FESDA: Fog-Enabled Secure Data Aggregation in Smart Grid IoT Network

Ahsan Saleem, Abid Khan, Saif Ur Rehman Malik, Haris Pervaiz, Hassan Malik, Masoom Alam and Anish Jindal

***Abstract*—With advances in Fog and edge computing, various problems such as data processing for large Internet of things (IoT) systems can be solved in an efficient manner. One such problem for the next generation smart grid IoT system comprising of millions of smart devices is the data aggregation problem. Traditional data aggregation schemes for smart grids incur high computation and communication costs, and in recent years there have been efforts to leverage fog computing with smart grids to overcome these limitations. In this paper, a new fog-enabled privacy-preserving data aggregation scheme (FESDA) is proposed. Unlike existing schemes, the proposed scheme is resilient to false data injection attacks by filtering out the inserted values from external attackers. To achieve privacy, a modified version of Paillier crypto-system is used to encrypt consumption data of the smart meter users. In addition, FESDA is fault-tolerant, which means, the collection of data from other devices will not be affected even if some of the smart meters malfunction. We evaluate its performance along with three other competing schemes in terms of aggregation, decryption and communication costs. The findings demonstrate that FESDA reduces the communication cost by 50%, when compared with the PPFA aggregation scheme.**

***Index Terms*—Smart grid, fog computing, aggregation, privacy, authentication, fault-tolerance.**

I. INTRODUCTION

INTERNET of things (IoT) has revolutionized various application domains by providing communication and computation capabilities at every node connected within the IoT network. One such next-generation network is the smart grid network which comprises of millions of smart appliances and can be perceived as one giant smart grid IoT network. Smart grid (SG) is no longer fiction, as a number of utility companies have replaced or implemented smart grid alongside existing power grid. This allows utility companies to significantly improve their power generation, transmission, distribution and control [1]. In addition, SG IoT network offers the utility company the capability to diagnose fault during generation, distribution, and transmission, which can help prevent power blackout, as well as the capability to forecast power demands, facilitate efficient billing process and smooth integration of distributed renewable power resources in the grid [2]. However, there are a number of potential security and privacy risks in SG [3]–[5]. Utility companies, for example, use metering data to perform data analytics to inform operational strategies such as power demand estimation and for real-time monitoring of end-point devices, as well as more effectively control and optimize power supply and distribution.

However, clearly such data analytics also give rise to privacy concerns [6], [7]. In addition, a malicious attacker could also seek to obtain users' consumption data and infer useful information from such data (e.g., profile a particular household in terms of their occupancy, living patterns, and economic status) [8]. In traditional data aggregation schemes for SG, a gateway is responsible for aggregating the SM's reading and sending it to the control center (CC). Because, sending individual smart meter (SM) reading may reveal an individual's privacy and therefore, data aggregation is a preferred choice. Typical SG IoT network architecture consists of a number of smart appliances connected to a SM, which records the consumption of each appliance and sends this data periodically to a gateway node called aggregator. The aggregator is a semi-honest entity, and to preserve user privacy, the SM readings are sent in encrypted form. The aggregator is responsible for aggregating the encrypted readings before sending such readings to a CC. This saves time and preserves privacy. In a SG IoT network architecture, smart meters (SMs) are the core components, which collect users consumption data and provide information about the electricity demands to the utility company. In recent times, a number of cloud-based SG architectures [9], [10] have been proposed, where the SMs transmit their consumption data to a cloud. The data can then be used for billing, predictive analytics to forecast power demands, and so on. However, for a large number of SMs, the transmission of such data incurs high latency at the cloud, and the cloud may not be capable of handling all these requests in a timely fashion. To mitigate such a limitation, we can leverage fog computing [11] by partially shifting the computational and storage capability of the cloud to the edge of the terminal devices. Fog computing has enabled the extension of cloud computing functions to the network edge by assisting the cloud and end users in terms of communication, computation and storage. In the cloud-fog based aggregation schemes [9], [10], [12], [13] the fog nodes (FNs) perform the data aggregation, which can efficiently reduce the computational and communication overhead at cloud. While, in traditional aggregation schemes, there exist an entity named 'aggregator' which is supposed to perform aggregation and provide the storage capability. However, the aggregator in traditional schemes is not a specific device and it can be any device in the network. By having a FNs to perform

Ahsan Saleem, Abid Khan (corresponding author) and Masoom Alam are with Department of Computer Science, COMSATS University Islamabad, Pakistan. Saif Ur Rehman Malik is with Cybernetica AS, Estonia. Haris Pervaiz and Anish Jindal are with School of Computing and Communications (SCC), Lancaster University, UK. Hassan Malik is with Thomas Johann Seebeck Department of Electronics, School of Information Technologies,Tallinn University of Technology, Estonia. E-mail: ahsansaleemr@gmail.com, {abidkhan, masoom.alam}@comsats.edu.pk, saif.rehmanmalik@cyber.ee, hassan.malik@taltech.ee and {h.b.pervaiz, a.jindal3}@lancaster.ac.uk.

the data aggregation, we can leverage the inherent capabilities of efficient communication, computation and storage provided by the fog computing paradigm as suggested by existing fog-enabled schemes for secure data aggregation [9], [10], [12], [13]. This leads to a fog-cloud interplay to optimally use the nodes deployed at fog layer for the purpose of providing data aggregation. In other words, FNs perform the aggregation of the readings from the connected SMs using homomorphic encryption, prior to forwarding the aggregated results to the cloud. Such fog-enabled data aggregation also resolves the latency problem, as well as supports privacy and security. In addition, such a deployment setup has the potential to avoid the bandwidth and latency challenges that exist in a cloud-based setup [9], [10]. In our proposed model, FN is responsible for the aggregation of users' consumption data and forwarding it to CC. FNs are *honest* but *curious* meaning, they follow the protocol but may try to extract the information from users consumption data. To avoid this, we have used Paillier cryptosystem to encrypt the consumption data, which allows FNs to aggregate the users data in encrypted form. The homomorphic property of Paillier cryptosystem enables us to perform addition on encrypted data without the need to decrypt it. This property can be used in secure data aggregation and therefore, it is essential for secure data aggregation. Moreover, FNs store the HMAC secret key for each SM, which enables the FNs to perform the data integrity and source authentication of the metering data.

A. Related Work

In this section, an overview of existing fog-enabled privacy preserving-data aggregation (e.g., [9], [10], [12]–[14] and traditional privacy-preserving data aggregation (e.g., [17-28]) schemes in SG is provided in detail. Existing fog-enabled data aggregation schemes in SG, such as those presented in [9], [10], have a number of limitations. Lyu *et al.* [9] proposed a privacy-preserving fog enabled data aggregation scheme (PPFA), which is based on One-time-pad (OTP) homomorphic encryption. Although, it is conjectured that OTP is unconditionally secure [15], however, OTP has a number of limitations. Firstly, it requires the key size to be as long as the message itself. Secondly, OTP requires a new key to be generated every time an encryption has to be performed. Thirdly, transporting or storing a large number of keys is a tedious task. The scheme proposed in [9] is fault-tolerant with lower computational cost, however, it requires an additional round of communication among the participants of the protocol (fog-node, trusted authority and the cloud); thus, further increasing the communication cost. Furthermore, the PPFA scheme does not provide any protection against false data injection (FDI) attacks. *Wang et al.* [10] proposed an anonymous fog enabled SG data aggregation scheme, which efficiently preserves the users' privacy and ensures data integrity and source authentication of metering data. In this scheme, authors have used Boneh *et al.* [16] pairing based signature scheme for data integrity and source authentication of users data. While, the Weil-pairing based signature scheme is computationally intensive in verifying the signatures. The authors have focused on achieving anonymity. Furthermore, revocation of malicious terminal devices and fog nodes can be achieved efficiently. Authors in [12] proposed an anonymous and privacy-preserving fog-enabled data aggregation that guarantee anonymity and authenticity of metering data by using pseudonym and pseudonym certificate. Liu *et al.* [13] presented a fog-enabled privacy preserving smart grid data aggregation which allows the service provider (SP) to launch various function queries on encrypted metering data.

On the other hand, the existing traditional data aggregation schemes are presented as follows. Lie *et al.* [17] proposed a privacy-preserving data aggregation scheme, which aggregates the metering data of users without need of trusted third party (TTP). Abdallah *et al.* [18] presented a quantum secured privacy-preserving metering data aggregation scheme. Authors have used lattice based cryptosystem which involves only simple arithmetic operations that reduces communication and computational cost effectively, and it is feasible for appliances have limited computational resources. Zeadally *et al.* [19] proposed a lightweight and efficient privacy preserving scheme which employs symmetric homomorphic encryption and ECDH-key exchange schemes. It is resilient against various session key attacks and it is feasible for resource constrained devices. Lu *et al.* [20] proposed a novel data aggregation scheme, which achieves differential privacy and fault tolerance for an arbitrary number of malfunctioning SMs. Similarly, the authors in [21], [22] have also proposed privacy-preserving data aggregation schemes for ensuring integrity of metering data and securing SG communication respectively. Grining *et al.* [23] presented a privacy-preserving data aggregation scheme, which is fault-tolerant for even against a massive number of malfunctioning nodes. Authors in [24], [25] have proposed fault-tolerant privacy-preserving data aggregation schemes which perform the aggregation of users' consumption data even if some of the SMs are malfunctioning. Many researchers have worked on privacy preservation in an aggregated data by collecting the data in multiple subsets [26]–[28].

However, these traditional aforementioned traditional data aggregation schemes do not leverage the capabilities of fog computing paradigm and as a result these schemes suffer from latency issues. Generally, in the smart grid IoT communication network, a CC requests for the consumption data every 15 minutes from the SM. This means that after every 15 minutes, a new key would have to be generated for the encryption and decryption of smart metering data. The creation, transportation, and storing of this new key is logistically challenging; thus, making OTP impractical for real life settings. To cater this problem, we have used Paillier homomorphic encryption, which efficiently preserves user privacy and also allows performing aggregation on encrypted data in this paper. To achieve data integrity and source authentication, we use hash-based message authentication codes (HMAC), which provides efficiency in terms of computational and communication cost. Furthermore, a time-stamp TS together with MAC is used in our scheme to prevent replay attacks. HMAC ensures the integrity and source authentication of metering data. Moreover, our scheme is fault-tolerant which means FNs can still aggregate users' consumption data and CC can decrypt the

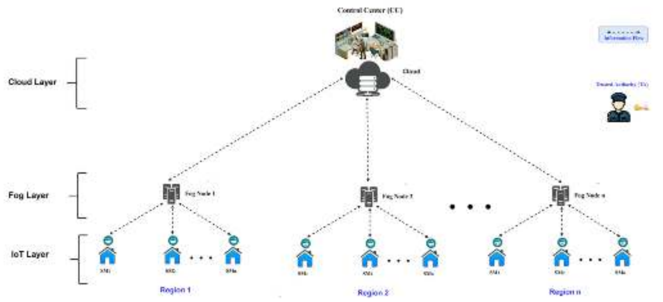

Figure 1: System model

aggregated data even if some of the SMs are malfunctioning. In contrast to the existing similar scheme presented in [9], our fault-tolerant approach does not require any extra round of communication between FN and the trusted authority (TA), which makes our scheme efficient in terms of communication and computation cost.

B. Contributions

The main contributions of our proposed FESDA scheme are summarized as follows:

- We propose an efficient fog-enabled privacy preservation data aggregation scheme in the SG IoT network. To encrypt the metering data, a modified version of Paillier cryptosystem is used and, no entity (except CC) can decrypt the aggregated consumption data of users. Moreover, individual privacy is protected against malicious CC.
- FESDA scheme is fault-tolerant, which means that the collection of data from other devices will not be affected even if 50% of the smart meters are malfunctioning.
- We demonstrate the effectiveness of the proposed scheme with performance and security analysis. The performance analysis shows that FESDA scheme is more efficient than the comparative schemes. To prove the proposed scheme is privacy-preserving and secured, we have performed comprehensive security and privacy analysis which shows that FESDA scheme is secured under the defined attacker model. Additionally, our scheme prevents the reply attack, FDI attack and, secured against malicious CC.

II. Preliminaries

This section provides an overview of the necessary background information on the important cryptographic primitives used in the paper.

A. Paillier Homomorphic Cryptosystem

Paillier cryptosystem [29] is a homomorphic cryptosystem which efficiently preserves the privacy of the data and also allows to perform computations on the encrypted data. It involves the following three algorithms:

- **Key Generation:** First, select two randomly large and independent prime numbers p and q. Then, compute $\lambda = lcm(p-1, q-1)$ and $N = p.q$ where λ is least common multiple of $p-1$ and $q-1$. Set a function $L(x) = \frac{x-1}{N}$ and select a random integer g where $g \in \mathbb{Z}^*_{N^2}$ and calculate $\mu = (L(g^\lambda \ mod \ N^2))^{-1} \ mod \ N$. Finally, get the public key (N, g) and the private key (λ, μ).
- **Encryption:** For given message $m \in \mathbb{Z}^*_N$, select a random number $r \in \mathbb{Z}^*_{N^2}$ and perform encryption using public key (N, g): $c = E(m) = g^m.r^N \ mod \ N^2$
- **Decryption:** For a given cipher c to be decrypted where $c \in \mathbb{Z}^*_{N^2}$, and get the plaintext m using private key (λ, μ): $m = (L(g^\lambda \ mod \ N^2)). \ \mu \ mod \ N$

In the proposed FESDA scheme, we have used Paillier Cryptosystem having key size of 1024 bits depending p and q values. Paillier supports three different sizes of keys 1024, 2048 and 4096 bits. However, as SMs are constrained devices and therefore, we have used a key size of 1024 bits. Although, larger key size such as 2048/4096 bits can be used for improved security, but these will slow down the system.

B. Message Authentication Codes (MAC)

MAC algorithm is a symmetric-key cryptographic technique used for message integrity and source authentication [30]. MAC ensures that received message is from the authenticated source and it is not tempered by any third party during transmission. The sender, Alice, first computes the MAC on the message m using shared secret key s_k. Afterwards, Alice sends the MAC-tag concatenated with the message to Bob. Bob first generates the MAC on the received message using same shared secret-key s_k and compares the generated MAC with the received MAC. If both MACs are same, then Bob keeps the received message m and knows that message is from authenticated source (Alice) and no third party has tempered with the message. In this way, MAC protects both integrity as well authenticity of the received message.

III. Models and goals

In this section, we define our system model, attacker model and identify security goals of our proposed scheme.

A. System Model

Fog computing extends the cloud computing and services to the edge of the network and acts as an intermediate layer between the Cloud data centers and the IoT devices. It offers computing networking, location awareness, and storage facilities so that cloud-based services can be extended closer to the IoT devices. We envision a three-tier network hierarchy framework for smart metering as depicted in Fig. 1. IoT devices are highly distributed devices which are located at the edge of the network along with real-time and latency sensitive service requirements. In this trend, we assume that there are numerous SMs attached to their respective FN in the bottom layer of the framework. The SMs collect consumption data from electric appliances, and forward the data to their respective FNs. The middle layer comprises of FNs, which are the computational and storage resources for the terminal devices. Since FNs has the computational capability, therefore, they can process or aggregate the incoming data from SMs, while cloud is assumed at the top layer. FNs are one hop away from network devices, thus the transmission of measurements from terminal devices to the FNs is more economical as compared to the transmission of all data from terminal devices to the cloud. Since the cloud is only concerned with the overall aggregation of all regions, therefore, FNs perform aggregation of users' data from their connected SMs and forward it to the cloud. Fog-cloud interplay overcomes the latency and bandwidth issues in data-intensive IoT. The system model of the proposed scheme consists of the following five entities, as depicted in Fig. 1.

- **Trusted authority (TA):** TA is the most reliable entity in the system, which generates the public and private keys to be used by the Paillier cryptosystem. Moreover, TA also generates additional secret parameters for each user as well for the CC. Afterwards, TA provides the secret parameter to all users and the CC. TA is no more required in the data aggregation process, once the key distribution is completed.
- **Smart Meters:** SMs are the devices which collect the consumption data of all the appliances on users premises. Moreover, SMs perform cryptographic operations for privacy preservation, data integrity and, source authentication. SMs provide their consumption data to their nearby FN.
- **Fog Node:** FN leverages the computational and storage capability of the cloud to the edge of the end devices. In our proposed model FN is responsible for the aggregation of users' consumption data and forwarding it to CC. FN also ensures data integrity and source authentication of the metering data.
- **Cloud:** Cloud first verifies the integrity and source authentication of the incoming aggregated data from the FNs and then stores the aggregated data.
- **Control center (CC):** CC has access to the cloud and, gets the aggregated data from FNs. CC performs decryption, using secret key and, gets the aggregated consumption data of the end users.

B. Attacker Model Assumptions

In the attacker model, we have assumed that FNs, CC and the cloud are *honest-but-curious*. More specifically, these entities will follow the protocol correctly, however, they may try to get private information of users. Moreover, the terminal devices (SMs) are tamper-resistant. Our attacker model consists of following assumptions:

1) FN and CC both are honest-but-curious. Meaning, they will follow the protocol, however, at the same time they are curious to know the values of the SMs readings.
2) Although, a user wants to have minimum electricity charges, however, we assumed that all the smart meter users are honest. In general scenarios, a user may temper with the SM, but this is not in our scope as we assumed the users to be honest.
3) An external adversary $\mathscr{A}$ may compromise the FN. The adversary goals may include knowing the aggregated and individual SM readings.
4) An external adversary $\mathscr{A}$ may compromise the CC. However, individual meter readings are not exposed.
5) An external attacker may launch a FDI attack. However, such injections of data will be detected at the FN and CC.

C. Security Goals

The FESDA scheme aims to achieve the following security goals:

- **Privacy:** The adversary $\mathscr{A}$ is not able to access the users' data, even if the attacker intercepts the communication data transmitted on the insecure channel. FN has no way to decrypt the user data, so FN cannot compromise the user's privacy. Moreover, CC performs decryption to get the aggregated data only, but has no way to get data of an individual SM. Therefore, the privacy of individual and aggregated SM reading should be preserved.
- **Resistance to false data injection attack:** In order to gain some monetary benefits, an adversary may attempt to inject some false values in the SM readings. Our proposed scheme efficiently filters out the false data injected from an external attacker. Any such modification attempts should be detected by the FN and CC. Therefore, injecting false or dummy values by an adversary should be detected.
- **Fault tolerance:** It is assumed that some of the SMs may be faulty for a period of time and may not provide their consumption readings to the FN. Our scheme is fault-tolerant which means that aggregation and decryption of the SM readings can be performed, even if some of the SMs are malfunctioning. Previous schemes have not considered the issues of fault tolerance, which results into high communication delays because then CC has to ask the TA for the missing values and as a result the data aggregation activity is halted. Our goal is to minimize this delay.
- **Integrity:** Integrity of the metering data is preserved and any modification from any illegitimate entity should be detected. FN and CC should detect If the meter's reading has been modified.
- **Authentication:** The FN and CC should check that the incoming data is from authorized source. Since, SM and the corresponding FN have the same shared secret key, therefore, FN can be ensured that incoming data is from an authenticated source. Likewise, CC verifies the source authentication of aggregated data from each FN. This is important to avoid any false values from malicious entities who may attempt to inject dummy values and to victimize an innocent user.

IV. Fog-Enabled Secure Data Aggregation

The proposed FESDA scheme consists of four algorithms: (i) Key generation, Encryption and MAC-tag generation (ii) Secure Aggregation and MAC verification at FN (iii) Decryption and MAC verification at CC (iv) Fault-Tolerant Aggregation and Fault-Tolerant decryption. The list of symbols used in proposed scheme along with their descriptions are shown in Table I.

Table I: List of Notations

Symbol	Definition
u_i	User
SM	Smart meter
FN	Fog node
x_i	Secret parameter of u_i
x_0	Secret parameter of CC
CC	Control center
TA	Trusted authority
$\mathscr{A}$	Adversary
p, q	Randomly generated large prime numbers
s_k	Secret key
(N,g)	Public key pair
$(\lambda,\ \mu)$	Private key pair
m	Message
r	Random number
TS	Time stamp
$H(T_s)$	Hash of T_s using SHA-256
c_i	Encrypted SM reading
MAC$_i$	MAC-tag generation on c_i at SM
MAC$_j$	MAC-tag generation on received c_i at FN
C_i	Aggregated value at n-th FN
$\widehat{C}$	Fault-tolerant aggregated cipher
$\widehat{U}$	Set of malfunctioning SMs
MAC$_{x_i}$	MAC-tag generated at n-th FN
MAC$_y$	MAC-tag generated at CC
M	Sum of all consumption data of all users

Algorithm 1 Key Generation, encryption and MAC-tag generation

```
1:  procedure
2:  Input: Large prime numbers $p$, $q$, Function $L(x) = \frac{x-1}{N}$, Message $m_i$, Hash $H(T_s)$, $N$
3:  Output: Public key $K_{pub} = (N, g)$, Private key $K_{pri} = (\lambda, \mu)$, Secret Parameters $x_0$, $x_i$, Cipher $c_i$, MAC-tag MAC$_i$
4:  TA chooses two random large and independent prime numbers $p$ and $q$
5:  Computes $\lambda = lcm(p-1, q-1)$ and $N = p.q$
6:  Set a function $L(x) = \frac{x-1}{N}$
7:  Select a random integer $g$ where $g \in \mathbb{Z}^*_{N^2}$
8:  Computes $\mu = (L(g^{\lambda}\ mod\ N^2))^{-1}\ mod\ N$
9:  Return $K_{pub} = (N, g)$ and $K_{pri} = (\lambda, \mu)$
10: Generates $x_i \in \mathbb{Z}^*_N$, $i = 1, 2, 3, ...., n$
11: Computes $x_0 \in \mathbb{Z}^*_N$ s.t. $x_0 + \sum_{i=1}^{n} x_i = 0\ mod\ \lambda$
12: For message $m_i \in \mathbb{Z}^*_N$ the user $u_i$ at each time interval computes
13:     for $(i = 1;\ i \le n;\ i++)$ do
14:         $c_i = E(m_i) = g^{m_i}.H(T_s)^{N.x_i}\ mod\ N^2$
15:         MAC$_i = s_k\ (H(c_i) \parallel TS)$
16:     end for
17: end procedure
```

The Algorithm 1 generates encryption and decryption keys for users and for CC respectively. Moreover, Algorithm 1 performs encryption of users consumption data and generates MAC-tag on each encrypted value. The Algorithm 1 works as follows, first, the TA selects two random large and independent prime numbers p and q and computes the public key (N, g) and private key (λ, μ) of Paillier cryptosystem. Afterwards, TA takes a pseudo-random number generator function to generate n random numbers $x_i \in \mathbb{Z}^*_N$, where $i = 1, 2, 3, ..., n$, and computes $x_0 \in \mathbb{Z}^*_N$ such that:

$$x_0 + \sum_{i=1}^{n} x_i = 0\ mod\ \lambda \tag{1}$$

Afterwards, TA forwards x_i and x_0 as additional secret parameter to each user u_i and to CC respectively, via a secure channel. Finally, TA computes $T_i = H(T_s)^{N\cdot\ x_i}$ and forward it to CC. After this initializing, TA has no further role in the aggregation process. Moreover, from lines 12 to 16 in Algorithm 1 show how each user u_i performs encryption and how it generates MAC-tag on c_i. SMs collect all the consumption data $m \in \mathbb{Z}^*_N$ from connected appliances, and encrypts it using secret key x_i.

$$c_i = E(m_i) = g^{m_i}.H(T_s)^{N.x_i}\ mod\ N^2 \tag{2}$$

The user u_i also generates MAC-tag of c_i using shared secret key s_k.

$$\text{MAC}_i = s_k\ (H(c_i) \parallel TS) \tag{3}$$

Where TS is the time-stamp, which enables the proposed scheme to prevent reply attack. Afterwards, each user u_i forwards $c_i \parallel$ MAC$_i$ to their respected FN. Algorithm 2 performs aggregation of users metering data and MAC verification at FN. The FN receive cipher c_i with MAC-tag MAC$_i$ from connected SMs. Then, FN creates it's own MAC$_j$ for every received cipher c_i, as shown in Eq. (4).

$$\text{MAC}_j = s_k\ (H(c_i) \parallel TS) \tag{4}$$

To verify the incoming data is from an authenticated source and it has not been tempered with while in transit, FN compares both MACs and checks whether MAC$_i$ == MAC$_j$. FN only accepts the cipher c_i, if MAC$_i$ == MAC$_j$ otherwise, FN discard that received cipher c_i. In this way, FN ensures that incoming metering data is from authenticated source and as well as filters-out the false data injected from external attacker, efficiently. In the same fashion, the FNs receives all ciphertexts c_i from connected SMs, and perform the aggregation of the received encrypted data, as depicted in Eqs. (5) and (6).

$$C = \prod_{i=1}^{n} c_i \tag{5}$$

$$C = g^{\sum_{i=1}^{n} m_i}.H(T_s)^{N\cdot \sum_{i=1}^{n} x_i}\ mod\ N^2 \tag{6}$$

Furthermore, after performing the aggregation, the FN creates MAC-tag on aggregated cipher C and forwards $C \parallel$ MAC$_x$ to the Cloud after embedding the current time stamp (TS), as shown in Eq. (4).

$$\text{MAC}_x = s_k\ (H(C) \parallel TS) \tag{7}$$

Algorithm 3 performs decryption and MAC verification at CC. The cloud verifies the integrity and source authentication of the incoming aggregated data. CC performs decryption and gets the all consumption data M of all users.

After receiving $C \parallel$ MAC$_x$, the Cloud generates it's own MAC$_y$.

$$\text{MAC}_y = s_k\ (H(C) \parallel TS) \tag{8}$$

Algorithm 2 Aggregation and MAC-verification at Fog Node (FN)

```
procedure
Input: Concatenation of $\text{MAC}_i \parallel c_i$
Output: Aggregated cipher $C$, Concatenation of $\text{MAC}_x \parallel C$
FN receives $\text{MAC}_i \parallel c_i$ from each user $u_i$
Generates its own $\text{MAC}_j = s_k\ (H(c_i) \parallel TS)$
    if $\text{MAC}_i == \text{MAC}_j$ then
        $\text{MAC}_i$ is verified and $c_i$ is accepted
    else
        $c_i$ is tempered so rejected
    end if
    for $(i = 1;\ i \leq n;\ i++)$ do
        $C = \prod_{i=1}^{n} c_i$
    end for
FN generates $\text{MAC}_x = s_k\ (H(C) \parallel TS)$
Forward $\text{MAC}_x \parallel C$ to CC
end procedure
```

Algorithm 3 Decryption and MAC verification at control centre (CC)

```
1: procedure
2: Input: Concatenation of $\text{MAC}_x \parallel C$
3: Output: Sum of consumption data $M$
4: CC receives $\text{MAC}_x \parallel C$ from FN
5: Generates $\text{MAC}_y = s_k\ (H(C) \parallel TS)$ and checks
6:     if $\text{MAC}_y == \text{MAC}_x$ then
7:         $\text{MAC}_x$ is verified and $C$ is accepted
8:     else
9:         $C$ is tempered so rejected
10:    end if
11: Performs decryption $V = C.\ H(T_s)^{N.x_0}$
12: CC gets $M = \sum_{i=1}^{n} m_i$
13: end procedure
```

Algorithm 4 Fault-tolerant aggregation and decryption

```
1: procedure
2: Input: Cipher $c_i$, Set of malfunctioning SMs $\widehat{U}$, Aggregated cipher $\widehat{C}$
3: Output: Aggregated cipher $\widehat{C}$, Sum of consumption data $M$
4: FN receives $c_i$ from $U_i \in U/\widehat{U}$ users
5:     for $(i = 1;\ i \leq U_i \in U/\widehat{U};\ \ i++)$ do
6:         $\widehat{C} = \prod_{U_i \in U/\widehat{U}} c_i$
7:     end for
8: CC receives $\widehat{U}$, $\widehat{C}$ from FN
9:     for $(i = 1;\ i \leq U_i \in \widehat{U};\ i++)$ do
10:        $\overline{C} = \prod_{U_i \in \widehat{U}} T_i$
11:    end for
12: CC calculates $C = \widehat{C}\ \overline{C}$
13: Performs decryption $V = C.\ H(T_s)^{N.x_0}$
14: CC get $M = \sum_{U_i \in U/\widehat{U}} m_i$
15: end procedure
```

The Cloud keeps the aggregated C only if MAC verification is successful otherwise discard it, as shown in lines $5 - 10$ of Algorithm 3. Thus, in this way data integrity and source authentication is achieved at CC. Furthermore, FESDA efficiently filters out the false data. CC has access to Cloud so it can get the aggregated C and performs the decryption process, which involves the following 2 steps:

Step1: CC uses it's secret key x_0 to compute:

$$V = C.\ H(T_s)^{N.x_0} \tag{9}$$

$$= g^{\sum_{i=1}^{n} m_i}.H(T_s)^{N.\sum_{i=1}^{n} x_i}.\ H(T_s)^{N.x_0}\ mod\ N^2$$

$$= g^{\sum_{i=1}^{n} m_i}.H(T_s)^{N.(x_0 + \sum_{i=1}^{n} x_i)}\ mod\ N^2$$

$$\because \quad x_0 + \textstyle\sum_{i=1}^{n} x_i\ =\ 0\ mod\ \lambda$$

Finally CC gets:

$$V = g^{\sum_{i=1}^{n} m_i}\ mod\ N^2 \tag{10}$$

Step 2: CC computes $\sum_{i=1}^{n} m_i$ using Pollard's Lambda method [31] and, gets the aggregated consumption data M of the users.

$$M = \sum_{i=1}^{n} m_i \tag{11}$$

A. Fault-Tolerant Aggregation and Decryption

In case, if some SMs are malfunctioning, FN would not be able to get consumption data from such SMs and ultimately, the CC would not be able to decrypt the aggregated data. It is necessary that CC can still perform the decryption even if some of the SMs are malfunctioning. Algorithm 4 shows the fault-tolerant aggregation and fault-tolerant decryption at FN and at cloud respectively. Let us assume that total number of users is U, and $\widehat{U}$ out of U number of users have failed to transmit their consumption data to FN, where $\widehat{U} \subset U$. Algorithm 4 from line 4 to 7, allows the FN to perform the aggregation on received reports and, calculate the fault-tolerant aggregated cipher $\widehat{C}$ as shown in Eqs. (12) and (13).

$$\widehat{C} = \prod_{U_i \in U/\widehat{U}} c_i \tag{12}$$

$$\widehat{C} = g^{\sum_{U_i \in U/\widehat{U}}\ m_i}.H(T_s)^{N.\sum_{U_i \in U/\widehat{U}}\ x_i}\ mod\ N^2 \tag{13}$$

Afterward FN forwards $\widehat{U}$ and $\widehat{C}$ to CC, and CC performs aggregation and gets $\overline{C}$.

$$\overline{C} = \prod_{U_i \in \widehat{U}} T_i \tag{14}$$

$$\overline{C} =\ H(T_s)^{N.\ \sum_{U_i \in \widehat{U}}\ \cdot\ x_i} \tag{15}$$

Finally, CC calculates aggregated cipher C.

$$C = \widehat{C}.\ \overline{C} \tag{16}$$

$$= g^{\sum_{U_i \in U/\widehat{U}}\ m_i}.H(T_s)^{N.\sum_{U_i \in U/\widehat{U}}\ x_i}\ H(T_s)^{N.\ \sum_{U_i \in \widehat{U}}\ \cdot\ x_i}$$

$$= g^{\sum_{U_i \in U/\widehat{U}}\ m_i}.H(T_s)^{N.\ (\sum_{U_i \in U}\ \cdot\ x_i)}$$

Moreover, Algorithm 4 from line 8 to 14, shows all the steps for fault-tolerant decryption. Likewise, as described above, CC performs decryption using its secret key x_0, which involves the following 2 steps:

Step1: CC uses it's secret key x_0 to perform decryption to obtain V.

$$V = C.\, H(T_s)^{N.x_0} \tag{17}$$

$$= g^{\sum_{U_i \in U/\widehat{U}}\ m_i}.H(T_s)^{N.\,(\sum_{U_i \in U}\,\cdot\, x_i)}.\, H(T_s)^{N.x_0}$$

$$= g^{\sum_{U_i \in U/\widehat{U}}\ m_i}.H(T_s)^{N.\,(x_0+\sum_{U_i \in U}\,\cdot\, x_i)}$$

$$\because \quad x_0 +\ \textstyle\sum_{i=1}^{n} x_i =\ 0\, mod\, \lambda$$

Finally, CC gets:

$$V = g^{\sum_{U_i \in U/\widehat{U}}\ m_i} \tag{18}$$

Step2: Similarly as described earlier, CC computes $\sum_{U_i \in U/\widehat{U}}\ m_i$ from g using Pollard's Lambda method [31] and, afterward computes the sum of all consumption data M.

$$M = \sum_{U_i \in U/\widehat{U}} m_i \tag{19}$$

Moreover, FN and subsequently the CC verifies the source authentication and data integrity using HMAC, in the same way as described earlier in Algorithms 1, 2 and 3, when no SM was malfunctioning.

V. Security and Privacy Analysis

The aim of this section is to show that the proposed scheme is privacy-preserving and secure under the defined attacker model. We evaluate our scheme on the basis of the privacy of users, integrity and source authentication of the metering data using five theorems. Theorems 1-4 are related to the privacy-preservation while, theorem 5 is related to protection of smart metering data against FDI and replay attacks.

Theorem 1. *An external adversary cannot compromise the privacy of individual SM reading.*

Proof. An adversary may $\mathscr{A}$ eavesdrops the communication between SMs to get the report c_i of user u_i. In FESDA, the SM reports its consumption data to FN in the form of $c_i = E(m_i) = g^{m_i}.H(T_s)^{N.x_i}\ mod\ N^2$. If we let the $r = H(T_s)^{x_i}$, then ciphertext expression will become $c_i = g^{m_i}.r^N\ mod\ N^2$. The resultant ciphertext is still the legal ciphertext of the Paillier cryptosystem. Since, Paillier is Indistinguishable under Chosen Plain-text Attack (IND-CPA), meaning even if an adversary $\mathscr{A}$ gets the c_i, he will not be able to recover the private data m_i of user u_i. Hence, the individual user's privacy is preserved. □

Theorem 2. *A colluding set of users cannot compromise the privacy of other users.*

Proof. If an adversary $\mathscr{A}$ compromises the users privacy, he can get the users private data and may reveal the private parameter x_i. In FESDA, TA randomly generates the private parameters $x_i \in \mathbb{Z}_N^*$, $i = 1, 2, 3, ..., n$ and there is no correlation between them. Therefore, compromising of private parameters of a few users (say $j \leq n-1$), will not reveal the secret parameters of remaining users.

Suppose an extreme situation occurs, where an adversary $\mathscr{A}$ succeeds in compromising $n-1$ users, and gets their corresponding secret parameters $x_1, x_2, ..., x_{n-1}$. Recalling from Eq. (1), the expression for n users can be expressed as follow:

$$x_0 +\ \sum_{i=1}^{n} x_i =\ 0\, mod\, \lambda$$

For $(n-1)$ users, Eq. (1) can be rewritten as:

$$x_0 +\ x_n + \sum_{i=1}^{n-1} x_i =\ 0\, mod\, \lambda \tag{20}$$

This means without having secret parameter (x_0) of CC and Paillier's secret key λ, an adversary $\mathscr{A}$ will not be able to compromise x_n. We can conclude that, no matter how many users are compromised, the adversary cannot disclose the private data of the other users. □

Theorem 3. *The privacy of c_i and C is preserved even if the FN is compromised.*

Proof. Compromising the FN, an adversary $\mathscr{A}$ can get the individual encrypted data, from Eq. (2) as follow:

$$c_i = g^{m_i}.H(T_s)^{N.x_i}\ mod\ N^2$$

Similarly, an adversary $\mathscr{A}$ can also get the encrypted aggregated data, from Eq. (6) as given below:

$$C = g^{\sum_{i=1}^{n} m_i}.H(T_s)^{N.\sum_{i=1}^{n} x_i}\ mod\ N^2$$

Paillier crypto-system is semantically secure under chosen plaintext attack, and cipher c_i reveals nothing about the plaintext. Since, FNs has no security parameter to perform decryption of metering data, meaning, the adversary $\mathscr{A}$ cannot compromise the privacy of individual as well as aggregated data C, by compromising the FN. Thus, even though an adversary has compromised the FN, still the privacy of individual as well as that of aggregated data is preserved. □

Theorem 4. *An individual privacy is preserved from a malicious CC.*

Proof. Since, CC only decrypts the aggregated consumption data of users, and he has no way of inferring the individual value of users from aggregated data. It means that the individual privacy of the users is preserved against the malicious CC. The CC receives the aggregated consumption data of users from FNs, and performs decryption using the Eqs. (9) and (11), to get the consumption data M of all users.

$$V = C.\, H(T_s)^{N.x_0}$$

$$V = g^{\sum_{i=1}^{n} m_i}.H(T_s)^{N.\sum_{i=1}^{n} x_i}.\, H(T_s)^{N.x_0}\ mod\ N^2$$

$$V = g^{\sum_{i=1}^{n} m_i}.H(T_s)^{N.(x_0+\sum_{i=1}^{n} x_i)}\ mod\ N^2$$

$$\because \quad x_0 + \textstyle\sum_{i=1}^{n} x_i\ =\ 0\, mod\, \lambda$$

$$V = g^{\sum_{i=1}^{n} m_i}\ mod\ N^2$$

Table II: Aggregation cost at Fog Nodes

Aggregation at FN 1		Aggregation at FN 2		Aggregation at FN 3		Aggregation at Cloud	
No. of SMs	Time (ms)	No. of SMs	Time (ms)	No. of SMs	Time (ms)	No. of SMs	Time (ms)
100	4	200	73	300	208	600	49
200	73	400	431	600	1108	1200	120
300	208	600	1108	900	2500	1800	213
400	431	800	1978	1200	4495	2400	310
500	690	1000	3071	1500	7432	3000	452

$$M = \sum_{i=1}^{n} m_i$$

Likewise, if some strong adversary $\mathscr{A}$ somehow compromises the CC, the adversary can get the aggregated value of consumption data M. Since, CC has no way to decrypt the individual user's consumption data, therefore, an adversary cannot infer the individual reading from the aggregated value. We can conclude that, even though the adversary has compromised the CC, however, he is still unable to compromise the individual privacy of the users. □

Theorem 5. *FN can detect false data values injected by external attacker.*

Proof. To authenticate the source of data for each time slot, HMAC has been used. Each user u_i creates MAC on the c_i with time stamp TS, and forward it to FN as given previously by Eq. (3)

$$\text{MAC}_i = \ s_k \ (H(c_i) \parallel \ TS)$$

While, the FN creates its own MAC_j on received c_i, and compares both MACs. If both MACs are equal, it shows that incoming data is from authenticated source, and data is not tampered with while in transit. For every time slot TS, the FN always receives a fresh MAC_i. If this MAC_i is not fresh at time slot TS, it indicates that a FDI has been launched by an external adversary. Therefore, We can conclude that FESDA is secured against FDI attacks launched by an external adversary. □

VI. Performance Evaluation

In this section, we evaluate the performance aspects of the fog-computing paradigm in the context of secure data aggregation in a SG IoT network. More specifically, we have:

- Analyzed the aggregation cost, decryption cost at fog node and at the CC respectively by providing a comparison with the existing schemes [9], [26], [27] as shown in the Fig. 2, 3 and 4.
- Evaluated the aggregation cost at FNs and cloud. The data aggregation cost at fog nodes and at the cloud, with the increasing number of SMs is shown in Table. II.
- Calculated the communication cost from SMs to FN, and provided a comparison with existing schemes [9], [26], [27] as shown in Fig. 5.

We have assumed three fog nodes in our experimentation and all the fog nodes have the same computational capability. Therefore, we have compared the results of aggregation cost with the existing scheme for one such fog node. The proposed FESDA scheme is implemented using Java cryptographic extension (JCE). For the implementation of FESDA scheme, we have used the data-set of Irish smart grid [32], while, parameters used in our implementation are shown in Table III. Although, there are some cloud-fog simulators i.e., iFogSim [33] and YAFS [34] which have been used in various studies to calculate the latency, network congestion, energy consumption, and cost, etc., for cloud-fog scenarios. However, these simulators do not have support to implement the cryptographic schemes. For this reason, we have performed a custom implementation of the proposed scheme. The objective of this was to measure the cost of various cryptographic primitives in FESDA scheme. The experimental results were obtained on a system with Intel Core i5-3210(M), 2.50 GHZ CPU, 6 GB DDR3 RAM, and Windows 10 OS. We have performed a comparison of proposed scheme with other state of art schemes such as [9], [26], [27]. For performance evaluation, we have compared the computational and communicational cost. We have performed our experiments for 100 to 1000 SMs, which collect and transfer their consumption data to their respective FN. The FNs perform data aggregation, and check for data integrity and source authentication of metering data and, forward the aggregated data to the cloud. Finally, cloud checks for source authentication, performs decryption, and get the users' consumption data.

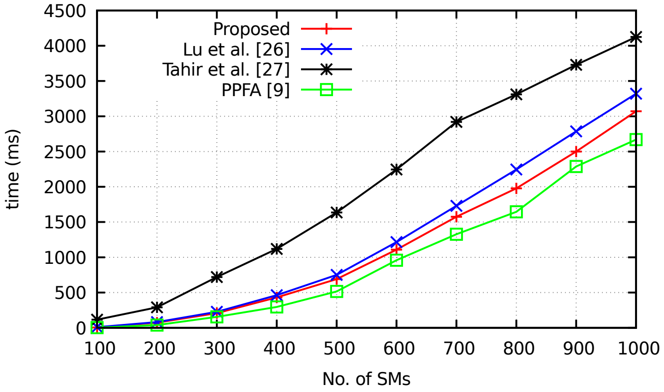

Figure 2: Aggregation cost comparison

Table III: Implementation parameters

Parameter	Value
Large prime p	512 bits
Large prime q	512 bits
Hash Algorithm	256 bits

A. Computational Cost

Computational cost is measured in terms of the time required for aggregation at FNs, CC and the time it takes for decryption at CC. For comparison with other schemes [9], [26], [27], we have performed our experiments for 100 to 1000 SMs, that are attached with FNs. The aggregation cost comparison computed using Algorithm 2 as shown in Fig. 2, the plot shows that the proposed scheme has less aggregation cost than the Lue *et al.* and Tahir *et al.* schemes and slightly higher cost than the aggregation cost of PPFA scheme. The proposed FESDA is fog-enabled scheme, where major computations are performed at FNs, therefore, proposed FESDA scheme has

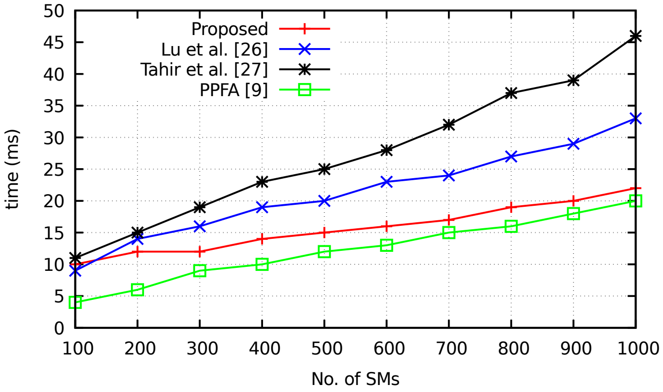

Figure 3: Decryption cost comparison

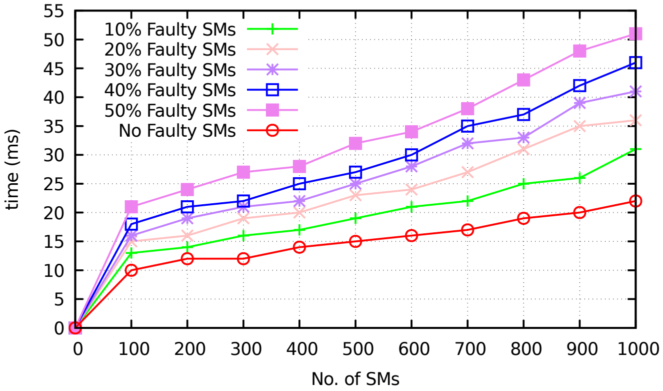

Figure 4: Fault-tolerant decryption cost

lower aggregation cost than [26], [27] schemes. The proposed FESDA scheme has slightly higher cost than PPFA [9]. PPFA [9] have used OTP to encrypt smart metering data, although, it is conjectured that OTP is computationally efficient and unconditionally secure, OTP has a number of security and performance limitations.

Similarly, the decryption cost at CC is computed using Algorithm 3, and Fig. 3 shows the decryption cost comparison of proposed scheme with existing schemes [9], [26], [27]. The proposed FESDA scheme has lower decryption cost than Lue *et al.* and Tahir *et al.* schemes and slightly higher decryption cost than PPFA scheme, for the same reason as mentioned earlier. The proposed FESDA scheme is fault-tolerant aggregation scheme, which means that the collection of data from other devices will not be affected even if 50% of the smart meters are malfunctioning. For implementation, we have considered that 10% to 50% SMs out of 1000 SMs, are faulty. Meaning, faulty SMs will not transmit their consumption data to FN. The fault-tolerant decryption cost at CC is computed using Algorithm 4 and shown in Fig. 4. The plot shows that fault-tolerant decryption cost increases at CC with the increasing number of malfunctioning SMs.

Furthermore, the computational cost of others operations (such as the cost of encryption, MAC-Tag generation and MAC-Tag verification etc.) is depicted in Table IV. The security properties achieved in FESDA scheme and the comparison with schemes [9], [26], [27] is depicted in Table V.

Table IV: Computational cost

Operation	Time (ms)
Encryption at SM	10
MAC$_i$ generation of c_i	2
MAC$_i$ verification at FN	1
MAC$_{x_1}$ verification at Cloud	417
MAC$_{x_2}$ verification at Cloud	695
MAC$_{x_3}$ verification at Cloud	1022

B. Communication Cost

The communication cost is calculated in terms of size of the message from SMs to FN, and from FN to CC. In FESDA, the size of cipher-text c_i is 1024-bits. Since, we are using HMAC with SHA-256 for integrity and source authentication, and the time stamp (TS) is of 32-bits, therefore, the communication cost of FESDA will become $1312 \times N$ bits from N SMs to the FN. The communication cost of Lu *et al.* [26] is $1024 \times N$ bits, while, the communication cost of Tahir *et al.* [27] scheme is $1152 \times N$, from N SMs to the aggregator. For implementation purposes, we have considered the message size is of 1024 bits in PPFA [9], therefore, the key require will be of 1024-bits, as OTP is used in this scheme. Moreover, the signature size is of 128-bits, therefore, the communication cost of PPFA with differential privacy considerations will be of $1184 \times N$ bits from SMs to FN. The communication cost of proposed scheme is much less than [9] and slightly higher than the schemes [26], [27]. However, we are achieving data integrity and source authentication, while the scheme [26] does not ensure data integrity and source authentication of metering data, and these schemes are not fault-tolerant. Likewise, [27] is not resistant against the wide array of attacks, as FESDA is. Additionally, in FESDA, no external party can inject false data in SG communication, therefore, FESDA prevents FDI and replay attacks by external attackers. Therefore, we can say that the communication overhead introduced by the proposed scheme is not avoidable, if FDI and replay attacks are to be prevented. The communication cost comparison with [9], [26], [27] is shown in Fig. 5.

We have compared proposed FESDA scheme with traditional aggregation schemes [26], [27] and with PPFA [9]. The APPA scheme [12] has focused on anonymity. To achieve anonymity, the authors have pseudonyms and certificates. For encryption the authors have used Paillier cryptosystem same as our scheme. However, our focus is on achieving fault-tolerant data aggregation. Therefore, we have chosen the scheme PPFA for the comparison. The proposed FESDA scheme is computationally more efficient than the schemes [26], [27] and has lower communication cost than the scheme PPFA [9]. The PPFA scheme has used OTP to encrypt smart metering data. In SG data aggregation schemes, CC requires aggregated data from SMs for better demand estimation. In this scenario, SMs need to encrypt their consumption data for every 15 minutes and forward it to their corresponding FN. While in public key cryptosystem, i.e., Paillier cryptosystem, we need to generate and transfer key for only one time. Consequently, it resolved

Table V: Comparison of security properties

Technique	Privacy	Data Integrity	Source Authentication	FDI Attack Resistance	Replay Attack Resistance	Fault-Tolerance	Forward Secrecy
PPFA [9]	✓	✓	✓	✗	✗	✓	✗
Lu *et al.* [26]	✓	✗	✗	✗	✗	✗	✗
Tahir *et al.* [27]	✓	✓	✓	✗	✗	✗	✗
FESDA	✓	✓	✓	✓	✓	✓	✓

the scalability, storage and, communication problems, which arise due to new keys generation, storage and transportation of it. Specifically, speaking FESDA has reduced the communication by 50% when compared with PPFA [9].

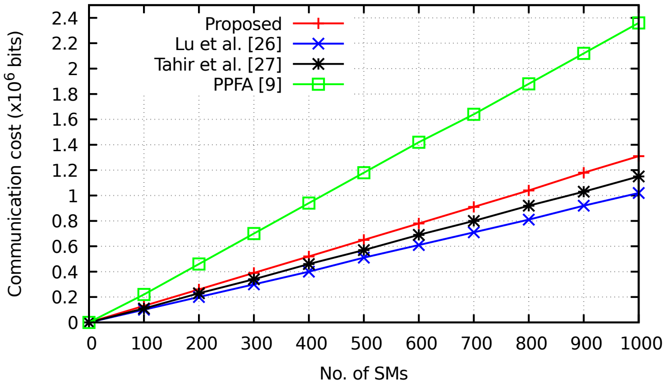

Figure 5: Communication Cost Comparison

VII. Conclusion and Future Work

In this work, we have proposed a privacy-preserving fog-enabled smart metering aggregation scheme in SG IoT network, FESDA, which achieves privacy preservation, data integrity, source authentication, and fault-tolerance. Unlike existing fog-enabled data aggregation schemes, which are based on either based on OTP or pseudonym certificate, FESDA uses HMAC to verify the integrity and source authentication of metering data. The use of OTP or pseudonym certificate is not practical in large scale distributed systems due to key/certificate generation, updation and storage costs. The proposed scheme filters out the false data injected by external attackers (i.e. FDI attack resilience). The use of Paillier and HMAC efficiently reduces the computational and communication overhead and improves the work efficiency of FNs and CC. In addition, the proposed scheme is proven to be fault-tolerant and computationally inexpensive in terms of aggregation, decryption, and communication costs as compared to its counterparts. Future research includes redesigning the privacy-preserving data aggregation so that it does not involve any trusted authority (TA), for example by using secure multi-party computation (SMPC) for data aggregation. In addition, designing a virtualised privacy-preserving billing mechanism by leveraging the software defined networking and network function virtualization for fog-enabled smart metering infrastructure can be interesting future extension of this work.

References

[1] B. Li, R. Lu, K. R. Choo, W. Wang, and S. Luo, “On reliability analysis of smart grids under topology attacks: A stochastic petri net approach,” *ACM Trans. on Cyberphysical Systems*, vol. 3, no. 1, pp. 10:1–10:25, 2019. [Online]. Available: https://dl.acm.org/citation.cfm?id=3127021

[2] X. Fang, S. Misra, G. Xue, and D. Yang, “Smart grid—the new and improved power grid: A survey,” *IEEE Commun. Surveys Tuts.*, vol. 14, no. 4, pp. 944–980, 2012.

[3] B. Li, R. Lu, W. Wang, and K. R. Choo, “Distributed host-based collaborative detection for false data injection attacks in smart grid cyber-physical system,” *Journal of Parallel and Distributed Computing*, vol. 103, pp. 32–41, 2017. [Online]. Available: https://doi.org/10.1016/j.jpdc.2016.12.012

[4] G. D. L. T. Parra, P. Rad, and K. R. Choo, “Implementation of deep packet inspection in smart grids and industrial internet of things: Challenges and opportunities,” *Journal of Network and Computer Applications*, vol. 135, pp. 32–46, 2019. [Online]. Available: https://doi.org/10.1016/j.jnca.2019.02.022

[5] S. Garg, K. Kaur, G. Kaddoum, J. J. P. C. Rodrigues, and M. Guizani, “Secure and lightweight authentication scheme for smart metering infrastructure in smart grid,” *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2019.

[6] S. Ge, P. Zeng, R. Lu, and K. R. Choo, “FGDA: fine-grained data analysis in privacy-preserving smart grid communications,” *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 966–978, 2018. [Online]. Available: https://doi.org/10.1007/s12083-017-0618-9

[7] V. K. PRASAD, M. BHAVSAR, and S. TANWAR, “Influence of montoring: Fog and edge computing,” *Scalable Computing: Practice and Experience*, vol. 20, no. 2, p. 365–376, May 2019.

[8] G. Wood and M. Newborough, “Dynamic energy-consumption indicators for domestic appliances: environment, behaviour and design,” *Energy and buildings*, vol. 35, no. 8, pp. 821–841, 2003.

[9] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, “Ppfa: Privacy preserving fog-enabled aggregation in smart grid,” *IEEE Trans. on Ind. Informat.*, 2018.

[10] H. Wang, Z. Wang, and J. Domingo-Ferrer, “Anonymous and secure aggregation scheme in fog-based public cloud computing,” *Future Generation Computer Systems*, vol. 78, pp. 712–719, 2018.

[11] A. Kumari, S. Tanwar, S. Tyagi, N. Kumar, M. S. Obaidat, and J. J. P. C. Rodrigues, “Fog computing for smart grid systems in the 5g environment: Challenges and solutions,” *IEEE Wireless Communications*, vol. 26, no. 3, pp. 47–53, June 2019.

[12] Z. Guan, Y. Zhang, L. Wu, J. Wu, J. Li, Y. Ma, and J. Hu, “Appa: An anonymous and privacy preserving data aggregation scheme for fog-enhanced iot,” *Journal of Network and Computer Applications*, vol. 125, pp. 82–92, 2019.

[13] J.-N. Liu, J. Weng, A. Yang, Y. Chen, and X. Lin, “Enabling efficient and privacy-preserving aggregation communication and function query for fog computing based smart grid,” *IEEE Trans. Smart Grid*, 2019.

[14] S. Garg, K. Kaur, G. Kaddoum, and K. R. Choo, “Towards secure and provable authentication for internet of things: Realizing industry 4.0,” *IEEE Internet of Things Journal*, pp. 1–1, 2019.

[15] A. M. Odlyzko, “Public key cryptography,” *AT&T Technical Journal*, vol. 73, no. 5, pp. 17–23, 1994.

[16] D. Boneh, B. Lynn, and H. Shacham, “Short signatures from the weil pairing,” in *International Conference on the Theory and Application of Cryptology and Information Security*. Springer, 2001, pp. 514–532.

[17] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, “A practical privacy-preserving data aggregation (3pda) scheme for smart grid,” *IEEE Trans. on Ind. Informat.*, 2018.

[18] A. Abdallah and X. Shen, “A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid,” *IEEE Trans. on Smart Grid*, 2016.

[19] M. Badra and S. Zeadally, “Lightweight and efficient privacy-preserving data aggregation approach for the smart grid,” *Ad Hoc Networks*, vol. 64, pp. 32–40, 2017.

[20] H. Bao and R. Lu, “Ddpft: Secure data aggregation scheme with differential privacy and fault tolerance,” in *IEEE International Conference on Communications (ICC)*, 2015, pp. 7240–7245.

[21] ——, “A lightweight data aggregation scheme achieving privacy preservation and data integrity with differential privacy and fault tolerance,”

Peer-to-Peer Networking and Applications, vol. 10, no. 1, pp. 106–121, 2017.

[22] V. Ford, A. Siraj, and M. A. Rahman, "Secure and efficient protection of consumer privacy in advanced metering infrastructure supporting fine-grained data analysis," *Journal of Computer and System Sciences*, vol. 83, no. 1, pp. 84–100, 2017.

[23] K. Grining, M. Klonowski, and P. Syga, "On practical privacy-preserving fault-tolerant data aggregation," *International Journal of Information Security*, pp. 1–20, 2018.

[24] F. Knirsch, D. Engel, and Z. Erkin, "A fault-tolerant and efficient scheme for data aggregation over groups in the smart grid," in *IEEE Workshop on Information Forensics and Security (WIFS)*, 2017, pp. 1–6.

[25] J. Ni, K. Zhang, K. Alharbi, X. Lin, N. Zhang, and X. S. Shen, "Differentially private smart metering with fault tolerance and range-based filtering," *IEEE Trans. on Smart Grid*, vol. 8, no. 5, pp. 2483–2493, 2017.

[26] R. Lu, K. Alharbi, X. Lin, and C. Huang, "A novel privacy-preserving set aggregation scheme for smart grid communications," in *IEEE Global Communications Conference (GLOBECOM)*, 2015, pp. 1–6.

[27] M. Tahir, A. Khan, A. Hameed, M. Alam, M. K. Khan, and F. Jabeen, "Towards a set aggregation-based data integrity scheme for smart grids," *Annals of Telecommunications*, vol. 72, no. 9-10, pp. 551–561, 2017.

[28] S. Li, K. Xue, Q. Yang, and P. Hong, "Ppma: Privacy-preserving multisubset data aggregation in smart grid," *IEEE Trans. on Ind. Informat.*, vol. 14, no. 2, pp. 462–471, 2018.

[29] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *International Conference on the Theory and Applications of Cryptographic Techniques*, 1999, pp. 223–238.

[30] M. Bellare, R. Canetti, and H. Krawczyk, "Keying hash functions for message authentication," in *Annual international cryptology conference*. Springer, 1996, pp. 1–15.

[31] J. Katz, A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of applied cryptography*. CRC press, 1996.

[32] E. A. of Ireland. (2019) Commission for energy regulation. [Online]. Available: http://www.ucd.ie/issda/data/commissionforenergyregulationcer/

[33] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, and R. Buyya, "ifogsim: A toolkit for modeling and simulation of resource management techniques in the internet of things, edge and fog computing environments," *Software: Practice and Experience*, vol. 47, no. 9, pp. 1275–1296, 2017.

[34] I. Lera, C. Guerrero, and C. Juiz, "Yafs: A simulator for iot scenarios in fog computing," *arXiv preprint arXiv:1902.01091*, 2019.