Different IEC 61850 Messages

Message Type	Application	Time (ms)	
		P1 Class	P2/P3 Class
1A	Fast message (off command)	10	3
1B	Fast message (other)	100	20
2	Medium speed message	100	100
3	Low speed message	500	500
4	Raw data message	10	3
5	File transfer message	1000	1000
6A	Time synchronize (control & retection)	1	0.1
6A	Time synchronize (measurement)	0.025	0.004
7A	Command message (access control)	500	500
7B	Command message (access control special task)	10	3

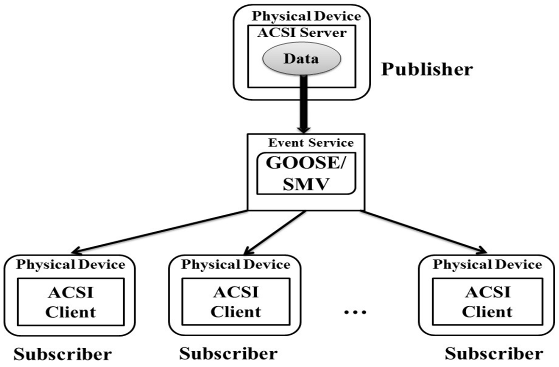


Fig. 2: Publisher-subscriber communication model.

periodic measurement data over the network. A communication scenario among different devices and applications are presented in Figure 3. It presents three types of communication in different directions using Manufacturing Message Specification (MMS), GOOSE, and SMV.

The MMS is an ISO 9506 standard that is used to transfer real-time process data and control information between the network devices, such as IED, and the HMI application running on a PC. The MMS is governed by IEC 61850-8-1, and follows a client-server model for the (vertical) communication between the SCADA and the IEDs. On the other hand, GOOSE, an event driven message, follows publisher-subscriber model for the asynchronous multicast communication (horizontal) between the Control & Protection (C&P)-IED to C&P-IED, and is also governed by IEC 61850-8-1. The SMV also follows a publisher-subscriber model, and is governed by IEC 61850-9-2. It is used for the asynchronous multicast communication with voltage and current values between the Merging Unit (MU)-IED to C&P-IED. These multicast messages use MAC addresses for the communication via bridge routing (brouting), and do not use IP-based routing. GOOSE/SMV messages supports the availability of information and high reliability by repeating transmissions for a number of times with sequence number increment until its Time-to-Live (TTL) expires and does not require to be acknowledged.

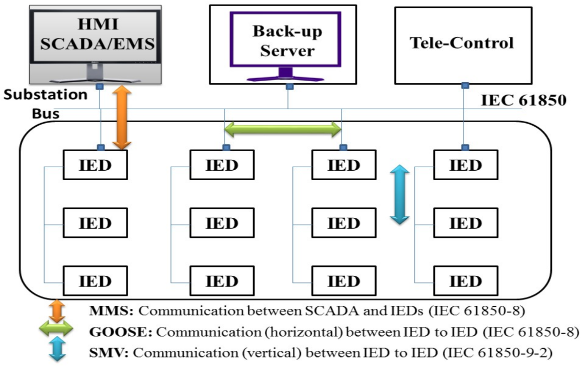

HMI SCADA/EMS
Back-up Server
Tele-Control
Substation Bus
IEC 61850
IED
IED
IED
IED
IED
IED
IED
IED
IED
IED
IED
IED
MMS: Communication between SCADA and IEDs (IEC 61850-8)
GOOSE: Communication (horizontal) between IED to IED (IEC 61850-8)
SMV: Communication (vertical) between IED to IED (IEC 61850-9-2)

Fig. 3: Types of communication in the IEC 61850.

The SMV (type 4) and GOOSE messages (type 1A) are time critical messages. The medium speed message (type 2), low speed message (type 3), file transfer functions (type 5), and command message with access control (type 7) are mapped to MMS protocol suits that follows TCP/IP stack. The time synchronization messages based on Simple Network Time Protocol (SNTP) (type 6) are broadcasted to all IEDs within a substation using UDP [4]. Even though, GOOSE messages are currently transmitted over LANs owned by the utilities, and there are some initiatives to use GOOSE messages over the WAN [5]. Since GOOSE messages are not protected, an adversary can generate forged GOOSE messages and transmit false status messages to listening devices. If the GOOSE messages are not authenticated, adversary can send a bogus message to open a breaker and disrupt the power supply [6].

B. *Adversary Model*

The publisher-subscriber architecture of the Substation Automation Systems (SAS) is vulnerable against Man-in-the-Middle (MITM), replay, and impersonation attacks. The system also suffers from the issues of the publisher (sender) authentication, subscribers (receivers) authorization, and data integrity. Security attacks on critical infrastructure, such as smart grid, are critical and the system has high importance to defeat them. We assume that an adversary is capable of performing security attacks over an insecure network. An adversary can make a malicious connection between the publisher and the subscriber to perform MITM attack. An adversary can try to impersonate the publisher and/or the subscriber or resend previously sent data over the network. The adversary can also make an attempt to maliciously authenticate itself and try to access unauthorized measurement data. In the worst case, the adversary can try to modify the measurement data sent over the network. The model assumes that middleware implementing the publiser-subscriber system is honest but curious.

IV. Proposed Restricted Publisher-Subscriber Architecture and Scheme

In this section, we propose and describe a new restricted publisher-subscriber architecture along with our scheme.

A. *Proposed Architecture Design*

In a generic publisher-subscriber model, the publishers need not to know the recipients of their data and subscribers do not know the publishers and their locations. However, in the smart grid scenario, it is important to authenticate each publisher and the subscriber, and allow only the legitimate subscribers with appropriate rights to access the data published by the publishers, especially during critical power operations. In order to address this challenge in the smart grid, we propose and present a new restricted yet dynamic publisher-subscriber architecture for the smart grid substation, as shown in Figure 4, where each publisher has an access of the list of its subscribers. This list could be anonymous, but the publisher should be able to access the public keys of all the subscribers. Figure 4(a) presents the basic architecture of the publisher-subscriber model. A dynamic list of publishers and subscribers is maintained, so they can change at any time, and join and leave the network as per their requirements. There are brokers (the ones dispatch messages) and filters to which subscribers registers in our system architecture. For simplicity, we did not mention them in Figure 4. There are generic filters that publisher can publish and to which subscribers can subscribe. Here, addresses and locations can be implemented as filters for the publishers and subcribers depending upon the topics or the contents. Generally, subscribers at the substation are connected through a LAN and we can provide enough security to this communication network, if it is not already there. This assumption is realistic and currently implemented at many power substations as per our understanding.

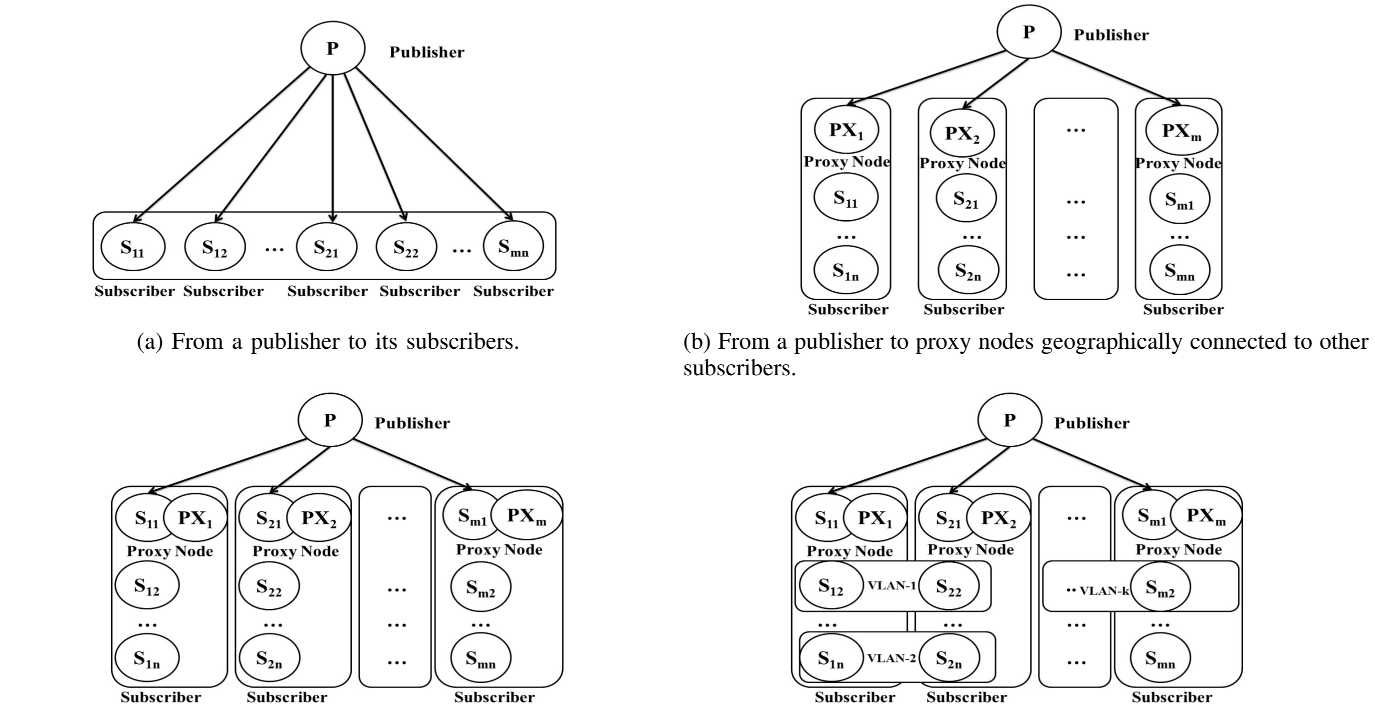
Here, publisher P publishes the messages whenever it has data to be sent to all its n-subscribers $S_i = \{S_1, S_2, ..., S_k, S_{k+1}, ..., S_n\}$ who had previously subscribed to the service. As shown in Figure 4(b), we modify the traditional architecture to propose a new architecture where a publisher P publishes the messages to its subscribers $S_{ij} = \{S_{11}, S_{12}, ..., S_{21}, S_{22}, ..., S_{mn}\}$ mentioned in the list for a specific topic or content depending upon the filter settings. Here, m are different geographical locations within the substation. Also, each geographic location has a special node, called proxy node *PX*, such as a Remote Terminal Unit (RTU), which receives the messages from the publisher on behalf of all the connected subscribers j ($j = 1, 2, ..., n$) in that particular location i ($i = 1, 2, ..., m$). In order to reduce the overhead of having an additional node at each location, one of the subscribers at each location can act as a proxy node, as shown in Figure 4(c). The selection of a proxy node can be done randomly or based on specific features, such as nearest location to the publisher, having more computing power, implementation specific, number of neighbor nodes and so on. The proxy node can also be changed depending upon the filters and brokers applied by the publisher. For special purposes, the subscribers at the substation can be configured with VLANs as shown in Figure 4(d). This will reduce the delay and improve the performance when there are only few subscribers at nearby locations. The assumption of proxy node *PX* is trusted for connected subscribers is fair, but the publisher does not trust proxy node or any other subscriber.

B. *Proposed Scheme*

We propose a new secure and efficient published message delivery scheme in a substation, based on the proposed architecture discussed in the previous section (Figure 4(c)) that also maintains confidentiality and/or message integrity. In the publisher-subscriber model of the smart grid, when the publisher multicasts a message to its subscribers, the subscribers need to verify whether the messages are correctly and accurately received from the publisher. In order to do so, the publisher in our scheme first retrieves an access structure of the targeted subscribers. Note that one location may have more than one access structure for the subscribers based on different services subscribed. The scheme considers that each publisher will multicast a message to each proxy node at different locations, and each proxy node will take appropriate actions, such as decryption or check message integrity, based on the type of message received. The proxy node will forward the original message to its authorized and connected subscribers. The publisher encrypts or hashes the transmitted message, and multicasts it to all the respective proxy nodes located at different locations so that only the authorized subsets of the proxy nodes at the substation can recover the message. Each proxy node first decrypts the message or verifies hash of the message. The proxy node forwards the message to its connected and authorized subscribers only if the message received from the publisher is valid. We present our scheme in two phases: scheme initialization and scheme execution.

(1) Scheme Initialization: We denote an authorized subset of the receiving subscribers at a proxy node as F_i, where $i = 1, 2, ..., m$ are the proxy nodes, each at different location. The overall access structure is defined as $F = \{F_1 + F_2 + ... + F_m\}$ where $F_1 = \{S_{11}, S_{12}, ..., S_{1n}\}$, $F_2 = \{S_{21}, S_{22}, ..., S_{2n}\}$, ..., $F_m = \{S_{m1}, S_{m2}, ..., S_{mn}\}$. Let E be the elliptic curve defined over a finite field F_p, and let G be a publicly known base point and generator with order p on E. We assume that each subscriber S_{ij} at the substation has a private key $x_{ij} \in [1, p-1]$ and a corresponding public key $y_{ij} = x_{ij}.G$, where $i = 1, 2, 3, ..., m$ and $j = 1, 2, 3, ..., n$. Only the authorized subscribers will receive the message by the *PX*.

First, the proxy node generates partial private and public keys for the subscriber nodes connected to it and sends these keys to the respective subscribers in a secure manner. Upon receiving the keys, each subscriber node generates its actual private and public keys, and sends its public key to the proxy



(a) From a publisher to its subscribers.

(b) From a publisher to proxy nodes geographically connected to other subscribers.

(c) From a publisher to subscribers acting as proxy nodes that are connected to other subscribers.

(d) From a publisher to subscribers acting as proxy nodes that are connected to other subscribers via VLANs.

Fig. 4: Different multicasting scenarios of publisher-subscriber communication architecture.

node. Consequently, all public keys are stored in an off-line repository accessible to the publisher. In particular, we consider a publisher P and a specific proxy *PX* at a location with subscribers S_j. The *PX* generates a random value r_{PX}, and passes a sum of r_{PX} and its private key as a token to the first subscriber connected to it. Each subscriber also generates a random value r_{S_j}. Then, when the first subscriber receives a token, it adds its private key x_{S_j} along with the generated random r_{S_j}, and then forwards it to the next subscriber. This process continues until the last subscriber repeats the same and passes the token computed as $Tok = r_{PX} + \sum_{j=1}^{n} x_{S_j} + \sum_{j=1}^{n} r_{S_j}$, back to the proxy node. Upon receiving the token from the last node, the proxy node computes Actual Token Value (*ATV*) by computing $ATV = Tok - r_{PX}$, and stores it in its memory. Also, all the involved nodes send their random values r_{S_j} directly to the proxy node, which keeps their sum as $r_{sum} = \sum_{j=1}^{n} r_{S_j}$. Whenever the proxy node receives data from the publisher, it first computes True Token Value (*TTV*) as $TTV = ATV - r_{sum} = \sum_{j=1}^{n} x_{S_j}$, and then uses this value to extract the actual data from messages on behalf of all its connected subscribers. The random numbers keep the actual secret keys protected from other subscribers, proxy node, or adversary. This process of generating keys and random numbers is repeated every few minutes to compute new values of *Tok* and *TTV*. This helps the system to quickly recover, even if the current state of the system is compromised. Finally, the proxy node transmits the message (in plaintext) to each of the authorized and connected subscribers over a secure network. The data can be encrypted by the publisher using the public key of each proxy node and decrypted by the proxy node using its private key. It ensures the sender authenticity. Asymmetric encryption is much slower than symmetric encryption. Therefore, we consider a symmetric algorithm for the encryption with a secret key shared between the publisher and each *PX*.

We also design a lightweight and secure cipher function for data encryption and decryption, which converts each plaintext *PT* into ciphertext *CT* and vice versa, as follows:
At the publisher: $CT = f_{enc}(PT, counter, sk)$;
At the proxy node: $PT = f_{dec}(CT, counter, sk)$;
where sk is a symmetric secret key generated by the publisher, each time it publishes data using a pseudo-random number generator. The *counter* is a number generated by a pre-shared function between the publisher and the proxy node. The *counter* changes every few minutes at both ends. The definitions of f_{enc} and f_{dec} for encrypting and decrypting each plaintext *PT* and ciphertext *CT* messages are as follows:

Encryption	**Decryption**
$f_{enc}(PT,\ counter,\ sk)$ {	$f_{dec}(CT,\ counter,\ sk)$ {
$CT = (PT + counter) \oplus sk$;	$PT = (CT \oplus sk) - counter$;
$counter++$;	$counter++$;
$return\ CT$; }	$return\ PT$; }

A shared key k is generated using Elliptic Curve Diffie Hellman (ECDH) algorithm and is shared between the publisher

($k = x_p.y_{S_1}$) and the current proxy node ($k = x_{S_1}.y_p$) among subscribers at each location, where (x_p, y_p) is the private-public key pair of the publisher. The purpose of generating the k key is to support symmetric encryption between the publisher and each proxy node for transmitting the sk key, because asymmetric encryption is almost 1000 times slower than symmetric encryption [13]. This k key is different for each publisher-proxy node pair in the substation. We believe that there are least chances of "ciphertext malleable" due to performing XOR and addition operations, as the sk key is generated for each new encryption and *counter* changes each time it is used. In the worset case if a ciphertext is compromed, there are no chances that the adversary could extract the original messages from the other ciphertexts as well.

(2) Scheme Execution: This phase involves the computations performed by the publisher and each proxy node on behalf of its authorized subscribers.

At Publisher: Each publisher does the following:

- **Step 1.** Chooses a random number $r_P \in [1, p-1]$ and computes a point $B = r_P.G$.
- **Step 2.** Extracts the public keys y_{ij} of the subscribers authorized for receiving the message and computes T_i for each different geographical location grouped by its subscribers, where $T_i = \{T_1, T_2, ..., T_m\}$, and $T_1 = \{y_{11}.r_P{+}...{+}y_{1n}.r_P\}$, ..., $T_m = \{y_{m1}.r_P{+}...{+}y_{mn}.r_P\}$.
- **Step 3.** Sends encrypted message M as $C_1 = E_k(M)$ for each location or hash of the message $H(M)$ and appends it as $C_1 = M||H(M)$ for all locations, depending on the communication latency requirement with each type of the message. Then computes $C_2 = (x_P.counter.(TS_i||F_i)+ ID_P).G$, where x_P, k, and ID_P are the private key, shared key, and identity of the publisher.
- **Step 4.** Converts T_i into a string [14] and computes $Z_{ij} = C_1 \oplus T_i$, where Z_{ij} = $\{Z_{11}, Z_{12}, ..., Z_{mn}\}$.
- **Step 5.** Finally, sends $(F_i, B, Z_{ij}, C_2, TS_i, ID_P)$ to the respective proxy nodes over the insecure network. Here, F_i is an access structure $\{F_1, F_2, ..., F_m\}$ for the respective proxy nodes and TS_i is the current timestamp value.

At Proxy Node (Substation's RTU): Each proxy node PX_i that receives the message performs the following steps:

- **Step 1.** Verifies the sender's signature by computing $R = ID_P.G$ and verifying $C_2 \overset{?}{=} R + y_P.\ counter.(TS_i||F_i)$, where $y_P = x_P.G$ is the public key of the sender.
- **Step 2.** Retrieves the received access structure F_i and previously computed TTV, computes $T_i^{'} = B.TTV$, converts $T_i^{'}$ into a string, and retrieves $C_1^{'} = Z_{ij} \oplus T_i^{'}$.
- **Step 3.** Retrieves $M^{'} = D_k(C_1^{'})$ or verifies message integrity as $H(M) \overset{?}{=} H(M^{'})$ depending on the communication latency requirement with each type of message.
- **Step 4.** Forwards M to all its authorized subscribers over a secure network.

AES-GCM (Galois/Counter mode) or the proposed cipher function can be used for encrypting and decrypting the message, whereas SHA256 is suitable for maintaining message integrity, depending upon the types of messages. The GCM mode provides both confidentiality and integrity, and the plaintext is XORed with output from the block cipher. The adversary cannot guess the output unless it already knows both, the plaintext and the ciphertext. Message integrity is provided for the messages that do not require confidentiality, such as alert messages. We assume here that time synchronization between all network nodes is maintained, if not, we can use nonce or random numbers in place of timestamp values.

V. Security Analysis

This section presents the security analysis of the proposed scheme.

A. Correctness Proof

Ww consider three subscribers as S_1, S_2, and S_3 connected with the *PX*. Subscribers S_1, S_2, and S_3 have (x_1, y_1), (x_2, y_2), and (x_3, y_3) private-public key pairs, respectively. The proxy node has a pre-computed *TTV* $= x_1 + x_2 + x_3$. First, the publisher chooses a random $r \in [1, p-1]$ and computes $B = r.G$. Thereafter, it computes $T = y_1.r{+}y_2.r{+}y_3.r$, $C_1 = E_k(M)$ or $C_1 = M||H(M)$, and $Z = C_1 \oplus T$. The publisher sends message $(F, B, Z, C_2, TS_i, ID_P)$ to the proxy node of the subscribers of a specific location. Here $F = \{S_1, S_2, S_3\}$. On receiving the message, the *PX* first verifies identity of the sender. If it matches, the *PX* computes $T^{'} = B.TTV = B(x_1 + x_2 + x_3)$. Thereafter, the *PX* extracts

$$\begin{aligned} C_1 &= Z \oplus T^{'} = (C \oplus T) \oplus T^{'} \\ &= C_1 \oplus r.(y_1 + y_2 + y_3) \oplus r.(y_1 + y_2 + y_3) = C_1. \end{aligned}$$

Finally, the *PX* retrieves $M^{'} = D_k(C_1)$ or verifies message integrity by comparing $H(M) \overset{?}{=} H(M^{'})$.

B. Resistance Against Security Attacks

This section discusses the security of the proposed scheme, *i.e.*, how our scheme is able to ensure secure and authorized data delivery and defeats attacks listed in the adversary model.

1) Secure and Authorized Data Delivery: The proposed scheme provides a *secure and authorized delivery* of published data from the authenticated publisher to all of its subscribers. Any unauthorized or malicious user at any location of the substation cannot (i) send (as a publisher) a malicious message (as sender authenticity is verified) or (ii) retrieve a message content as a subscriber (as access structure of each subscriber is verified). On receiving the messages, the proxy node (substation's RTU) verifies the signature of the publisher by computing $R = ID_P.G$ and comparing $C_2 \overset{?}{=} R + y_P.counter.(TS_i||F_i)$, where $y_P = x_P.G$. If the verification in successful, then the identity of the publisher and the authenticity of the access structure is proved.

2) Prevention Against Replay Attack: The proposed scheme is secure against *replay attack* as a timestamp value *TS*$_i$ is used with each transmitted message. The proxy node ensures that the timestamp is unique and correct. The proxy node can easily verify TS_i in each $C_2 \overset{?}{=} R + y_P.counter.(TS_i||F_i)$. Also, the adversary do not have the knowledge of *counter* value. If the system is not synchronized, we can use nonces instead of timestamp values.

3) Prevention Against Man-in-the-Middle Attack: We consider that there is no direct communication between the publisher and the proxy node as subscriber. Instead, all communication is routed through an adversary. The proposed scheme is indistinguishable against Chosen Plaintext Attack (CPA) as it can be proved by the following property:

Definition 1. Let $SE = (\mathcal{K}, \mathcal{E}, \mathcal{D})$ be a symmetric encryption scheme, and let $\mathcal{A}$ be an adversary that has access to an oracle that takes input a pair of messages and returns a message. Then $Adv_{SE}^{ind-cpa}(\mathcal{A}) = 2.Pr[Guess_{SE}^{\mathcal{A}} \Rightarrow true] - 1$.

In this game, using the oracle provided to an adversary $\mathcal{A}$, $\mathcal{A}$ can guess if a specific bit b (say last) is 0 or 1, where b is chosen at random $\in \mathbb{Z}_n$. Thus, $Pr[Guess_{SE}^{\mathcal{A}} \Rightarrow true]$ is the probability that $\mathcal{A}$ correctly guesses the message bit and this value is 1/2 when $\mathcal{A}$ gains no advantage. The random oracle can compute $cipher_1 = \mathcal{E}_K(msg_1)$ and $cipher_2 = \mathcal{E}_K(msg_2)$ for the given input messages msg_1 and msg_2, respectively. The scheme is secure if $\mathcal{A}$ is not able to correctly guess the message, which was encrypted by the oracle out of these two. Our scheme uses a secret key for each session to encrypt and decrypt the transmitted message. The scheme also uses a unique *counter* for each operation and $\mathcal{A}$ has no idea about its starting value. This all make the scheme to be indistinguishable for any chosen message, hence, secure against CPA.

The adversary cannot perform MITM attack, as the data values are encrypted using a cipher algorithm and can only be decrypted by the respective *PX*. No secret or public information is sent from the subscribers to the publisher using which the adversary could establish an active but malicious connection between the publisher and the subscribers. An adversary also cannot retrieve the actual content of the message.

4) Prevention Against Impersonation Attack: In this section, we discuss how the proposed scheme prevents impersonation attack when an adversary tries to impersonate the publisher by sending a forged C_2 to the proxy.

Definition 2. Let $DS = (\mathit{KeyGen}, \mathit{Sign}, \mathit{Ver})$ be a signature scheme with a message space $\mathcal{M}$.
KeyGen() $\rightarrow$ (x_P, y_P): A probabilistic key generation algorithm that takes no input, and generates a set of signing (private) and public keys of the publisher.
$Sign(x_P, msg) \rightarrow C_2$: A probabilistic signing algorithm that takes as input a signing key x_P and a message $msg \in \mathcal{M}$, and outputs $C_2 = (x_P.counter.(TS_i||F_i) + ID_P).G$.
Ver$(y_P, msg, C_2) \rightarrow \{0,1\}$: A deterministic verification algorithm that takes as input a public key y_P, a message msg, and a signature C_2, verifies $C_2 \overset{?}{=} R + y_P.counter.(TS_i||F_i)$, and outputs either 0 (invalid) or 1 (valid).

The adversary also cannot perform *impersonation* attack. An attempt of impersonating the publisher will fail as the adversary does not have the private key x_p of the publisher. Hence, the signature of the adversary will not be successfully verified at the proxy node (RTU). If the adversary sends a forged message, it will not be correctly decrypted by the *PX*, and as a result, the *PX* will discard the message.

VI. Performance Evaluation

In this section, we discuss different possible communication scenarios based on the number of control devices at different geographical locations at the substation. In our scheme, the processes of generating keys and random numbers are fast enough to enable quick system recovery even under attack. We also compare the total computations and the execution time of the proposed scheme with the existing schemes.

A. Communication Scenarios

Each proxy node has an access instance from a set $\{F_1, F_2, ..., F_m\}$, each of which is having different subscribers as $\{S_1, S_2, ..., S_n\}$. We discuss three different scenarios ($n > m$, $n = m$ and $n < m$) with n control devices at m different geographical locations at the substation: (i) $n < m$: more encryption/hash operations are executed by the publisher than decryption operations by the proxy nodes, and this case can be used for specific services using proposed architecture with VLAN shown in Figure 4(d), (ii) $n = m$: lower overhead than (i) as a smaller number of decryption operations are needed by each proxy node, and (iii) $n > m$: less decryption operations performed by the proxy nodes, hence the overall overhead is reduced by using the proposed architecture in Figure 4(c).

B. Computation Overhead and Execution Time

The proposed scheme performs $(n + 1)$-addition, 2-XOR, $(n + 4)$-multiplication, and 1 encryption and 1 decryption or 2 hash operations depending upon the operation performed on a specific type of the message. Based on the execution times of different operations, the total computation time of our scheme is 0.83477+0.00285×n +0.000933×$(n-1)$ using the proposed cipher function at each location. Table II summarizes the comparison of computation overhead of our scheme with the scheme [12]. The schemes [10] and [11] did not clearly describe the functions used by these schemes. Therefore, we cannot evaluate their computation overhead. By observing the table, it can be concluded that our scheme is lightweight and efficient as compared to the scheme in [12].

We develop an experimental setup with a co-simulator [15]. The co-simulator uses JDK1.7 with Gridsim, PowerWorld, MATLAB, and Java Agent Development Framework (JADE) on Intel Core i3-4005U CPU 1.7GHz with Win7 and 4GB RAM to implement communication and power system scenarios between the publisher and the proxy (subscriber) node. We consider a 24 substations system case with 42 buses, 62 lines, 7 generators, 27 loads, 6 transformers, and 9 shunts. Table III describes the selected ranges of the communication parameters for our simulation, which supports data packets with message passing using C37.118 protocol. We simulate the system by varying the baud rate from 20-24 mbits/sec, Maximum Transmission Unit (MTU) from 32-1024 bytes (transport segment 1-249), and packet size 50-1500 bytes.

Generation of a random number, addition, and XOR take 0.69 ms, 0.000933 ms, and 0.029132 ms, respectively, while the hardware implementation of elliptic curve multiplication operation takes 0.00258 ms [16]. AES-GCM, proposed cipher

TABLE II: Computational Analysis

Parameters	Our Scheme	Heimgaertner et al. [12]
Key generation operations	(n+1)RAND	nKDF
Key generation time with one subscriber (ms)	1.38	4.1
Scheme Operations	(n+1)ADD, 2XOR, (n+4)MUL, 1E, 1D	4nMAC, 2nRSA, 2AES
Scheme execution time with one subscriber (ms)	0.84	1032

TABLE III: Parameters for Simulation Setup

Parameters	Range Value	Unit
Baud rate	20 - 24	megabits/sec
MTU	32-1024	bytes
Packet size	50-1500	bytes

function, and RSA take (13 ms, 4 ms), (0.032 ms, 0.032 ms), and (16 ms, 15 ms) for encryption and decryption operations, respectively. Hash function SHA256, MAC function HMACSHA256, and hash (SHA256)-based Key Derivation Function (KDF) take 4, 246, and 4.1 ms, respectively. We run a simulation with 32 bytes MTU, 50 bytes packet size, and 24 megabits/sec as data transmission rate between the publisher and the *PX*, which took 0.0166 ms time. We can consider this communication time negligible (close to zero ms) for our analysis. The total computation times between a publisher and a proxy node (RTU) are 0.84, 0.87, 1.19, and 2.94 ms, respectively, for 1, 10, 100, and 600 subscribers. Based on these parameter setting, our scheme can support up to 600 subscribers at one location, when the latency requirement is 3 ms for message types 1A (P2/P3) and 4 (P2/P3). Note that the total time needs to include data transmission between publisher-*PX* and *PX*-subscriber, and computation time at publisher and *PX*. For other message types, the scheme can support delivery of data with the proposed cipher and hash functions. A comparison of execution times with the schemes in [10] and [11] is shown in Table IV. Our scheme is much efficient than these both schemes.

VII. Conclusion and Future Directions

We proposed a novel lightweight, secure, and efficient data delivery scheme for the publisher-subscriber architecture in smart grid. The scheme provides accurate and correct delivery of published messages with data from the publisher to all its authorized subscribers over an insecure network. The scheme is scalable up to 600 subscribers when the latency requirement is 3 ms, and can support a large number of subscribers to receive measurement data simultaneously under other less stringent latency requirements. The scheme reduces the overall overhead compared to the traditional publisher-subscriber model and is feasible to implement for real-time applications at substations. The presented system model considers that all the subscribers at a location trust each other (including proxy node) and are connected with a secure LAN. Hence, as a future work, we will extend this work by exploring the possibilities and their impact when the proxy node, brokers, filters and subscribers cannot be trusted. This scheme is not suitable when a subscriber does not provide it's correct private key.

TABLE IV: Comparison of Execution Times (ms)

# of Subscribers	Fateri et al. [10]	Kumar et al. [11]	Our Scheme
1	2.17	6.3	0.84
10	2.18	6.5	0.87
100	2.22	7.3	1.19
600	4.01	8.1	2.94

VIII. Acknowledgments

This research was funded by the MSIP, Korea, under the ”ICT Consilience Creative Program” (IITP-2015-R0346-15-1007) supervised by the IITP, by the KEIT, Korea, under the ”Global Advanced Technology Center” (10053204), by the NRF, Korea, under the ”Basic Science Research Program” (NRF-2015R1C1A1A01053788), and by the Bournemouth University, UK (CID04).

References

[1] Pavel Polityuk, Ukraine sees Russian hand in cyber attacks on power grid, Feb 12, 2016. [Online]. http://www.reuters.com/article/us-ukraine-cybersecurity-idUSKCN0VL18E.

[2] Y.-J. Kim, M. Thottan, V. Kolesnikov, and W. Lee, “A secure decentralized data-centric information infrastructure for smart grid,” *IEEE Communication Magazine*, vol. 48, no. 11, pp. 58-65, Nov. 2010.

[3] D. Hou and D. Dolezilek, “IEC 61850 what it can and cannot offer to traditional protection schemes,” *SEL Journal of Reliable Power*, vol. 1, no. 2, pp. 1-11, Oct. 2010.

[4] T.-H. Yeh, S.-C. Hsu, C.-K. Chung, and M.-S. Lin, “Conformance test for IEDs based on IEC 61850 communication protocol,” *Journal of Power and Energy Engineering*, vol. 3, pp. 289-296, 2015.

[5] C. H. R. de Oliveira and A. P. Bowen, “IEC 61850 GOOSE message over WAN, in *Proc. WorldComp*, 2012, pp. 1-3.

[6] P. Weerathunga, “Security Aspects of Smart Grid Communication,” The School of Graduate and Postdoctoral Studies Western University London, Ontario, Canada, 2012, 98 pages.

[7] C. R. Ozansoy, A. Zayegh, and A. Kalam, “The real-time publisher/ subscriber communication model for distributed substation systems,” *IEEE Trans. on Power Delivery*, vol. 22, no. 3, 2007, pp. 1411-1423.

[8] H. Falk, “Securing IEC 61850,” in *Proc. Power and Energy Society General Meeting*, Pittsburgh, USA, Jul. 2008, pp. 1-3.

[9] S. Hong, D.-Y. Shin, and S.-J. Lee, “Experimenting security algorithms for the IEC 61850-based substation communication,” in *Proc. China-Korea Forum on Protective Relaying*, Beijing, China, Sep. 2009, pp. 1-9.

[10] S. Fateri, Q. Ni, G. A. Taylor, S. Panchadcharam, and I. Pisica, “Design and analysis of multicast-based publisher/subscriber models over wireless platforms for smart grid Communications,” in *Proc. IEEE TrustCom*, Liverpool, UK, 2012, pp. 1617-1623.

[11] K. Kumar, M. Radhakrishnan, K. M. Sivalingam, D. P. Seetharam, and M. Karthick, “Comparison of publish-subscribe network architectures for smart grid wide area monitoring,” in *Proc. IEEE SmartGridComm*, Tainan, Taiwan, 2012, pp. 611-616.

[12] F. Heimgaertner, M. Hoefling, B. Vieira, E. Poll, and M. Menth, “A security architecture for the publish/subscribe C-DAX middleware,” in *Proc. IEEE ICC Workshop*, London, UK, 2015, pp. 2616-2621.

[13] Hardjono, “Security in wireless LANS and MANS,” Artech House Publishers, 2005.

[14] D. R. L. Brown, “SEC 1: elliptic curve cryptography,” Certicom Research, May 21, 2009. [online]. http://www.secg.org/sec1-v2.pdf.

[15] N. Saxena, V. Chukwuka, L. Xiong, and S. Grijalva, “CPSA: a cyber-physical security assessment tool for situational awareness in smart grid,” in *Proc. ACM CCS Workshop - CPS-SPC*, Dallas, USA, 2017.

[16] A. Sghaier, M. Zeghid, B. Bouallegue, A. Baganne, and M. Machhout, “Area time efficient hardware implementation of elliptic curve cryptosystem,” Cryptology ePrint Archive, 2015. [Online]. https://eprint.iacr.org/2015/1218.pdf.