Actively Secure Authenticated Data Acquisition Protocols: Advancing Substation Automation Security

Shabnam Saderi Oskouei, Oyonika Samazder,
Gaurav Vinay Uttarkar and Kalikinkar Mandal

November 24, 2024

Actively Secure Authenticated Data Acquisition Protocols: Advancing Substation Automation Security

Shabnam Saderi Oskouei[0009-0000-3422-2216], Oyonika Samazder[0009-0007-6990-6942], Gaurav Vinay Uttarkar, and Kalikinkar Mandal[0000-0002-8228-5016]

Canadian Institute for Cybersecurity
University of New Brunswick
Faculty of Computer Science
Fredericton, NB, E3B 5A3, CANADA
{sh.saderi,oyonika.samazder,gauravvinay.u8,kmandal}@unb.ca

Abstract. The deployment of sensors and electronic devices has become instrumental in realizing the functionality of a smart grid. Secure operational data collection via communication protocols from field devices by the supervisory control and data acquisition (SCADA) at the utility control center is fundamental for the secure and reliable operation of smart grids. In this work, we propose two efficient transport-layer protocols for authenticated acquisition of data from smart grid devices. Our first protocol is based on a pre-shared key, and the second protocol is certificate-based. The constructions of our protocols are based on computationally efficient cryptographic primitives such as lightweight authenticated encryption, digital signatures, and elliptic curve Diffie-Hellman (ECDH) computations. We formally prove the security of our protocols against both active and passive adversaries. Finally, to demonstrate their practicality, we implement our protocols and perform comparative analysis with other contemporary protocols.

Keywords: Secure transport protocol · Authenticated data collection · SCADA · Smart grid security.

1 Introduction

Smart grid is a modernized version of the power grid where advanced computing and information and communication technologies are deployed to enable improved real-time demand-response management, high-quality energy delivery, reliability, automation, and customer services. In the smart grid, key components include advanced digital communication networks, supervisory control and data acquisition (SCADA), and advanced metering infrastructure (AMI). Communication within the smart grid relies on both wired and wireless technologies, encompassing wide area network (WAN), neighborhood area network (NAN), and home area network (HAN). A SCADA in the smart grid plays a crucial role in real-time monitoring and controlling smart grid assets using centralized data acquisition and supervisory control, while simultaneously collecting and analyzing the power grid's data. The communication and control in the smart grid is hierarchical. For instance, a SCADA control server placed at a control center can control field devices such as remote terminal units (RTUs) and/or programmable logic controllers (PLCs) that control actuators and/or monitor sensors [38].

The electricity network consists of digital devices in components such as electricity generation, distribution, transmission, and electricity consumers/customers domain. Over the decades, numerous communication protocols have been designed to facilitate the exchange of information within the power grid. For instance, a SCADA server communicates with substation field devices through different networks and protocols. Examples of industrial standard protocols include Modbus/TCP [1], IEC 61850 [9], ICCP [10] and DNP3 [6].

Data collection is an integral task for the smart grid operation. The SCADA collects data in (near) real-time from various grid components such as smart meters, switches, reclosers, capacitor banks, intelligent electronic devices (IEDs), transformers, relays, and actuators. With the advent of industrial internet of things (IIoT), IIoT devices are deployed in various components of the smart grid, namely substation automation, grid distribution management, network management, grid asset management, and AMI. The deployment of IIoT in the smart grid offers numerous benefits, including better planning, improved decision making, enhanced safety, and reliability.

The rapid adoption of commercialized technologies and (public) Internet, cybersecurity is a growing concern for the power grid as the network-connected operational technology (OT) and information technology (IT) are integrated to modernize power grids, making them resilient, efficient, and smarter with automated operations. The cybersecurity landscape is dynamic due to the development of new and sophisticated attack vectors. Recent instances have evidenced blackouts nationwide due to cyberattacks [11,22].

In recent years, several standardization initiatives for securing the smart grid or industrial control systems (ICS) have been taken by organizations, for instance, by the national institute of standards and technology (NIST) [38,21] and north american electric reliability corporation (NERC) [4]. The NIST recommends the security criteria of ICS for different types of information such as sensors data, admin information and SCADA system data [38]. For instance, sensors' data integrity and availability are of primary importance than the confidentiality of data. On the other hand, the SCADA system's data integrity and availability are high and confidentiality is moderate [38].

As many of the industrial standard communication protocols such as Modbus/TCP, DNP3, and ICCP were designed decades ago, there is no built-in security or security was not a priority due to the communication in private or isolated networks [24,19,35,20]. Typically, these protocols do not have any encryption or authentication enabled. Thus, a man-in-the-middle (MITM) or any protocol manipulation attacks will be successful [35]. To overcome the security issues in these protocols, secure variants of these protocols such as Modbus/TCP security [2], DNP3-secure authentication (DNP3-SA) [7] and secure ICCP [8] have been proposed. The blueprint to secure many of these application layer protocols is to tunnel via the transport layer security (TLS) protocol to provide authentication and encryption [2,8]. Following that, PLC and RTU manufacturers have implemented TLS or Internet protocol security (IPSec) (e.g., Modicon M580) for secure communication in the smart grid [3]. Few attacks on the secure variant of DNP3 were found, e.g., in [14].

The secure transmission of information is critical for secure smart grid data acquisition. In the smart grid, the control center needs to collect data periodically — every 15 or 30 minutes or more frequently — from devices located in substations or fields for various purposes. Ensuring the authenticity and integrity of this data is of paramount importance. For faster data collection, the key constraints are: 1) the computational complexity of cryptographic algorithms which should be efficient, and 2) the control center/server side's computational complexity and parallelization to run multi-instances of security protocols.

As data is collected periodically, running a TLS or IPSec like heavy secure communication protocol designed for the dynamic Internet environment is expensive due to the number of communication rounds and computationally heavy cryptographic primitives.

Given that the communication patterns and device topology in smart grids are fixed, it is crucial to use efficient security protocols for secure data collection and exchange. Inspired by the TLS protocol blueprint, our goal in this paper is to design transport security protocols specifically tailored for smart grid applications. These protocols aim to reduce the number of communications by leveraging inherent smart grid properties, such as periodic data collection and the implementation of only one or two ciphers per device, which contrasts with the more heterogeneous environments of Internet devices.

The application of these protocols is to collect data from PLCs/RTUs and sensors, sending control commands to actuators, sensors, and RTUs/PLCs in an authenticated manner, and also securely collecting logs for security monitoring. The main requirements for secure smart grid data collection are timeliness, integrity, confidentiality protection, and scalability.

Our contribution. Similar to other security protocols, such as TLS, at a high level, our security protocol has two phases: 1) a mutual entity authentication and generation of a session key; and 2) a secure data acquisition phase which assures the protection of measurement data traffic with confidentiality and integrity. The key objective of our work is to design an efficient mutual authentication and key establishment protocol tailored to the needs of the smart grid. Our contributions are summarized as follows:

- We propose two efficient mutual entity authentication and key establishment protocols for secure smart grid data collection. Like the CoAP security protocol [36], we propose two variants, namely pre-shared key (PSK-ADA) and certificate-based (Cert-ADA) protocols to support a variety of smart grid devices. Our first protocol is based on the pre-shared key which improves over the SSTP protocol in [27] in terms of cryptographic primitive computations and communication overhead. The construction of the PSK-ADA protocol is based on two primitives namely a lightweight authenticated encryption scheme and an ECDH key agreement protocol. The construction of our Cert-ADA protocol is based on signatures, a lightweight authenticated encryption with associated data (AEAD) scheme, and the Diffie–Hellman (DH) key agreement. Both protocols enjoy minimal rounds, including the TCP connection request step. We prove the security of our protocols against external adversaries, and estimate the computational and storage complexity of our protocols.

- We implement and evaluate our protocols in C++ using OpenSSL [5]. In our experiment, for our protocols, we use the lightweight authenticated cipher ASCON as AEAD and ASCON hash as a KDF to derive session keys, and OpenSSL is for the ECDH computation. We present the results for the execution time and data transfer overhead for cryptographic operations to perform entity authentication and establish session keys. Finally, we compare our protocols with SSTP which is closest to ours, and TLS.

2 Related Work

Transport Protocols. Transmission control protocol (TCP) is intended for use as a highly reliable host-to-host protocol in packet-switched computer communication networks, and in interconnected systems of such networks [34]. The stream control transmission protocol (SCTP) [37], offers a reliable, point-to-point, and connection-oriented data transport service over IP networks, similar to TCP. SCTP is implemented in the kernel of the operating system, providing a dependable and secure protocol for transporting critical data. In [28], a transmission control protocol designed for delay-sensitive smart grid applications was introduced. This protocol aims to minimize the end-to-end delay, although it does not address congestion control. In [26], the authors mentioned that TCP is unsuitable for data collection since it suffers from excessive signaling messages and packet retransmissions. They designed a mechanism with a TCP aggregator node, which collects and aggregates data in a less number of connections. They claim that congestion and flow control can be performed effectively. The problem with their scheme is that confidentiality will not be maintained.

Secure Data Collection. Devices need extreme computational power and massive memory capacity to do cryptographic operations in the TLS protocol [32]. That is why TLS may not be suitable for lightweight data collection. Moreover, when designing a data collection protocol, interoperability, and legacy compliance are critical aspects [16]. As a result, using internet security protocols is not promising for data collection purposes. For instance, IPsec [12] is not a suitable scheme for data collection as it does not offer scalability and extensibility.

In [42], a data collection scheme has been proposed that uses homomorphic encryption. They claim that using their method can detect whether the data packet is tampered by the adversary during transmission. This method as they mentioned needs a third party and also the user authentication is not considered. In [33], an efficient scheme that enables data collection while preserving consumer privacy was proposed. This scheme primarily uses lightweight symmetric key cryptography and hashing operations to collect data. This design only uses asymmetric key cryptography for key management. Further, a PUF-based data collection in smart grid proposed to guarantee communication between smart meters and the control center [13]. The authors in this work claimed that the scheme can provide physical security for smart meters and add a lesser computational overhead to system.

Hierarchical Data Collection. In [41] and [18], a hierarchical architecture consisting of measurement devices, data collectors, and power operators was proposed. The measurement devices encrypt generated data, and the data collectors relay those data from the measurement devices to the power operator. A similar approach was used in [25] where the authors focused on reducing the data collection time within a tree-based smart grid data collection environment. Another work that addressed the time issue is [31] where authors analyzed the delay minimization problem and proposed a time-efficient algorithm to set up direct cellular links on a subset of nodes in the power line sensor net (PLSN) to minimize the data collection delay. In [40], measurement devices transmit data to the power operator through intermediary data collectors. The protocol employs DH key exchange and asymmetric cryptography for establishing hop-by-hop and end-to-end keys.

End-to-end Secure Communication. The work [27] proposed a protocol called SSTP using AES and the DH protocol for the smart grid data collection. REMP [29] is a designed approach for end-to-end secure and scalable communication for resource constrained devices. Authors in their work on cyber physical systems (CPS), proposed a transparent end-to-end encryption scheme that requires a trusted key server to distribute the topic-specific key to all authorized clients [15]. In [23], an end-to-end encryption approach was presented for securing communications and ensuring the confidentiality and integrity of shared data.

Table 1: Comparison of Our Protocols with Other Related Secure Data Transport Protocols.

Features	**TCP** [34]	**SSTP** [27]	**SELINDA** [18]	**SSDC** [41]	**PSK-ADA**	**Cert-ADA**
Security Scheme	IPsec/TLS	Built-in	Built-in	Built-in	Built-in	Built-in
Connection Establishment	3 Message	4 Message	4 Message	5 Message	4 Message	4 Message
In-order Delivery	Mandatory	None	None	None	None	None
Reliable Delivery	ACK	ACK	Signature	Signature	ACK	ACK
End-to-End Secure	No	Yes	Yes	Yes	Yes	Yes
Identity Hiding	No	No	No	No	Yes	Yes

Table 1 presents a comprehensive comparison of the security features across various data transport schemes, ranging from traditional methods like TCP and SSTP to our proposed protocols, PSK-ADA and Cert-ADA.

Key advantages of PSK-ADA and Cert-ADA include inherent *identity hiding* to ensure that the identities of communicating entities are protected throughout the transmission process, and *end-to-end encryption*, which secures data from source to destination without relying on intermediate nodes. Additionally, the protocols support reliable delivery through mandatory ACK messages, ensuring critical data is transmitted securely and received as intended, mitigating the risks of data manipulation or loss. By integrating these security enhancements, PSK-ADA and Cert-ADA provide a comprehensive solution tailored to the specific needs of smart grid systems, positioning them as a significant contribution to the field by enabling secure, scalable, and resilient communication channels that enhance privacy and data integrity against evolving cyber threats.

3 Background

In this section, we provide a brief background on the cryptographic primitives and algorithms that will be used in our protocols.

3.1 Cryptographic Primitives

Authenticated Encryption. An authenticated encryption with associated data AEAD scheme consists of three distinct algorithms $\mathsf{AEAD} = (\mathsf{AEAD.KeyGen}, \mathsf{AEAD.Enc}, \mathsf{AEAD.Dec})$. On a security parameter κ, the key generation algorithm $K \leftarrow \mathsf{AEAD.KeyGen}(1^\kappa)$ samples a symmetric-key (K) that is used in the encryption and decryption algorithms. The encryption algorithm $(C, \mathsf{tag}) \leftarrow \mathsf{AEAD.Enc}(K, AD, M)$ accepts a key K, an associated data (AD) and a plaintext message (M) to be encrypted and produces a ciphertext (C) and a tag (tag). Similarly, the decryption algorithm $\{M, \perp\} \leftarrow \mathsf{AEAD.Dec}(K, AD, C, \mathsf{tag})$ takes a key (K), an associated data (AD), a ciphertext (C) and a tag (tag) as input, and outputs the plaintext message (M) or $\perp$. An AEAD scheme should have indistinguishability under chosen-plaintext attack (IND-CPA) and integrity of ciphertext (INT-CTXT) security to realize an authenticated channel.

Message Authentication Code. A message authentication code (MAC) is a tuple of two deterministic algorithms $\mathsf{MAC} = (\mathsf{MAC.TGen}, \mathsf{MAC.Verify})$ where the tag generation algorithm $\mathsf{MAC.TGen} : \mathcal{K} \times \mathcal{M} \rightarrow \mathcal{T}$ accepts a key from $\mathcal{K}$ and a message from $\mathcal{M}$ and outputs a tag in $\mathcal{T}$, and the tag verification algorithm $\mathsf{MAC.Verify} : \mathcal{K} \times \mathcal{M} \times \mathcal{T} \rightarrow \{\text{yes}, \text{no}\}$ accepts a key from $\mathcal{K}$, a message from $\mathcal{M}$ and a tag from $\mathcal{T}$ and outputs either yes or no if the verification succeeds or fails, respectively. For a message $M \in \mathcal{M}$ and $K \in \mathcal{K}$, $\mathsf{MAC.TGen}$ produces a tag, $\mathsf{tag} \leftarrow \mathsf{MAC.TGen}(K, M)$, and the verification algorithm outputs $\{\text{yes}, \text{no}\} \leftarrow \mathsf{MAC.Verify}(K, M, \mathsf{tag})$. We require the unforgeability under chosen message attack (UF-CMA) security for the MAC algorithm.

Digital Signature. A digital signature algorithm is a tuple of three algorithms: $\mathsf{SIG} = (\mathsf{SIG.KeyGen}, \mathsf{SIG.Sign}, \mathsf{SIG.Verify})$. On a security parameter κ, the key generation algorithm $(sk, vk) \leftarrow \mathsf{SIG.KeyGen}(1^\lambda)$ generates a signing key sk (private) and a verification key vk (public). The signing algorithm $\sigma \leftarrow \mathsf{SIG.Sign}(sk, m)$ accepts the private signing key (sk) and a message (m) and produces a signature (σ). The verification algorithm $\{\text{yes}, \text{no}\} \leftarrow \mathsf{SIG.Verify}(vk, m, \sigma)$ accepts the verification key (vk), the message (m) and a signature (σ) and output a decision that the signature verification is successful (yes) or failed (no). We require the existential unforgeability under chosen message attack (EUF-CMA) security for the SIG algorithm

3.2 Key Agreement Protocol

A key agreement protocol is a tuple of three algorithms $\mathsf{KE} = (\mathsf{KE.ParamGen}$, $\mathsf{KE.KeyGen}$, $\mathsf{KE.KeyAgree})$. As an example, we use an elliptic curve (EC) variant to explain the DH key agreement protocol. Given a security parameter κ, the parameter generation algorithm $(\mathbb{G}, q, P) \leftarrow \mathsf{KE.ParamGen}(1^\kappa)$ samples an elliptic curve over a field $\mathbb{F}_q$ of order q and a generator P. For a user A, the key generation algorithm $(x_A, P_A) \leftarrow \mathsf{KE.KeyGen}(\mathbb{G}, q, P)$ generates a secret key $x_A \leftarrow \mathbb{Z}_q$ and a public key $P_A = x_A P$, and similarly, the key (x_B, P_B) is for user B. The key agreement function $P_{AB} \leftarrow \mathsf{KE.KeyAgree}(x_A, P_B) = x_A x_B P$ computes a pairwise key $K_{AB} = \mathsf{KDF}(P_{AB})$ where $\mathtt{KDF}$ is a key derivation function. The security of the key agreement protocol follows from the decisional Diffie-Hellman (DDH) assumption.

4 Our Authenticated Data Acquisition Protocols

Our SCADA system model, similar to the one considered by the NIST [39], consists of a SCADA server and multiple field devices connected via a communication network, enabling data collection, device control, and urgent data reporting between them (see Figure 1). We present two secure data collection/acquisition protocols that run over TCP for the SCADA system.

Our primary objective is to achieve integrity by verifying the authenticity of both the client and server. Moreover, we place the utmost importance on safeguarding the client's identity during the initial communication establishment by implementing robust measures for protection. To ensure security, we also rely on the ECDH algorithm to establish cryptographic keys for the subsequent data acquisition phase, thus ensuring the confidentiality of all transmitted information. The protocols we designed not only strengthen security but also enhance scalability. They allow seamless accommodation of an increasing number of clients and servers while maintaining the integrity and effectiveness of the security measures.

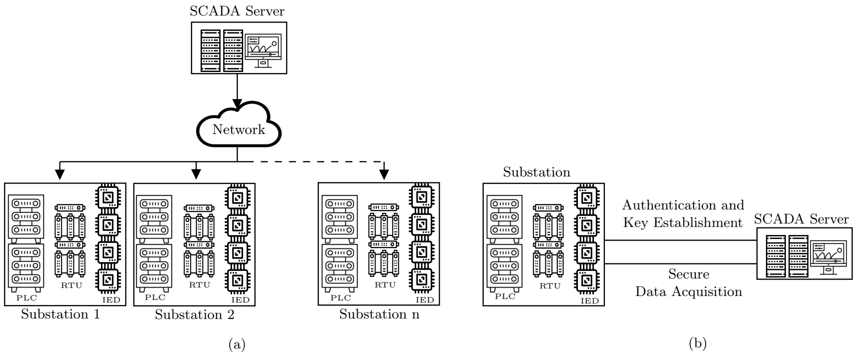

Fig. 1: An Overview of SCADA System Model and Secure Data Acquisition.

4.1 PSK-ADA: Pre-shared Key based Protocol

Our protocol, illustrated in Figure 2, is based on a pre-shared key approach and ensures secure communication and interoperability among devices in the smart grid. We assume that all devices in the network have agreed upon the same Elliptic curve parameters. The protocol generates a shared session key K_C^e from the Diffie-Hellman (DH) value $P_{SC} = xyP$. This key is used in the AEAD algorithm to secure the client's critical measurement data m^t during the data acquisition phase (in the time interval t).

Description of the PSK-ADA protocol. The $\mathsf{PSK\text{-}ADA}$ protocol uses a lightweight authenticated encryption scheme based on nonces. This scheme has two primary functions: it computes message authentication codes (MACs) and encrypts data. For simplicity, the nonce is not shown in Figure 2.

- **Shared Key and Device Identification:** The server (S) and client (C) share a long-term master key K_C, installed by the device manufacturer. Each client device is identified by an ID (ID_C).
- **ECDH Key Establishment:** To minimize communication overhead, the protocol uses the ECDH variant of the DH key exchange protocol. All messages during connection establishment, including TCP SYN, ACK, and state information, are protected, similar to the SSTP protocol [27].
- **Initial Client SYN Protection:** In the first round, the client's SYN packet is protected by a tag generated with the pre-shared key K_C. This ensures that only authorized clients can start the session key setup with the server, protecting the integrity of the connection and preventing unauthorized tampering or illegitimate connections.
- **Identity Anonymity and Message Freshness:** In the second round, AEAD is used to ensure identity anonymity and message freshness through a timestamp and counter. This round encrypts the identity ID_C for confidentiality and uses an associated data $AD_S = (P_x||ACK_1||ctr_1||ts_1)$ and creates a tag for integrity.

- **Session Key Establishment:** In the third step, after receiving $AD_C = P_y||SYN_2||ctr_2||ts_2$, TKN_1^C and tag_2^C from the server, the client derives the session key $KS_C = \mathsf{KDF}(xyP)$ using a key derivation function (KDF). Similarly for the client. From this point, the session key KS_C will be used for further communication.
- **Acknowledgment from Server:** The server sends an acknowledgment (ACK) and a tag to the client, notifying it that the session key has been successfully established.

Security. Our PSK-ADA protocol is designed to guarantee an authenticated TCP connection establishment between the client and the server while protecting their identities and ensuring authenticated key establishment. Once the secure key establishment is complete, the server securely acquires the data from a client with confidentiality and integrity protection. Theorem 1 summarizes the security of the PSK-ADA protocol.

Theorem 1. *Suppose the AEAD algorithm is IND-CPA secure, the MAC algorithm is UF-CMA secure, and* KDF *is secure under the random oracle model. Our PSK-ADA protocol in Figure 2 is secure against active adversaries.*

Proof. To establish the security of the PSK-ADA protocol, we must consider the integration and interaction of its cryptographic primitives — AEAD, MAC, and KDF — under the stated security models. We assume an active adversary, denoted as $\mathcal{A}$, is capable of intercepting, modifying, and injecting messages.

- An AEAD scheme in the PSK-ADA protocol is IND-CPA secure if no polynomial time adversary $\mathcal{A}$ can distinguish between the encryption of two chosen inputs $C_0 = (AD_0, p_0)$, and $C_1 = (AD_1, p_1)$, where $AD_0 = (P_x \parallel ACK_1 \parallel ctr_1 \parallel ts_1)$, $p_0 = (ID_C)$, and similarly for $C_1 = (p_1, AD_1)$. This is formally defined as:

  $$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{AEAD},\mathcal{A}}(\kappa) = \Big|\Pr[\mathcal{A}(\mathsf{AEAD.Enc}(K_C, C_0)) = 1] - \Pr[\mathcal{A}(\mathsf{AEAD.Enc}(K_C, C_1)) = 1]\Big| \leq \mathsf{negl}(\kappa)$$

  Here, $\mathsf{AEAD.Enc}(K_C, C)$ is the encryption function with key K_C and security parameter κ. If $\mathcal{A}$ can distinguish $\mathsf{AEAD.Enc}(K_C, C_0)$ from $\mathsf{AEAD.Enc}(K_C, C_1)$ with non-negligible advantage, it implies:

  $$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{AEAD},\mathcal{A}}(\kappa) > \mathsf{negl}(\kappa),$$

  which contradicts the IND-CPA security definition of the AEAD scheme.
- The MAC scheme is UF-CMA secure if no adversary $\mathcal{A}$ can forge a valid MAC tag tag_1^C without access to K_C or querying the MAC oracle.
- If adversary $\mathcal{A}$ is capable of deriving KS_C from publicly observable or intercepted values (such as the public components of an ECDH), this indicates that $\mathcal{A}$ has effectively found a way to invert or predict the output of the KDF. This capability would demonstrate that the KDF does not conform to the random oracle model, as it fails to provide output that is indistinguishable from random.

Hence, the PSK-ADA protocol is secure against active adversaries under the IND-CPA security of AEAD, UF-CMA security of MAC, and the KDF's random oracle model adherence.

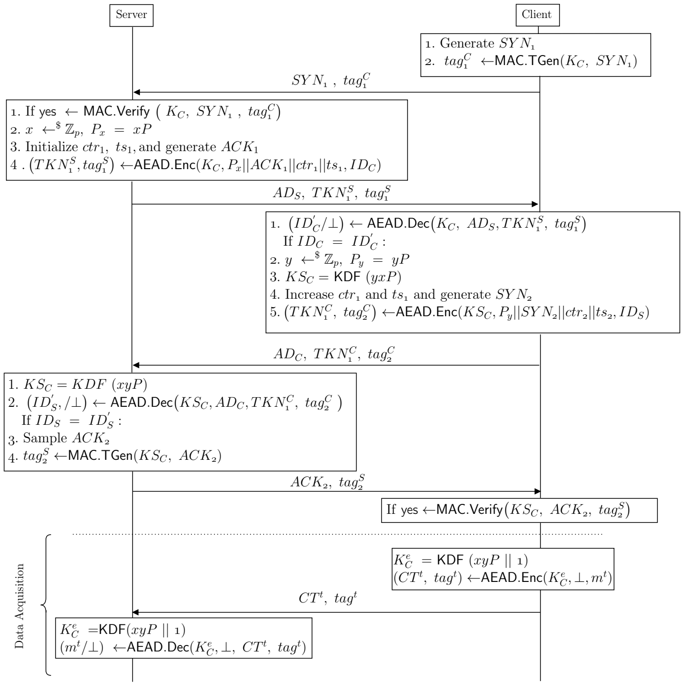
Server
Client
1. Generate SYN_1
2. tag_1^C $\leftarrow$MAC.TGen$(K_C,\ SYN_1)$
SYN_1 , tag_1^C
1. If yes $\leftarrow$ MAC.Verify $(\ K_C,\ SYN_1\ ,\ tag_1^C)$
2. $x \xleftarrow{\$} \mathbb{Z}_p,\ P_x\ =\ xP$
3. Initialize $ctr_1,\ ts_1,$ and generate ACK_1
4 . (TKN_1^S, tag_1^S) $\leftarrow$AEAD.Enc$(K_C, P_x||ACK_1||ctr_1||ts_1, ID_C)$
$AD_S,\ TKN_1^S,\ tag_1^S$
1. $(ID_C^{'}/\bot)$ $\leftarrow$ AEAD.Dec$(K_C,\ AD_S, TKN_1^S,\ tag_1^S)$
If $ID_C\ =\ ID_C^{'}$:
2. $y \xleftarrow{\$} \mathbb{Z}_p,\ P_y\ =\ yP$
3. $KS_C =$ KDF (yxP)
4. Increase ctr_1 and ts_1 and generate SYN_2
5. $(TKN_1^C,\ tag_2^C)$ $\leftarrow$AEAD.Enc$(KS_C, P_y||SYN_2||ctr_2||ts_2, ID_S)$
$AD_C,\ TKN_1^C,\ tag_2^C$
1. $KS_C = KDF\ (xyP)$
2. $(ID_S^{'}, /\bot)$ $\leftarrow$ AEAD.Dec$(KS_C, AD_C, TKN_1^C,\ tag_2^C\ )$
If $ID_S\ =\ ID_S^{'}$:
3. Sample ACK_2
4. tag_2^S $\leftarrow$MAC.TGen$(KS_C,\ ACK_2)$
$ACK_2,\ tag_2^S$
If yes $\leftarrow$MAC.Verify$(KS_C,\ ACK_2,\ tag_2^S)$
Data Acquisition
$K_C^e\ =$ KDF $(xyP\ ||\ 1)$
$(CT^t,\ tag^t)$ $\leftarrow$AEAD.Enc$(K_C^e, \bot, m^t)$
$CT^t,\ tag^t$
$K_C^e\ =$KDF$(xyP\ ||\ 1)$
$(m^t/\bot)$ $\leftarrow$AEAD.Dec$(K_C^e, \bot,\ CT^t,\ tag^t)$

Fig. 2: PSK-ADA: Pre-shared Key based Authenticated Data Acquisition Protocol.

4.2 Cert-ADA: Certificate-based Protocol

Our certificate-based protocol, shown in Figure 3, uses the notations described in Section 3. Each device and the server possess a certificate that verifies the signature verification key, and all devices agree on the same elliptic curve parameters. To protect the identities of communicating devices, we leverage the Sigma protocol [30] with the TCP protocol to design a transport security protocol. In the secure data acquisition or transfer phase, we employ a lightweight single-pass AEAD scheme for encryption and authentication, utilizing a single session key. By efficiently encrypting and authenticating protocol messages using a tweaked Sigma protocol and a lightweight AEAD scheme, we ensure a secure communication.

Description of the Cert-ADA protocol. The Cert-ADA protocol uses a combination of lightweight AEAD, a digital signature scheme, and the DH key agreement scheme. Each step (shown in Figure 3) builds upon these elements to secure communication between the client and server.

- **SYN and ACK Authentication:** In Steps 1 and 2 (each box in Protocol II is a step), SYN and ACK messages (SYN_1 and ACK_1) are included as associated data for authentication in steps 3 and 4 (SYN_2 and ACK_2),

without needing to resend these messages. Key confirmation in these steps is automatically provided by the Sigma protocol.
- **State Information Protection:** This protocol encrypts and authenticates critical state information, such as ID_C, timestamp (ts), and counter (ctr), using an AEAD scheme in steps 2 and 3. The associated data for step 2 is $AD_C = (SYN_1||SYN_2||ctr_1||ts_1||P_y||\sigma_C)$, and for step 3, it is $AD_S = (ACK_1||ACK_2||ctr_2||ts_2||\sigma_S)$. This ensures identity anonymity and communication freshness.
- **Master Key and Session Key Usage:** The pre-shared master key K_C is used in the first two rounds, while the agreed DH session key $KS_C = \mathsf{KDF}(xyP)$ is used in the last two rounds for key confirmation. This prevents any modifications or MITM attacks during the DH key agreement.
- **Shared Session Key Derivation:** The protocol output is a shared session key KS_C, derived from the DH value $P_{SC} = xyP$, using $KS_C = \text{KDF}(P_{SC} \mid 1)$. This session key is used to secure the client's measurement data mt during the data acquisition phase.

Security. Our $\mathsf{Cert\text{-}ADA}$ protocol is designed to ensure an authenticated TCP connection establishment between the client and the server while protecting their identities and enabling an authenticated key establishment process. The protocol assures that once the key establishment is completed, the server can securely acquire data from the client, maintaining both confidentiality and integrity. Theorem 2 comprehensively outlines the security measures embedded within the $\mathsf{Cert\text{-}ADA}$ protocol. Certificates play a critical role in this protocol by authenticating public keys, guaranteeing the security during the initial key exchange and enhancing the reliability of identity verification processes.

Theorem 2. *Suppose the AEAD algorithm is IND-CPA secure, the SIG algorithm is EUF-CMA secure, and KDF is secure under random oracle model. Our $\mathsf{Cert\text{-}ADA}$ protocol in Figure 3 is secure against active attacks.*

Proof. To establish the security of the $\mathsf{Cert\text{-}ADA}$ protocol, we must consider the integration and interaction of its cryptographic primitives — AEAD, MAC, SIG and KDF — under the stated security models. We assume an active adversary, denoted as $\mathcal{A}$, is capable of intercepting, modifying, and injecting messages.

- An AEAD scheme in the $\mathsf{PSK\text{-}ADA}$ protocol is IND-CPA secure if no polynomial-time adversary $\mathcal{A}$ can distinguish between the encryption of two chosen inputs $C_0 = (AD_0, p_0)$ and $C_1 = (AD_1, p_1)$, where $AD_0 = (SYN_1||SYN_2||ctr_0|| ts_0||P_y||\sigma_C)$, $p_0 = ID_C$, and similarly for $C_1 = (p_1, AD_1)$. This is formally defined as:

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{AEAD},\mathcal{A}}(\kappa) = \left| \Pr[\mathcal{A}(\mathsf{AEAD.Enc}(KS_C, C_0)) = 1] - \Pr[\mathcal{A}(\mathsf{AEAD.Enc}(KS_C, C_1)) = 1] \right| \leq \mathsf{negl}(\kappa)$$

Here, $\mathsf{AEAD.Enc}(K_C, C)$ is the encryption function under key K_C, with κ as the security parameter. If an adversary $\mathcal{A}$ is capable of distinguishing between $\mathsf{AEAD.Enc}(K_C, C_0)$ and $\mathsf{AEAD.Enc}(K_C, C_1)$ with non-negligible advantage, this implies:

$$\mathsf{Adv}^{\mathsf{ind}}_{\mathsf{AEAD},\mathcal{A}}(\kappa) > \mathsf{negl}(\kappa),$$

which contradicts the IND-CPA security definition of the AEAD scheme.

- Assume that adversary $\mathcal{A}$ can forge a MAC. This means $\mathcal{A}$ has successfully computed tag_1^C, without access to the session key KS_C and without making a query for the MAC oracle during the learning phase. By the security definition of UF-CMA, the probability that $\mathcal{A}$ succeeds should be negligible.
- If adversary $\mathcal{A}$ is capable of deriving KS_C from from observed or intercepted values, this means that $\mathcal{A}$ has effectively found a way to invert or predict the output of the KDF. This capability would suggest that the KDF fails to adhere to the random oracle model, as it does not provide outputs that are indistinguishable from random.

Together, these properties ensure the Cert-ADA protocol is secure against active attacks.

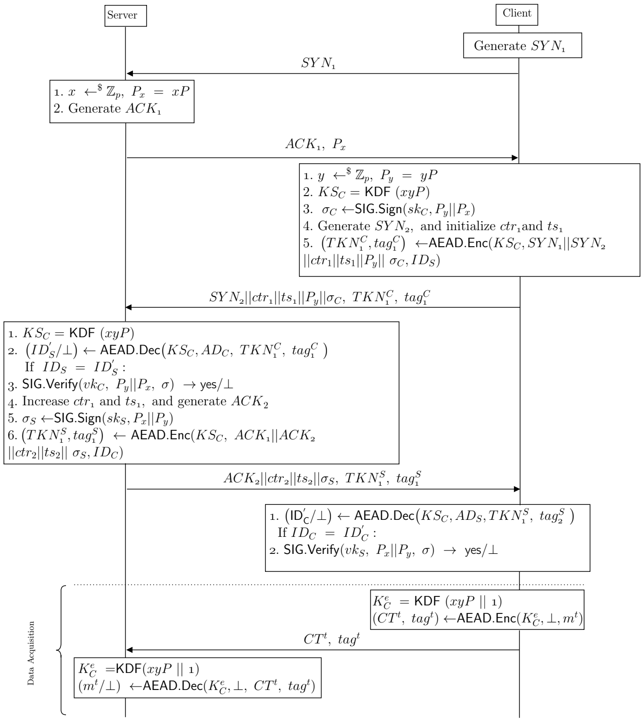
Server
Client
Generate SYN_1
SYN_1
1. $x \xleftarrow{\$} \mathbb{Z}_p,\ P_x = xP$
2. Generate ACK_1
$ACK_1,\ P_x$
1. $y \xleftarrow{\$} \mathbb{Z}_p,\ P_y = yP$
2. $KS_C = \mathsf{KDF}\ (xyP)$
3. $\sigma_C \leftarrow \mathsf{SIG.Sign}(sk_C, P_y||P_x)$
4. Generate SYN_2, and initialize $ctr_1$and ts_1
5. $(TKN_1^C, tag_1^C) \leftarrow \mathsf{AEAD.Enc}(KS_C, SYN_1||SYN_2$
$||ctr_1||ts_1||P_y||\ \sigma_C, ID_S)$
$SYN_2||ctr_1||ts_1||P_y||\sigma_C,\ TKN_1^C,\ tag_1^C$
1. $KS_C = \mathsf{KDF}\ (xyP)$
2. $(ID'_S/\perp) \leftarrow \mathsf{AEAD.Dec}(KS_C, AD_C,\ TKN_1^C,\ tag_1^C\ )$
If $ID_S = ID'_S$:
3. $\mathsf{SIG.Verify}(vk_C,\ P_y||P_x,\ \sigma) \rightarrow \mathsf{yes}/\perp$
4. Increase ctr_1 and ts_1, and generate ACK_2
5. $\sigma_S \leftarrow \mathsf{SIG.Sign}(sk_S, P_x||P_y)$
6. $(TKN_1^S, tag_1^S) \leftarrow \mathsf{AEAD.Enc}(KS_C,\ ACK_1||ACK_2$
$||ctr_2||ts_2||\ \sigma_S, ID_C)$
$ACK_2||ctr_2||ts_2||\sigma_S,\ TKN_1^S,\ tag_1^S$
1. $(\mathsf{ID}'_\mathsf{C}/\perp) \leftarrow \mathsf{AEAD.Dec}(KS_C, AD_S, TKN_1^S,\ tag_2^S\ )$
If $ID_C = ID'_C$:
2. $\mathsf{SIG.Verify}(vk_S,\ P_x||P_y,\ \sigma) \rightarrow \mathsf{yes}/\perp$
$K_C^e = \mathsf{KDF}\ (xyP\ ||\ 1)$
$(CT^t,\ tag^t) \leftarrow \mathsf{AEAD.Enc}(K_C^e, \perp, m^t)$
$CT^t,\ tag^t$
$K_C^e = \mathsf{KDF}(xyP\ ||\ 1)$
$(m^t/\perp) \leftarrow \mathsf{AEAD.Dec}(K_C^e, \perp,\ CT^t,\ tag^t)$
Data Acquisition

Fig. 3: Cert-ADA: Certificate based Authenticated Data Acquisition Protocol.

5 Performance Evaluation

In this section, we evaluate the performance of our data transport security protocols in Figures 2 and 3 for a 128-bit security level. We provide the timing and communication overhead results for both protocols.

Experimental setup. We implement our protocols in C++ using OpenSSL 3.0.5 [5] in a desktop environment. In our experiment, we instantiate AEAD by the sponge-based lightweight ASCON cipher [17] standardized by the NIST. We also use ASCON for the message authentication code (MAC) and the KDF in the XOF mode to derive session keys. We leverage the OpenSSL library for the ECDH key agreement computation and use the curve "P-256" in our experiment. We enabled a client-server communication in the LAN at a speed of 1 Gbps. To achieve the best efficiency, we implement ASCON's AEAD, MAC and XOF mode using the SIMD instruction (SSE2) using Intel intrinsics. As the SSTP protocol is similar to our first protocol, we implement it using the primitives recommended by the authors in [27].

Our experiments were conducted on two desktops with 2.90GHz Intel i7-10700 CPU and 32 GB RAM where one desktop is configured as a client and another desktop is configured as the server. The codes were compiled using `g++ 9.4.0` with `-fomit-frame-pointer -funroll-all-loops -march=native` flag.

Experimental results. We report the execution time and data transfer overhead by the cryptographic operations. The execution time involves performing entity authentication and establishing session keys. Finally, we compare our protocols with the SSTP and TLS protocols. Table 2 presents the execution time and the amount of data transfer by our protocols. We run our experiment 10 times to captures the execution time. The execution time is computed by summing up the time taken by all cryptographic primitives and the communication time.

Table 2: Comparing Performance and Data Exchange of Various Protocols.

Protocols	**Time (ms)**	**Parameter size (byte)**	**Total data exchanged (byte)**
Protocol I	5.5	256	544
Protocol II	3.0	256	544
Plain TCP/IP (no security)	0.0036	492	544
TLS	3.4	2695	544
SSTP	7.1	256	2050

Comparison. As seen in Table 2, plain TCP/IP, which lacks security, transmits 544 bytes in 0.0036 ms. Protocol I's total parameter size, calculated as 2 × (SYN size + ACK size + 2 × Tag size + DH PK size + CTR + TS + ID size), employs a 128-bit (16 bytes) configuration for the counter (CTR), timestamp (TS), and device ID. Protocol II has a similar configuration, computed as 2 × (SYN size + ACK size + Tag size + DH PK size + CTR + TS + ID size).

The SSTP protocol uses modular arithmetic-based DH, leading to a total of 2050 bytes due to the inclusion of modulus prime, DH public key, and re-sending the DH shared session key. Protocol II offers the fastest runtime at 3.0 ms with a 544-byte data exchange, the same as Protocol I and TLS. TLS performs well at 3.4 ms for 544 bytes, while SSTP is slower at 7.1 ms and exchanges the largest data volume at 2050 bytes.

For large-scale smart grid systems like AMI and distributed energy resource (DER) management, scalability and low latency are critical. These environments require efficient protocols to support numerous concurrent connections for real-time data monitoring and control, minimizing delay and resource consumption. While TLS provides robust security, its computational demands can increase latency and resource usage, posing challenges for fast, reliable communication across interconnected substations, DERs, and centralized control centers.

Our proposed protocols address these challenges by leveraging lightweight cryptographic primitives and reducing data overhead to ensure secure, low-latency communication under high connection loads. For instance, Protocol II's optimized data exchange and processing make it ideal for real-time data acquisition from smart meters and DERs, where transmission speed and data integrity are essential. Similarly, AMI systems benefit from the reduced overhead, enabling millions of meters to communicate securely without overwhelming the network.

In summary, our protocols provide scalable, efficient end-to-end security solutions tailored to the performance and reliability demands of modern smart grid environments, facilitating the secure and scalable deployment of future-proof energy systems.

6 Conclusion

In this work, we presented two transport layer protocols for the SCADA system to securely acquire data from smart grid devices. The construction of our two protocols is based on computationally-cheap cryptographic primitives such as lightweight single-pass AEAD, digital signatures, and ECDH computations, with the use of a minimal number of primitives. Our protocols are intended to be designed enabling lightweight transport layer security for the smart grid standardized protocols such as Modbus and DNP3. We implemented and presented experimental results on the execution time and communication cost, along with a comparison.

Acknowledgment. This work was supported by the NB Power Cybersecurity Research Chair grant. We extend our sincere gratitude to the reviewers for their valuable feedback, which have contributed to improving this paper.

References

1. Modbus, https://modbus.org/
2. Modbus/tcp security: Protocol specification. https://modbus.org/docs/MB-TCP-Security-v21_2018-07-24.pdf
3. Modicon m580 high-end controller. https://www.se.com/ca/en/product-range/62098-modicon-m580-highend-controller/#overview
4. North american electric reliability corporation, https://www.nerc.com/Pages/default.aspx
5. Openssl. the openssl library., https://www.openssl.org/
6. Overview of dnp3 protocol. https://www.dnp.org/About/Overview-of-DNP3-Protocol

7. Overview of dnp3 security version 6. `http://surl.li/gveqr`
8. Secure iccp. `http://surl.li/gverj`
9. IEC 61850: Communication networks and systems for power utility automation (2010), `https://webstore.iec.ch/publication/6028`, accessed: 2024-09-08
10. Iec 60870-6 (iccp). `https://webstore.iec.ch/publication/3760` (2014)
11. Ukraine power cut 'was cyber-attack' (2021), `https://www.bbc.com/news/technology-38573074`
12. Barker, E., Dang, Q., Frankel, S., Scarfone, K., Wouters, P.: Guide to ipsec vpns (2020-06-30 00:06:00 2020). `https://doi.org/https://doi.org/10.6028/NIST.SP.800-77r1`
13. Cao, Y.N., Wang, Y., Ding, Y., Zheng, H., Guan, Z., Wang, H.: A puf-based lightweight authenticated metering data collection scheme with privacy protection in smart grid. In: 2021 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom). pp. 876–883. IEEE (2021)
14. Cremers, C., Dehnel-Wild, M., Milner, K.: Secure authentication in the grid: A formal analysis of dnp3: Sav5. In: Foley, S.N., Gollmann, D., Snekkenes, E. (eds.) Computer Security – ESORICS 2017. pp. 389–407. Springer International Publishing, Cham (2017)
15. Dahlmanns, M., Pennekamp, J., Fink, I.B., Schoolmann, B., Wehrle, K., Henze, M.: Transparent end-to-end security for publish/subscribe communication in cyber-physical systems. In: Proceedings of the 2021 ACM Workshop on Secure and Trustworthy Cyber-Physical Systems. pp. 78–87 (2021)
16. Dán, G., Sandberg, H., Ekstedt, M., Björkman, G.: Challenges in power system information security. IEEE Security & Privacy Magazine **10**(4), 62–70 (2012)
17. Dobraunig, C., Eichlseder, M., Mendel, F., Schläffer, M.: Ascon v1. 2: Lightweight authenticated encryption and hashing. Journal of Cryptology **34**(3), 1–42 (2021)
18. Dán, G., Lui, K.S., Tabassum, R., Zhu, Q., Nahrstedt, K.: Selinda: A secure, scalable and light-weight data collection protocol for smart grids. In: 2013 IEEE International Conference on Smart Grid Communications (SmartGridComm). pp. 480–485 (2013). `https://doi.org/10.1109/SmartGridComm.2013.6688004`
19. East, S., Butts, J., Papa, M., Shenoi, S.: A taxonomy of attacks on the dnp3 protocol. In: Palmer, C., Shenoi, S. (eds.) Critical Infrastructure Protection III. pp. 67–81. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
20. East, S., Butts, J., Papa, M., Shenoi, S.: A taxonomy of attacks on the dnp3 protocol. In: Palmer, C., Shenoi, S. (eds.) Critical Infrastructure Protection III. pp. 67–81. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
21. Greer, C., Wollman, D.A., Prochaska, D., Boynton, P.A., Mazer, J.A., Nguyen, C., FitzPatrick, G., Nelson, T.L., Koepke, G.H., Hefner Jr, A.R., et al.: Nist framework and roadmap for smart grid interoperability standards, release 3.0 (2014)
22. threaten smart power grids, H.: (accessed in 2021), `https://www.politico.eu/article/smart-grids-and-meters-raise-hacking-risks/`
23. Gupta, S., Sacchetti, T., Crispo, B.: End-to-end encryption for securing communications in industry 4.0. In: 2022 4th IEEE Middle East and North Africa COMMunications Conference (MENACOMM). pp. 153–158. IEEE (2022)
24. Hoyos, J., Dehus, M., Brown, T.X.: Exploiting the goose protocol: A practical attack on cyber-infrastructure. In: 2012 IEEE Globecom Workshops. pp. 1508–1513 (2012). `https://doi.org/10.1109/GLOCOMW.2012.6477809`

25. Jin, H., Uludag, S., Lui, K.S., Nahrstedt, K.: Secure data collection in constrained tree-based smart grid environments. In: 2014 IEEE International Conference on Smart Grid Communications (SmartGridComm). pp. 308–313. IEEE (2014)
26. Khalifa, T., Naik, K., Alsabaan, M., Nayak, A., Goel, N.: Transport protocol for smart grid infrastructure. In: 2010 Second International Conference on Ubiquitous and Future Networks (ICUFN). pp. 320–325. IEEE (2010)
27. Kim, Y.J., Kolesnikov, V., Kim, H., Thottan, M.: Sstp: A scalable and secure transport protocol for smart grid data collection. In: 2011 IEEE international conference on smart grid communications (SmartGridComm). pp. 161–166. IEEE (2011)
28. Kim, Y.J., Thottan, M.: Sgtp: Smart grid transport protocol for secure reliable delivery of periodic real time data. Bell Labs Technical Journal **16**(3), 83–99 (2011)
29. Kim, Y., Kolesnikov, V., Thottan, M.: Resilient end-to-end message protection for cyber-physical system communications. IEEE Transactions on Smart Grid **9**(4), 2478–2487 (2016)
30. Krawczyk, H.: Sigma: The 'sign-and-mac' approach to authenticated diffie-hellman and its use in the ike protocols. In: Boneh, D. (ed.) Advances in Cryptology - CRYPTO 2003. pp. 400–425. Springer Berlin Heidelberg, Berlin, Heidelberg (2003)
31. Li, P., Guo, S.: Delay minimization for reliable data collection on overhead transmission lines in smart grid. In: 2013 Computing, Communications and IT Applications Conference (ComComAp). pp. 147–152. IEEE (2013)
32. McKay, K., Cooper, D.: Guidelines for the selection, configuration, and use of transport layer security (tls) implementations (2019-08-29 2019). `https://doi.org/https://doi.org/10.6028/NIST.SP.800-52r2`
33. Mohammed, H., Tonyali, S., Rabieh, K., Mahmoud, M., Akkaya, K.: Efficient privacy-preserving data collection scheme for smart grid ami networks. In: 2016 IEEE Global Communications Conference (GLOBECOM). pp. 1–6. IEEE (2016)
34. Postei, J.: Transmission control protocol-darpa internet program protocol specification. RFC 793 **9**, 1–85 (1981)
35. Reda, H.T., Ray, B., Peidaee, P., Anwar, A., Mahmood, A., Kalam, A., Islam, N.: Vulnerability and impact analysis of the iec 61850 goose protocol in the smart grid. Sensors **21**(4) (2021). `https://doi.org/10.3390/s21041554`, `https://www.mdpi.com/1424-8220/21/4/1554`
36. Shelby, Z., Hartke, K., Bormann, C.: Rfc 7252, the constrained application protocol (coap). Tech. rep. (2014), `https://www.rfc-editor.org/rfc/rfc7252`
37. Stewart, R., Metz, C.: Sctp: new transport protocol for tcp/ip. IEEE Internet Computing **5**(6), 64–69 (2001)
38. Stouffer, K., Falco, J., Scarfone, K., et al.: Guide to industrial control systems (ics) security. NIST special publication **800**(82), 16–16 (2011)
39. Stouffer, K., Falco, J., Scarfone, K., et al.: Guide to industrial control systems (ics) security. NIST special publication **800**(82), 16–16 (2011)
40. Uludag, S., Lui, K.S., Ren, W., Nahrstedt, K.: Practical and secure machine-to-machine data collection protocol in smart grid. In: 2014 IEEE Conference on Communications and Network Security. pp. 85–90. IEEE (2014)
41. Uludag, S., Lui, K.S., Ren, W., Nahrstedt, K.: Secure and scalable data collection with time minimization in the smart grid. IEEE Transactions on Smart Grid **7**(1), 43–54 (2015)
42. Yukun, N., Xiaobin, T., Shi, C., Haifeng, W., Kai, Y., Zhiyong, B.: A security privacy protection scheme for data collection of smart meters based on homomorphic encryption. In: Eurocon 2013. pp. 1401–1405. IEEE (2013)