Privacy-Preserving Data Falsification Detection in Smart Grids using Elliptic Curve Cryptography and Homomorphic Encryption

Sanskruti Joshi†, Ruixiao Li†, Shameek Bhattacharjee‡, Sajal K. Das$, and Hayato Yamana†
†Department of Computer Science and Computer Engineering, Waseda University, Tokyo, Japan
‡Department of Computer Science, Western Michigan University, Kalamazoo, USA
$Department of Computer Science, Missouri University of Science and Technology, Rolla, USA
E-mail: sanskrutijoshi@fuji.waseda.jp, {liruixiao, yamana}@yama.info.waseda.ac.jp,
shameek.bhattacharjee@wmich.edu, sdas@mst.edu

***Abstract*— In an advanced metering infrastructure (AMI), the electric utility collects power consumption data from smart meters to improve energy optimization and provides detailed information on power consumption to electric utility customers. However, AMI is vulnerable to data falsification attacks, which organized adversaries can launch. Such attacks can be detected by analyzing customers' fine-grained power consumption data; however, analyzing customers' private data violates the customers' privacy. Although homomorphic encryption-based schemes have been proposed to tackle the problem, the disadvantage is a long execution time. This paper proposes a new privacy-preserving data falsification detection scheme to shorten the execution time. We adopt elliptic curve cryptography (ECC) based on homomorphic encryption (HE) without revealing customer power consumption data. HE is a form of encryption that permits users to perform computations on the encrypted data without decryption. Through ECC, we can achieve light computation. Our experimental evaluation showed that our proposed scheme successfully achieved 18 times faster than the CKKS scheme, a common HE scheme.**

Keywords—Elliptic Curve Cryptography, Homomorphic Encryption, smart grids, bilinear pairing

I. INTRODUCTION

Advanced Metering Infrastructure (AMI) refers to the entire infrastructure, from smart meters to two-way communication networks, controlling electric appliances and transferring energy usage data. AMI enables two-way communication with customers and serves as the smart grid's backbone. A smart grid allows bidirectional energy flow and integrates two-way communication and control capabilities, offering several new features and applications [1].

At the same time, customer data from smart meters raises privacy concerns and confidentiality issues [2]. The major concern is that it is vulnerable to cyber-attacks. As the energy consumption data collected from smart meters are sensitive consumer information, providing data privacy is a key concern. Hence, the data must be protected from malicious parties that attack the system to generate falsified data to manipulate the customer's power consumption data.

Ishimaki et al. [3] proposed a privacy-preserving anomaly-based attack detection scheme, which adopts the CKKS scheme for privacy-preserving anomaly detection. Note that the CKKS scheme is one of the homomorphic encryption (HE) schemes enabling the calculation over encrypted data. Other approaches include differential privacy and blockchain. Wen et al. [4] performed privacy-preserving anomaly detection for power grids by adopting local differential privacy (LDP) and a deep learning model. Keshk et al. [5] used blockchain technology to verify the data integrity and deep learning technology to perform anomaly detection. Though the methods above protect the security and privacy of the consumers' data, the computational overhead of previous works [3-5] still has been an issue.

Therefore, this paper proposes an elliptic curve cryptography (ECC) based HE scheme, which needs smaller memory space and shorter computation time than the HE schemes like CKKS, while ensuring the privacy of the data. Moreover, ECC supports bilinear pairing over encrypted data, detecting computation manipulations over meter readings. Whereas, homomorphic operations can be performed on encrypted data, which safeguards the secret information from unauthorized access.

Contributions: Our contributions are as follows.

i) We first propose an ECC-based HE scheme for privacy-preserving data falsification detection in smart grids, enabling faster computation than the previous CKKS-based schemes.
ii) We perform validation checking for encrypted data using pairing operations over encrypted data, which can detect maliciously encrypted data. That is, maliciously encrypted meter readings can be detected not to send to the utility. The solution uses the bilinear pairing property of ECC, which is not possible for other encryption schemes.
iii) For a fair comparison, our proposed scheme and the CKKS-based scheme are implemented on the same platform to compare the performance.

The rest of the paper is organized as follows. Section II describes related work, followed by Section III describing preliminary knowledge. The system architecture is explained in Section IV. Section V proposes our ECC-based HE scheme. The experimental evaluation is performed in Section VI. Finally, we conclude our work in Section VII.

II. RELATED WORK

The existing privacy-preserving techniques and privacy-preserving anomaly detection schemes are summarized.

A. Privacy-Preserving Techniques

The literature includes three privacy-preserving techniques, differential privacy (DP), secure multiparty computation (SMC), and homomorphic encryption (HE).

DP preserves privacy by adding a controlled amount of randomness (noise) to the raw data. Since the randomness is controlled, the resulting data is still accurate. However, DP adds much noise when the data has large diversity, which results in reducing the data utility. Moreover, balancing the best trade-off is an open problem [6].

Secure multiparty computation (SMPC) is a cryptographic approach that allows two or more parties to jointly compute without revealing any information to one another in decentralized scenarios. Though smart meters can outsource the desired computation to a set of servers, it is assumed that non-colluding servers are controlled by distinct third parties [7-8]. One of the disadvantages of SMPC is communication overhead among parties.

To perform data aggregation [9] and billing [10] computations in smart grids, additive HE (AHE), which can perform addition and constant multiplication, is sufficient. However, identifying anomalous behavior requires complex calculations such as division and logarithms, which prevents HE implementation.

B. Privacy-Preserving Anomaly Detection

A framework for privacy-preserving anomaly-based attack detection was proposed by Ishimaki et al. [3]. The CKKS scheme, one of the popular HE schemes, was adopted for privacy-preserving anomaly detection with the *harmonic to the arithmetic mea*n (HM-AM) metric. However, the HM-AM ratio involves various HE-incompatible operations, resulting in computational overhead in execution time.

Wen et al. [4] tackled the issue of energy theft detection in smart grids with a novel privacy-preserving federated learning framework, FedDetect. The local differential privacy (LDP) scheme is adopted to preserve consumers' data privacy. Besides, a deep learning model called the temporal convolutional network (TCN) is used to detect energy thefts in smart grids. However, LDP adds noise to the original data, resulting in degradation of the detection accuracy. Moreover, noise-added data cannot be used for billing calculations.

Keshk et al. [5] proposed a privacy-preserving framework to protect data and find anomalous behavior in smart power networks. The framework uses blockchain technology to verify the integrity of the data and a deep learning technique for anomaly detection. For privacy-preserving, they adopt a variational autoencoder to transform raw data into an encoded format before inputting it into the deep learning model. However, the raw data might be revealed before transforming the raw data to an encoded format.

In this paper, we tackle the heavy computation load of HE by adopting an ECC-based HE scheme as ECC provides the same security as a 3,072-bit RSA key with a 256-bit ECC key and supports the property of bilinear pairing. Then, we ensure the integrity of the data and detect data falsification over encrypted data. In the experiment, we compare our proposed scheme with Ishimaki's scheme [3].

III. PRELIMINARIES

This section explains the fundamental concepts required for our proposed scheme, including preliminaries of ECC, Elliptic Curve Discrete Logarithm Problem (ECDLP), bilinear pairing, and anomaly detection ratio metric.

A. Elliptic Curves and ElGamal Encryption

Elliptic curve cryptography is based on the properties of algebraic curves over fields [11]. To keep the comprehensiveness of this paper, the Elliptic curves and ElGamal encryption is described by quoting the explanation by Deepak et al. [11]. Mathematically, an elliptic curve is represented by an equation of the form: $y^2 = x^3 + ax + b$ with a constraint that the discriminant $\Delta = -16(4a^3+27b^2)$ is non-zero. The security of elliptic curve cryptography is based on the ECDLP [12]. In other words, given two points, P and Q, on the curve such that one is a scalar multiple of the other, i.e., $P = x.Q$ (here '.' (dot) represents scalar multiplication), it is computationally difficult to find x. The ElGamal encryption scheme with additive homomorphism can be implemented using elliptic curve cryptography.

B. Elliptic Curve Discrete Logarithm Problem (ECDLP)

The ECDLP [12] is the fundamental assumption for elliptic-curve-based protocols. Computing the discrete logarithm of a random elliptic curve element concerning a publicly known base-point is infeasible. The inability to compute the multiplicand given the original and product points is required for elliptic curve encryption. To be secure, the inability of the potential to compute an elliptic curve scalar multiplication is also required. The difficulty of the problem is determined by the size of the elliptic curve, as measured by the total number of discrete integer pairs satisfying the curve equation.

Consider an elliptic curve E defined over a finite field F_p. Let A be a point of order n on the elliptic curve, where A ∈ E (F_p). The ECDLP is based on identifying integer z, where $0 \leq z \leq n - 1$. For a given point B on the elliptic curve, B ∈ ⟨A⟩ and B is a scalar multiplication of the integer z and the point on elliptic curve A, such that B = z·A. Here '·' is the scalar multiplication. The property of bilinear pairing is supported by ECC. In this paper, we adopt bilinear pairing to detect malicious meter readings over encrypted data.

C. Bilinear Pairing

Let G_1 be an additively written group of order n with identity ∞, and let G_T be a multiplicatively written group of order n with identity 1. A bilinear pairing on (G_1, G_T) is a map $\hat{e}$: $G_1 \times G_1 \rightarrow G_T$ that satisfies the following conditions [13].

i) (bilinearity) For all $R, S, T \in G_1$, $\hat{e}(R + S, T) = \hat{e}(R, T), \hat{e}(S, T)$. This is equivalent to $\hat{e}(aS, bT) = \hat{e}(S, T)^{ab}$

ii) (non-degeneracy) $\hat{e}(P, P) \neq 1$

iii) (computability) $\hat{e}$ can be efficiently computed.

D. Anomaly Detection Metric

The harmonic to arithmetic mean (HM-AM) ratio is an efficient standard for identifying anomalous behavior in smart grid data [3]. Hence, the HM-AM ratio is adopted for anomaly-based attack detection in the proposed scheme because i) it deals with additive, deductive, and camouflage attacks, and ii) it can detect the minute changes in data that occurred due to data falsification attacks.

Here, N is denoted as the total number of smart meters in an AMI located in a neighborhood area network, and each timeslot is denoted as t. A set of timeslots t is represented by T, where ($\forall t \in T$). The power consumption of the i-th smart

meter is represented as $p_t^{(i)}$, where $p_t^{(i)} \in R^+$. The i-th smart meter performs natural logarithm transformation ($P_t^{(i)}$) on each power consumption $p_t^{(i)}$ and computes its inverse ($P_t^{'(i)}$) as follows: $P_t^{(i)} = log(p_t^{(i)} + 2)$, $P_t^{'(i)} = 1/P_t^{(i)}$. Finally, the HM-AM ratio Q_d for the d-th date is computed.

$$Q_d = \frac{\sum_{t \in T} HM_t}{\sum_{t \in T} AM_t}, \quad (1)$$

$$\text{where } AM_t = \frac{\sum_{i=1}^{N}(P_t^{(i)})}{N} \text{ and } HM_t = \frac{N}{\sum_{i=1}^{N}(\frac{1}{P_t^{(i)}})} \quad (2)$$

IV. System Architecture

The system architecture is shown in Figure 1 which consists of three main components: the utility, a computational server (operated by a third party), and N smart meters, which is the same as the model used by Ishimaki et al. [3]. In Figure 1 SM_i is the i-th smart meter, HAN is Home Area Network, and NAN is Neighborhood Area Network.

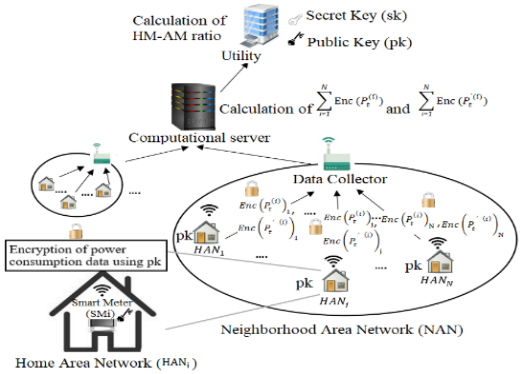

Figure 1: System Architecture

The function of each component is as follows:

Utility:

- The utility performs the system initialization step of key generation (public and secret keys).
- The utility sends the public key (only known to the smart meters) to the smart meters to perform encryption operations and keeps the secret key.
- The utility computes the HM-AM ratio (Equation 1) and performs the anomaly detection.

Smart meters:

- The smart meters use the public key provided by the utility to encrypt the power consumption data to send to the computational server.

Computational server:

- The computational server receives the encrypted power consumption data from the smart meters and computes the summation of AM (Equation 2) and the summation of HM (Equation 3).
- The computed summations are then sent to the utility to perform anomaly detection.

The details of the scheme are provided in Section 4.

Threat Model The proposed scheme attempts to protect the consumer's private data from both the computational server and the utility if the following conditions hold.

The utility, the computational server, and the smart meters are assumed to be semi-honest, i.e., honest but curious [3]. They obey the protocol; however, they try to collect the consumers' data while communicating the data from the smart meters to the utility through the computational server. Another assumption is that the computational server does not collude with the utility that has the secret key. Note that if the server and a subset of smart meters collude, only the meter readings of those smart meters are revealed, whereas readings from other smart meters are protected. Thus, we do not restrict the collusion among smart meters and between smart meters and the computational server.

Data integrity threat, where an adversary attacks the smart meters to falsify the meter readings, is assumed to occur before encrypting the power consumption data in smart meters.

V. Proposed Scheme

We propose a novel privacy-preserving data falsification detection scheme with ECC-based HE to encrypt data and perform HM-AM ratio calculations over the encrypted data. ECC-based encryption is additively homomorphic, which uses smaller keys to improve performance end-to-end and supports bilinear pairing. Moreover, the bilinear pairing function over the elliptic curve group allows us to validate the encrypted data without decryption.

The proposed scheme consists of four phases, as shown in Figure 2: 1) system initialization by the utility, 2) meter report generation by smart meters, 3) HM-AM computation over encrypted data by the computational server, and 4) anomaly attack detection by the utility. Table I shows the variables used in this paper.

TABLE I. Description of Variables

Variable	Description
P	Independent point on the elliptic curve
Q	Independent point on the elliptic curve
G_1	Elliptic curve group
T	A set of timeslots in a day
t	A timeslot
r_t	A random value generated at timeslot t
$p_t^{(i)}$	Power consumption of i-th smart meter at timeslot t, where $1 \leq i \leq$ N.
$P_t^{(i)}$	Natural log transformation of $p_t^{(i)}$ $(log(p_t^{(i)} + 2))$
$P_t^{'(i)}$	Inverse of $P_t^{(i)}$ $(1/log(p_t^{(i)} + 2))$
SHA-256	A cryptographic hash function that outputs a 256-bit long value
$\hat{e}$	Bilinear map
AM_{sum}^t	Summation of Enc $(P_t^{(i)})$ at timeslot t $(1 \leq i \leq N)$
HM_{sum}^t	Summation of Enc $(P_t^{'(i)})$ at timeslot t $(1 \leq i \leq N)$
N	Total number of smart meters
Q_d	HM-AM ratio for a day

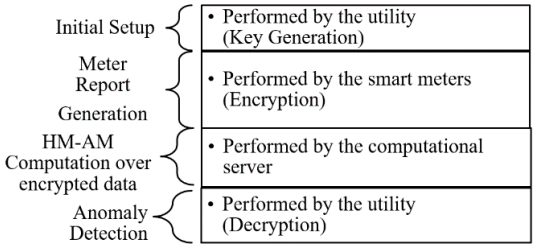

Figure 2: Proposed Scheme

A. System Initialization

The utility performs the initial setup. First, the utility chooses an appropriate elliptic curve group G_1 and two independent points on the elliptic curve, P, $Q \in G_1$ of order n. Second, the utility generates two keys (key_1 and key_2) that are shared between the smart meters and the utility only. Both the keys are kept secret from the computational server.

B. Meter Report Generation

The HM-AM ratio involves $\log(p_t^{(i)}+2)$ and its inverses that are not HE-friendly operations. Therefore, we first need to compute $P_t^{(i)} = \log(p_t^{(i)} + 2)$ and $P_t^{'(i)} = 1/P_t^{(i)}$ for i-th smart meter. Log-transformation results in decimal places; however, we can perform encryption over integers only. Thus, we first round both $P_t^{(i)}$ and $P_t^{'(i)}$ up to four decimal places and then remove decimal points to make it an integer.

First, every smart meter computes a random value r_t using the shared key_1 and corresponding timeslot t. Note that for timeslot t, every smart meter uses the same random value r_t.

$$r_t = \text{SHA256}(key_1 \| t) \quad (4)$$

Second, we need to encode $P_t^{(i)}$ into a group element to apply ECC-based ElGamal encryption. We round $P_t^{(i)}$ to three decimal places and then convert $P_t^{(i)}$ into integer. After that, we encode it as $(P_t^{(i)} - 1)r_t\text{P}$, where r_t is the random value generated by Equation 4. The corresponding encryption of $P_t^{(i)}$ is shown s below.

$$\text{Enc}(P_t^{(i)}) = (r_tP, ((P_t^{(i)} - 1)r_tP + r_tP) = (r_tP, P_t^{(i)}r_t) \quad (5)$$

Similarly, encryption of $m' = P_t^{'(i)}$ is shown below.

$$\text{Enc}(m') = (r_t' Q, m' r_t' Q), \quad (6)$$
$$\text{where } r_t' = \text{SHA256}(key_2 \| t) \quad (7)$$

The algorithm of smart meter-side encryption is shown below.

Algorithm 1 (Smart meter-side encryption)

Input:
- $p_t^{(i)}$: Power consumption data of i-th meter at time t
- pk: Elgamal public key

Output: $Enc(p_t^{(i)})$, $Enc(P_t^{(i)})$, $Enc\left(P_t^{'(i)}\right)$

1: $c_1 = Enc_{pk}(p_t^{(i)})$ ▶ Encryption using Elgamal public key
2: Compute $P_t^{(i)} = log(p_t^{(i)} + 2)$
3: $c_2 = Enc_{pk}(P_t^{(i)})$
4: Compute $P_t^{'(i)} = 1/P_t^{(i)}$
5: $c_3 = Enc_{pk}(P_t^{'(i)})$
6: return c_1, c_2, c_3

C. HM-AM Computation over Encrypted Data

After receiving the encrypted data from each smart meter, the computational server first checks for the validity of each pair ($\text{Enc}(P_t^{(i)})$, $\text{Enc}(P_t^{'(i)})$) using bilinear pairing as follows:

$$\hat{}(r_tP, r_t' Q) = \hat{e}(mr_tP, m' r_t' Q), \quad (8)$$
$$\text{where } m = P_t^{(i)} \text{ and } m' = P_t^{'(i)}$$

Note that since $m \times m' = 1$, we have the following equation.

$$\hat{}(mr_tP, m' r_t' Q) = \hat{e}(P, Q)^{mr_tm' r_t'} = \hat{e}(P, Q)^{r_tr_t'}$$
$$= \hat{e}(r_tP, r_t' Q) \quad (9)$$

If and only if the above equation does not hold for each reading set of $\text{Enc}(P_t^{(i)})$ and $\text{Enc}(P_t^{'(i)})$, where $1 \le i \le \text{N}$, the reading set has been manipulated, which results in discarding the data set to detect the malicious meter readings. After confirming the above equation holds, the computational server computes AM_{sum}^t and HM_{sum}^t for each timeslot t as shown below, where N is the total number of smart meters.

$$AM_{sum}^t = \sum_{i=1}^{N} \text{Enc}(P_t^{(i)}) \quad (10)$$
$$HM_{sum}^t = \sum_{i=1}^{N} \text{Enc}(P_t^{'(i)}) \quad (11)$$

The computational server then sends $\{AM_{sum}^t, HM_{sum}^t\}_{\forall t \in T}$ to the utility to calculate the HM-AM ratio. The algorithm of server-side computation is shown below.

Algorithm 2 (Server-side computation for each timeslot)

Input:
- $\left\{Enc(P_t^{(i)})\right\}_{1 \le i \le N}$: A set of encrypted log power consumption data in an area
- $\left\{Enc\left(P_t^{'(i)}\right)\right\}_{1 \le i \le N}$: A set of encrypted inverse log power consumption data in an area

Output: $Enc\left(\sum_{i=1}^{N} P_t^{(i)}\right), Enc\left(\sum_{i=1}^{N} P_t^{'(i)}\right)$

1: $\textbf{fracsum}_t \leftarrow 0$, $\textbf{sum}_t \leftarrow 0$
2: **for** $i \leftarrow 1$ to N **do**
3: $\quad \textbf{sum}_t \leftarrow \textbf{sum}_t \boxplus Enc(P_t^{(i)})$
4: $\quad \textbf{fracsum}_t \leftarrow \textbf{fracsum}_t \boxplus Enc\left(P_t^{'(i)}\right)$
5: **end for**
6: return $\textbf{sum}_t$, $\textbf{fracsum}_t$

D. Anomaly Detection by the Utility

After receiving $\{AM_{sum}^t, HM_{sum}^t\}_{t \in T}$ from the computational server, the utility first decrypts them and then computes AM_t and HM_t as follows:

$$AM_t = \frac{Dec(AM_{sum}^t)}{N}, \quad HM_t = \frac{N}{Dec(HM_{sum}^t)}, \quad (12)$$

where N is the total number of smart meters.

Finally, the utility computes the HM-AM ratio as follows:

$$Q_d = \frac{\sum_{t\in\ T} HM_t}{\sum_{t\in\ T} AM_t} \qquad (13)$$

Then, the utility adopts the residual under the curve (RUC) metric proposed in [14] to detect the anomaly. The algorithm of utility-side computation is shown below.

Algorithm 3 (Utility-side computation)

Input:

- $\{\boldsymbol{sum_t}\}_{t\in T}$: A set of the encrypted sum of log power consumption data in an area
- $\{\boldsymbol{fracsum_t}\}_{t\in T}$: A set of the encrypted sum of inverse log power consumption data in an area

Output: Anomaly detection result

1: HM ← 0, AM ← 0

2: **for** $t \in T$ **do**

3: AM ← AM + $\frac{Dec(sum_t)}{N\ \ N}$

4: $HM \leftarrow$ HM + $\frac{}{Dec(fracsum_t)}$

5: Calculate $\frac{HM}{AM}$

6: Apply the RUC metric [14] to detect the anomaly.

E. System Goals

The goal of our proposed scheme is as follows:

- Perform anomaly-based attack detection in a secured manner without disclosing each consumer's power usage details to the server and the utility.
- Verify the validity of the encrypted data to ensure that the consumers' data are not manipulated to pass through the anomaly-based detection process.

F. Security Analysis

The proposed scheme ensures the security of the consumers' data from both the computational server and the utility, as shown below.

- The proposed scheme adopts the ECC-based El-Gamal system, and its security depends on the discrete logarithm (DL) problem [12] in the elliptic curve (EC) group. Therefore, the proposed scheme is as secure as the DL in the EC group.
- As the computational server evaluates encrypted data, the privacy of the data is maintained, and consumers' data is protected from leaking.
- In the proposed scheme, AM_t and HM_t (the arithmetic mean and the harmonic mean of power consumption data at each time slot) will be visible to the utility; however, this does not leak individual readings of the meter; thereby, the utility cannot find the reading of any meter from either AM_t or HM_t.

VI. EXPERIMENTAL EVALUATION

To confirm the effectiveness of our proposed scheme, we compared our scheme with the CKKS scheme, which is one of the popular HE schemes.

A. Setup

For all the experiments, we used a Windows 10 operating system, 11th Gen Intel(R) Core (TM) i5 (2.4 GHz) processor, 8 GB RAM, SageMath 9.2 compiler, and Python 3.8.10.

The proposed scheme was implemented in Jupyter Notebook and NumPy library in Python-based SageMath[1]. The proposed scheme uses elliptic curve E: $y^2 = x^3 - 4$ over a finite field F_p with prime of form p = $36u^4 + 36u^3 + 24u^2 + 6u + 1$, where $u = 2^{114}+2^{101} - 2^{14} - 1$, which is the recommended setting to achieve 128-bit security for bilinear pairing and ECC-based encryption [15].

For the CKKS-based scheme, the HEAAN Python library[2] and the Python wrapper[3] for the HEAAN C++ library[4] were used. For division over ciphertext operation, an inbuilt cipher inverse function of the HEAAN library is used. Parameters are set as (n, log Q, p) = (2^{15}, 491, 35), in which a fresh ciphertext size is calculated as $2n \cdot logQ$ bits. As $2^{15} = 32{,}768$, it supports 128-bit security [16].

We used the same dataset used in the paper [3], a smart grid dataset collected from the Pecan Street Project. The dataset includes the power consumption data of 200 households in Texas, USA, over three years (2014–2016). The differences in the implementation of the proposed scheme and the CKKS scheme are listed in Table II.

TABLE II. IMPLEMENTATION COMPARISON

	ECC-based HE (proposed)	CKKS
Library	NumPy	HEAAN C++
Tool	Python binding for C++ libraries	Python based SageMath
Inverse function	not supported over ciphertext	supported over ciphertext
Linking	supported through bilinear pairing	unsupported
HM-AM ratio	computed by the utility	computed by the computational server

B. Performance comparison

In this experiment, we compared the performance of use-side encryption, server-side computation, utility-side decryption, and the total execution time.

a) Performance of user-side encryption

Each smart meter performs three encryptions per timeslot, i.e., Enc ($p_t^{(i)}$), Enc ($P_t^{(i)}$) and Enc ($P_t^{'(i)}$). Table III shows the runtime for three times encryption, including Enc ($p_t^{(i)}$), Enc ($P_t^{(i)}$) and Enc ($P_t^{'(i)}$).

TABLE III. RUNTIME COMPARISON OF USER-SIDE ENCRYPTION

Scheme	Runtime of user-side encryption (sec)
ECC-based HE (proposed)	0.074
CKKS (Enc(0) +m)	0.016
CKKS(Enc(m))	2.112

When encrypting with the CKKS, direct encryption takes longer than an addition with ciphertext; thereby, Enc(0) is pre-computed, consuming 0.984 sec, followed by addition.

[1] https://www.sagemath.org/download.html
[2] https://awesomeopensource.com/project/Huelse/HEAAN-Python
[3] https://github.com/Huelse/HEAAN-Python
[4] https://github.com/snucrypto/HEAAN

Table III shows that the ECC-based HE scheme has a longer encryption time than the CKKS with pre-computation; however, the ECC-based HE is much faster than the CKKS without pre-computation.

b) Performance of server-side computation

Table IV shows the server-side computation runtime, which confirms that the ECC-based HE scheme performs better with and without bilinear pairing operation than the CKKS scheme. Our proposed scheme can adopt bilinear pairing on the server to check if the individual encryptions Enc ($P_t^{(i)}$) and Enc ($P_t^{'(i)}$) are related to each other. In the CKKS scheme, the computational server computes the HM-AM ratio; thereby, the runtime without HM-AM ratio computation is also shown in Table IV for a fair comparison. Note that the ECC-based HE scheme omits the calculation of the HM-AM ratio in the computational server to delegate it to the utility.

TABLE IV. RUNTIME COMPARISON OF SERVER-SIDE COMPUTATION

Scheme	Runtime of server-side computation (sec)
ECC-based HE w/o bilinear pairing (proposed)	0.051
ECC-based HE w/ bilinear pairing (proposed)	112.457
CKKS (w/ HM-AM ratio computation)	191.590
(w/o HM-AM ratio computation)	(63.962)

c) Performance of utility-side computation

Table V shows the runtime of utility-side computation, which confirms that the CKKS scheme is faster than the ECC-based HE scheme. The reason is that the ECC-based HE scheme needs to compute the HM-AM ratio after the decryption of HM and AM, while the CKKS scheme needs only the decryption of the HM-AM ratio.

TABLE V. RUNTIME COMPARISON OF UTILITY-SIDE COMPUTATION

Scheme	Runtime of utility-side computation(sec)
ECC-based HE (proposed)	10.377
CKKS	0.273

d) Total performance

The total execution time is compared in Table VI, which confirms that the ECC-based HE scheme performs better with and without bilinear pairing operation than the CKKS scheme. Especially without bilinear pairing, our proposed ECC-based HE scheme performed 18 times faster than the CKKS scheme. Furthermore, we still have 1.56 times speedup even with bilinear pairing to check if the individual encryptions Enc ($P_t^{(i)}$) and Enc ($P_t^{'(i)}$) are related to each other to increase the security level.

TABLE VI. TOTAL RUNTIME

Scheme	Total runtime (sec)
ECC-based HE w/o bilinear pairing (proposed)	10.502
ECC-based HE w/ bilinear pairing (proposed)	122.908
CKKS	191.879

VII. CONCLUSION

We proposed the first implementation of ECC-based HE with HM-AM ratio-based anomaly detection and confirmed 18 times speedup of the anomaly detection compared to the previously proposed CKKS scheme. Our proposed scheme is as secure as the CKKS scheme and ensures that the consumers' private data is protected from the server and the utility.

ACKNOWLEDGMENT

This work was supported by Japan-US Network Opportunity 2 by Commissioned Research of the National Institute of Information and Communications Technology (NICT), Japan, and NSF grants SATC-2030611, SATC-2030624, DGE-1433659, CNS-1818942.

REFERENCES

[1] NIST, Smart grid: A beginner's guide, https://www.nist.gov/el/smart-grid/about-smart-grid/smart-grid-beginners-guide

[2] M.R. Asghar and D. Miorandi, "A holistic view of security and privacy issues in smart grids," Proc. of *Int'l Workshop on Smart Grid Security*, LNCS, Springer, vol. 7823 pp. 58-71, 2012.

[3] Y. Ishimaki, S. Bhattacharjee, H. Yamana, and S. K. Das, "Towards Privacy-preserving Anomaly-based Attack Detection against Data Falsification in Smart Grid," Proc. of *2020 IEEE Int'l Conf. on Communications, Control, and Computing Technologies for Smart Grids*, pp. 1-6, 2020.

[4] M. Wen, R. Xie, K. Lu, L. Wang, and K. Zhang, "FedDetect: A Novel Privacy-Preserving Federated Learning Framework for Energy Theft Detection in Smart Grid," *IEEE Internet of Things J.*, vol. 9, no. 8, pp. 6069-6080, 2022.

[5] M. Keshk, B. Turnbull, N. Moustafa, D. Vatsalan, and Kim-Kwang R. Choo, "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks," *IEEE Trans. on Industrial Informatics*, vol. 16, no. 8, pp. 5110-5118, 2019.

[6] M. R. Asghar, G. Dán, D. Miorandi, and I. Chlamtac, "Smart Meter Data Privacy: A Survey," *IEEE Comm. Surveys & Tutorials*, vol. 19, no. 4, pp. 2820-2835, 2017.

[7] G. Danezis, C. Fournet, M. Kohlweiss, and S. Zanella-B´eguelin, "Smart meter aggregation via secret-sharing," Proc. of the *First ACM Workshop on Smart Energy Grid Security*, pp. 75–80, 2013.

[8] M. A. Mustafa, S. Cleemput, A. Aly and A. Abidin, "A Secure and Privacy-Preserving Protocol for Smart Metering Operational Data Collection," *IEEE Trans. on Smart Grid,* vol. 10, no. 6, pp. 6481-6490, 2019.

[9] F. Li, B. Luo, and P. Liu, "Secure Information Aggregation for Smart Grids Using Homomorphic Encryption," Proc. of *First IEEE Int'l Conf. on Smart Grid Communications*, pp. 327-332, 2010.

[10] X. Liang, X. Li, R. Lu, X. Lin, and X. Shen, "UDP: Usage-Based Dynamic Pricing with Privacy Preservation for Smart Grid," *IEEE Trans. on Smart Grid*, vol. 4, no. 1, pp. 141-150, 2013.

[11] K. Deepak and K. Chandrasekaran, "Investigating Elliptic Curve Cryptography for Securing Smart Grid Environments," Proc. of *Third ISEA Conf. on Security and Privacy (ISEA-ISAP),* pp. 1-7, 2020.

[12] D. Hankerson and A. Menezes, "Elliptic Curve Discrete Logarithm Problem," *Encyclopedia of Cryptography and Security*, Springer, pp. 397-400, 2011.

[13] PS. Barreto, HY. Kim, B. Lynn, and M. Scott, "Efficient algorithms for pairing-based cryptosystems," Proc. of *Ann. Int'l Cryptology Conf.*, pp. 354-369, 2002.

[14] S. Bhattacharjee and S. K. Das, "Detection and Forensics against Stealthy Data Falsification in Smart Metering Infrastructure," *IEEE Trans. on Dependable and Secure Computing*, vol. 18, no. 2, pp. 356-371, 2021.

[15] R. Barbulescu and S. Duquesne, "Updating Key Size Estimations for Pairings," *J. of Cryptology*, vol. 32, pp.1298–1336, 2018

[16] M. Chase, H. Chen, J. Ding, S. Goldwasser, S. Gorbunov, J. Hoffstein, K. Lauter, S. Lokam, D. Moody, T. Morrison, A. Sahai, and V. Vaikuntanathan, "Security of Homomorphic Encryption," Draft Homomorphic Encryption Standard, HomomorphicEncryption.org, Redmond WA, Tech. Rep, 2017