Privacy-Preserving Data Provenance for Smart Meter Communications

Rohini Poolat Parameswarath
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
rohini.p@nus.edu.sg

Biplab Sikdar
Department of ECE
College of Design and Engineering
National University of Singapore
Singapore
bsikdar@nus.edu.sg

***Abstract*—Smart meters play an important role in the smart grid infrastructure. They monitor the electrical consumption data of consumers and transmit it to a server. This data is used to make critical decisions in the smart grid. Smart meter communications face several security and privacy concerns, and due to untrusted environments, the server cannot trust the origin of the data. Thus, the server should be able to verify the provenance of the received data so that it can be certain of the origin of the data. This paper proposes a protocol for data provenance with authentication that preserves the privacy of consumers in smart meter communications. Decentralized Identifiers (DIDs) generated by users themselves and Verifiable Credentials (VCs) that can be cryptographically verified are the building blocks of the proposed protocol. The use of user-controlled identities contributes significantly to privacy preservation. We present security, privacy, and performance analyses to demonstrate the robustness of the proposed protocol. We also provide a formal security verification of the protocol using the Automated Validation of Internet Security Protocols and Applications (AVISPA) tool.**

***Index Terms*—Data provenance, privacy, security, smart meters.**

I. Introduction

Smart grid networks achieve reliable and efficient management and distribution of electricity by employing advanced technologies [1]. The traditional power grid is designed in such a way that the electricity flows from the energy supplier to the consumers. On the other hand, the smart grid enables bidirectional power flow between the power supplier and the consumers [1]. The smart grid monitors the energy consumption of customers to improve energy efficiency and maintain the demand-supply balance [1].

Smart meters are essential building blocks of the smart grid infrastructure [2]. Smart meters are deployed to collect the electrical consumption data of consumers and transmit it to a server [2], [3]. The server uses the received consumption data from smart meters in various applications such as electricity billing, demand-side management, etc [3].

Since smart meters and the server communicate through the Internet, an insecure channel, smart meter communications face several security and privacy challenges. Lack of mutual authentication before exchanging data and various cyber-attacks are some of the security issues smart meter communications [1]. As a result, cyber security measures are required in smart meter communications [4]. Further, data provenance in smart meter communications is an important, but less investigated security aspect [5]. Data provenance enables establishing trust about the source and location of the data's origin. The energy usage data sent to a server must have originated from a specific smart meter as claimed and it should give the actual consumption data without any modifications. Also, disclosure of customers' sensitive information to unauthorized parties is another challenge that needs to be considered in smart meter communications. Since the collected data is used in various applications, it can result in the privacy violation of the users [6]. The smart meter data can reveal the usage pattern of electricity, energy consumption data, or even the presence of a customer in a building at a given time. As a result, the privacy of consumers should be preserved in smart meter communications. Also, ensuring the integrity of messages is equally important since the messages received by the receiving entity should be the same as the ones sent by the sender [7]. Hence, this paper proposes a privacy-preserving data provenance protocol that protects smart meter communications from several attacks. We use Decentralized Identifier (DID) [8] and Verifiable Credential (VC) [9] to build the protocol. DID enables the creation of decentralized digital identities. A VC is a digital credential [9]. Cryptographic methods can be applied to verify VCs making them trustworthy [9].

A. Related Work

In this section, we present some of the research works that have been published in the literature about data provenance in various fields. The authors of [10] investigated the significance of data provenance in security and privacy. They pointed out that there has been limited research in ensuring provenance while protecting privacy. Their study emphasized that it is essential to ensure provenance information to trust the data. Aman et al. [11] proposed protocols for data provenance in Internet of Things (IoT) systems using Physical Unclonable Functions (PUFs) and characteristics of wireless links. This solution is lightweight and preserves the privacy of users. However, it requires additional hardware, PUF. Similarly, a protocol based on PUFs to secure the communication between the smart meters and neighborhood gateways was proposed in [12]. This protocol also requires additional hardware and does not preserve the privacy of

customers. To secure microgrid operations, several best practices that must be integrated into the system architecture have been provided in [13]. Kumar et al. proposed a security scheme for smart metering infrastructure employing hybrid cryptography (Elliptic Curve Cryptography and symmetric cryptography) in [3]. This scheme ensures authentication and key agreement between a smart meter and a Neighbourhood Area Network (NAN) gateway while maintaining privacy of the electricity consumers. However, the smart meters do not generate their own identities in [3]. Instead, the NAN assigns identities to smart meters. Since there is a dependency on a central server for identities, there is a possibility of a single point of failure in these systems. Data provenance also was not addressed in [3]. Chai et al. proposed an authentication scheme for smart meters based on the Shangyong Mima 2 authentication key exchange protocol [14]. The proposed scheme is specifically for resource-constrained devices. Garg et al. proposed a mutual authentication scheme in [15]. However, according to [14], the scheme in [15] does not provide sufficient anonymity.

In a nutshell, the limitations of most of the above-listed existing works can be summarized as requiring separate hardware such as PUFs, customers depending on a central authority for their identities, not establishing data provenance, and not preserving the anonymity and privacy of customers. Since the data generated by smart meters are used in many critical applications, the trustworthiness of the data and hence establishing data provenance are essential requirements in smart meter communications. The source of data and the location of the data's origin should be verified. To address the above issues, this paper leverages the concepts of DID and VC to establish data provenance in smart meter communications. The proposed protocol does not require any additional hardware such as PUF.

B. Contributions

The key contributions of this paper are as follows:

- **Data provenance:** We leverage DID and VC to confirm data provenance in smart meter communications. The smart meter and the server verify the legitimacy of each other and establish a session key for secure communication. After that, the smart meter sends the data securely using the session key. The proposed method offers protection from several attacks as well.
- **Privacy:** By using DIDs, users have complete control over their identities in the proposed protocol. Users can create their identities without depending on a central authority and share their identities at their own discretion. This results in a high level of privacy for the customers.
- **Security and privacy analyses:** We provide a formal privacy analysis and informal security analysis to demonstrate that the proposed protocol achieves the desired security properties and privacy of the customers. We also provide formal verification of the proposed protocol using the Automated Validation of Internet Security Protocols and Applications (AVISPA) [16] tool.

II. Preliminaries

In this section, we briefly present the basics of decentralized identifiers and verifiable credentials.

A. Decentralized Identifiers

DIDs enable decentralized digital identities [8]. In the DID framework, individuals create and manage their own identities without depending on any centralized party. Different DIDs can be created for different applications to achieve unlinkability. A DID is associated with a public key and a private key. A DID resolves to a DID document stored on a distributed ledger. The public key associated with the DID is stored in the DID document. The DID owner is responsible for storing his/her private key in a secure manner.

B. Verifiable Credentials

VCs are digital credentials that can be verified cryptographically and hence cannot be tampered with. The VC ecosystem consists of three participants: an *Issuer* who issues the VC, a *Holder* who holds the VC, and a *Verifier* who verifies the holder's VC before providing services to confirm that the holder owns the required credentials.

III. System and Adversary Models

A. System Model

The system model is illustrated in Figure 1. For the efficient management of resources, the proposed scheme considers a decentralized architecture where multiple servers are deployed. Each server is in charge of the smart meters in a particular area. This architecture helps to reduce the latency. Smart meters are directly connected to the server through a wireless network. We consider a residential area where a smart meter is installed in each house. The smart meters collect and record data such as voltage levels and electric energy consumption of the households, and transmit the collected data to the server in charge of that particular residential area. We denote the smart meters as $\mathcal{SM}_x$ for $x \in \{1, 2, \ldots\}$ and servers as $\mathcal{S}_y$ for $y \in \{1, 2, \ldots\}$. The smart meters and servers are registered with a trusted authority $\mathcal{TA}$. The smart meters, the servers, and the $\mathcal{TA}$ exchange messages through the Internet. The smart meters, the servers, and the $\mathcal{TA}$ create their DIDs. The DID documents corresponding to DIDs are stored on the blockchain. The DID owner's public key is stored in the DID document while the private keys are stored with the owner.

B. Adversary Model

Smart meters send the collected data to servers. Since the smart meters communicate with the server through an insecure medium (the Internet), an adversary may carry out various attacks on communication channels. We assume the Dolev-Yao model (DY model) [17] where an adversary has complete control over the communication between smart meters and servers and may listen to, modify, or delete the transmitted messages. An adversary may impersonate a registered smart meter to send malicious data to the server. The adversary may also capture and replay the exchanged messages. An adversary may drop the messages between

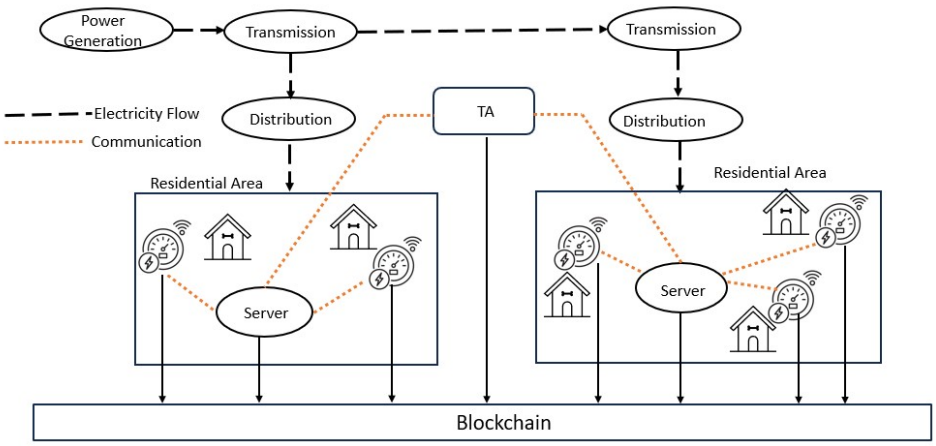

Fig. 1. System model.

the smart meter and the server to carry out a Denial-of-Service (DoS) attacks. Since the server makes decisions based on the received data, these malicious attempts by an adversary will have an impact on the server's decisions. Hence, it is important to verify data provenance by ensuring the source of data and location of smart meters and to provide protection from other cyber attacks.

IV. Proposed Data Provenance Protocol

In this section, we present the protocol. The proposed protocol consists of the following phases: system initialization, server registration, smart meter registration, and data transfer. The registration phase is executed only once for each participant and is assumed to take place through a secure channel. The data transfer phase is executed whenever the smart meter wants to send information to the server. In the data transfer phase, the smart meter and the server authenticate each other first. After successful authentication and establishing a session key, the smart meter sends data to the server.

A. System Initialization Phase

Step 1: Every entity creates its DID. As mentioned previously, a DID is associated with a public key and a private key. Let ID_{TA}, ID_S, and ID_{SMx} represent the DIDs of the $\mathcal{TA}$, server $\mathcal{S}$, and smart meter $\mathcal{SM}_x$, respectively.

Step 2: Let the pair of public and private keys, p_{TA} and s_{TA}, respectively, correspond to ID_{TA}. Similarly, p_S and s_S correspond to ID_S and p_{SMx} and s_{SMx} are associated with ID_{SMx}.

B. Server Registration Phase

The registration of server $\mathcal{S}$ takes place through the following steps:

Step 1: $\mathcal{S}$ sends a registration request to the $\mathcal{TA}$ with its DID, ID_S.

Step 2: The $\mathcal{TA}$ registers $\mathcal{S}$ and stores ID_S on the blockchain.

C. Smart Meter Registration Phase

The registration of smart meter $\mathcal{SM}_x$ takes place through the following steps:

Step 1: $\mathcal{SM}_x$ sends a registration request to the $\mathcal{TA}$ with ID_{SMx}. The $\mathcal{TA}$ stores ID_{SMx} on the blockchain.

Step 2: The $\mathcal{TA}$ generates a credential $cred'$ for $\mathcal{SM}_x$. Let L_x denote the location where $\mathcal{SM}_x$ is installed. The $\mathcal{TA}$ appends $cred'$ with L_x to generate $cred$ and signs it with s_{TA} to generate the verifiable credential V_x for $\mathcal{SM}_x$. The $\mathcal{TA}$ assigns $\mathcal{SM}_x$ to $\mathcal{S}$. After that, the $\mathcal{TA}$ generates a secret symmetric key k_i and a set of emergency keys $k = \{k_1, k_2, \ldots, k_n\}$ for $\mathcal{SM}_x$. Finally, the $\mathcal{TA}$ sends V_x to $\mathcal{SM}_x$ and ID_{SMx} to $\mathcal{S}$. The $\mathcal{TA}$ also shares k_i and k with $\mathcal{S}$ and $\mathcal{SM}_x$.

Step 3: $\mathcal{SM}_x$ stores V_x, k_i, and k in its memory. $\mathcal{SM}_x$ generates a pseudo-identity from ID_{SMx} and k_i as $PID_i = ID_{SMx} \oplus k_i$ and stores it to use during the authentication phase. $\mathcal{S}$ also stores k_i, k, and ID_{SMx} in its memory. Now, $\mathcal{SM}_x$ is registered and ready to transfer data to the server.

D. Data Transfer Phase

The ith round of the data transfer phase between $\mathcal{SM}_x$ and $\mathcal{S}$ consists of the following steps:

Step 1: $\mathcal{SM}_x$ generates a random number R_i of length n. Then, $\mathcal{SM}_x$ applies XOR operation on R_i with k_i to compute $R_i^* = R_i \oplus k_i$. After that, $\mathcal{SM}_x$ composes a message D_1 with a data transfer request, its pseudo-identity, PID_i, and R_i^* as $D_1 = \{Req, PID_i, R_i^*\}$. $\mathcal{SM}_x$ sends D_1 to $\mathcal{S}$.

Step 2: Upon receiving D_1, $\mathcal{S}$ extracts PID_i from D_1 and computes $ID_{SMx} = PID_i \oplus k_i$. $\mathcal{S}$ first checks if ID_{SMx} exists on the blockchain. If it does not exist, the protocol will be terminated. Otherwise, $\mathcal{S}$ decodes $R_i = R_i^* \oplus k_i$. Then, $\mathcal{S}$ generates a random number N_i. Subsequently, $\mathcal{S}$ calculates $N_i^* = N_i \oplus k_i$ and an authentication parameter $\alpha_i = h(N_i \parallel k_i \parallel ID_S)$. After that, $\mathcal{S}$ composes a message $D_2 = \{\alpha_i, N_i^*, ID_S\}$ and sends it to $\mathcal{SM}_x$.

Step 3: $\mathcal{SM}_x$ decodes $N_i = N_i^* \oplus k_i$. Then, $\mathcal{SM}_x$ computes $\alpha_i' = h(N_i \parallel k_i \parallel ID_S)$ and verifies whether

$\alpha'_i = \alpha_i$. Successful verification of the authentication parameter indicates that the message has not been tampered with. If the verification is successful, $\mathcal{SM}_x$ obtains the public key p_S of $\mathcal{S}$ from the blockchain. $\mathcal{SM}_x$ encrypts the verifiable credential V_x with p_S to compute V_x^* and calculates an authentication parameter $\beta_i = h(N_i \parallel V_x^*)$. Finally, $\mathcal{SM}_x$ composes $D_3 = \{\beta_i, V_x^*\}$ and sends it to $\mathcal{S}$.

Step 4: $\mathcal{S}$ computes the authentication parameter $\beta'_i = h(N_i \parallel V_x^*)$ and verifies if $\beta'_i = \beta_i$. $\mathcal{S}$ decrypts V_x^* using its private key s_S to get V_x. Then, $\mathcal{S}$ obtains the public key p_{TA} of the signing trusted authority $\mathcal{TA}$ from the blockchain and verifies the signature on V_x. Successful verification of the signature indicates that $\mathcal{SM}_x$ is a legitimate smart meter registered with the $\mathcal{TA}$ and is located at L_x. After that, $\mathcal{S}$ generates a new key k_{i+1} for the next round of authentication. Then, the $\mathcal{S}$ computes a session key $SK = h(R_i \parallel N_i \parallel k_i \parallel k_{i+1})$, $SK^* = SK \oplus k_i$, $k_{i+1}^* = k_{i+1} \oplus k_i$, and $PID_{i+1} = ID_{SMx} \oplus k_{i+1}$. Fianlly, $\mathcal{S}$ composes a message with the acknowledgement Ack, SK^*, and k_{i+1}^* as $D_4 = \{Ack, SK^*, k_{i+1}^*\}$ and sends it to $\mathcal{SM}_x$.

Step 5: From D_4, $\mathcal{SM}_x$ extracts $SK = SK^* \oplus k_i$. Then, $\mathcal{SM}_x$ computes $k_{i+1} = k_{i+1}^* \oplus k_i$ and $SK' = h(R_i \parallel N_i \parallel k_i \parallel k_{i+1})$. After that, $\mathcal{SM}_x$ verifies whether the computed and received session keys are equal, i.e., $SK' = SK$. If the session key verification is successful, both $\mathcal{SM}_x$ and $\mathcal{S}$ have established a session key. $\mathcal{SM}_x$ generates the pseudo-identity for the next authentication iteration as $PID_{i+1} = ID_{SMx} \oplus k_{i+1}$. To send the data D to $\mathcal{S}$, $\mathcal{SM}_x$ encrypts D with SK to get D_{SK}, computes an authentication parameter $\gamma_i = h(D_{SK} \parallel SK \parallel ID_S)$, and composes $D_5 = \{\gamma_i, D_{SK}\}$. Then, D_5 is sent to $\mathcal{S}$.

Step 6: $\mathcal{S}$ computes the authentication parameter $\gamma'_i = h(D_{SK} \parallel SK \parallel ID_S)$ and verifies if $\gamma'_i = \gamma_i$. Then, $\mathcal{S}$ decrypts D_{SK} with SK to extract D. Finally, $\mathcal{S}$ computes an authentication parameter $\delta_i = h(D_{SK} \parallel SK)$, and composes $D_6 = \{\delta_i\}$. After that, D_6 is sent to $\mathcal{SM}_x$. Upon receiving D_6, $\mathcal{SM}_x$ verifies δ_i. If the verification is successful, it can be concluded that the data sent by $\mathcal{SM}_x$ has been received successfully by $\mathcal{S}$. Then, $\mathcal{SM}_x$ terminates the session.

Thus, a session key is established between $\mathcal{S}$ and $\mathcal{SM}_x$ after verifying the verifiable credential during the data transfer phase. This session key enables secure data transfer and establishes data provenance by verifying the source of the data and the location of the smart meter. The data transfer phase is illustrated in Figure 2.

Smart Meter	**Server**
Generate: R_i	
$R_i^* = R_i \oplus k_i$	
$D_1 = \{Req, PID_i, R_i^*\}$	
$\xrightarrow{D_1}$	$ID_{SMx} = PID_i \oplus k_i$
	Verify: ID_{SMx}
	$R_i = R_i^* \oplus k_i$
	Generate: N_i
	$N_i^* = N_i \oplus k_i$
	Compute:
	$\alpha_i = h(N_i \parallel k_i \parallel ID_S)$
	$D_2 = \{\alpha_i, N_i^*, ID_S\}$
$N_i = N_i^* \oplus k_i$	$\xleftarrow{D_2}$
Compute:	
$\alpha'_i = h(N_i \parallel k_i \parallel ID_S)$	
Verify: $\alpha'_i ? = \alpha_i$	
Encrypt V_x with p_S to get V_x^*	
Compute: $\beta_i = h(N_i \parallel V_x^*)$	
$D_3 = \{\beta_i, V_x^*\}$	
$\xrightarrow{D_3}$	Compute: $\beta'_i = h(N_i \parallel V_x^*)$
	Verify: $\beta'_i ? = \beta_i$
	Decrypt V_x^* and Verify V_x
	Generate: k_{i+1}
	$SK = h(R_i \parallel N_i \parallel k_i \parallel k_{i+1})$
	Compute: $SK^* = SK \oplus k_i$
	Compute: $k_{i+1}^* = k_{i+1} \oplus k_i$
	$PID_{i+1} = ID_{SMx} \oplus k_{i+1}$
	$D_4 = \{Ack, SK^*, k_{i+1}^*\}$
Compute: $SK = SK^* \oplus k_i$	$\xleftarrow{D_4}$
Compute: $k_{i+1} = k_{i+1}^* \oplus k_i$	
$SK' = h(R_i \parallel N_i \parallel k_i \parallel k_{i+1})$	
Verify: $SK' ? = SK$	
$PID_{i+1} = ID_{SMx} \oplus k_{i+1}$	
Encrypt: D with SK to get D_{SK}	
Compute:	
$\gamma_i = h(D_{SK} \parallel SK \parallel ID_S)$	
$D_5 = \{\gamma_i, D_{SK}\}$	
$\xrightarrow{D_5}$	Compute:
	$\gamma'_i = h(D_{SK} \parallel SK \parallel ID_S)$
	Verify: $\gamma'_i ? = \gamma_i$
	Decrypt: D_{SK}
	$\delta_i = h(D_{SK} \parallel SK)$
	$D_6 = \{\delta_i\}$
Verify: δ_i	$\xleftarrow{D_6}$

Fig. 2. Data transfer phase.

V. Security Analysis

We provide a formal privacy analysis and informal security analysis of the proposed protocol in this section.

A. Formal Privacy Analysis

Now, we formally analyze the security and privacy provided by the proposed protocol using the security model given in [18]. To define the attacks formally, we consider an adversary $\mathcal{A}$ who can eavesdrop and modify the messages exchanged between smart meters and the server. $\mathcal{A}$ can run the following queries.

- *Query1(i)*: With this query, $\mathcal{A}$ can eavesdrop and capture all the messages exchanged between a smart meter and the server in the ith session. *Query1(i)* models a passive attack.
- *Query2(m, i)*: With this query, $\mathcal{A}$ impersonates a smart meter and sends a message m to the server in the ith session. *Query2* models an active attack.
- *Query3(m)*: $\mathcal{A}$ uses this query to retrieve the secrets stored in the smart meter's memory.
- $Query4(\mathcal{SM}_0, \mathcal{SM}_1, i)$: This query is used to verify whether the protocol offers indistinguishable privacy to the smart meters. First, $\mathcal{A}$ selects two smart meters $\mathcal{SM}_0$ and $\mathcal{SM}_1$. After that, $\mathcal{A}$ sends $Query4(\mathcal{SM}_0, \mathcal{SM}_1, i)$ to a challenger. Then, the challenger randomly chooses a bit b as 0 or 1 and gives $\mathcal{SM}_b$ from the set $\{\mathcal{SM}_0, \mathcal{SM}_1\}$ to $\mathcal{A}$. $\mathcal{A}$'s aim is to

guess b correctly.

Proof: A game played between an adversary $\mathcal{A}$ and smart meters as well as the server is used to demonstrate that the proposed scheme offers indistinguishable privacy. The game G has three phases:

- *Learning phase*: $\mathcal{A}$ chooses two smart meters $\mathcal{SM}_1$ and $\mathcal{SM}_2$ and executes $Query1(i)$ to eavesdrop on the messages on their ith round of authentication. $\mathcal{A}$ learns the exchanged parameters for both $\mathcal{SM}_1$ and $\mathcal{SM}_2$.
- *Challenge phase*: $\mathcal{A}$ executes $Query4(\mathcal{SM}_0, \mathcal{SM}_1, i)$. Then, the challenger randomly chooses $\mathcal{SM}_b$ where the random bit $b \in \{0, 1\}$ and gives it to $\mathcal{A}$. After that, $\mathcal{A}$ executes $Query1(i+1)$ to eavesdrop on the messages of $\mathcal{SM}_b$ on its $(i+1)$th round of authentication and learns the exchanged parameters for $\mathcal{SM}_b$.
- *Guess phase*: In this phase, $\mathcal{A}$ needs to determine b. $\mathcal{A}$ has learnt the parameters for both $\mathcal{SM}_1$ and $\mathcal{SM}_2$ in session i and the parameters for $\mathcal{SM}_b$ in session $i+1$. Suppose $b = 0$, i.e., the challenger chose $\mathcal{SM}_0$ as $\mathcal{SM}_b$. $\mathcal{SM}_0$ generates its own DID without depending on a third party. Further, $\mathcal{SM}_0$ generates its pseudo-identities from its DID as $PID_i = ID_{SMx} \oplus k_i$ and $PID_{i+1} = ID_{SMx} \oplus k_{i+1}$ in rounds i and $i+1$, respectively. Since $k_i \neq k_{i+1}$, $PID_i \neq PID_{i+1}$. Similarly, $R_i \neq R_{i+1}$ and $N_i \neq N_{i+1}$ as they are completely random. Hence, $\mathcal{A}$ has to make a random guess on the bit b even after learning the parameters in session i. $\mathcal{A}$ guesses a bit $d \in \{0, 1\}$. $\mathcal{A}$ wins the game if $b = d$. Here, the advantage of $\mathcal{A}$ in breaking the indistinguishable privacy is the advantage over random guessing of the bit. The advantage of the adversary in this game is $Adv_A = Pr((b = d) - \frac{1}{2}) = 0$. Hence, the proposed scheme offers indistinguishable privacy.

B. Informal Security Analysis

Data Provenance: During the data transfer phase, the sender should know V_x, encrypt it with p_S to get V_x^*, and compose the message D_3 with V_x^* to prove that he/she is the legitimate sender. An attacker does not know V_x. Hence, the attacker cannot generate a valid D_3 and proceed with session key generation for successful data transfer. Thus, the proposed protocol ensures the authenticity of the data's origin and hence establishes data provenance.

Integrity of Messages: The authentication parameter $\alpha_i = h(N_i \parallel k_i \parallel ID_S)$ sent in D_2 is computed and verified by the smart meter and the authentication parameter $\beta_i = h(N_i \parallel V_x^*)$ sent in D_3 is computed and verified by the server before agreeing on a session key. Similarly, $\gamma_i = h(D_{SK} \parallel SK \parallel ID_S)$ sent by the smart meter in D_5 is computed and verified by the server while receiving the data. Finally, $\delta_i = h(D_{SK} \parallel SK)$ is verified by the smart meter. Hence, if an adversary attempts to modify the exchanged messages, these verifications will fail. Thus, the proposed protocol can detect an adversary's attempts to tamper with the messages.

Protection Against Eavesdropping Attacks: The parameters exchanged in the messages R_i, N_i, and SK are XORed with k_i. Similarly, V_x and the data are encrypted with p_S and SK, respectively. Hence, an adversary will not be able to decode and understand the messages. Thus, the proposed protocol is secure against eavesdropping attacks.

Mutual Authentication: Only a legitimate smart meter and server know the symmetric key k_i to generate valid messages to get authenticated. Further, only legitimate smart meters hold verifiable credentials signed by the trusted authority. The server authenticates the smart meter by verifying the signature on V_x using the public key of the trusted authority. Since only the trusted authority knows its private key to do the signing, the adversary cannot generate a valid signature. Thus, the proposed protocol enables mutual authentication between a legitimate smart meter and a server.

Protection Against Impersonation Attacks: To impersonate a smart meter, the adversary must compose valid messages using the key k_i and the VC, V_x. The key k_i is shared only between the smart meter and the server, and only the smart meter knows V_x. Thus, the adversary cannot generate valid messages and the protocol provides protection against impersonation attacks.

Privacy: Users do not depend on third parties to generate their identities. They create their DIDs. Since the real identity of the user is not used during message exchange, even if an adversary listens to the exchanged messages, he/she cannot link them to a specific customer. Hence, the energy consumption data and usage patterns are not available to an attacker, thereby preserving the privacy of customers.

Session Key Security: The proposed protocol enables the generation of a session key between a smart meter and server as $SK = h(R_i \parallel N_i \parallel k_i \parallel k_{i+1})$. Only a legitimate smart meter and server know the parameters required to generate the session key. Thus, the proposed protocol provides session key security.

Desynchronization and DoS Attacks Resistance: An adversary may execute a DoS attack by desynchronizing the secrets between the smart meter and the server. The adversary may do this by dropping the message D_4 sent by the server to desynchronize k_{i+1} between the smart meter and the server. In such a scenario, the value of k_{i+1} will not be received at the smart meter and the subsequent authentication events will fail as the smart meter expects k_{i+1}. To overcome such a scenario, the smart meter stores a list of emergency values k during the registration phase. If an authentication request is rejected due to desynchronization, the smart meter can use one of the values from the emergency list. Hence, the synchronization between the smart meter and the server will not be affected. Thus, the protocol provides desynchronization and DoS attacks resistance.

VI. Formal Security Verification Using AVISPA

Next, we provide formal security verification of the proposed protocol. We use the AVISPA tool [16] which implements the DY model [17]. The backends of AVISPA can check replay and man-in-the-middle (MITM) attacks. To do the security verification, first, we set up Security Protocol ANimator (SPAN) and AVISPA on Ubuntu running in VirtualBox. Then, we implemented the proposed protocol in

TABLE I
COMPARISON BASED ON SECURITY FEATURES

Scheme	SF1	SF2	SF3	SF4	SF5	SF6	SF7	SF8	SF9	SF10
Kumar et al. [3]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Kaveh et al. [12]	No	Yes	Yes	Yes	No	Yes	Yes	No	No	No
Chai et al. [14]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Garg et al. [15]	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	Yes
Proposed Scheme	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
SF1: Privacy; **SF2:** Mutual Authentication; **SF3:** Session key; **SF4:** Protection against replay attacks;										
SF5: Protection against impersonation attacks; **SF6:** Protection against eavesdropping attacks; **SF7:** Integrity;										
SF8: Data Provenance; **SF9:** Does not need clock synchronization ; **SF10:** Protection against DoS attacks										

the High-Level Protocol Specification Language (HLPSL), which is a role-oriented, formal language [16]. We defined two roles (the smart meter and the server) in the HLPSL implementation for registration and data transfer phases. We also defined the necessary roles for the session, goal, and environment to specify scenarios involving the interaction of the smart meter and the server. An attacker knows all the public parameters and participates in the execution of the protocol. We used the OFMC and CL-AtSe backends to evaluate the security of the proposed protocol. The results demonstrate that the proposed protocol is secure against replay and MITM attacks.

VII. PERFORMANCE ANALYSIS

In this section, we first compare the proposed protocol with other existing protocols in terms of the security properties achieved. Then, we analyze the computation cost of the proposed protocol and compare it with that of other protocols.

A. Comparison of Security Properties

The main features that set the proposed protocol apart from others are data provenance, privacy, and not requiring clock synchronization. A comparison of the security features of the proposed protocol with the features of the schemes in [3], [12], [14], and [15] is provided in Table I. The proposed protocol ensures data provenance, strong privacy protection, mutual authentication, session key agreement for secure data transfer, and integrity of messages. It also offers protection against replay, impersonation, DoS, and eavesdropping attacks. The protocols in [3], [12], [14], and [15] do not address data provenance. These protocols expect the clocks of the sender and receiver to be synchronized. Hence, we can see that the proposed protocol offers more security features compared to other similar protocols.

B. Computation Cost

The registration phase is executed only once for each participant. Hence, the computation cost of the proposed protocol depends on the execution time taken during the data transfer phase. We run the experiments on a Raspberry Pi 3B using Python. We do not consider the time taken by XOR and concatenation operations since the execution time of these operations is negligible. We use Rivest–Shamir–Adleman (RSA) asymmetric cryptographic technique for signature generation/verification and encryption/decryption with a key size of 512 bits. In our analysis, T_h, T_{Verify}, and $T_{E/D}$ represent the time taken by the hash, RSA signature verification, RSA encryption/decryption operations, respectively. From the experiments, $T_h = 1.16$ ms, $T_{Verify} = 15.4$ ms, and $T_{E/D} = 0.1$ ms. The total time taken by the proposed protocol is $10T_h + T_{Verify} + 4T_{E/D} = 27.4$ ms.

Next, we compare the proposed protocol with the protocols in [3], [12], [14], and [15]. Let T_M, T_{MAC}, and T_{PUF} represent the time taken by the point multiplication operation, Message Authentication Code (MAC), and a PUF operation, respectively. From the experiments, $T_M = 5.1$ ms and $T_{MAC} = 2.1$ ms. T_{PUF} = 19.4 μs is the time taken for a PUF operation by a 256-bit PUF [12]. The total time taken by the protocol in [3] can be approximated as $6T_M + 9T_h + 4T_{MAC} + 4T_{E/D} = 49.84$ ms. The total time taken by the protocol in [12] is $2T_{PUF} + 6T_h = 6.998$ ms and by the protocol in [14] is $4T_M + 10T_h = 32$ ms. The protocol in [15] takes $4T_M + 8T_h = 29.68$ ms.

TABLE II
COMPUTATION COST DURING AUTHENTICATION

Scheme	Cost
Kumar et al. [3]	$6T_M + 9T_h + 4T_{MAC} + 4T_{E/D} = 49.84$ ms
Kaveh et al. [12]	$2T_{PUF} + 6T_h = 6.998$ ms
Chai et al. [14]	$4T_M + 10T_h = 32$ ms
Garg et al. [15]	$4T_M + 8T_h = 29.68$ ms
Proposed Scheme	$10T_h + T_{Verify} + 4T_{E/D} = 27.4$ ms

The computation costs of the protocols in [3], [12], [14], and [15] are summarised in Table II. We have also plotted the total computation costs for different schemes in Figure 3. From Figure 3, it can be concluded that the computation cost of the proposed protocol is less than that of most other schemes.

VIII. CONCLUSION

In this paper, we presented a protocol that ensures data provenance for smart meter communications. The protocol establishes data provenance in terms of the source authenticity of the data and the location of the smart meter. The protocol is built on the concepts of decentralized identifiers and verifiable credentials. Through security analysis and comparison with similar protocols, we demonstrated that the

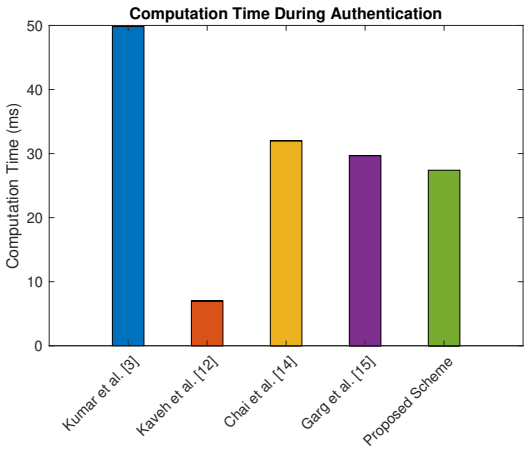

Fig. 3. Total computation cost.

proposed protocol offers better security features compared to other similar schemes. We have also provided security verification of the proposed protocol using AVISPA. Our performance analysis showed that the computation cost of the proposed protocol is reasonable. Hence, we can conclude that the proposed protocol establishes data provenance for smart meter communications and offers enhanced security features at a reasonable computation cost.

IX. Acknowledgement

This work was supported in part by Ministry of Education, Singapore under the Tier 2 grant MOE-T2EP20121-0011.

References

[1] N. Saxena and B. J. Choi, "State of the art authentication, access control, and secure integration in smart grid," *Energies*, vol. 8, no. 10, pp. 11 883–11 915, 2015.

[2] D. Abbasinezhad-Mood, A. Ostad-Sharif, M. Nikooghadam, and S. M. Mazinani, "A secure and efficient key establishment scheme for communications of smart meters and service providers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 3, pp. 1495–1502, 2020.

[3] P. Kumar, A. Gurtov, M. Sain, A. Martin, and P. H. Ha, "Lightweight authentication and key agreement for smart metering in smart energy networks," *IEEE Transactions on Smart Grid*, vol. 10, no. 4, pp. 4349–4359, 2019.

[4] Nist framework and roadmap for smart grid interoperability standards, release 1.0. [Online]. Available: https://www.nist.gov/system/files/documents/public_affairs/releases/smartgrid_interoperability_final.pdf

[5] M. H. Chia, S. L. Keoh, and Z. Tang, "Secure data provenance in home energy monitoring networks," in *Proceedings of the 3rd annual industrial control system security workshop*, 2017, pp. 7–14.

[6] S. Aggarwal, N. Kumar, S. Tanwar, and M. Alazab, "A survey on energy trading in the smart grid: Taxonomy, research challenges and solutions," *IEEE Access*, vol. 9, pp. 116 231–116 253, 2021.

[7] L. Ge, W. Yu, P. Moulema, G. Xu, D. Griffith, and N. Golmie, "Detecting data integrity attacks in smart grid," *Security and Privacy in Cyber-Physical Systems: Foundations, Principles and Applications*, pp. 281–303, 2017.

[8] "Decentralized Identifiers (DIDs)," Online, https://www.w3.org/TR/did-core/, [Accessed: Aug 2023].

[9] "Verifiable Credentials Data Model 1.0," Online, https://www.w3.org/TR/vc-data-model/, [Accessed: Aug 2023].

[10] B. Pan, N. Stakhanova, and S. Ray, "Data provenance in security and privacy," *ACM Computing Surveys*, 2023.

[11] M. N. Aman, M. H. Basheer, and B. Sikdar, "Data provenance for iot with light weight authentication and privacy preservation," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 441–10 457, 2019.

[12] M. Kaveh and M. R. Mosavi, "A lightweight mutual authentication for smart grid neighborhood area network communications based on physically unclonable function," *IEEE Systems Journal*, vol. 14, no. 3, pp. 4535–4544, 2020.

[13] F. Sangoleye, J. Johnson, A. Chavez, E. E. Tsiropoulou, N. L. Marton, C. R. Hentz, and A. Yannarelli, "Networked microgrid cybersecurity architecture design guide: A new jersey transitgrid use case," Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2022.

[14] S. Chai, H. Yin, B. Xing, Z. Li, Y. Guo, D. Zhang, X. Zhang, D. He, J. Zhang, X. Yu, W. Wang, and X. Huang, "Provably secure and lightweight authentication key agreement scheme for smart meters," *IEEE Transactions on Smart Grid*, vol. 14, no. 5, pp. 3816–3827, 2023.

[15] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, "Secure and lightweight authentication scheme for smart metering infrastructure in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3548–3557, 2020.

[16] "Automated Validation of Internet Security Protocols and Applications," Online, https://www.avispa-project.org/, [Accessed: Mar 2024].

[17] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[18] K. Ouafi and R. C. W. Phan, "Privacy of recent rfid authentication protocols," in *Information Security Practice and Experience: 4th International Conference, ISPEC 2008 Sydney, Australia, April 21-23, 2008 Proceedings 4*. Springer, 2008, pp. 263–277.