IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 1 January 2023, accepted 8 January 2023, date of publication 16 January 2023, date of current version 31 January 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3237643

RESEARCH ARTICLE

A Privacy-Preserving Scheme for Smart Grid Using Trusted Execution Environment

METE AKGÜN[1,2,3], ELIF USTUNDAG SOYKAN[4], (Member, IEEE), AND GURKAN SOYKAN[5], (Member, IEEE)

[1]Medical Data Privacy and Privacy-Preserving ML on Healthcare Data, Department of Computer Science, University of Tübingen, 72070 Tübingen, Germany
[2]Institute for Bioinformatics and Medical Informatics, University of Tübingen, 72070 Tübingen, Germany
[3]Computer Engineering Department, Izmir Institute of Technology, 35430 Izmir, Türkiye
[4]Ericsson Product Security, 164 83 Stockholm, Sweden
[5]Energy Systems Engineering Department, Bahcesehir University, 34349 Istanbul, Türkiye

Corresponding author: Elif Ustundag Soykan (elif.ustundag.soykan@ericsson.com)

This work was supported by the Open Access Publishing Fund of the University of Tübingen.

ABSTRACT The increasing transformation from the legacy power grid to the smart grid brings new opportunities and challenges to power system operations. Bidirectional communications between home-area devices and the distribution system empower smart grid functionalities. More granular energy consumption data flows through the grid and enables better smart grid applications. This may also lead to privacy violations since the data can be used to infer the consumer's residential behavior, so-called power signature. Energy utilities mostly aggregate the data, especially if the data is shared with stakeholders for the management of market operations. Although this is a privacy-friendly approach, recent works show that this does not fully protect privacy. On the other hand, some applications, like nonintrusive load monitoring, require disaggregated data. Hence, the challenging problem is to find an efficient way to facilitate smart grid operations without sacrificing privacy. In this paper, we propose a privacy-preserving scheme that leverages consumer privacy without reducing accuracy for smart grid applications like load monitoring. In the proposed scheme, we use a trusted execution environment (TEE) to protect the privacy of the data collected from smart appliances (SAs). The scheme allows customer-oriented smart grid applications as the scheme does not use regular aggregation methods but instead uses customer-oriented aggregation to provide privacy. Hence the accuracy loss stemming from disaggregation is prevented. Our scheme protects the transferred consumption data all the way from SAs to Utility so that possible false data injection attacks on the smart meter that aims to deceive the energy request from the grid are also prevented. We conduct security and game-based privacy analysis under the threat model and provide performance analysis of our implementation. Our results demonstrate that the proposed method overperforms other privacy methods in terms of communication and computation cost. The execution time of aggregation for 10,000 customers, each has 20 SAs is approximately 1 second. The decryption operations performed on the TEE have a linear complexity e.g., 172800 operations take around 1 second while 1728000 operations take around 10 seconds. These results can scale up using cloud or hyper-scalers for real-world applications as our scheme performs offline aggregation.

INDEX TERMS Smart grid, load monitoring, privacy, security, trusted execution environment.

I. INTRODUCTION

Smart grid is a paradigm shift as it enables digitalized, dynamic, and the more intelligent electric grid. With the advent of smart grid, the bi-directional flow of information is supported among appliances, devices, and the grid. Data-driven operations like advanced metering and load monitoring in smart grid have become an inevitable feature as they increase customer involvement in sustainable and clean energy by making them prosumers who produce energy and sell it back on the energy market. It is also essential for the distribution system operators (DSO, will be referred as

The associate editor coordinating the review of this manuscript and approving it for publication was Jiafeng Xie.

Utility from now on) which eager to have improved customer satisfaction and balanced load.

In smart grid, there are a lot of data producing and transferring points, including the home area network. If the data is not adequately protected on the communication line or where it is stored, unauthorized access to data might end up with privacy violations [1], [2]. Since the data can be linked with the natural person using the metering identifiers, it reveals daily behavior, physical presence, and socio-economic status of the person. The data can be used as a valuable asset for consumer profiling that will create competence between energy companies. Therefore, it is crucial to build trust and confidence in applying the proper protection mechanisms for data protection.

From the regulative perspective, general data privacy is covered by regulations e.g., in the EU by GDPR (General Data Protection Regulation) [3] which stipulates among others the responsibility for the energy data. On the other hand, a domain-specific data protection framework is required for smart grid data. To fulfill this need, the recast version of the Energy Directive was drafted by the commission aiming at the integration of the relevant GDPR provisions to ensure customer data protection. Although these legislative frameworks put the measures and principles to protect data, actual solutions must be developed providing functionalities to meet these provisions, which lead us to privacy-preserving techniques, which is also pointed out by the ‘‘data protection by design’’ in the Article 25 of the GDPR.

A. RELATED WORK

In the literature, privacy-preserving techniques for smart grid are proposed by several researchers with different perspectives [1], [4]. Many solutions have been proposed in the literature to protect customer privacy in smart grids. We can evaluate these in two categories. In the first category, the intention is to provide privacy without changing the real energy consumption data. For this purpose, anonymization, aggregation, and obfuscation techniques are applied. In [5], anonymization is provided using a trusted third party (TTP). TTP sends the data received over the secure channel to the Utility after making anonymization. However, TTP knows the measurement data and data source. This problem was solved in [6] by sending the measurement data in encrypted form with pseudo identifiers. In [7], each smart meter first splits the measurement data into shares and then sends each share to a different concentrator. The concentrator replaces the actual identifier with a pseudonym and sends the data to the Utility. Thus, the concentrators do not learn the measurement data, and the data source is hidden from the Utility. Bohli et al. [8] use TTP as an aggregator. After TTP receives the data with the pseudonym from the secure channel, it performs summations and sends them to the Utility. TTP can not learn the data source. After the data is divided into shares, aggregation can be performed on multiple TTPs [9]. Thus, data privacy is provided against TTPs, and the Utility has to make aggregation to obtain total consumption.

There are several studies in which the measurement data is included in the aggregation by homomorphic encryption without disclosing it to other users. In [10], TTP realizes the aggregation process on the encrypted measurement data and sends the encrypted aggregation data to the Utility. The Utility obtains the aggregated data by decrypting. In [11], aggregation is performed in the Utility. Each smart meter picks a random key, and the aggregator is selected randomly from the group of smart meters in each round. The aggregated key is sent to the Utility manager. Smart meters have to update their keys after each aggregation round.

In [12], multi-party computation for data aggregation is carried out using homomorphic encryption. However, the computation and communication costs of this scheme are very high. It is also vulnerable to data forgery attacks due to the malleability property of homomorphic encryption. Mustafa et al. [13] apply selective data aggregation with homomorphic encryption. Lattice-based homomorphic encryption [14] is used for aggregation to be performed between smart appliances [15], [16]. This provides data confidentiality and customer privacy. Smart meters and base stations control the authenticity of the received encrypted messages before forwarding them to the control center. Many more studies have been proposed to make smart grid systems more secure and protect users' privacy using cryptographic systems like Paillier cryptosystem [17] and Lattice-based cryptography. However, all of them suffer from high computation and communication costs. In [18], the authors use Intel SGX to prevent leakage of customers' private data by protecting aggregated data on the intermediate gateway and the central control center. They provide secure key establishment between parties using remote attestation functionality of the SGX architecture. Their results are demonstrated for three scenarios; data aggregation, dynamic pricing, and load forecasting, and they present the trade-off between privacy and accuracy.

In the second approach, privacy is provided by using energy storage units and alternative energy sources and applying changes and irregularities in measurement sampling frequency. In [19], alternative energy sources are used to conceal the energy demand. The energy demanded and spent are different from each other. The difference is covered by the alternative energy source. Energy storage units are used in conjunction with selective privacy protection algorithms [20]. These algorithms satisfy some consumption determined by the user preference from the storage unit or simulate some consumption to fill the storage unit. Danezis et al. [21] apply differential privacy techniques [22] by adding noise to the measurement data. Acs et al. [23] apply differential privacy on smart metering data level where noise is added before aggregation. In [24], Soykan et al. use differential privacy on load forecasting application.

Several privacy proofing models have been proposed to determine the privacy of smart metering systems. Bohli et al. [8] define a privacy game to separate two smart meter groups. In [25], a privacy model that measures information disclosure is proposed. In [26], the F-Test is applied to show the privacy differences between normal and noise-added consumption profiles. Buescher et al. [27] propose a privacy game to measure the privacy of a single load profile contained in an aggregate. This privacy game tries to distinguish the individual load profile from the aggregated data. In this article, all privacy measurements were made using real consumption data sets.

Readers are recommended to read surveys in [4], [28], and [29] for more information on this topic.

B. MOTIVATION AND OUR CONTRIBUTIONS

There are several privacy preserving schemes proposing homomorphic encryption techniques based on computationally expensive operations which are too expensive to be realistically employed by resource constrained smart meters or home appliances. Approaches that aim to mask the consumption data via integrating energy storage add extra hardware costs to Utility which are eventually passed on the consumer. Solutions that offer privacy on the aggregated data can not enable fine-grained data management; e.g., when a consumer wants to delete specific appliance data from the Utility databases, it is not possible to accurately extract and delete the desired data since it is aggregated. Aggregating data at the home area level also prevents accurate load management. Trust models that is adopted by most of the studies assume that the smart meter - smart appliance interface is trusted. These approaches ignore the fact that the device is physically accessible. Therefore, privacy protection starts from the smart meters in end-to-end communication.

In this study, we propose a privacy preserving solution for smart grid applications like load profiling. Main contribution of the study are as follows:

- A new customer-oriented aggregation method is proposed. The method allows the aggregation over the selected customer within the selected time frame. Consumption data is not aggregated at home area or substation level, rather the data is aggregated for a specific customer on the Utility site when it is required for load monitoring, billing or other types of added value services using the TEE e.g., providing information about user's SA level consumption. As privacy protection starts in smart appliances, protected consumption data is conveyed through the smart grid connection without any interference and stored in the encrypted database. Consumption data can not be retrieved as plaintext by any parties since decryption operation can only be performed by TEE.
- A novel key initialization mechanism is provided so that SA credentials can not be inferred during the initialization process by any party except the trusted TEE.
- The scheme leverages the accuracy of smart grid applications that use load disaggregation since the data is aggregated on the TEE, e.g., the Utility can program TEE to perform the required tasks without losing any accuracy resulting from disaggregation.
- The scheme enables the Utility to fulfill the data privacy requirements when households request data rights like deletion and give consumer's better control over their data.

The paper continues as follows; we explain the system model and requirements in Section II. We provide the solution architecture in Section III and security and privacy analysis in Section IV. Performance evaluation of the solution is given in Section V-A. Then we conclude the paper in Section VI.

II. SYSTEM MODEL AND REQUIREMENTS

The proposed system model provides services to measure and analyze energy consumption data on the smart grid with the aim of remote management of the metering, balanced power usage, and running diagnostics for a healthy grid and customer satisfaction. The key components in the system are Distribution Management System (DMS), Computational Service Provider (CSP), Short Message Service (SMS) Gateway, Smart Meter (SM), and Smart Appliances (SAs). The relation between actors are depicted in Figure 1. A brief description of their roles is given as such:

- Utility: As stated in [30], "a natural or legal person responsible for operating, ensuring the maintenance of and, if necessary, developing the distribution system in a given area and, where applicable, its interconnections with other systems and for ensuring the long-term ability of the system to meet reasonable demands for the distribution of electricity."
- DMS: A system that provides services to monitor and control a distribution grid from a centralized center.
- CSP: A system that provides computational services to the DMS. It comprises a Trusted Execution Environment which is an isolated and encrypted computation platform and a database to store encrypted consumption data.
- SM: Utility's metering end device at customer's premises that has bidirectional communication functionality.
- SA: It is a connected smart device (dishwasher, ventilation, refrigerator, etc.) that has bidirectional communication capability with smart meter.
- SMS Gateway: Provides SMS services to the Utility to communicate with the Customer.

A. PRIVACY REQUIREMENTS

In this section, we would like to highlight the privacy concerns of smart grids. Smart grid systems are deemed multi-directional networks for communications and energy transfer, allowing electricity service providers, consumers, or third-party energy management programs to access consumption data. In this context, smart grid systems with a high frequency of data collection can generate large amounts of information. Since this data contains sensitive information, utilities must ensure it is stored in secure, disaster-proof

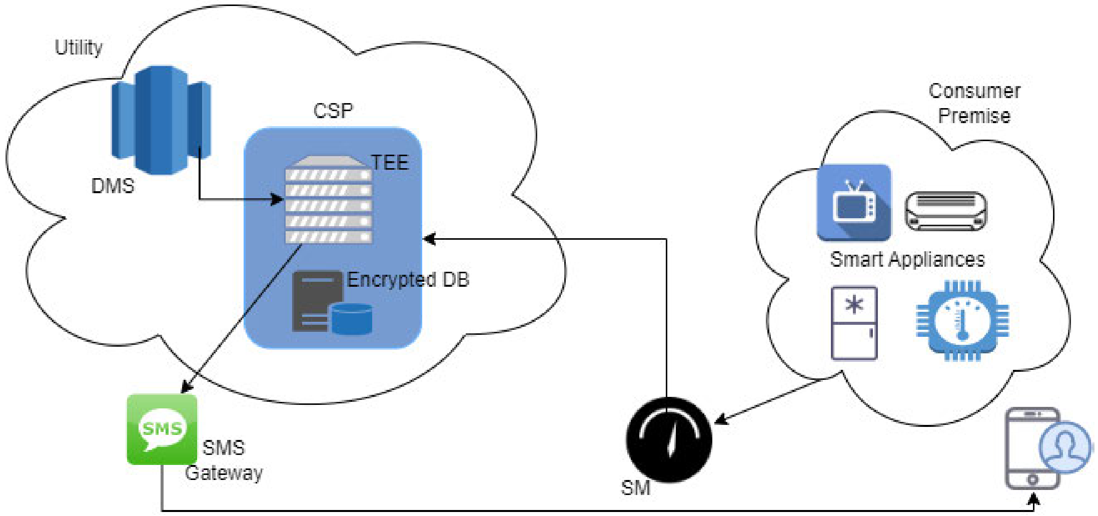

FIGURE 1. High level system view of distribution grid with its actors and their interactions.

facilities and have business continuity and contingency plans. However, even if the data is kept secure, analyzing customers' consumption data can reveal detailed information about their private lives, such as their time spent at home, work schedule, vacations, specific appliance usage, and habits, known as "consumer profiling". This information is valuable to third parties, such as insurance companies. On the other hand, misuse of the information may lead to targeted home invasions, unwanted surveillance, or tracking threats and put individuals' privacy at risk.

The power data flowing in the grid is required for several reasons but can be classified mainly into billing and operational purposes. From the operational perspective, especially for load management, the granularity of the data is important for the quality of grid operations. A high frequency of data gathering paves the way to the accurate monitoring of the grid, like detection of power failures, balancing the demand, and reacting to sudden changes in consumption. This also enables Utilities to provide services to consumers to optimize their energy consumption and offer better tariffs. Although these functions are crucial for the health of the grid, they also pose a risk to the privacy of the consumer and may lead to unwanted malicious events. Nonintrusive load monitoring (NILM), [31], [32], is a methodology to interpret aggregated consumption data so that distinguishing loads of the individual appliances is possible. NILM allows extracting when the appliances are on/off. In other words, one can say that when the residents use the kettle, have breakfast or if they are at home or on vacation. It is easy to guess that this information can be of use for thieves and other people aiming to harm. Most innocently, marketing activities can be performed. Therefore, protecting residential data from untrusted entities is a must.

Taking the above-mentioned considerations into account, we elaborate the privacy requirements into four categories as follows:

Linkability: An adversary should not be able to connect separate consumption data without knowing the identity of the customer.

Identifiability: An adversary should not be able to identify a customer's behaviour from consumption data.

Disclosure of Information: An adversary should not be able to learn the consumption data of a specific customer.

Compliance: The handling of consumption data should comply with legislation, regulation, or data protection policy.

B. SECURITY REQUIREMENTS

Although we mainly focus on privacy needs in the scope of this study, there are obvious security requirements to sustain a reliable system. We briefly give these requirements and our position to meet them as such:

End-to-end confidentiality: The data-in-transit over the system should be protected from unauthorized access with proper encryption techniques. Since each component in the smart grid communicates with different standards, like DLSM/COSEM or IEC 61334 between the smart meter and the Utility, Zigbee for SAs, there is no unique network level protection mechanism that can apply for all. Besides that, consumption data should not be disclosed in cleartext to intermediary data concentrators. Having these objectives in mind, we designed our security scheme on an end-to-end basis such that encryption is performed on the SAs and only the DMS can use the data. Our scheme does not rely on network layer protections, but provides confidentiality at the application level.

Integrity: The data-in-transit over the system should be protected from unauthorized changes. This means that the security scheme should provide a detection mechanism if any unwanted changes occur. We provide this protection implicitly with the authenticated encryption scheme so that we do not need to put an additional integrity mechanism that will bring extra computational and communication costs to the system.

False Data Injection Attacks: The lack of end-to-end confidentiality and integrity causes the system to be vulnerable to false data injection attacks. False data injection attacks can be classified as (i) False data injection attacks against grid system state estimation and (ii) False data injection attacks against energy distribution. In the first category, the adversary targets the operational health of the grid to create instabilities by injecting bad data in both alternating and direct current state estimations. In this study, we do not focus on the state estimation data collected for operational purposes; rather focus on the energy consumption data. On the other hand, preventing and detecting these malformed activities is an active research area [33]. In the second category of attacks, also called energy-request deceiving attacks [34], the adversary compromises smart meters and injects a modified demand request. In the case of several smart meters compromised and controlled by the attacker, the Utility may be misled into making a wrong decision on the requested demand of the area. In our solution, compromising smart meters does not give any advantage to the attacker as the data is already protected while it is being conveyed over the smart meters. Hence our solution prevents energy-request deceiving attacks caused by forging the demand request of a smart meter by providing end-to-end confidentiality and integrity.

III. THE PROPOSED SOLUTION

In the proposed solution, consumption data is encrypted in the household smart appliances and sent to the encrypted database through the Utility's network. The encrypted database's keys are protected by the TEE. Hence, during the reading process, no intermediary device is aware of the actual content since only the intended parties can access the metering data. The solution does not require any change on the smart meters as meters do not play a part in the encryption process. Therefore, the adoption of the solution does not bring any extra cost to Utility as it does not require meter replacement.

The cryptographic architecture of the solution uses asymmetric, symmetric algorithms and keyed hash function for key distribution. A symmetric block cipher algorithm is used for encryption in authenticated encryption mode so that any additional integrity mechanism is not needed. The whole process comprises key distribution, consumption data reading, and consumption data usage protocols. Next, preliminary information about the cryptographic building blocks that are used in the protocols is given.

A. PRELIMINARIES

Definitions of asymmetric encryption function, keyed hash function, and authenticated encryption scheme (AEAD) [35] based on a symmetric block cipher are given as follow:

Definition 1: EPUB() is an asymmetric function that encrypts a message M under the public key PK and decrypts a ciphertext C under the private key PrK.

Definition 2: H() is a one-way keyed hash function that is based on a cryptographic hash function. H() takes two inputs (Source Key (SK), Context specific data (CTX)) and produces one output (Derived key (K)), $K = H(SK, CTX)$

Definition 3: AEAD provides both confidentiality and authenticity of the plaintext. It consists of one authenticated encryption and one inverse function of encryption which is authenticated decryption function.

Inputs:

- *Encryption key, K of size k bits*
- *Data subject to encryption, called plaintext P,*
- *Initialization vector, IV,*
- *Additional authentication data, A, optional*

Outputs: Encrypted text, C, and Authentication tag, T. Then the AEAD function is defined as:

$$\tilde{E}(K, P, IV, A) = C||T(C, A, IV)$$

where $C = E(K, P, IV)$

B. THREAT MODEL

Here, we give the adversarial assumptions and considerations we made while designing the security posture of our system.

We consider a malicious adversary that controls the full software stack on CSP, except the code inside TEE. This includes all security threats from the system administrator controlling all systems, including the operating system and database server. The adversary can make any changes to data on disk and data over the network. The adversary can also make arbitrary numbers of queries to TEE.

We assume that TEE is trusted. This means the adversary can not extract any secrets inside TEE. The code inside TEE is assumed to be trusted. Note that TEEs may subject to side-channel attacks [36], [37], [38], [39] but this is out of our scope as we rely on TEE vendors' security claims. We consider that being trusted against these attacks is still an open problem. We consider that TEE has the ability to communicate securely over TLS, which is the case for Intel SGX. So all interactions with TEE will be carried out over TLS. We also assume that SMs and SAs are malicious. The adversary corrupting a SA can extract its secrets and use them to get access to previous consumption data of the SA. In this case, the adversary has also to corrupt CSP or make collaboration with the administrator of the CSP, that is a rare situation to happen. We assume that DMS is trusted which means that the DMS does not send crafted queries to CSP to abuse the information provided by TEE.

The home area network is assumed to be a trusted and personal space. So we assume the customer premise is protected from external actors. Moreover, we expect that the households do not violate or launch attacks on their own appliances to manipulate the consumption data. In case the households act malicious, they need to obtain the keys of the SA to modify the information on the SA. This issue can be resolved by using tamper-proof hardware, but the costs of SAs will increase. The prevention of this attack is not in the scope of this paper because it does not leak any privacy related information about the household.

We also do not focus on possible faults or vulnerabilities that may exist on smart appliances either by design or via their external interfaces e.g., if there are weak web or mobile applications for appliance management or monitoring. We assume the mobile application (provisioning app) used for SA key provisioning is served and maintained securely and communicates with CSP over a secure TLS channel. In addition, our scheme does not try to solve all cyber-security issues that may occur in the smart grid, e.g., a threat to SCADA system or False data injection attacks against power system state estimation. Our solution guarantees that the privacy, confidentiality, and integrity of the consumption data are protected, hence implicitly protecting against energy-request deceiving attacks.

C. KEY DISTRIBUTION

For setting up a secure system, the first step is to generate and distribute the keying material to all related devices. In the proposed architecture, Utility provides a mobile application for SA provisioning, which we call provisioning app from now on. Provisioning app pave the way for the key initialization and the re-keying containing the public key of TEE (PK). It only takes part in the SA initialization process, is not involved data reading or data usage processes.

On the Utility side, CSP, which is a component that interfaces with TEE and encrypted database, can not access any credential like the keys or one time password, *OTP*. Security functions performed by TEE and served to CSP. We assume that customers are registered to the Utility with their mobile phone number, PN. The customer has a unique identifier given by the Utility, which we call *CustomerID*, and has the ability to receive SMS messages via mobile phone. CSP is governed by Utility. Customer's mobile phone number and *CustomerID* is known by the CSP. We also assume that each SA has a unique identifier SA_{ID} that is given during the appliance manufacturing phase, e.g., a unique hardware identifier.

The whole key distribution comprises three phases, indicated as P1, P2, and P3. In the first phase, P1, the customer initiates the process over provisioning app when an SA joins the system. Note that the key distribution flow described below should be performed for each SA in the home.

P1.1 Customer enters his/her mobile phone number (PN) and a verification text (VT) of choice via provisioning app interface.

P1.2 Provisioning app encrypts the customer's PN and the VT with TEE's public key PK. Then it sends the ciphertext C to the CSP over TLS.
$C = EPUB(PK, PN \,||VT)$

P1.3 CSP conveys the C to the TEE, along with customer phone number, PN.

P1.4 Since the private key PrK is only known by TEE, TEE decrypts the ciphertext and extract mobile phone number and verification text. Then TEE compares the extracted PN with the one received from the CSP. If these are not equal, TEE aborts the process.

P1.5 TEE generates a short-lived *OTP*, concatenates with VT, adds a timestamp, T, then sends to SMS Gateway over secure TLS connection to initiate SMS submission.

P1.6 SMS gateway submits the SMS to the Customer.

P1.7 After receiving the SMS, Customer checks if the VT is correct and the message is timely.

P1.8 Then customer enters the received *OTP*, VT, and the given *CustomerID* to SA on the SA's management interface.

P1.9 Steps above are performed for each SA so that each SA is initialized with an *OTP*.

At this point, SA is initialized with the short-lived *OTP* via its management interface so that it can receive keying material for future secure communications. TEE has the ability to communicate securely over TLS with the SMS Gateway so that neither CSP nor DMS can interfere with *OTP*. After SMS Gateway is invoked by the TEE, *OTP* is conveyed via SMS message over the mobile operator's network.

In the second phase, P2, TEE proceeds with the key distribution as follows:

P2.1 TEE computes a distribution key for the customer:
$K_{Dist} = H(OTP, CustomerID)$.

P2.2 TEE generates a random number using its True Random Number Generator for the SA which will be the SA's channel encryption key, CK_{SA}.

P2.3 TEE encrypts the CK_{SA} along with the *CustomerID* using the K_{Dist} and computes the encrypted channel key ECK.
$ECK_{SA} = AEAD(K_{Dist}, (CK_{SA}||CustomerID))$.

P2.4 TEE sends ECK_{SA} to CSP and CSP dispatches it to each SA for the associated *CustomerID*.

P2.5 Steps above are performed for each SA so that each SA can receive its own ECK_{SA}.

In the third phase, P3, after receiving its ECK_{SA}, each SA performs the following operations:

P3.1 SA computes the distribution key by using its *CustomerID* and the *OTP* that is entered from the management interface during initialization:
$K_{Dist} = H(OTP, CustomerID)$

P3.2 SA decrypts the ECK_{SA} using the evaluated K_{Dist}, checks if the *CustomerID* matches. If successful, extracts the CK_{SA} for further communications

P3.3 SA encrypts its SA_{ID} and *CustomerID* with the CK_{SA} sends to the TEE.
$ESA_{ID} = AEAD(CK_{SA}, (SA_{ID}||CustomerID))$.

P3.4 In this state, each SA has its own channel encryption key CK_{SA}. As a last confirmation step, TEE decrypts the ESA_{ID} extracts the SA_{ID} then binds the CK_{SA} with the SA_{ID} and *CustomerID* values and updates its registry with these values.

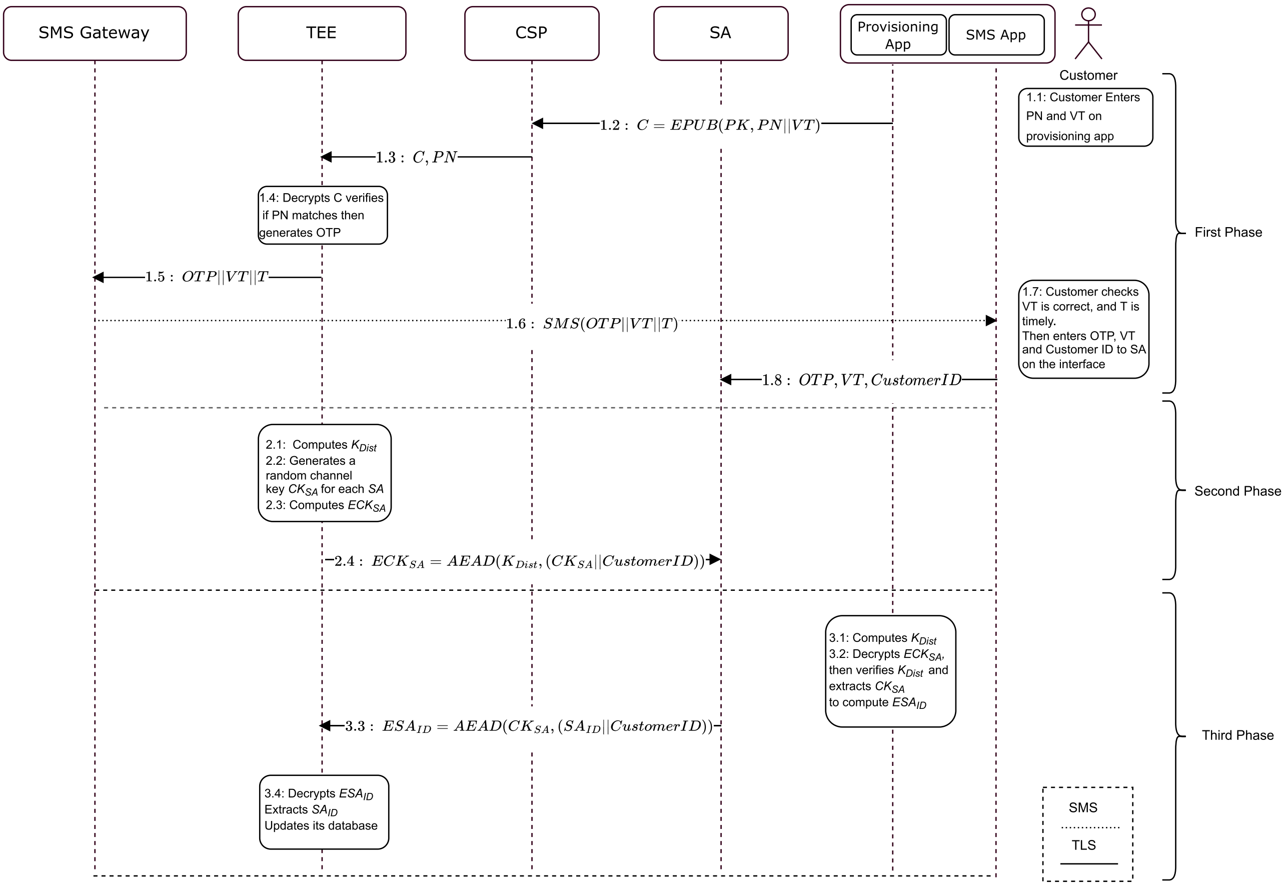
SMS Gateway
TEE
CSP
SA
Provisioning App
SMS App
Customer
1.1: Customer Enters PN and VT on provisioning app
1.2 : $C = EPUB(PK, PN||VT)$
1.3 : C, PN
1.4: Decrypts C verifies if PN matches then generates OTP
1.5 : $OTP||VT||T$
1.6 : $SMS(OTP||VT||T)$
1.7: Customer checks VT is correct, and T is timely. Then enters OTP, VT and Customer ID to SA on the interface
1.8 : $OTP, VT, CustomerID$
First Phase
2.1: Computes K_{Dist}
2.2: Generates a random channel key CK_{SA} for each SA
2.3: Computes ECK_{SA}
2.4 : $ECK_{SA} = AEAD(K_{Dist}, (CK_{SA}||CustomerID))$
Second Phase
3.1: Computes K_{Dist}
3.2: Decrypts ECK_{SA}, then verifies K_{Dist} and extracts CK_{SA} to compute ESA_{ID}
3.3 : $ESA_{ID} = AEAD(CK_{SA}, (SA_{ID}||CustomerID))$
3.4: Decrypts ESA_{ID} Extracts SA_{ID} Updates its database
Third Phase
SMS
TLS

FIGURE 2. Key distribution flow including three phases.

D. CONSUMPTION DATA READING

The consumption data, *CD*, measurement proceeds as follows:

1) Smart meter sends a reading request to each SA
2) Each SA encrypts its consumption data using their CK_{SA} with an authenticated symmetric encryption mechanism.

$$ECD_{SA} = AEAD(CK_{SA}, CD\,||SA_{ID} \times ||timestamp||CustomerID)$$

3) The encrypted consumption data ECD_{SA}, SA_{ID}, *timestamp*, and *CustomerID* is sent to the smart meter in the payload of the read response.
4) Smart meter transfers the received payload to the CSP so that CSP stores ECD_{SA}, SA_{ID}, and *timestamp* into an encrypted database for the customer identified by *CustomerID*.

E. CONSUMPTION DATA USAGE

When DMS needs to do an operation like billing or load profile monitoring purposes, it triggers the CSP and starts the process. Then DMS sends a query to TEE along with the *CustomerID* of the Customer in question over a protected tunnel. DMS must specify a time frame for the request. After receiving the trigger:

1) CSP receives the query from TEE that was sent by DMS.
2) CSP queries the required information from the encrypted database with the *CustomerID* for the given time frame in order to retrieve the encrypted consumption data and pulls all ECD_{SA}'s and SA_{ID}'s for the *CustomerID*.
3) CSP sends ECD_{SA}'s and SA_{ID}'s to the TEE since only the TEE knows the CK_{SA}.
4) TEE decrypts ECD_{SA}'s, then compares the information with the query received from DMS to verify whether CSP provided the expected database record. If successful, TEE aggregates the results and send the aggregated data to the DMS.
5) DMS then can use aggregated data for the required processes.

Since the database is encrypted at the record level, data-at-rest is protected, so customer data will not be leaked in case of unauthorized physical access. Additionally, TEE provides data-in-use protection functionality; therefore, data

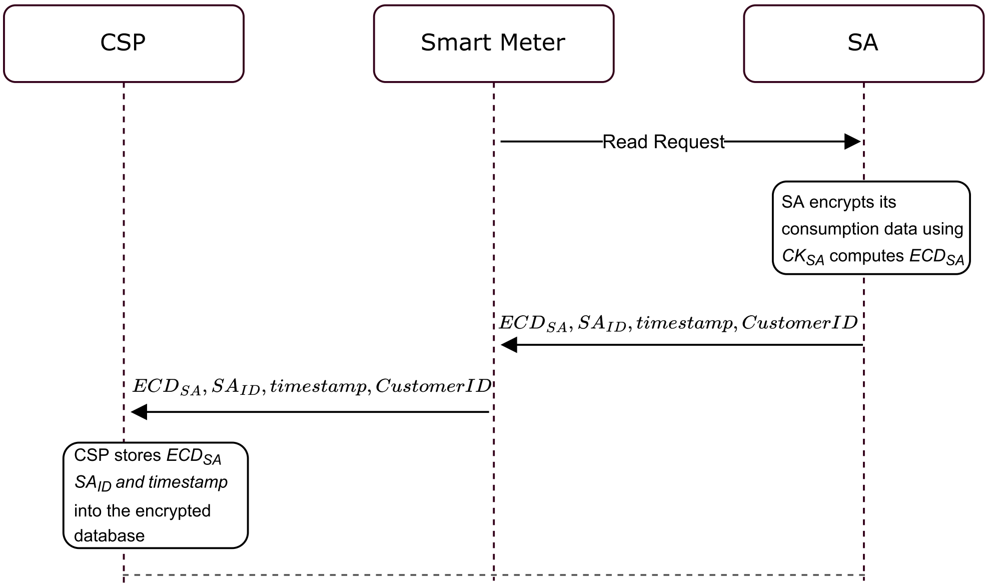
CSP
Smart Meter
SA
Read Request
SA encrypts its consumption data using CK_{SA} computes ECD_{SA}
$ECD_{SA}, SA_{ID}, timestamp, CustomerID$
$ECD_{SA}, SA_{ID}, timestamp, CustomerID$
CSP stores ECD_{SA} SA_{ID} and timestamp into the encrypted database

FIGURE 3. Consumption data reading.

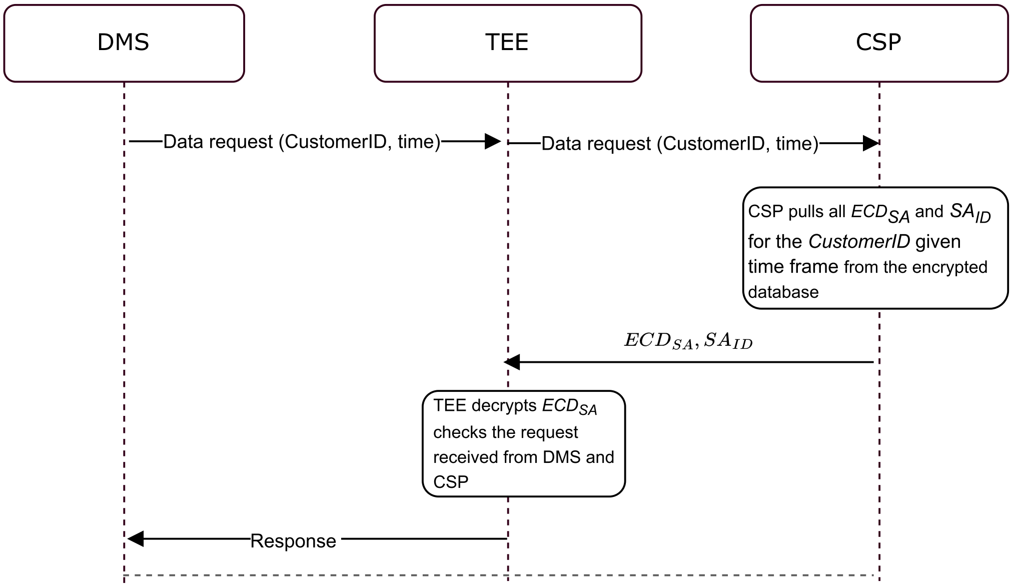
DMS
TEE
CSP
Data request (CustomerID, time)
Data request (CustomerID, time)
CSP pulls all ECD_{SA} and SA_{ID} for the CustomerID given time frame from the encrypted database
ECD_{SA}, SA_{ID}
TEE decrypts ECD_{SA} checks the request received from DMS and CSP
Response

FIGURE 4. Consumption data usage.

in the memory is protected from vulnerabilities while being processed.

IV. SECURITY AND PRIVACY ANALYSIS

In this section, we give the security and privacy analysis of our protocol. We analyze the possibility of an adversary learning a single of consumption data by monitoring the smart grid network and tampering with the TEE. We investigate the privacy of our protocol using an indistinguishability notion by defining two privacy games in which two consumption data that are known to the adversary are trying to be distinguished by the adversary.

A. SECURITY ANALYSIS

Lemma 1: Let $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ of obtaining the consumption data without tampering with the TEE is negligible.

Proof: We assume that there is an adversary $\mathcal{A}$ who gets the consumption data without tampering with the TEE. Each SA encrypts the consumption data and sends it to the CSP. To obtain consumption data, $\mathcal{A}$ must decrypt the encrypted message. Doing this operation without knowing the encryption key contradicts with the security of the encryption algorithm. Security of the encryption algorithm as follows:

Consumption data CD is encyrpted with AEAD, implemented as AES-GCM, under key CK_{SA} that is shared between SA and TEE. This encryption is IND-CCA2 (INDistinguishability under adaptive Chosen Ciphertext Attack) [40] and any p.p.t (probabilistic polynomial-time) adversary cannot break the confidentiality without knowing the CK_{SA}.

SAs generally do not supply tamper-proof design. $\mathcal{A}$ can obtain the encryption key by tampering with these devices. On the other hand, SAs are located in the house, which is a private area for consumers. For this reason, such attacks against these devices are not very feasible. The proposed system provides security for all areas outside the home, which we have considered as personal space.

Lemma 2: Let $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ of forging encrypted consumption data by breaking the ciphertext integrity is negligible.

Proof: AEAD provides ciphertext integrity (CI), meaning that the attacker cannot craft new ciphertexts with encryption function $\mathcal{E}$ that can be successfully decrypted with decryption function $\mathcal{D}$ using the key CK_{SA}, if the Adv is negligible under encryption oracle $\mathcal{E}_k(.)$ for every p.p.t adversary $\mathcal{A}$.

$$\mathrm{Adv}_{CI}(\mathcal{A}, \mathcal{E}) \overset{\mathrm{def}}{=} \mathrm{P}\left[k \overset{\mathrm{R}}{\leftarrow} K; c \leftarrow A^{\mathcal{E}_k(.)} : \mathcal{D}_k(c) \neq \perp\right] \leq \epsilon$$

Meaning that $\mathcal{A}$ cannot find a valid decryption with c resulting from oracle using random k [40]. The ciphertext integrity guarantees that the ECD_{SA} originated from SA. If it is forged, it will be detected during decryption within the TEE.

Lemma 3: Let $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ of acting man-in-the-middle and re-sending the encrypted consumption data that is obtained from previous sessions is negligible.

Proof: To prevent replay attacks, the encrypted data is supported by a timestamp so that the TEE can validate when the encryption is performed. The timestamp is integrity protected as it is included the payload of the ECD_{SA}. Hence the security of the session freshness relies on the Lemma 1 and 2, which are proven to be secure. So if any previously obtained session content is used by the adversary to lunch a replay attack on TEE to reveal sensitive information, TEE will first decrypt and verify the integrity of the content. If it is successfully verified, then TEE will check the decrypted content, detect if the timestamp is outdated, and generate an alarm.

Lemma 4: Let $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ of obtaining the consumption data by tampering with the TEE is negligible.

Proof: We assume that there is an adversary $\mathcal{A}$ who gets the consumption data by tampering with the TEE. When DMS requires consumption data for any purpose, all computations are secured by TEE. All computations are running inside TEE, and any data used in the computations do not leave TEE. In order to get any consumption data, $\mathcal{A}$ has to corrupt TEE, which contradicts the security of TEE.

Intel's Software Guard Extensions (SGX) is used as TEE [41], which consists of a set of hardware and software architectures and provides sensitive data analysis in a protected area [42].

Lemma 5: Let $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ of obtaining the consumption data by intercepting the SMS is negligible.

Proof: We assume that there is an adversary $\mathcal{A}$ who gets the consumption data by intercepting the SMS. When the adversary obtains the short-live OTP, VT, and T from the SMS content, it cannot use this information to get consumption data without physically accessing and tampering with the SA. In order to get any consumption data, $\mathcal{A}$ has to corrupt SA and extract CK_{SA}. This contradicts our assumption that SAs are located in the house, which is a private area for consumers as explained in Lemma 1.

Lemma 6: Let $\mathcal{A}$ be an adversary. The advantage of $\mathcal{A}$ of obtaining the CK_{SA} by breaking the key distribution flow is negligible.

Proof: Adversary $\mathcal{A}$ cannot obtain the SA key CK_{SA} without breaking the security of TEE, which is proven in Lemma 4. The adversary may try to analyze ECK_{SA} to extract CK_{SA}, but $\mathcal{A}$ cannot win as proven in Lemma 1 and 2 without knowing K_{Dist}. The adversary then may try to obtain K_{Dist}, but this contradicts with Lemma 5.

B. PRIVACY ANALYSIS

Many solutions based on the aggregation method have been proposed to protect customer privacy. In order to analyze the privacy aspects of these solutions, game-based definitions are very convenient and widely used in the literature. Two privacy games have been proposed to examine the aggregation-based methods in [8] and [27]. In these models, privacy is modeled based on the difficulty of distinguishing two load profiles known to the attacker in the aggregated data.

The first privacy game for smart metering [8] is depicted in Figure 5. The game defines the experiment between the adversary and the challenger and corresponds to the Linkability requirement given in Section II-A, as follows:

1) The adversary and the challenger agree on the consumption data generator C_{gen}.
2) The adversary is receiving two consumption scenario $c_0 = (e^0_{ij})$ and $c_1 = (e^1_{ij})$ from C_{gen}. These scenarios is sent to the challenger.
3) The challenger draws a random bit $r \in \{0, 1\}$ and sends $m \leftarrow E_{SA}(c_r)$ to the adversary.

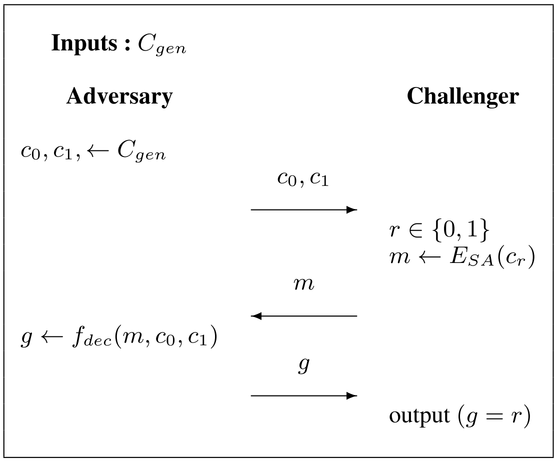
Inputs : C_{gen}
Adversary
Challenger
$c_0, c_1, \leftarrow C_{gen}$
c_0, c_1
$r \in \{0, 1\}$
$m \leftarrow E_{SA}(c_r)$
m
$g \leftarrow f_{dec}(m, c_0, c_1)$
g
output $(g = r)$

FIGURE 5. Privacy Game I.

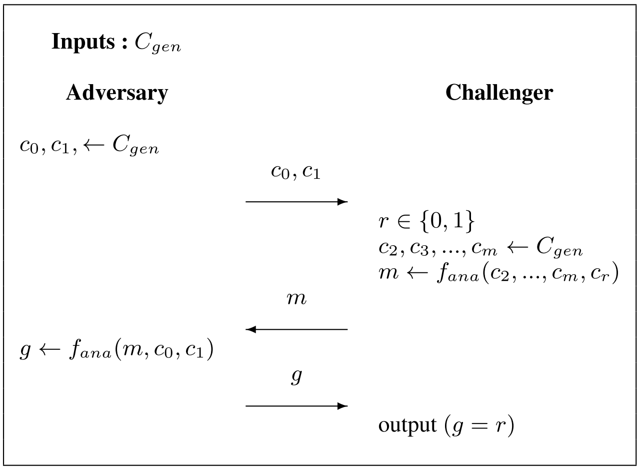
Inputs : C_{gen}
Adversary
Challenger
$c_0, c_1, \leftarrow C_{gen}$
c_0, c_1
$r \in \{0, 1\}$
$c_2, c_3, ..., c_m \leftarrow C_{gen}$
$m \leftarrow f_{ana}(c_2, ..., c_m, c_r)$
m
$g \leftarrow f_{ana}(m, c_0, c_1)$
g
output $(g = r)$

FIGURE 6. Privacy Game II.

4) The adversary computes decision function $f_{dec}(m, c_0, c_1)$ and sends a bit $g \in \{0, 1\}$ which shows the encrypted value is belonged to c_0 or c_1.
5) The challenger outputs true if $g = r$, otherwise false.

In the second privacy game [27] that is shown in Figure 6, the adversary tries to decide which consumption data is used in the calculation of analysis function f_{ana}. The adversary knows the result of the analysis. In this game, the adversary, whose power is not fully known, must define measurement data from a result of the aggregate function, which takes multiple consumption data as input. The purpose of the challenger is to provide privacy by using the aggregation function. The advantage of the adversary over random guessing is used as a privacy measure, and it depends on the followings:

- The background knowledge of the adversary (past load profiles of households, past billing information, weather conditions and some auxiliary information about household's energy consumption) [27].
- The distribution of energy consumption data used in an aggregate.
- The distinction level of two energy consumption chosen by the adversary.
- The size of aggregation.

Theorem 1: The proposed solution provides privacy for each individual consumer if the TEE is a tamper-proof hardware.

Proof: We first model the privacy of our solution based on Privacy Game I. An adversary $\mathcal{A}$ chooses two consumption scenarios $c_0 = (e^0_{ij})$ and $c_1 = (e^1_{ij})$ where $\sum_{i=1}^{s} e^0_{ij} = \sum_{i=1}^{s} e^1_{ij}$, $\sum_{j=1}^{t} e^0_{ij} = \sum_{j=1}^{t} e^1_{ij}$, t is the number of time intervals and s is the number of SAs. Assume that $\mathcal{A}$ can eavesdrop on the communication between SAs and TEE. In this case, the challenger $\mathcal{C}$ gives all encrypted consumption data $E_{SA}(e^r_{ij})$ in c_r to $\mathcal{A}$. The protocol is secure because all e^r_{ij} values are encrypted with secure block cipher (Lemma 1). In the second case, assume that the adversary can not eavesdrop on the communication between SAs and TEE. The challenger gives $\sum_{i=1}^{t} e^r_{ij}$ and $\sum_{i=1}^{s} e^r_{ij}$ to $\mathcal{A}$. It is oblivious that the adversary can not distinguish c_0 and c_1 by using $\sum_{i=1}^{t} e^r_{ij}$ and $\sum_{i=1}^{s} e^r_{ij}$. In the third case, $\mathcal{A}$ may try to corrupt the TEE to obtain the plain-text value of m. However, $\mathcal{A}$ can not obtain any data from the device TEE (Lemma 4).

The Utility eventually uses the consumption data stored in the encrypted database for different purposes. An adversary $\mathcal{A}$ at the Utility may try to infer information from the analysis result. To demonstrate this case, we model the privacy of our solution based on Privacy Game II, which refers to Identifiability requirement given in Section II-A. In [27], it was shown that for an aggregate of size 20, $\mathcal{A}$ identifies an energy consumption with 50% probability. This is an average advantage for all consumption. The advantage of the $\mathcal{A}$ having some external information is higher. In our solution, we can control the distribution of energy consumption in aggregate and the size of the aggregate and bring some restrictions on them. The queries that our system support is in the following form:

SELECT $f(*)$ FROM Consumption_Data
WHERE (AP_ID IN $(A_1, A_2, \ldots, A_N)$
OR SM_ID IN $(S_{1,2}, \ldots, S_N)$
OR RG_ID IN $(R_1, R_2, \ldots, R_N)$)
AND DT > dt$_1$ AND DT ≤ dt$_2$

We express a sample query in SQL format. $f()$ could be any type of aggregate function predetermined by the system. $f()$ cannot be chosen or controlled by the user. Consumption data can be searched according to appliance id, smart meter id or a specific region that is denoted in the query as *AP_ID*, *SM_ID*, *RG_ID* respectively, and also for a particular time interval. Different SQL queries can be derived depending on the need.

From the privacy point of view, the most critical aspect regarding queries is the size of the aggregate. As the number of data contributing to the aggregate function and the time interval increase, the level of privacy, which defines the degree of indistinguishability of customer behavior, will also increase. As an example, suppose that there are five houses in an area, and the adversary, who is navigating around the area, knows that only one house consumes in a given time interval. The adversary can easily explore from the query that the result will only be related to the data belonging to this house by just investigating the query. To avoid this, more consumption data should be used in the queries than the adversary can observe.

In our scheme, we assume that the DMS is a trusted entity. On the other hand, if DMS is considered an honest-but-curious entity, it can perform some analysis on the query results, e.g., by doing a differential analysis of two query results, it can infer the consumption data for a chosen time interval of queries. In order to prevent such cases, TEE can implement a differential privacy method before revealing the aggregated data to the DMS. The other countermeasures can restrict the queries so that only a specified set of queries can be submitted by DMS. The submitted query can then be validated by the TEE to check if the query is a member of this set before executing.

In our solution, we have taken privacy by design principle into account so that GDPR requirements can easily be applied and the Compliance requirement given in Section II-A can be met. Appliance and smart meter level data can be used for any kind of analysis within the scope of the consent to be taken from the customer. The customer can give these permissions within the data included in the specified time period. Since our solution protects the data without aggregation, it will be straightforward to delete these data at the request of the customer, which is not possible on the systems working with aggregated data.

V. PERFORMANCE EVALUATION

In this section, we analyze the performance of our scheme in terms of computation and communication overhead. In order to better explain the evaluation results, we highlight the differences, given in Table 1 between our aggregation solution and the most adapted aggregation methods in the literature.

Most of the studies in the related work section employ aggregation strategies for privacy protection either on SM or using a dedicated aggregator. The aggregator can behave as a mediator or trusted third party to collect the data and provide security functionalities depending on the scheme. Having an aggregator in the middle of the communication puts an extra overhead. Moreover, the data has to be sent in bigger chunks since it is aggregated. In our solution, we do not perform aggregation during the data reading, so we do not have an additional actor in our approach. Instead, we transfer each measurement to the Utility in an encrypted format and let the Utility perform the aggregation whenever needed. In our case, the number of transfers will be more, but the data size sent over the network will be much less.

In terms of the computation cost of the data reading, the studies that use aggregation need to encrypt the data on the SA or the SM using homomorphic encryption techniques before it arrives at the aggregator. Then aggregator performs the aggregation on the encrypted data using homomorphic encryption. When the Utility receives the aggregated data, it decrypts them. In our scheme, we encrypt the data on the SA, but we do not use homomorphic encryption to aggregate the data; rather, we use a symmetric encryption technique that is less costly than homomorphic encryption. The encrypted data is stored on the database after being received by the Utility and aggregated based on the need as a batch process. Hence aggregation does not affect the data reading cost.

Lastly, we compare the number of aggregations. Since aggregation is part of the data reading process, it must be done for every data reading round for studies that rely on aggregation for privacy protection. In our study, aggregation functionality is agnostic from data reading. This gives Utilities a lot of flexibility, increased accuracy in operations, and performance gain. As the data is in disaggregated format, they can perform aggregation for billing and load forecasting or directly use the disaggregated data for load monitoring purposes. Since the aggregation is not part of the data reading, the computation cost can be optimized using back-end hyper-scalers if needed.

In the following sections, we give some of the implementation results. We have implemented the TEE functionality using Azure confidential computing service, Standard_DC2s_v2 instance. The machine has a 3.7 GHz Intel Xeon E-2288G processor with SGX technology, with two physical cores, and 8 GB Ram. The platform runs Ubuntu 18.04 LTS. We use the 2019-04-2 Intel SGX Linux 2.5 Release, and the Enclave Page Cache is set to the maximum available size of 128 MB. We implemented a SA simulator on the same Linux machine for SA measurement encryption. We benefit from publicly available individual household electric power consumption data set provided by UCI repository [43], which includes minutely measured consumption values over four years period.

A. DATA READING

First, we focus on the performance of the data reading phase. We compare the communication and computation cost of our scheme's data reading phase with two previous schemes. Abdallah et al.'s scheme [15] and Qian et al.'s scheme [16] use lattice-based homomorphic encryption (HE), and aggregation of encrypted consumption data is done during their data reading phase with homomorphic operations. Homomorphic operations are costly both in terms of computation and communication due to their intrinsic properties compared with symmetric encryption schemes. The reasons behind this are; (i) the cost of the operations, including bootstrapping step used to reduce the noises in the ciphertext, and (ii) the big increase in the size of the resulting ciphertext after encryption. These are the biggest limitations for HE when it comes to time-critical applications, and reducing them is an active research area [44]. Thus, as shown in Figure 7 and 8, our protocol outperforms Abdallah et al.'s scheme and Qian et al.'s scheme in terms of both communication and computation performance in the data reading phase with the increasing number of customers. The main reason for this difference is that our scheme uses symmetric encryption and does not

TABLE 1. Cost comparison for different aggregation models.

Cost/Aggregation model	Aggregation during data reading	Aggregation on demand (proposed approach)
Communication cost	Communication cost between SA and aggregator + Communication cost between aggregator and Utility	Communication cost between SA and Utility
Computation cost of data reading	Encryption cost + Aggregation cost + Decryption cost	Encryption cost by SA
Number of aggregation during data reading	Performed on every data reading round	Performed on the Utility when needed

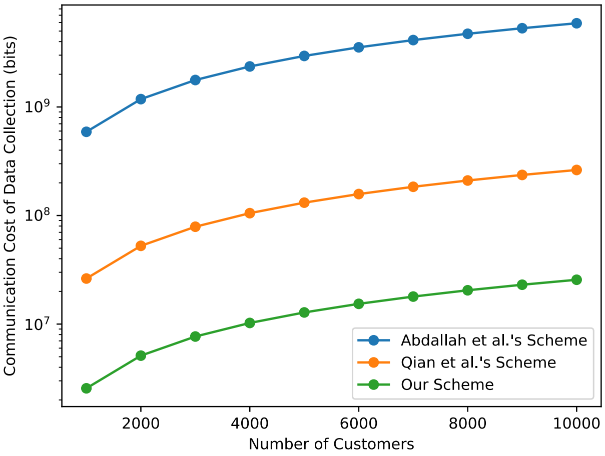

FIGURE 7. Comparison of the data reading phase of our scheme with those of previous protocols in terms of communication cost.

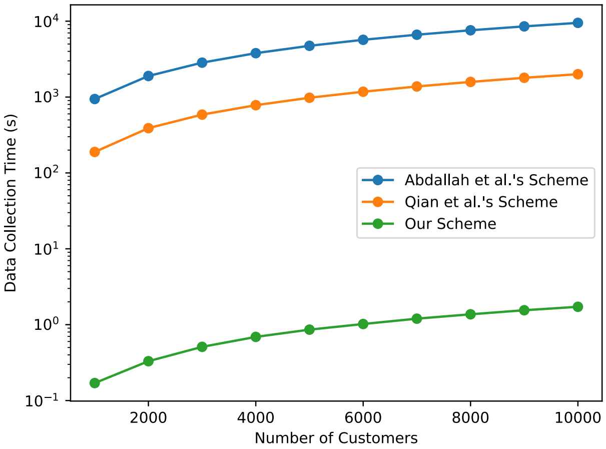

FIGURE 8. Comparison of the data reading phase of our scheme with those of previous protocols in terms of computation cost.

perform aggregation on consumption data during the data reading. The size of each measurement data encrypted on SAs fits into 128 bits, corresponding to one block of ciphertext, 128 bits, when we use the AES-GCM as AEAD scheme, while the ciphertext is around 72k bits in other schemes due to HE operations. This means that the communication overhead provided by ciphertext is very low in our approach, as seen in Figure 8. For computation overhead, we have evaluated the real-time computation cost for data reading, which is the sum HE cost, HE aggregation cost, and decryption cost for the related studies, and the total symmetric encryption cost is done by SAs for our study. The aggregation cost is not considered in our case as it is not part of the reading process but rather done offline when required. The results demonstrate that the performance of our approach surpasses due to the fast operation of AES-GCM, as shown in Figure 7.

B. DATA AGGREGATION

In this section, we present the execution times of aggregation that is performed on encrypted consumption data stored in the Utility's database. In our experiment, we assume that each customer has 20 SAs. Figure 9 shows that the execution time of aggregation scales linearly with the number of customers. The execution time of aggregation for 10,000 customers, each has 20 SAs is approximately 1 second. These results show the efficiency and practicality of our scheme. As we pointed out before, aggregation can be done anytime after the data reading phase in our scheme. This means aggregation is not part of the data reading phase. Even if we sum the execution times of the data reading phase and aggregation, both take approximately 2 seconds in total. This result shows how efficient our scheme is in terms of computation time compared to homomorphic encryption-based schemes.

C. DATA AGGREGATION FOR BILLING

In the billing application, DMS sends a query to TEE. In this experiment, DMS wants to know the total energy consumption of a specific user in a given time interval. In our experiments, we assume that the user has four SAs in the home. Each SA sends measurements periodically. Measurements periods in our experiments are minute, 15 min, or hour. We specify the time interval as 30 days, considering the billing period. For example, TEE needs to aggregate 172,800 entries in total for four SAs from one consumer, leading to 172,800 decryption and addition operations as stated in

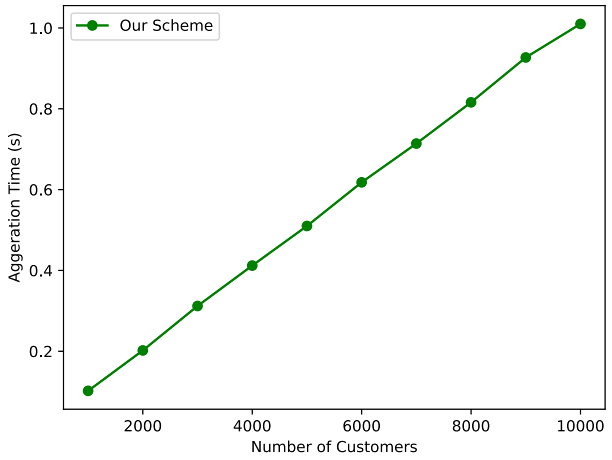

FIGURE 9. Execution times of performing aggregation on consumption data.

TABLE 2. Execution times of computing total energy consumption for a given time interval for different MPs with the increasing NoC (Time interval is 30 days and each customer has 4 SAs.).

	MP		
NoC	1 min	15 min	1 hour
1	1.032s	0.063s	0.015s
10	10.471s	0.658s	0.164s
100	106.933s	6.859s	1.774s
1000	1,082.247s	70.391s	18.117s
10000	10,975.495s	707.437s	182.592s

Equation 1.

$$NoE = NoSA \times NoC \times 30 \times 24 \times 60 \frac{1}{MP} \tag{1}$$

where

NoE = Number of entries in the encrypted database within a month

NoSA = Number of SAs

NoC = Number of consumers

MP = Measurement period e.g., 1 for minutely, 15 for quarterly, 60 for hourly measurements

As we can see from our performance evaluation given in Figure 10, the run-time of the billing application scales linearly with the NoE by changing NoC. We demonstrate the run-time complexity in the worst-case scenario in terms of memory utilization such that we have performed a single TEE operation for each time, which can be optimized using batch processing considering the SGX enclave page cache. In Table 2 we give the exact outcome of our experiments. The minimum number of operations is 2880 for 1 NoC, for hourly measurements, and the maximum is 172,800,000 for 10000 NoC for minutely measurements. The measurement period can be selected by the Utility based on the desired load profiling granularity. In our experiments, we used a single SGX platform. It is possible to increase throughput by running queries in parallel on several SGX platforms, or using hyper-scalers.

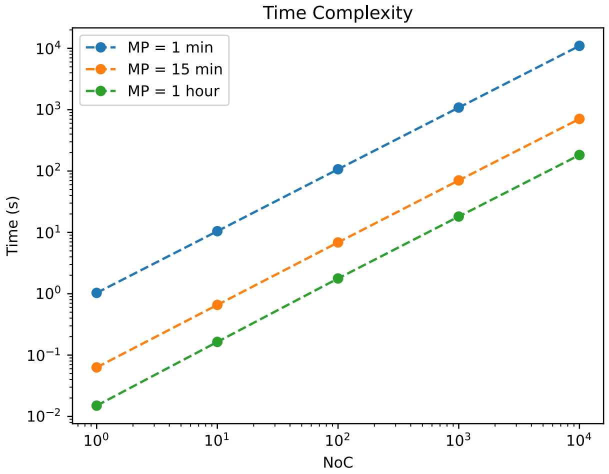

FIGURE 10. Execution times of computing total energy consumption for a given time interval complexity for different MPs with the increasing NoC (Time interval is 30 days and each customer has 4 SAs.).

VI. CONCLUSION

In this study, we proposed a privacy-preserving method to protect consumer data without compromising smart grid functionalities. We provided a novel customer-oriented aggregation scheme using Trusted Execution Environments. In the proposed scheme, customer data is protected all the way from the SAs through the Utility's trusted server, CSP. To enable this, we proposed a novel key initialization scheme. The encrypted database preserves the encrypted customer data coming from the SAs and serves the CSP when needed by DMS. TEE acts as a trusted key store for the cryptographic credentials of the SAs and performs security functions during consumer data usage for operational purposes like load profiling.

We elaborated on the fundamental security and privacy requirements and specific requirements to mitigate false data injection attacks. We presented security, privacy, and performance analysis to show that the proposed scheme is secure and viable. We also indicated how we addressed the identified requirements. We performed a privacy analysis by using game-based privacy definitions to formalize the methodology. The analysis results in some remarks to the distribution system operators regarding aggregation size, which affects the level of privacy. One of the advantages of the scheme is that it does not affect the smart meter deployment models of Utilities. Since we adopted end-to-end confidentiality in our privacy model, no functional change is required on smart meters or other intermediary devices.

Performance analysis shows the runtime scales linearly with the number of decryption operations and can scale up using cloud or hyper-scalers for real-world applications. We demonstrated that our method outperforms other privacy methods that use homomorphic encryption in terms of communication and computation cost. The scheme paves the way for industry acceptance as smart product designs will reach more mature states, and companies will embrace more privacy-aware techniques. From the legal point of view, the scheme meets the legally binding data privacy requirements, such as the right to erasure, as we do not use aggregation.

REFERENCES

[1] S. Zeadally, A.-S. K. Pathan, C. Alcaraz, and M. Badra, "Towards privacy protection in smart grid," *Wireless Pers. Commun.*, vol. 73, no. 1, pp. 23–50, Nov. 2013.

[2] G. Giaconi, D. Gündüz, and H. V. Poor, "Smart meter data privacy," *CoRR*, vol. abs/2009.01364, pp. 1–36, Sep. 2020.

[3] *Regulation (EU) 2016/679 of the European Parliament and of the 27 April 2016, General Data Protection Regulation*, Eur. Commission, 2016.

[4] M. A. Ferrag, L. A. Maglaras, H. Janicke, and J. Jiang, "A survey on privacy-preserving schemes for smart grid communications," *CoRR*, vol. abs/1611.07722, pp. 1–30, Nov. 2016.

[5] A. Molina-Markham, P. Shenoy, K. Fu, E. Cecchet, and D. Irwin, "Private memoirs of a smart meter," in *Proc. 2nd ACM Workshop Embedded Sens. Syst. Energy-Efficiency Building (BuildSys)*, New York, NY, USA, Nov. 2010, pp. 61–66, doi: 10.1145/1878431.1878446.

[6] R. Petrlic, "A privacy-preserving concept for smart grids," in *Proc. Sicherheit Vernetzten Systemen DFN Workshop*. Norderstedt, Germany: Books on Demand, 2010, pp. B1–B14.

[7] C. Rottondi, G. Mauri, and G. Verticale, "A data pseudonymization protocol for smart grids," in *Proc. IEEE Online Conf. Green Commun. (GreenCom)*, Sep. 2012, pp. 68–73, doi: 10.1109/GreenCom.2012.6519618.

[8] J.-M. Bohli, C. Sorge, and O. Ugus, "A privacy model for smart metering," in *Proc. 1st IEEE Int. Conf. Workshops Smart Grid Commun.*, May 2010, pp. 1–5.

[9] C. Rottondi, G. Verticale, and C. Krauss, "Privacy-preserving smart metering with multiple data consumers," *Comput. Netw.*, vol. 57, no. 7, pp. 1699–1713, May 2013, doi: 10.1016/j.comnet.2013.02.018.

[10] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012, doi: 10.1109/TPDS.2012.86.

[11] F. G. Mármol, C. Sorge, O. Ugus, and G. M. Pérez, "Do not snoop my habits: Preserving privacy in the smart grid," *IEEE Commun. Mag.*, vol. 50, no. 5, pp. 166–172, May 2012, doi: 10.1109/MCOM.2012.6194398.

[12] F. D. Garcia and B. Jacobs, "Privacy-friendly energy-metering via homomorphic encryption," in *Proc. 6th Int. Conf. Secur. Trust Manage. (STM)*. Berlin, Germany: Springer-Verlag, 2011, pp. 226–238. [Online]. Available: http://dl.acm.org/citation.cfm?id=2050149.2050164

[13] M. A. Mustafa, N. Zhang, G. Kalogridis, and Z. Fan, "DEP2SA: A decentralized efficient privacy-preserving and selective aggregation scheme in advanced metering infrastructure," *IEEE Access*, vol. 3, pp. 2828–2846, 2015, doi: 10.1109/ACCESS.2015.2506198.

[14] J. Hoffstein, J. Pipher, and J. H. Silverman, "NTRU: A ring-based public key cryptosystem," in *Proc. 3rd Int. Symp. ANTS*, Portland, OR, USA, Jun. 1998, pp. 267–288, doi: 10.1007/BFb0054868.

[15] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Trans. Smart Grid*, vol. 9, no. 1, pp. 396–405, Jan. 2018, doi: 10.1109/TSG.2016.2553647.

[16] J. Qian, Z. Cao, X. Dong, J. Shen, Z. Liu, and Y. Ye, "Two secure and efficient lightweight data aggregation schemes for smart grid," *IEEE Trans. Smart Grid*, vol. 12, no. 3, pp. 2625–2637, May 2021.

[17] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, Prague, Czech Republic, May 1999, pp. 223–238, doi: 10.1007/3-540-48910-X_16.

[18] S. Li, K. Xue, D. S. L. Wei, H. Yue, N. Yu, and P. Hong, "SecGrid: A secure and efficient SGX-enabled smart grid system with rich functionalities," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 1318–1330, 2020.

[19] D. Gündüz and J. Gómez-Vilardebó, "Smart meter privacy in the presence of an alternative energy source," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Budapest, Hungary, Jun. 2013, pp. 2027–2031, doi: 10.1109/icc.2013.6654823.

[20] G. Kalogridis, R. Cepeda, S. Z. Denic, T. Lewis, and C. Efthymiou, "ElecPrivacy: Evaluating the privacy protection of electricity management algorithms," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 750–758, Dec. 2011, doi: 10.1109/TSG.2011.2160975.

[21] G. Danezis, M. Kohlweiss, and A. Rial, "Differentially private billing with rebates," in *Proc. 13th Int. Conf. IH*, Prague, Czech Republic, May 2011, pp. 148–162.

[22] C. Dwork, "Differential privacy," in *Automata, Languages and Programming*, M. Bugliesi, B. Preneel, V. Sassone, and I. Wegener, Eds. Berlin, Germany: Springer, 2006, pp. 1–12.

[23] G. Ács and C. Castelluccia, "I have a DREAM! (Differentially private smart metering)," in *Proc. 13th Int. Conf. IH*, Prague, Czech Republic, May 2011, pp. 118–132, doi: 10.1007/978-3-642-24178-9_9.

[24] E. U. Soykan, Z. Bilgin, M. A. Ersoy, and E. Tomur, "Differentially private deep learning for load forecasting on smart grid," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2019, pp. 1–6.

[25] M. A. Lisovich, D. K. Mulligan, and S. B. Wicker, "Inferring personal information from demand-response systems," *IEEE Security Privacy*, vol. 8, no. 1, pp. 11–20, Jan./Feb. 2010, doi: 10.1109/MSP.2010.40.

[26] L. Sankar, S. R. Rajagopalan, S. Mohajer, and H. V. Poor, "Smart meter privacy: A theoretical framework," *IEEE Trans. Smart Grid*, vol. 4, no. 2, pp. 837–846, Jun. 2013, doi: 10.1109/TSG.2012.2211046.

[27] N. Büescher, S. Boukoros, S. Bauregger, and S. Katzenbeisser, "Two is not enough: Privacy assessment of aggregation schemes in smart metering," *Proc. Privacy Enhancing Technol.*, vol. 2017, no. 4, pp. 198–214, Oct. 2017, doi: 10.1515/popets-2017-0045.

[28] M. Jawurek, F. Kerschbaum, and G. Danezis, "SOK: Privacy technologies for smart grids—A survey of options," *Microsoft Res.*, vol. 1, pp. 1–16, 2012.

[29] S. Sultan, "Privacy-preserving metering in smart grid for billing, operational metering, and incentive-based schemes: A survey," *Comput. Secur.*, vol. 84, pp. 148–165, Jul. 2019, doi: 10.1016/j.cose.2019.03.014.

[30] *Directive 2009/72/EC of the European Parliament and of the Council of 13 July 2009 Concerning Common Rules for the Internal Market in Electricity and Repealing Directive 2003/54/EC*, Eur. Commission, 2009.

[31] G. W. Hart, "Residential energy monitoring and computerized surveillance via utility power flows," *IEEE Technol. Soc. Mag.*, vol. 8, no. 2, pp. 12–16, Jun. 1989.

[32] H. Y. Lam, G. K. S. Fung, and W. K. Lee, "A novel method to construct taxonomy electrical appliances based on load signaturesof," *IEEE Trans. Consum. Electron.*, vol. 53, no. 2, pp. 653–660, May 2007, doi: 10.1109/TCE.2007.381742.

[33] E. Naderi, A. Aydeger, and A. Asrari, "Detection of false data injection cyberattacks targeting smart transmission/distribution networks," in *Proc. IEEE Conf. Technol. Sustainability (SusTech)*, Apr. 2022, pp. 224–229.

[34] J. Lin, W. Yu, X. Yang, G. Xu, and W. Zhao, "On false data injection attacks against distributed energy routing in smart grid," in *Proc. IEEE/ACM 3rd Int. Conf. Cyber-Phys. Syst.*, Apr. 2012, pp. 183–192.

[35] *Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC*, Standard NIST SP 800-38D, NIST, 2007.

[36] S. Shinde, Z. L. Chua, V. Narayanan, and P. Saxena, "Preventing page faults from telling your secrets," in *Proc. 11th ACM Asia Conf. Comput. Commun. Secur.*, Xi'an, China, May/Jun. 2016, pp. 317–328, doi: 10.1145/2897845.2897885.

[37] Y. Xu, W. Cui, and M. Peinado, "Controlled-channel attacks: Deterministic side channels for untrusted operating systems," in *Proc. IEEE Symp. Secur. Privacy*, San Jose, CA, USA, May 2015, pp. 640–656, doi: 10.1109/SP.2015.45.

[38] S. Chen, X. Zhang, M. K. Reiter, and Y. Zhang, "Detecting privileged side-channel attacks in shielded execution with Déjàvu," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, Abu Dhabi, United Arab Emirates, Apr. 2017, pp. 7–18, doi: 10.1145/3052973.3053007.

[39] F. Brasser, U. Müller, A. Dmitrienko, K. Kostiainen, S. Capkun, and A. Sadeghi, "Software grand exposure: SGX cache attacks are practical," in *Proc. 11th USENIX Workshop Offensive Technol. (WOOT)*, Vancouver, BC, Canada, Aug. 2017, pp. 1–12. [Online]. Available: https://www.usenix.org/conference/woot17/workshop-program/presentation/brasser

[40] D. Boneh and V. Shoup, "A graduate course in applied cryptography," Draft 0.5, 2020.

[41] I. Anati, S. Gueron, S. P. Johnson, and V. R. Scarlata, "Innovative technology for CPU based attestation and sealing," in *Proc. 2nd Int. Workshop Hardw. Architectural Support Secur. Privacy*, vol. 13, no. 7. New York, NY, USA: ACM, 2013.

[42] Y. Lindell, "The security of Intel SGX for key protection and data privacy applications," Tech. Rep., 2018. [Online]. Available: https://cdn2.hubspot.net/hubfs/1761386/security-of-intelsgx-key-protection-data-privacy-apps.pdf

[43] *UCI Repository Individual Household Electric Power Consumption Data Set*. Accessed: Jul. 21, 2019. [Online]. Available: https://archive.ics.uci.edu/ml/machine-learning-databases/00235/

[44] Ö. Özerk, C. Elgezen, A. C. Mert, E. Öztürk, and E. Savaş, "Efficient number theoretic transform implementation on GPU for homomorphic encryption," *J. Supercomput.*, vol. 78, no. 2, pp. 2840–2872, Feb. 2022.

METE AKGÜN received the B.Sc. degree in computer engineering from Bahcesehir University, Turkey, in 2005, and the M.Sc. and Ph.D. degrees in computer engineering from Bogazici University, Turkey, in 2009 and 2016, respectively. He worked as a Senior Researcher at the Center of Research for Advanced Technologies of Informatics and Information Security (TUBITAK BILGEM), Kocaeli, Turkey. He is a Computer Scientist and a Security Researcher with the University of Tübingen, where he leads the Medical Data Privacy and Privacy-preserving ML (MDPPML) Group. His research interests include applied cryptography, data privacy, security protocols, and machine learning.

ELIF USTUNDAG SOYKAN (Member, IEEE) received the M.S. and Ph.D. degrees in computational science and engineering from Istanbul Technical University. She had worked at the Scientific and Technological Research Council, National Cryptology Institute for 13 years in the security domain. In 2018, she joined as a Senior Security Researcher with Ericsson Research and became a Master Security Researcher. Currently, she is with Ericsson Product Security as a Senior Security Technology Specialist. She has published several papers at international conferences, mostly on information security and privacy. Her research interests include privacy-enhancing technologies, ML/AI security, 6G, and the IoT security.

GURKAN SOYKAN (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering and the Ph.D. degree in computational science and engineering from Istanbul Technical University, Istanbul, Turkey, in 1998, 2001, and 2012, respectively. Currently, he is an Assistant Professor with Bahçeşehir University, Istanbul. His research interests include modeling and simulation of power systems, distributed energy resources, microgrids, and parallel simulation.