Article

Certificate Based Authentication Mechanism for PMU Communication Networks Based on IEC 61850-90-5

Shaik Mullapathi Farooq ¹ , S. M. Suhail Hussain ^{2,*} , Siddavaram Kiran ¹ and
Taha Selim Ustun ²

¹ Department of Computer Science and Engineering, YSR Engineering College, Yogi Vemana University, Andhra Pradesh 516360, India; smfarooq@ieee.org (S.M.F.); rkirans125@yogivemanauniversity.ac.in (S.K.)

² Fukushima Renewable Energy Institute, AIST (FREA), Koriyama 963-0215, Japan; selim.ustun@aist.go.jp

* Correspondence: suhail.hussain@aist.go.jp; Tel.: +81-804-373-5146

Received: 24 October 2018; Accepted: 23 November 2018; Published: 2 December 2018

Abstract: Smart grids are becoming increasingly popular thanks to their ability to operate with higher precision and smaller margins. Dynamic operation control in smart grids can be achieved with phasor measurement unit (PMU) based wide area monitoring and control systems. The data communication requirements for the PMU based applications are well addressed in the IEEE C37.118.2 and IEC 61850-90-5 standards. Due to the higher probability of cyberattacks and the scale of their impact, data security is a critical requirement in PMU communication networks. The IEC 61850-90-5 communication standard addresses this security concern and proposes the HMAC (hash based message authentication code) with key distribution center (KDC) scheme for achieving information authentication and integrity. However, these IEC 61850-90-5 security recommendations do not consider the mechanism for attacks such as man-in-the-middle (MITM) attacks during KDC key exchanges. MITM attacks can be easily implemented and may have a large impact on the grid operation. This paper proposed an explicit certificate-based authentication mechanism to mitigate MITM attacks in PMU communication networks. The proposed certificate-based authentication mechanisms were implemented in real-time using Python-based terminals to observe their performance with different signature algorithms.

Keywords: cybersecurity in power systems; synchrophasors; authentication; man-in-the-middle-attack; explicit certificate mechanism; X.509

1. Introduction

State estimation, monitoring, controlling, and protection of smart grids can be accomplished by phasor measurement unit (PMU) based wide area monitoring and control systems. The PMU is an intelligent electronic device (IED) in a smart grid that periodically records data pertaining to the power system dynamics. It then sends them along with the GPS synchronized time stamp data to phasor data concentrators (PDCs) with a high sampling rate (6–60 samples/s) [1]. The recorded data include voltage and current phasors (amplitude and angle), frequency and rate-of-change-of-frequency (ROCOF) as well as other parameters [2]. The time stamped measurements are called synchrophasors.

Synchrophasor measurements were first standardized with IEEE 1344 in 1995 [3]. Later in 2005, IEEE 1344 was replaced with the IEEE C37.118 standard [4]. In 2011, the IEEE C37.118 standard was split into IEEE C37.118.1 and IEEE C37.118.2 standards [5,6]. The former deals with how to measure synchrophasor values while the latter deals with the data transfer requirements of recorded synchrophasor measurements. The split enables the harmonization of IEEE Std C37.118-2005 with IEC 61850. IEC 61850-90-5 was developed in 2012 to specify data communication requirements between PMUs, PDCs, wide area monitoring, protection, and control applications, and control centers

based on IEC 61850 [7]. It adopted IEEE C37.118.1 as a whole, followed its own communication standardization, i.e., replaced C37.118.2 with its own communication procedures such as generic object-oriented substation event (GOOSE) and sample value (SV) messages.

The main role of the PMU is to monitor voltage instability in the smart grid, which may lead to a blackout [8]. According to a North American electric reliability corporation (NERC) report, the shortage of data on grid status awareness was the main reason for the 2003 blackout which led to a loss of billions of dollars [9]. Security is of paramount importance in smart grids as the cyber-threat is real [10]. The authors in [11] reported that a myriad of attacks were possible, e.g., network congestion, ping of death, open port scan, address resolution protocol (ARP) spoofing, and penetration attacks. There are different tools (e.g., the network mapper (NMAP), open vulnerability assessment system (OpenVAS) to scan the ports to identify the services offered by the devices and to remove unused ports [12,13]. With the help of such tools, it is possible to obtain unauthorized access to user profiles in PMU communication networks. The communication system may be vulnerable to denial of service (DoS) attack by taking complete control of the meters [14]. The authors in [15] reported several vulnerabilities such as unencrypted communication channels and weak password management. Such kind of vulnerabilities can be exploited for a structured query language (SQL) injection attack and cyber risks [16]. National institute of standards and technology (NIST) identifies PMU security as one of the key research and development (R&D) themes for smart grid cybersecurity [17].

IEEE C37.118 is still the most widely used protocol for the exchange of synchrophasor data between PMUs and PDCs. However, it suffers from security vulnerabilities. The reason is that IEEE C37.118 does not have any cyber-security specifications for PMU communication networks. Having replaced C37.118.2 with its own requirements, IEC 61850-90-5 addresses this security gap and specifies a hash based message authentication code (HMAC) with key distribution center (KDC) scheme to achieve information authentication and integrity of the PMU data. KDC provides symmetric keys to different nodes in the PMU communication network using the multicast mechanism. A new framework called the group domain of interpretation (GDOI) extended the IEC 61850-90-5 based KDC key exchanges [18]. GDOI eliminated most of the security vulnerabilities that are relevant to integrity and confidentiality such as those mentioned in [15,16]. That being said, GDOI does not touch upon authentication and refers to RFC 2409 for authentication mechanisms [19]. IEC 61850-90-5 recommends the use of a node authentication mechanism during KDC key exchanges, yet, it does not specify a certain mechanism. This paper addresses this gap and proposes the implementation of certificate-based authentication as a solution for node authentication during KDC key exchanges. There are several authentication methods in the literature such as key based and certificate based. The motivation behind choosing certificate-based authentication is that it is the only authentication mechanism that can prevent MITM attacks during the KDC key exchanges [20,21]. Furthermore, the computational time performances of different signature algorithms for certificate-based node authentication mechanism have been presented.

The major contribution of the paper is to propose a certificate authority (CA) based explicit certificate mechanism to perform identity management. In addition to fulfilling IEC 61850-90-5's authentication requirements, this method also prevents MITM attacks. The efficacy of the explicit certificate mechanism against an MITM attack has been demonstrated through the experimental implementation of an authentication scheme using openssl Python libraries [22]. The rest of the paper is organized as follows. Section 2 describes the IEEE C37.118 and IEC 61850-90-5 based PMU communication frameworks. Section 3 discusses the GDOI security mechanism, MITM attack, and the proposed certificate mechanism. Section 4 shows the implementation results while Section 5 draws the conclusions.

2. PMU Communication Standards

PMUs measure electrical quantities and send them to a substation PDC. The substation PDC forward the collected data to its regional PDC. Regional PDCs forward the collected data to the grid

control center for further processing by the super PDC. Figure 1 illustrates the block diagram of PMU and PDC communication.

Figure 1. Synchrophasor communication between PMUs and PDCs.

IEEE C37.118.2 and IEC 61850-90-5

Having evolved from IEEE 1344, IEEE C37.118 is considered to be the improved synchrophasor communication standard. The present standard defines the format of the messages to be exchanged, methods of evaluating synchrophasor measurements, and timing values.

The purpose of the standard is to facilitate synchronized phasor measurement data exchange between the PMUs and PDCs. The standard does not stipulate the implementation details such as mode of communication, physical medium, transport protocol, and leaves them to the user. IEEE C37.118.2 defines four types of messages such as data, configuration, header, and command. Data messages are used to send PMU's phasor and frequency measurements. Configuration messages consist of calibration factors and other information to properly decode the data messages in machine readable format. Configuration (CFG) messages come in three types: CFG1, CFG2, and CFG3. CFG1 represents the type of data and reporting capability of the PMU. CFG2 gives information about the synchrophasor values to be sent. CFG3 is similar to CFG2 and has additional information about the PMU's characteristics and measurements. Header messages contain descriptive information normally specified by the user. Command messages are sent to the data source to control its operation and transmission of data, header, and configuration messages. Example command messages are "data_on" and "data_off".

The sequence of messages exchanged between the PMU and PDC is illustrated in Figure 2. PDC, first, sends a command message to the PMU to get the configuration information. The format of the command message consists of the CMD field, which is used by the PMU to recognize the type of configuration information. After reading the command message, PMU sends its configuration message. Then, PDC sends another command message requesting the PMU sends data frames. Data frames contain synchrophasor measurements. After receiving the command message, the PMU continuously sends data frames until another command is sent by the PDC to stop transmission of the data frames.

IEEE C37.118 has many limitations. It lacks standard data names which enable auto-discovery and self-description without configuration messages. Devices may have different features, which leads to a lack of interoperability and integration support. Finally, the standard does not have any security mechanisms. The IEC 61850 gives self-description and object auto discovery capability with its structured meta-data. The main objective of IEC 61850 is to achieve interoperability among different components from different vendors [23]. The standard specifies a reliability mechanism which performs re-transmission, in the case of data loss. However, it lacks cyber-security mechanisms and is restricted to local networks. IEC 61850-90-5 is derived from IEC 61850 and deals with the transmission of

synchrophasors. As shown in Figure 3, IEC 61850-90-5 allows the transmission of time critical protocols such as SV and GOOSE over wide area networks using network and transport layer protocols. In IEC 61850-90-5, local area network protocols such as SV and GOOSE are extended to R-SV (Routable-SV) and R-GOOSE (Routable-GOOSE), which are compliant with wide area network communication.

Figure 2. Exchange of IEEE C37.118 messages between the PMU and PDC.

Figure 3. Communication exchanges based on IEC 61850-90-5.

IEEE C37.118.2 does not provide any integrity check other than cyclic redundancy check (CRC), which can be exploited easily. Considering timing performance, IEC 61850-90-5 recommends a separate mechanism: cryptographic hash functions and message authentication codes (MACs) such as hash based message authentication code (HMAC) for data authentication. Section 3 discusses the GDOI implementations and certificate-based authentication scheme to mitigate the MITM attack in IEC 61850-90-5 PMU communication networks.

3. Cyber-Security Vulnerability of PMU Networks Operating with IEC 61850-90-5 GDOI

3.1. IEC 61850-90-5 Cyber Security Considerations

IEC 61850-90-5 specifies the security model where information authentication and integrity are of paramount importance. Furthermore, the data flow from the PMU to PDC should not be interrupted due to the specified security model. Therefore, a KDC mechanism is recommended to manage the required symmetric key exchanges between the KDC and PMUs/PDCs. The KDC can be implemented

as a stand-alone function (centralized) or as an entity (distributed) in the network, that is either in the PMU or PDC. Figure 4 shows the default packet formats of IEC 61850-9-2 and IEEE C37.118.2. It also shows how KDC and KDC based on GDOI mechanisms are implemented in the session layer of the open systems interconnect (OSI) reference model for the IEC 61850-90-5 packet format. The GDOI security mechanism is a group key management protocol that supports common security and keying policy for secure group and multicast applications [24].

Figure 4. Comparison of IEEE C37.118.2 and IEC 61850-90-5 with GDOI.

In this communication model, participants are the group controller and key server (GCKS) as well as a group member (GM). GCKS can be a device that defines group policy and distributes the keys. GCKS can also be called KDC as it manages the secure communication among the group members by distributing the keys. It can be the PDC or regional PDC or any other third-party entity within the PMU communication network. The GM is an authorized member of the group which can communicate with other GMs. It can be a PMU or PDC in the communication network.

As depicted in Figure 5, the GDOI key distribution model, also referred to as the GDOI-KDC security mechanism, consists of two major phases called the GROUPPULL and GROUPPUSH phases. In the GROUPPULL phase, participating entities register with KDC with authentication. Here, mutual authentication is performed using key pairs. In the GROUPPUSH phase, KDC pushes the keying material such as key encryption key (KEK) and traffic encryption key (TEK) to GMs via an authenticated and encrypted session. The security policies and keying material used are not fixed for communication between PMU and PDC, but instead, are replaced periodically for secure transmission of data. The pairwise key also protects the GROUPKEY PULL exchange used to acquire KEK and TEK from KDC. The goal of the GROUPKEY PULL exchange is to establish and secure phase GROUPPUSH communication. For example, the KEK protects GROUPKEY PUSH exchanges whereas the TEK protects communication among GMs.

Figure 5. GDOI key distribution model.

The security mechanism has three key features: (i) authentication, (ii) freshness, and (iii) secrecy. Authentication ensures that only allowed GMs such as PMU or PDC can send and receive packets securely. Freshness ensures the periodic update of key credentials to protect from

cryptanalysis. The GDOI-KDC security mechanism ensures perfect forward secrecy to protect the previous communications, if the key is solved with a successful cryptanalysis. It also ensures backward and forward access control so that after any GM leaves the group, the required credentials are updated to other group members.

In the literature, functional specifications for GROUPPUSH are elaborated in detail [18]. GROUPPULL is a crucial phase of authentication. Security breaches of this phase may result in the full compromise of GM's communication during GROUPPUSH.

RFC 6407 [23] does not specify any authentication technique between GM and GCKS, and the choice is up to the developer. The Diffie-Hellman public key cryptography technique, shown in Figure 6, is widely implemented for encryption [18]. It assumes that neither of the devices has prior knowledge of the secret key. It can be understood from the figure that both parties, i.e., GM (such as PMU or PDC) and KDC, agree on prime p and generator g . They pick respective private keys a and b , then, using modular division, calculate public keys A and B mathematically. The public keys can take any value between 1 and $p - 1$. Devices exchange their public keys and derive a common pairwise key which will be used to generate KEK and TEK in the GDOI security mechanism. In Figure 6, generated pairwise key is k . However, this Diffie-Hellman authentication scheme can be compromised using MITM attack, as explained below.

Figure 6. Diffie-Hellman key exchange for authentication.

3.2. Man-in-the-Middle (MITM) Attack

MITM is one of the major threats that can be used to gain access to data communicated between the PMU and PDC. It has been treated as a high-risk attack in PMU communication which hampers authentication and may lead to severe damage to the critical infrastructure. Figure 7 demonstrates a possible MITM attack that can happen between the PMU and KDC. An intruder PMU can impersonate a real PMU and communicate with the KDC. In this case, it is assumed that the system uses the Diffie-Hellmann key exchange authentication. An MITM attack is launched in the authentication process as explained below.

Suppose an intruder wants to impersonate a real PMU, it will send its own computed public key A' as if it is requesting the KDC to generate a pairwise key. The KDC assumes that some device wants to become a GM and generates a public key B and sends it to the intruder. Both the KDC and intruder PMU generate a pairwise secret key k through the modular mathematical model shown in Figure 7. When a real PMU communicates with the KDC and wants to become a GM, it will compute its public key A and send it to the KDC. However, the intruder device impersonates the KDC and sends its computed public key A' to the real PMU. The real PMU generates a pairwise key k with KDC. Here, the intruder PMU has both pair-wise keys for communication between the GM and KDC. The intruder PMU communicates with the KDC as if it is an authorized GM and communicates with the real PMU as if it is the KDC. The real PMU sends its data to the intruder, thinking that the data is sent to the KDC. The KDC receives data from the intruder PMU devices as if it is received from an authorized GM. Here, the problem is that the intruder acts in the middle, causing a security breach called the

MITM attack. This stems from the fact that the KDC does not have any mechanism to find the identity of the requesting device and the same applies to the real PMU.

This paper proposed a certificate mechanism to mitigate this vulnerability and prevent MITM attacks in the PMU networks. The proposed certificate-based authentication is explained in the Section 3.3.

Figure 7. Intruder attempt of the MITM attack.

3.3. Implementing Certificate Based Authentication Mechanism with IEC 61850-90-5

The problem in the above attack is that the KDC and PMU do not have any mechanism to mutually identify each other and whether the public key belongs to these respective devices. The developed certificate mechanism solves this problem. A certificate signed by a trusted authority ensures that a device claiming its public key has its corresponding private key. If a PMU wants to register with the KDC entity; first, both devices are mutually authenticated with the certificate mechanism explained in this section. If an intruder PMU sends its own certificate to the KDC for authentication, then the KDC verifies the intruder PMU's certificate with the CA. If the certificate is not valid, then the request of the intruder PMU will be discarded.

A certificate is similar to binding a PMU's identity such as its name and serial number to its public key value. A certificate has a format defined by X.509 [25]. A certificate is issued by a trusted central authority called the CA. The format of the certificate consists of the version, serial number, subject name, certificate issuer information, validity, public key of PMU, and signature. A PMU keeps its corresponding private key inside its memory and generates a public key certificate with the CA. Figure 8 illustrates the certificate request and signing process. A PMU or PDC sends a certificate request which includes all of the necessary information according to the X.509 format. The CA generates a signature using a message digest algorithm (MDA). The outcome of the MDA is the digest MD1, which is further encrypted with the CA's private key. The encryption digest (ED) is the signature to be placed in the certificate format.

Once a PMU receives a signed certificate from the CA, the PMU sends its signed certificate to the KDC for authentication. Once the KDC receives a certificate of a PMU, it verifies the received certificate with the CA. The CA verifies the certificate of the PMU under discussion and sends a verified message as either "Accept" or "Reject" depending on the verification outcome. Figure 9 depicts the certificate verification mechanism. The CA maintains the database of issued certificates as well as a revocation

list. The certificate is said to be revoked if the validity of the certificate expires. This database is useful in verifying the legitimacy of certificates. If the certificate is valid, then the CA picks the signature ED from the certificate field and decrypts it to obtain the MD1. Furthermore, the CA generates a new signature MD2 using the MDA. If the MD1 matches the MD2, then authentication is successful, and vice-versa.

Figure 8. Signing of digital certificate by a certificate authority (CA).

Figure 9. Certificate verification by CA.

The implementation of the above mechanism requires specific algorithms to generate, sign, and verify the certificates. The authentication mechanism proposed in this paper implements the three distinct algorithms developed below.

Algorithm 1 is developed for certificate request generation (CSR) by the PMU. Initially, the PMU generates public–private key pairs. Furthermore, the PMU constructs a certificate following the X.509 format. The X.509 certificate fields such as name, issuer_name, serial number, validity period, public key and other credentials are filled. Finally, CSRX509Cert is a request message generated to be endorsed by the CA. The input of the Algorithm 1 is *X509Cert* and its output is CSR, *CSRX509Cert*.

Algorithm 1. CSRGGen (X509Cert)

```

1: (PMU_PUB_KEY, PMU_PR_KEY)  $\leftarrow$  Keygen()
2: X509Cert.name  $\leftarrow$  'PMU_NAME'
3: X509Cert.issuer_name  $\leftarrow$  'CA_NAME'
4: X509Cert.Srln  $\leftarrow$  rand ().
5: X509Cert.validity  $\leftarrow$  'VALID_DATE ()'
6: X509Cert.Pubkey  $\leftarrow$  'PMU_PUB_KEY'
7: return CSRX509Cert

```

Algorithm 2 is developed for the certificate signing process in the CA. After the certificate (X509Cert) is received at the CA, it generates a hash value ($h1$) for the part that corresponds to the PMU identity (PMUID). The PMUID includes the credentials of the PMU such as its name, issuer name, serial number, validity, etc. The hash value, $h1$, is generated by a signing function $SIG_k()$. The signing function is a message digest generation algorithm. Next, the digest is encrypted by an encryption algorithm ($ENC_k(h1)$) with the CA's private key. The encrypted message is copied to the signature field of the certificate.

Algorithm 2. Signing (X509Cert)

```

1: PMUID  $\leftarrow$  X509Cert.credentials
2:  $h1 \leftarrow SIG_k(PMUID)$ 
3:  $x \leftarrow ENC_{CAPrKey}(h1)$ 
4: X509Cert.Signature  $\leftarrow x$ 
5: return SX509Cert

```

Algorithm 3 is developed for the certificate verification mechanism in the CA. The sent, signed certificate $SX509Cert$ is verified by its signature. The CA generates the message digest $h2$ of the received certificate credentials. The signature x of the received certificate is decrypted with the CA's public key to obtain $h1$. Furthermore, the decrypted value $h1$ is compared with $h2$. If both match, then the PMU is authenticated, otherwise the certificate is rejected.

Algorithm 3. Verify(SX509Cert)

```

1: PMUID  $\leftarrow$  X509Cert.credentials
2:  $h2 \leftarrow SIG_k(PMUID)$ 
3:  $x \leftarrow SX509Cert.Signature$ 
4:  $h1 \leftarrow DEC_{CAPubKey}(x)$ 
5: if  $h1 = h2$  then
6:   return Accept
7: else
8:   return Reject
9: end if

```

4. Python-Based Implementations and Results

The proposed explicit certificate authentication mechanism was implemented using Python client and server programs. The Python-based implementation was done for the authentication phase of the GDOI security mechanism discussed in Section 3.1. In the implementation, first, a pair of public and private keys for PMU and CA are generated. Next, a certificate with the X.509 format is generated with the required credentials and is sent to the CA for signing. Using the secure hash algorithm (SHA256) with a 256-bit key, a hash value is generated. The hash value is further encrypted using Rivest–Shamir–Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) public key

cryptographic algorithms. Here, RSA algorithms with different key sizes and various elliptic curves defined by the National Institute of Standards and Technology (NIST) for ECDSA are considered. Table 1 shows the sizes of the CSR and certificates generated by the RSA algorithms with different key sizes and by ECDSA with various curves. Once the signing process is completed, the signed certificate is sent to the PMU. Upon receiving the signed certificate from the PMU, the KDC verifies it with the CA for authenticity. Table 1 gives the computational time required for verifying these certificates.

Table 1. Computational times for explicit certificate verification with different key sizes of RSA and ECDSA.

Type of Encryption		Private Key (bytes)	CSR Size (bytes)	Certificate Size (bytes)	Certificate Verification Computational Time (ms)
Algorithm	Key Size/Curve				
RSA	1024	891	745	1029	7
	2048	1769	1147	1371	7
	3072	2459	1496	1753	8
	7680	5973	3081	3321	8
	15,360	11,823	5701	5986	13
ECDSA	secp224r1	278	627	895	8
	secp521r1	436	838	1099	9
	prime192v1	270	627	912	8
	prime256v1	302	619	839	7
	brainpoolP384r1	367	725	981	10
	brainpoolP512r1	436	806	1070	12
	brainpoolP384r1	367	741	985	10
	brainpoolP512r1	436	826	1123	12

An intruder PMU tries to authenticate itself by either changing the signature in the certificate or generating a self-signed certificate. In Figure 10, an intruder PMU intercepts the signature of the certificate and replaces it with its own signature to authenticate itself. However, thanks to the certificate verification process explained in Section 3.3, the CA based certificate mechanism identifies this fake certificate since the signature is a mismatch.

Figure 10. Intruder PMU trying to authenticate itself by changing signature.

In Figure 11, an intruder PMU sends its own certificate for authentication. When the PMU sends its signed certificate to the KDC for authentication, it encrypts the entire certificate with its own private key. If an intruder PMU intercepts the encrypted certificate and replaces it with its own encrypted certificate, then the KDC tries to decrypt the received file with the public key of the original PMU. This results in a corrupted file, indicating that the certificate is invalid for the authentication. Thus, the certificate mechanism effectively eliminates MITM attack in PMU communication networks.

Figure 11. Intruder PMU sending its own certificate for authentication.

Figure 12 shows the results of the successful signature verification of a legitimate PMU by the CA. Figure 13 shows the detection of an intruder PMU certificate during verification with the message: “unable to get local issuer certificate”. This error message is displayed when the signed certificate cannot be verified. Hence, this concludes that this PMU is not legitimate and not signed by the CA.

```
Signature ok
subject=/C=IN/ST=AP/L=VIJ/O=signer /OU=development/CN=ssil/emailAddress=ssil@gmail.com
Getting CA Private Key
B54A397394DFA35D
0.00user 0.00system 0:00.00elapsed 100%CPU (0avgtext+0avgdata 4084maxresident)k
0inputs+0outputs (0major+206minor)pagefaults 0swaps
pmu-rsa.crt: OK
crypto@crypto-HP-2000-Notebook-PC:~/finalizedprograms/RSA/RSA1024/2048$
```

Figure 12. Legitimate PMU authentication when the CA signature is OK.

```
Signature ok
subject=/C=IN/ST=AP/L=Kdp/O=SSIL LIMITED/OU=RESEARCH/CN=SSIL/emailAddress=ssil@gmail.com
Getting CA Private Key
D9AA4DF0D854D549
intru-pmu-rsa.crt: C = IN, ST = AP, L = Kakinada, O = Jawaharlal Nehru University technolog
jntuk@gmail.com error 20 at 0 depth lookup:unable to get local issuer certificate
crypto@crypto-HP-2000-Notebook-PC:~/finalizedprograms/RSA/RSA1024$
```

Figure 13. Intruder PMU signature verification fail.

5. Conclusions

Real time monitoring, protection, and control in smart grids is of paramount importance. These capabilities highly depend on PMU measurements and their secure transmission over wide area PMU communication networks. With the increased connectivity and intelligence introduced to smart grids, cyber-security in PMU communication networks is a real concern. Authentication of devices in the communication network should be ensured to avoid cyber-attacks. Of the two existing PMU communication standards, IEEE C37.118 does not touch on cyber-security issues while IEC 61850-90-5 only recommends different security schemes without specific details of the implementation.

Addressing this knowledge gap, this paper developed a certificate-based node authentication method for the PMU networks. The benefit of using this particular method is that it mitigates MITM attacks during key exchanges in PMU networks. The developed solution was implemented with a Python programmed server and clients. The results showed that the proposed certificate-based

mechanism could effectively mitigate MITM attacks during key exchanges and ensure the safe operation of PMU networks based on IEC 61850-90-5.

As future work, it is possible to extend this solution by adding encryption and integrity check mechanisms such as AES-256 and HMAC, respectively. These will make use of the symmetric key exchanged during the certificate based mechanism developed in this paper. These two mechanisms will ensure the confidentiality and integrity of PMU measurements. In this fashion, other attacks such as SQL injection, Spoofing, and DoS attacks can also be mitigated.

Author Contributions: Conceptualization and methodology, S.M.S.H. and T.S.U.; software and validation, S.M.F.; writing—original draft preparation, S.M.F. and S.M.S.H.; writing—review and editing, S.M.S.H. and T.S.U.; supervision, T.S.U. and S.K.; funding acquisition, T.S.U.

Funding: This work was supported by the Research and Innovation Fund 2018 and by KEIDANREN (Japan Business Federation) Promotion of Environmental Protection Foundation's Research Grant-2018.

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

References

1. Ali, I.; Aftab, M.A.; Hussain, S.M.S. Performance comparison of IEC 61850-90-5 and IEEE C37.118.2 based wide area PMU communication networks. *J. Mod. Power Syst. Clean Energy* **2016**, *4*, 487–495. [CrossRef]
2. Phadke, A.; Throp, J. *Synchronized Phasor Measurements and Their Applications*; Springer: New York, NY, USA, 2008; ISBN 978-0-387-76537-2.
3. *IEEE Standard for Synchrophasor for Power Systems*; IEEE Std 1344-1995 (R2001); IEEE: Piscataway, NJ, USA, 1995. Available online: <https://ieeexplore.ieee.org/document/943067> (accessed on 26 November 2018).
4. *IEEE Standard for Synchrophasor for Power Systems*; IEEE Std C37.118-2005; IEEE: Piscataway, NJ, USA, 2005.
5. *IEEE Standard for Synchrophasor Data Transfer for Power Systems*; IEEE Std C37.118.2-2011; IEEE: Piscataway, NJ, USA, 2011.
6. *IEEE Standard for Synchrophasor measurements for Power Systems*; IEEE C37.118.1-2011; IEEE: Piscataway, NJ, USA, 2011.
7. Communication Networks and Systems for Power Utility Automation—Part 90–95. In *Use of IEC 61850 to Transmit Synchrophasor Information according to IEC C37.118*; IEC: Geneva, Switzerland, 2012.
8. Vournas, C.D.; Lambrou, C.; Mandoulidis, P. Voltage Stability Monitoring from a Transmission Bus PMU. *IEEE Trans. Power Syst.* **2017**, *32*, 3266–3274. [CrossRef]
9. DOE. *Final Report on the August 14, 2003 Blackout in the United States and Canada*. U.S.—Canada Power System Outage Task Force. 2004. Available online: http://www.nerc.com/docs/docs/blackout/NERC_Final_Blackout_Report_07_13_04.pdf (accessed on 26 November 2018).
10. Musleh, A.S.; Khalid, H.M.; Muyeen, S.M.; Al-Durra, A. A Prediction Algorithm to Enhance Grid Resilience Toward Cyber Attacks in WAMCS Applications. *IEEE Syst. J.* **2017**. [CrossRef]
11. Morris, T.; Pan, S.; Lewis, J.; Moorhead, J.; Younan, N.; King, R.; Freund, M.; Madani, V. Cybersecurity Testing of Substation Phasor Measurement Units and Phasor Data Concentrators. In Proceedings of the ACM Seventh Annual Workshop on Cyber Security and Information Intelligence Research, Oak Ridge, TN, USA, 12–14 October 2011.
12. NMAP Security Scanner. Available online: <http://www.nmap.org> (accessed on 26 November 2018).
13. Open Vulnerability Assessment System (OpenVAS). Available online: <http://www.openvas.org/> (accessed on 26 November 2018).
14. Manandhar, K.; Cao, X.; Hu, F.; Liu, Y. Detection of Faults and Attacks Including False Data Injection Attack in Smart Grid Using Kalman Filter. *IEEE Trans. Control Netw. Syst.* **2014**, *1*, 370–379. [CrossRef]
15. Coppolino, L.; Antonio, S.D.; Romano, L. Exposing Vulnerabilities in Electric Power Grids: An Experimental Approach. *Int. J. Crit. Infrastruct. Prot.* **2014**, *7*, 51–60. [CrossRef]
16. D'Antonio, S.; Coppolino, L.; Elia, I.; Formicola, V. Security Issues of a Phasor Data Concentrator for Smart Grid Infrastructure. In Proceedings of the 13th ACM European Workshop on Dependable Computing, Pisa, Italy, 11–12 May 2011.
17. Guidelines for Smart Grid Cyber Security, NIST, September 2014. Available online: <https://nvlpubs.nist.gov/nistpubs/ir/2014/NIST.IR.7628r1.pdf> (accessed on 26 November 2018).

18. Khan, R.; McLaughlin, K.; Lavery, D.; Sezer, S. Design and Implementation of Security Gateway for Synchrophasor Based Real-Time Control and Monitoring in Smart Grid. *IEEE Access* **2017**, *5*, 11626–11644. [CrossRef]
19. Harkins, D.; Carrel, D. The Internet Key Exchange (IKE). In *Internet Engineering Task Force (IETF)—Request for Comments (RFC) No. 2409*; 1998. Available online: <https://www.rfc-editor.org/rfc/pdf/rfc2409.txt.pdf> (accessed on 26 November 2018).
20. Yang, W.; Li, X.; Feng, Z.; Hao, J. TLSsem: A TLS Security-Enhanced Mechanism against MITM Attacks in Public WiFis. In *Proceedings of the 2017 22nd International Conference on Engineering of Complex Computer Systems (ICECCS)*, Fukuoka, Japan, 5–8 November 2017; pp. 30–39.
21. Braun, J. Ubiquitous support of multi path probing: Preventing man in the middle attacks on Internet communication. In *Proceedings of the 2014 IEEE Conference on Communications and Network Security*, San Francisco, CA, USA, 29–31 October 2014; pp. 510–511.
22. Python Interface to OpenSSL. Available online: <https://pyopenssl.org/en/stable/api.html> (accessed on 26 November 2018).
23. Ustun, T.S. Interoperability and interchangeability for microgrid protection systems using IEC 61850 standard. In *Proceedings of the 2016 IEEE International Conference on Power and Energy (PECon)*, Melaka, Malaysia, 28–29 November 2016; pp. 7–12.
24. Weis, B.; Rowles, S.; Hardjono, T. The Group Domain of Interpretation. In *Internet Engineering Task Force (IETF)—Request for Comments (RFC) No. 6407*; 2011. Available online: <http://www.rfc-editor.org/info/rfc6407> (accessed on 26 November 2018).
25. Cooper, D.; Santesson, S.; Farrell, S.; Boeyen, S.; Housley, R.; Polk, W. Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) profile. In *Network Working Group—Request for Comments (RFC) No. 5280*; 2008. Available online: <https://www.rfc-editor.org/info/rfc5280> (accessed on 26 November 2018).

© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).