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

A Thesis Submitted to the Department of Computer Science and Communications Engineering,
the Graduate School of Fundamental Science and Engineering of Waseda University in Partial Fulfillment of the Requirements for the Degree of
Master of Engineering

Submission Date: January 24, 2022

Sanskruti Joshi
Student ID: 5120FG06 – 6
Advisor: Professor Hayato Yamana
Research Guidance: Research on Parallel and Distributed Architecture

Abstract

In an advanced metering infrastructure (AMI), the utility collects power consumption data from smart meters to improve energy optimization and provides detailed information on power consumption to utility customers. However, AMI is vulnerable to data falsification attacks, which can be launched by organized adversaries. Such attacks can be detected by analysing fine-grained power consumption data from customers, however, they violate the privacy of each customer in the grid. To strike a balance between privacy and security, a framework for privacy-preserving anomaly-based attack detection was proposed in the previous work, which uses homomorphic encryption (HE) scheme to address the issue of data falsification.

HE is a form of encryption that permits users to perform computations on the encrypted data without having to decrypt the data. However, the downside of HE is computational overhead in terms of execution time. This thesis proposes a method for privacy-preserving and attack detection of data generated by smart meters to shorten the execution time. Our method applies elliptic curve cryptography (ECC) based HE for anomaly-based attack detection for data falsification over encrypted data. Through ECC, we can achieve the same security as a 3,072-bit RSA key with a 256-bit ECC key. Therefore, ECC requires less memory space to implement the encryption and decryption algorithms, which in turn reduces the time required to perform encryption and decryption operations.

The proposed scheme and the CKKS-based method are implemented on the same platform using Python 3.8.10 to compare the execution times for user-side computation, server-side computation, and utility-side computation. In the proposed scheme, the user side computation is 10 times faster and the server-side computation is more than 100 times faster compared to the CKKS (HE) scheme.

Contents

1. Introduction

Advanced Metering Infrastructure (AMI) refers to the entire infrastructure, from smart meters to two-way communication networks, to control electric appliances and all applications that enable the real-time gathering and transfer of energy usage data. AMI enables two-way communication with customers and serves as the smart grid's backbone. The smart grid provides an extraordinary opportunity to enhance the energy industry into a new era of reliability, availability, and efficiency that will contribute to our economic growth and sustainable environmental development. The objective of a smart grid is to provide a comprehensive architecture for the complete life-cycle management of energy resources based on the development of intelligent, dependable, secure, and cost-effective technology. 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, consumers' data from smart meters raises privacy concerns and confidentiality issues [2]. The major concern with the smart grid 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 generated by the smart meters must be protected from the malicious parties that attack the system to generate falsified data to manipulate the customers' power consumption data.

The information gathered by the smart meters could be used for an unforeseen purpose: invading users' privacy. According to current research, individual appliances (based on their load signatures [3]) can be recognized via extensive examination of energy usage traces [4]-[5]. Periodic meter readings can also be used to estimate a household's occupancy, and data mining algorithms can be used to violate users' privacy in more nuanced ways, such as by disclosing their activities and economic status [6]-[7]. Thus, it is crucial to ensure that malicious parties cannot access and modify the data in the smart meters.

The data management of smart meters should fulfil several security requirements, which are as follows [8]:

- Confidentiality: During transmission (data-in-transit), storage (data-at-rest), and computing, meter data should not be exposed to unauthorized individuals or processes (data-in-use). To achieve cryptographic privacy, data-in-transit, data-at-rest, and data-in-use must all be kept confidential.
- Integrity: During transmission, storage and computation, the accuracy and validity of the meter data should be preserved, and any modifications to the data should be detectable.
- Authenticity: The meter data receiver should be able to authenticate the source of the data.
- Non-Repudiation: The meter data source should not be able to refuse that it is the source of the data. It conveys authenticity and integrity.

- Auditability: The response to a request (meter data or a computation based on meter data) should be able to be verified.

Ishimaki et al. [9] proposed a framework for privacy-preserving anomaly-based attack detection. The proposed scheme uses the CKKS scheme for privacy-preserving anomaly detection by adopting a homomorphic encryption (HE) scheme based on the harmonic to the arithmetic mean (HM-AM) ratio. The HM-AM ratio is a metric that has been recently demonstrated as an effective indicator for detecting anomalous behaviour in smart metering data [9]. Wen et al. [10] performed privacy-preserving anomaly detection for power grids by using a local differential privacy (LDP) scheme and a deep learning model called temporal convolutional network (TCN). Keshk et al. [11] used blockchain technology to verify the integrity of the data and deep learning technology to perform anomaly detection. The previous methods, [9]-[11] focus on protecting the security and privacy of the consumers' data. However, the computational overhead depending on previous work is still an issue.

Therefore, in this thesis, elliptic curve cryptography (ECC) based HE scheme that not only ensures the security of the data but also detects data falsification over encrypted data is proposed. Through ECC, we can achieve the same security as a 3,072-bit RSA key with a 256-bit ECC key. Thus, ECC takes less memory space compared to the HE schemes to implement the encryption and decryption algorithms, which reduces the time required to perform encryption and decryption operations. Whereas homomorphic operations can be performed on encrypted data, which safeguards the secret information from unauthorized access. Thus, the ECC based HE scheme provides faster computations while ensuring the security of the data.

Contributions: The proposed scheme makes the following contributions:

i. The proposed scheme provides an ECC-based HE scheme for privacy-preserving data falsification detection in smart grids.
ii. It performs validation checking for different encryptions using pairing operations over encrypted data. The solution uses the bilinear pairing property of ECC (which is not possible for other encryption schemes).
iii. For a fair comparison, the proposed scheme and the CKKS based method are implemented on the same platform.

The rest of the thesis is organized as follows. The related work is introduced in Section 2. The preliminaries are described in Section 3. The system goals are explained in Section 4. The details of the proposed scheme are in Section 5. The experimental evaluation of the proposed scheme is in Section 6. Finally, the conclusion of the work is in Section 7.

2. Related Work

In this section, the existing privacy-preserving techniques and privacy-preserving anomaly detection schemes are briefly reviewed.

2.1 Privacy-Preserving Techniques

Privacy-preserving is an important concept because when the data is transferred between different parties, it is necessary to provide security to that data. Providing security ensures that data being communicated between the original parties is protected from untrusted third parties. There are various types of privacy-preserving schemes that use different methods to achieve privacy preservation.

We will address three types of privacy-preserving schemes, which are as follows:
(i) differential privacy (DP), (ii) secure multiparty computation (SMC), and
(iii) homomorphic encryption (HE).

In a DP scheme, privacy is preserved by adding a controlled amount of randomness (noise) to the data. As the randomness is controlled, the resulting data is still accurate and sensitive information is not revealed. But the downfall of DP is that it only works for interactive scenarios (where users can directly send the queries to the original database) and cannot provide good results for complex queries. When there is diversity in data, DP includes too much noise, which ultimately reduces the data utility. Moreover, balancing the best trade-off is an open problem [8].

Secure multi-party computation (also known as multi-party computation, SMPC, or MPC) is a cryptographic approach that allows two or more parties to do a computation using their private data without revealing their private information to one another. While smart meters can outsource the desired computation to several servers in a reasonable amount of time, it is assumed non-colluding servers are controlled by distinct third parties [12]-[13]. The disadvantages of SMC are (i) communication overhead: The SMC method requires communication between parties, which can lead to high communication costs. (ii) It is vulnerable to attacks from colluding parties; when the parties collude, data might be leaked.

To perform data aggregation [14] and billing [15] computations in smart grids, Additive HE (AHE) which can perform addition and constant multiplication is sufficient. However, for identifying anomalous behaviour in smart metering data, it is necessary to perform operations like division and logarithms.

2.2 Privacy-Preserving Anomaly Detection

A framework for privacy-preserving anomaly-based attack detection was proposed by Ishimaki et al. [9]. The proposed uses the CKKS scheme for privacy-preserving anomaly detection by adopting HE based on the *harmonic to arithmetic me*an (HM-AM) ratio. The HM-AM ratio involves various HE-incompatible operations. As a result, naive adoption of the HE results in inefficiency in regard to memory, communication cost and computational cost. The CKKS scheme optimizes both encoding and encryption procedures. In the CKKS scheme, the major issue is computational overheads in terms of efficiency and execution time. Adversarial assumptions for security threat model and security analysis in the related work [9] are as follows:

The adversarial assumption for the security threat model: An adversary can give a falsified reading as input to the smart meter but cannot modify the program within the smart meter.

Security analysis: CKKS encryption scheme (HE scheme) is used to ensure no information about the underlying messages is revealed to the semi-honest stakeholder, and smart meter readings of individual customers are not revealed to anyone at any stage. The paper uses an anomaly detection ratio for anomaly-based attack detection.

Wen et al. related work [10] tackles the issue of energy theft detection in smart grids. In this paper, a novel privacy-preserving federated learning framework, FedDetect is used for energy theft detection. A local differential privacy (LDP) scheme has been used to preserve the privacy of local consumers' data. A deep learning model called the temporal convolutional network (TCN) has been used for detecting energy thefts in smart grids.

Keshk et al. [11] addressed the security and privacy issues in smart power networks. A privacy-preserving framework to protect data and find anomalous behaviour in smart power networks has been introduced. To achieve privacy and security, a privacy module that consists of two levels and an anomaly detection module is proposed. This scheme uses blockchain technology to verify the integrity of the data and deep learning technique to perform anomaly detection.

To summarize, the aforementioned privacy-preserving techniques focus on protecting the security and privacy of the consumers' data. But the computational overhead depending on the mentioned privacy-preserving schemes is still an issue. Hence, in the proposed scheme, ECC based HE scheme for privacy-preserving data falsification detection in smart grids is adopted to reduce the computational overhead of execution time.

3. Preliminaries

In this section, the fundamental concepts required for the proposed scheme are explained. This section contains preliminaries of ECC, Elliptic Curve Discrete Logarithm Problem (ECDLP), bilinear pairing and anomaly detection ratio metric.

3.1 Elliptic curves and the Elgamal encryption

Elliptic curve cryptography is based on the properties of algebraic curves over fields [16]. To keep the comprehensiveness of this thesis, the Elliptic curves and Elgamal encryption is described by quoting the explanation by Deepak et al. [16].

Mathematically, an elliptic curve is represented by an equation of the form:

$$y^2 = x^3 + ax + b$$

with a constraint that the determinant $\Delta = -16(4a^3+27b^2)$ is non-zero. The security of elliptic curve cryptography is based on the ECDLP [17]. In other words, given two points A and B on the curve such that one is a scalar multiple of the other, i.e. $A = k.B$ (here '.' (dot) represents scalar multiplication), it is computationally difficult to find k.

The Elgamal encryption scheme with additive homomorphism can be implemented using elliptic curve cryptography as follows:

Key Generation:

- Choose a base point P of order N on an elliptic curve E over a finite field.
- Choose f: $x \rightarrow P_x$, which converts plaintexts x into points P_x on E. To realize the properties of additive homomorphism, the function f is defined as $f(x) = x.P$, where '.' (dot) represents the scalar multiplication of the point P with x
- Select a random secret key $k \in Z_N$. The points P and $Y = k.P$ are published as the public key.

Encryption:

- Choose a random number $a \in Z_N$. Calculate $P_x = f(x)$, where x is the plain text to be encrypted.
- The ciphertext is the pair of points $(a.P, a.Y + P_x)$

Decryption:

- From the received ciphertext (B_1, B_2), calculate $B_1' = k.B_1$ using the private key k.
- Compute $P_x = B_2 - B_1'$ and retrieve the original plaintext x as $f^{-1}(P_x)$

Additive Homomorphism:

- Consider two ciphertexts $c = (c_1, c_2)$, $d = (d_1, d_2)$, where c and d are the encryptions of messages x and y respectively under the same key k. For random a and b,

let $c = (a.P, a.Y + x.P)$, $d = (b.P, b.Y + y.P)$

- Compute the new ciphertext $e = c + d = ((a + b).P, (a + b).Y + (x + y).P)$ corresponding to the encryption of the message $(x + y)$ under key k [16].

3.2 Elliptic curve discrete logarithm problem (ECDLP)

The ECDLP [17], 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 potential to compute an elliptic curve scalar multiplication and the inability to compute the multiplicand given the original and product points are both required for elliptic curve encryption to be secure. 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 which is 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 the integer z, where z is in the range, $0 \leq z \leq n - 1$. 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.

3.3 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 [18].

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.

3.4 Anomaly Detection Ratio Metric

The harmonic to arithmetic mean (HM-AM) ratio has subsequently proven to be an efficient standard for identifying anomalous behavior in smart grid data [9]. Hence, the HM-AM ratio has been used for anomaly-based attack detection in the proposed scheme because i) it deals with numerous attacks such as 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 neighbourhood area network and each timeslot is denoted by 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 the inverse of the natural logarithm transformation ($P_t'^{(i)}$) 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 by

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

where,

$$AM_t = \frac{\sum_{i=1}^{N}(P_t^{(i)})}{N} \quad \ldots\ldots (2)$$

$$HM_t = \frac{N}{\sum_{i=1}^{N}(\frac{1}{P_t^{(i)}})} \quad \ldots\ldots (3)$$

3.5 Pseudo code for the CKKS scheme [9]

The following are the pseudo code for the CKKS schemes proposed by Ishimaki et al. [9]

Algorithm 1 Homomorphic Evaluation of the Daily Ratios [9]

Input:

- Encrypted log power consumption in an area, $\left\{\text{Enc}\left(P_t^{(i)}\right)\right\}_{i\in[N],t\in T}$
- Encrypted inverse log power consumption in an area, $\left\{\text{Enc}\left(P_t'^{(i)}\right)\right\}_{i\in[N],t\in T}$

Output: Encrypted HM-AM ratio

1: **HM** ← 0, **AM** ← 0

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

3: $\text{fracsum}_t \leftarrow 0$, $\text{sum}_t \leftarrow 0$

4: **for** i←1 to N **do**

5: $\text{sum}_t \leftarrow \text{sum}_t \boxplus \text{Enc}\left(P_t^{(i)}\right)$

6: $\text{fracsum}_t \leftarrow \text{fracsum}_t \boxplus \text{Enc}\left(P_t'^{(i)}\right)$

7: **end for**

8: HM ← HM ⊞ (Inv(fracsum_t) ⊡ N)

9: AM ← AM ⊞ (sum_t ⊡ $\frac{1}{N}$)

10: **end for**

11: return HM ⊡ Inv (AM)

Algorithm 2 Privacy-preserving Anomaly Detection Protocol [9]

1) **Data Transmission:** In a year *y* on the *d-th* date at each timeslot *t* in an area, the *i-th* smart meter SM_i does the following:

- Obtain the power consumption $p_{y,d,t}^{(i)}$
- Compute $P_{y,d,t}^{(i)} := \ln(p_{y,d,t}^{(i)} + 2)$ and $P_{y,d,t}^{\prime(i)} := \frac{1}{P_{y,d,t}^{(i)}}$
- Encrypt $p_{y,d,t}^{(i)}$, $P_{y,d,t}^{(i)}$ and $P_{y,d,t}^{\prime(i)}$ using preinstalled ***pk***
- Send Enc($p_{y,d,t}^{(i)}$), Enc($P_{y,d,t}^{(i)}$) and Enc($P_{y,d,t}^{\prime(i)}$) to the computational server through a data collector in NAN and other higher-level network

2) **Homomorphic Evaluation of HM-AM Ratio:** Upon receiving the ciphertexts, the computational server does the following:

- Call Algorithm 1 and obtain Enc(Q_d) if the daily amount of ciphertexts is available (Enc($p_{y,d,t}^{(i)}$)) is used for other tasks as discussed
- Send Enc (Q_d) to the utility

3) **Anomaly Detection:** The utility does the followings:

- Decrypt Enc(Q_d) using ***sk***, and locally save Q_d
- Perform training if a sufficient amount of Q_d's is available
- Perform test if the training phase has been performed, and obtain a decision bit

4. System Goal

This section provides the system goal, system architecture, and threat model for the proposed system. In the proposed scheme, the same architecture, and the threat model as in the paper by Yu Ishimaki *et al*. [9] are used.

The goal of our proposed scheme is as follows:

- Perform anomaly-based attack detection in a secured manner without disclosing the consumers' power usage details to the server and the utility.

- Verification of the validity of the encrypted data to ensure that the consumers' data is not manipulated to pass through the anomaly-based detection process.

4.1 System Architecture

The system architecture of the proposed system is the same as that in the paper by Yu Ishimaki *et al*. [9]. The system architecture shown in Figure 1 consists of three main components: the utility, a computational server (operated by a third party) and N smart meters.

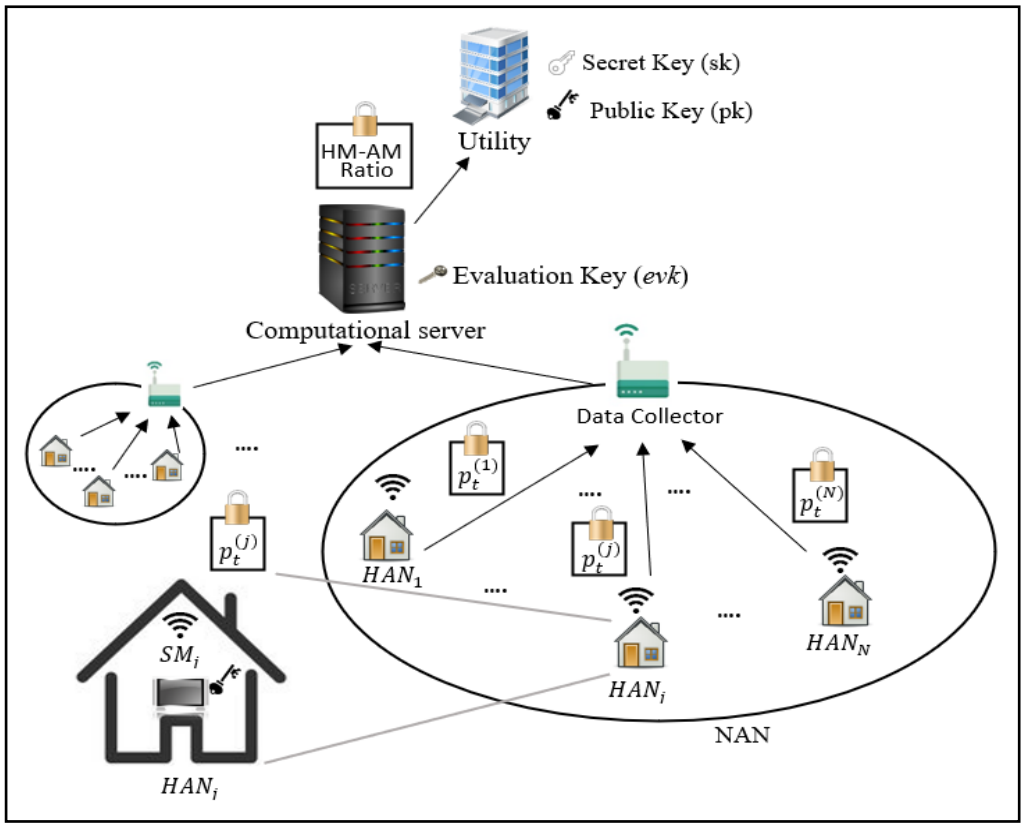

Figure 1: System overview
(Traced from Figure 1 in [9])

The function of each component is as follows:

Utility:

- The utility performs the system initialization step of key generation.
- 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 with itself.
- The utility computes the HM-AM ratio (Equation 1) and performs the anomaly detection ratio.

Smart meters:

- The smart meters use the public key provided by the utility to encrypt the smart meter readings and send the encrypted reading to the computational server.

Computational server:

- The computational server receives the encrypted reading from the smart meters and evaluates the encrypted data to compute the summation of AM and HM.
- The computed summation of AM (Equation 2) and HM (Equation 3) is then sent to the utility to perform anomaly detection.

The details of the scheme are provided in Section 5.

4.2 Threat Model

The proposed system attempts to protect the consumers' private data from both the computational server and the utility. It is assumed that the utility, computational server, and smart meters are semi-honest (honest but curious) [9]. They obey the protocol but as they are curious, 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. Moreover, if the server and a subset of smart meters collude, only the meters reading of those smart meters are revealed, whereas readings from other smart meters are protected.

Data integrity threat where an adversary attacks the smart meters to falsify the meter readings is assumed to occur before the encryption of power consumption data by the smart meters. Thus, the privacy requirement can be ensured if the assumptions such as the smart meters are honest but curious, the utility and the server do not collude and even when the server and the subset of smart meters collude, only readings of the colluded subset of smart meters are revealed are satisfied.

5. Proposed Scheme

In this section, a new privacy-preserving data falsification detection scheme is proposed. An ECC based HE scheme to encrypt data and perform HM-AM ratio over encrypted data is proposed. ECC based encryption is additively homomorphic. Moreover, the bilinear pairing function over the elliptic curve group allows us to check the validity of the encrypted data without decrypting it.

The four phases of the proposed scheme are system initialization (performed by the utility), meter report generation (performed by smart meters), HM-AM computation over encrypted data (performed by the computational server), and anomaly attack detection (performed by the utility).

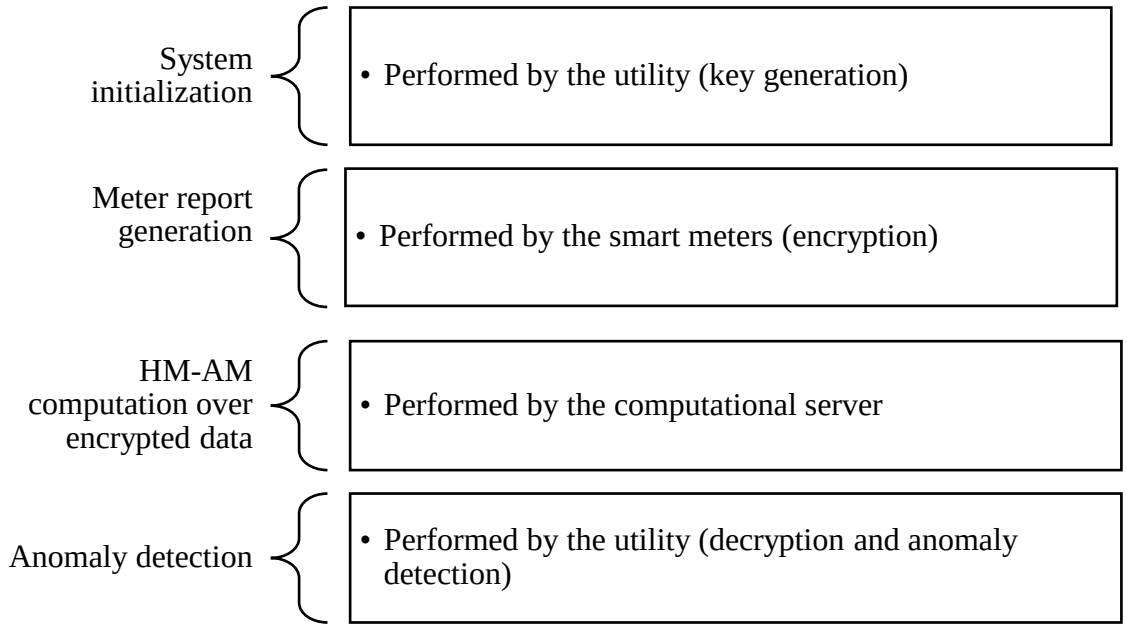

Figure 2: Proposed Scheme

Table 1: Variable and Description

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	Each timeslot
r_t	The random value generated for timeslot t
$p_t^{(i)}$	Power consumption of i^{th} smart meter ($1 \leq i \leq$ N) for timeslot t
$P_t^{(i)}$	Natural log transformation of $p_t^{(i)}$ ($log(p_t^{(i)} + 2)$) for i^{th} smart meter
$P_t'^{(i)}$	The inverse of natural log transformation $1/P_t^{(i)}$ for i^{th} smart meter
SHA-256	A cryptographic hash function that outputs a 256 bits 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 the day

5.1 System initialization

The initial setup is performed by the utility. It selects appropriate elliptic curve group G_1 and two independent points on the elliptic curve P, Q $\in G_1$ of order n.
After this setup, the utility generates two keys key_1 and key_2 which will be shared between the smart meters and the utility only. Both the keys will be kept secret from the computation server.

5.2 Meter report generation

Note that HM-AM ratio involves log ($p_t^{(i)}$+ 2) and its inverses which 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 always results in decimal places, and we can perform encryption over integers only. Therefore, we first round both $P_t^{(i)}$ and $P_t'^{(i)}$ up to certain decimal places and then remove decimal points to make it an integer.

Now to encrypt $P_t^{(i)}$, the i^{th} smart meter first computes r_t using the shared key key_1 and corresponding timeslot t.

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

We need to encode $P_t^{(i)}$ into 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 $(P_t^{(i)} - 1)r_t$P where r_t is a random value generated for timeslot t. The corresponding encryption of $P_t^{(i)}$ is

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

Note that for a timeslot t, each smart meter uses the same random value r_t as each has access to the shared key key_1 and without the key key_1 others cannot learn about r_t.

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

$$\text{Enc}(m') = (r_t'Q, m'r_t'Q) \quad \ldots..(6)$$

where,

$$r_t' = \text{SHA256}(\text{key}_2 || t) \quad \ldots\ldots.(7)$$

5.3 HM-AM computation over encrypted data

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

$$\hat{e}(r_tP, r_t'Q) = \hat{e}(mr_tP, m'r_t'Q) \quad \ldots..(8)$$

where, $m = P_t^{(i)}$ and $m' = P_t'^{(i)}$

Note that since $m * m' = 1$, we have

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

If the above equation does not hold for any readings, then it means that the data had been manipulated. After this, for each timeslot *t*, it computes

$$AM_{sum}^t = \sum_{i=1}^{N} \text{Enc}(P_t^{(i)}) \quad \ldots\ldots.(10)$$

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

Here N is the total number of smart meters.

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

5.4 Anomaly Detection by the Utility

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

$$AM_t = \frac{Dec(AM_{sum}^{t})}{N} \quad \ldots. (12) \qquad HM_t = \frac{N}{Dec(HM_{sum}^{t})} \quad \ldots. (13)$$

Here, N is the total number of smart meters.

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

$$Q_d = \frac{\sum_{t\in T} HM_t}{\sum_{t\in T} AM_t} \quad \ldots.. (14)$$

5.5 Security Analysis

The proposed system ensures the security of the consumers' data from both the computational server and the utility.

- The proposed system uses ECC based El-Gamal system, and its security depends on the discrete logarithm (DL) problem 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 will be visible for each time slot (that is summations of each time slot will be visible) to the utility.

- However, this does not leak individual readings of the meter and the utility cannot find the reading of any meter from either AM_t or HM_t

Hence, the proposed system does not leak any private data of the consumers either to the server or to the utility.

5.6 Pseudo code for the proposed scheme

The pseudo codes for the proposed scheme are shown as follows.

Algorithm 1 (Smart meter encryption)

Input:

- Data at time t, $p_t^{(i)}$,
- Elgamal public key, pk

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

1: $c_1 = Enc_{pk}\ (p_t^{(i)})$

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

Algorithm 2 (Server Computation for each timeslot)

Input:

- Encrypted log power consumption in an area, $\left\{Enc\left(P_t^{(i)}\right)\right\}_{1\le i\le N}$
- Encrypted inverse log power consumption in an area, $\left\{Enc\left(P_t'^{(i)}\right)\right\}_{1\le i\le N}$

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

1: **fracsum$_t$** ← 0, **sum$_t$** ← 0

2: **for** i ← 1 to N **do**

3: **sum$_t$** ← **sum$_t$** $\boxplus\ Enc\left(P_t^{(i)}\right)$

4: **fracsum$_t$** ← **fracsum$_t$** $\boxplus\ Enc\left(P_t'^{(i)}\right)$

5: **end for**

6: return **sum$_t$, fracsum$_t$**

Algorithm 3 (Utility computation)

Input:

- Encrypted sum of log power consumption in an area, $\{\boldsymbol{sum_t}\}_{t\in T}$
- Encrypted sum of inverse log power consumption in an area, $\{\boldsymbol{fracsum_t}\}_{t\in T}$

Output: HM-AM ration (Q_d)

1: HM ← 0, AM ← 0

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

3: AM ← AM + $\frac{Dec(sum_t)}{N}$

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

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

6. Evaluation

6.1 Experimental setup

For all the experiments, a computer with the specifications mentioned in Table 2 is used.

Table 2: Experimental setup

Operating system and version	Windows 10
Processor	11th Gen Intel(R) Core (TM) i5 (2.4 GHz)
RAM	8 GB
L1 cache size	320 KB
L2 cache size	5.0 MB
L3 cache size	8.0 MB
Compiler and version	SageMath 9.2
Programming language	Python 3.8.10

The proposed system is 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 are the recommended parameters to achieve 128-bit security for bilinear pairing and ECC based encryption [19].

For CKKS based method, the HEAAN Python library[2]is used. It is Python binding for a HEAAN Library, an open-source implementation of the CKKS scheme in C++.
To implement the CKKS method on the same platform, a Python wrapper[3] for the HEAAN C++ library[4] is used. Through the Python wrapper, we can import HEAAN functions in Python. For division over ciphertext operation, an inbuilt cipher inverse function of the HEAAN library is used. Parameter set as (n, log Q, p) = (2^{15}, 491, 35), in which a fresh ciphertext size is calculated as 2n log Q bits is used. As $2^{15} = 32{,}768$, it supports 128-bit security [20]. HEAAN related libraries are listed in Table 3.

The same dataset used in the paper Ishimaki et al. [9] is a smart grid dataset collected from the Pecan Street Project, which consists of a dataset from 200 households in Texas, USA over three years (2014–2016) is used for implementation.

[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 3: HEAAN related libraries

Library	Version
HEAAN	HEAAN-2.1
GMP	gmp-6.1.2
NTL	ntl-11.4.3
pybind	pybind11

Table 4: Differences in implementation of proposed scheme and CKKS scheme

	Proposed Scheme	CKKS scheme
Library	NumPy	HEAAN C++
Tool	Python binding for C++ libraries	Python based SageMath
Functionality	Does not support inverse operation over ciphertext	Supports inverse operation over ciphertext
Linking	Supports linking through bilinear pairing	Does not support linking
HM-AM ratio	Computed by the utility	Computed by the server

6.2 Computation and communication performance results

There are six important points in the evaluation part: comparison of user side encryption, comparison of server-side computation, comparison of utility side decryption, comparison of total execution time, comparison of ciphertext size for anomaly detection and maximum memory usage.

6.2.1 Comparison of user-side computation

Each smart meter performs three encryptions per timeslot Enc ($p_t^{(i)}$), Enc ($P_t^{(i)}$) and Enc ($P_t^{\prime(i)}$) in both CKKS and ECC based HE schemes. As shown in Table 5 the proposed scheme (ECC based HE scheme) performs 10 times better than the CKKS scheme for user-side encryption. In the CKKS method, to speed up the encryption time, the pre-computation of Enc (0) is performed and the optimized encryption is performed by adding power consumption data to Enc (0).

Table 5: Runtime comparison of user-side encryption in seconds

Scheme	Runtime of user-side encryption (in seconds)
ECC based HE	0.148
CKKS	2.112
Pre-computation (Enc (0))	0.984
Optimized Encryption (Enc (0) + m)	0.016

6.2.2 Comparison of server-side computation

In the CKKS scheme, the server computes the HM-AM ratio as the inverse function in the HEAAN library supports the computation of inverse operation over encrypted data. Whereas in ECC based HE scheme, division operation over encrypted data is not possible. Therefore, the server just calculates the summation of HM and AM and sends it to the utility to compute the HM-AM ratio. In the proposed scheme, we also perform bilinear pairing on the server-side to check if the individual encryptions Enc $(P_t^{(i)})$ and Enc $(P_t^{\prime(i)})$ are related to each other. In the CKKS scheme, in the last time slot, the HM-AM ratio is calculated. As shown in Table 6, the proposed scheme performs better both with and without bilinear pairing operation on the server-side. The proposed scheme performs 100 times better than the CKKS scheme for server-side computation.

Table 6: Runtime for server-side computation (in seconds) for the ECC based HE scheme

ECC based HE scheme	Runtime (in seconds)
Server computation (without bilinear pairing)	0.051
Server computation (with bilinear pairing)	112.457

Table 7: Runtime for server-side computation (in seconds) for the CKKS scheme

CKKS scheme	Runtime (in seconds)
Server computation (per time slot)	63.962
Server computation (for last time slot)	127.628

6.2.3 Comparison of utility-side decryption

In the CKKS scheme, the utility decrypts the HM- AM ratio (Q_d) to perform anomaly detection. Whereas in the proposed scheme, the utility computes the HM-AM ratio and performs decryption operation as well. Hence, as shown in Table 8, the CKKS scheme performs better for utility side computation.

Table 8: Runtime comparison of utility-side decryption in seconds

Scheme	Runtime of utility-side decryption (in seconds)
ECC based HE	10.377
CKKS	0.273

6.2.4 Comparison of total execution time

The total execution time of the proposed scheme as shown in Table 9, both with and without bilinear pairing is much better compared to the CKKS scheme.

Table 9: Total execution time (in seconds) for the ECC based HE scheme

ECC based HE scheme	Runtime (in seconds)
Total execution time (without bilinear pairing)	10.576
Total execution time (with bilinear pairing)	122.982

Table 10: Total execution time (in seconds) for the CKKS scheme

CKKS scheme	Runtime (in seconds)
Total execution time (without optimized encryption)	196.087
Total execution time (without optimized encryption)	191.879

6.2.5 Comparison of ciphertext size for anomaly detection

Table 11 shows the ciphertext size for the proposed scheme and the CKKS scheme. The ciphertext in both the schemes is first transferred from a household to the server and then from the server to the utility. As shown in Table 11, the ciphertext size for the proposed scheme is much smaller compared to the ciphertext size of the CKKS scheme. Hence, the cost of ciphertext size for the proposed scheme is less compared to the CKKS scheme.

Table 11: Comparison of ciphertext size for anomaly detection

Scheme	Household ⟶ Server	Server ⟶ Utility
ECC based HE	$3 \cdot 2 \cdot 2 \cdot 464$ bit = 0.696 KB	$24 \cdot 2 \cdot 2 \cdot 2 \cdot 464$ bit = 11.136 KB
CKKS	$3 \cdot 2 \cdot 2^{15} \cdot 491$ bit = 11,784 KB	$2 \cdot 2^{15} \cdot 36$ bit = 288 KB

6.2.6 Comparison of maximum memory usage

Table 12 shows the comparison of maximum memory usage of the proposed scheme and the CKKS scheme. As shown in the table, the proposed scheme performs better than the CKKS scheme in terms of maximum memory usage.

Table 12: Comparison of maximum memory usage

Scheme	Maximum Memory Usage (in MB)
ECC based HE	2.088 MB
CKKS	33.783 MB

6.3 Summary of evaluation

To summarize, the proposed scheme performs better than the CKKS scheme in terms of encryption time, server-side computation, total execution time, the cost of ciphertext size and the maximum memory usage. The proposed scheme is as secure as the CKKS scheme and ensures that the consumers' private data is protected from the server and the utility.

7. Conclusion

In this thesis, I have proposed an elliptic curve-based privacy-preserving anomaly detection scheme for data falsification in smart metering. I have compared our proposed scheme with the existing CKKS based method by implementing both the schemes on a similar platform and system. The proposed scheme is computationally efficient compared to the CKKS method for user side computation and server-side computation as well. For our proposed scheme, the user side computation is 10 times faster and the server-side computation is more than 100 times faster.

References

[1] NIST, Smart grid: A beginner's guide, http://www.nist.gov/smartgrid/beginnersguide.cfm

[2] Asghar, Muhammad Rizwan, Daniele Miorandi, "A holistic view of security and privacy issues in smart grids," Proc. of International Workshop on Smart Grid Security, Springer, Berlin, Heidelberg, LNCS, vol. 7823 pp. 58-71, 2012.

[3] H. Y. Lam, G. S. K. Fung and W. K. Lee, "A Novel Method to Construct Taxonomy Electrical Appliances Based on Load Signaturesof," Proc. of IEEE Transactions on Consumer Electronics, vol. 53, no. 2, pp. 653-660, May 2007, doi: 10.1109/TCE.2007.381742.

[4] N. Batra, Jack Kelly, Oliver Parson, Haimonti Dutta, Willian Knottenbelt, Alex Rodgers, Amarjeet Singh, Mani Srivastava, "NILMTK: An opensource toolkit for nonintrusive load monitoring," Proc. of ACM International Conference Future Energy Syst. (e-Energy), Cambridge, U.K., pp. 265–276, 2014.

[5] G. W. Hart, "Nonintrusive appliance load monitoring," Proc. of the IEEE, vol. 80, no. 12, pp. 1870-1891, Dec. 1992, doi: 10.1109/5.192069.

[6] G. Wood, M. Newborough, "Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design," Energy and Buildings, vol. 35, Issue 8, pp. 821-841, 2003 ISSN 0378-7788, https://doi.org/10.1016/S0378-7788(02)00241-4

[7] G. Kalogridis, R. Cepeda, S. Z. Denic, T. Lewis and C. Efthymiou, "ElecPrivacy: Evaluating the Privacy Protection of Electricity Management Algorithms," Proc. of IEEE Transactions on Smart Grid, vol. 2, no. 4, pp. 750-758, Dec. 2011, doi: 10.1109/TSG.2011.2160975.

[8] M. R. Asghar, G. Dán, D. Miorandi and I. Chlamtac, "Smart Meter Data Privacy: A Survey," Proc. of IEEE Communications Surveys & Tutorials, vol. 19, no. 4, pp. 2820-2835, Fourthquarter 2017, doi: 10.1109/COMST.2017.2720195.

[9] 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 International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm), pp. 1-6, 2020 doi: 10.1109/SmartGridComm47815.2020.9303009.

[10] 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," Proc. of IEEE Internet of Things Journal, pp. 1-13, 2021, doi: 10.1109/JIOT.2021.3110784.

[11] M. Keshk, Benjamin Turnbull, Nour Moustafa, Dinusha Vatsalan, Kim-Kwang Raymond Choo, "A privacy-preserving-framework-based blockchain and deep learning for protecting smart power networks," Proc. of IEEE Transactions on Industrial Informatics 16, no. 8, pp. 5110-5118, 2019.

[12] 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. ACM, pp. 75–80, 2013.

[13] M. A. Mustafa, S. Cleemput, A. Aly and A. Abidin, "A Secure and Privacy-Preserving Protocol for Smart Metering Operational Data Collection," Proc. of IEEE Transactions on Smart Grid, vol. 10, no. 6, pp. 6481-6490, Nov. 2019, doi: 10.1109/TSG.2019.2906016.

[14] F. Li, B. Luo and P. Liu, "Secure Information Aggregation for Smart Grids Using Homomorphic Encryption," Proc. of First IEEE International Conference on Smart Grid Communications, pp. 327-332, 2010 doi: 10.1109/SMARTGRID.2010.5622064.

[15] X. Liang, X. Li, R. Lu, X. Lin and X. Shen, "UDP: Usage-Based Dynamic Pricing with Privacy Preservation for Smart Grid," Proc. of IEEE Transactions on Smart Grid, vol. 4, no. 1, pp. 141-150, March 2013, doi: 10.1109/TSG.2012.2228240.

[16] K. Deepak and K. Chandrasekaran, "Investigating Elliptic Curve Cryptography for Securing Smart Grid Environments," Proc. of Third ISEA Conference on Security and Privacy (ISEA-ISAP), pp. 1-7, 2020, doi: 10.1109/ISEA-ISAP49340.2020.234993.

[17] D. Hankerson, Alfred Menezes, "Elliptic Curve Discrete Logarithm Problem," pp. 397-400, 2011.

[18] G. Patil, Varsha Galande, Vedant Kekan, Kalpana Dange, "International Journal of Innovative Research in Computer and Communication Engineering," Sentiment analysis using support vector machine 2, no. 1, pp: 2607-2612, 2014.

[19] R. Barbulescu, S. Duquesne, "Updating Key Size Estimations for Pairings," Journal of Cryptology, vol. 32, pp.1298–1336, 2018, https://doi-org.ez.wul.waseda.ac.jp/10.1007/s00145-018-9280-5

[20] Melissa Chase, Hao Chen, Jintai Ding, Shafi Goldwasser, Sergey Gorbunov, Jeffrey Hoffstein, Kristin Lauter, Satya Lokam, Dustin Moody, Travis Morrison, Amit Sahai, Vinod Vaikuntanathan, "Security of Homomorphic Encryption," HomomorphicEncryption. org, Redmond WA, Tech. Rep, 2017.