Efficient and Privacy Preserving User Data Aggregation and Billing Scheme for Smart Grid

By

NOSHABA NAEEM

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Islamabad, Pakistan, in partial fulfillment of the requirements for the degree of MS in Information Security

October 2022

THESIS ACCEPTANCE CERTIFICATE

This is to certify that the research work presented in this thesis, entitled "Efficient and Privacy Preserving User Data Aggregation and Billing for Smart Grids" written by Ms. Noshaba Naeem Student of MS IS Registration No. 00000329797, of Military College of Signals has been vetted by undersigned, found complete in all respects as per NUST Statutes/ Regulations/ MS Policy, is free of plagiarism, errors, and mistakes, and is accepted as partial fulfillment for the award of MS degree. It is further certified that necessary amendments as pointed out by GEC members and foreign/ local evaluators of the scholar have also been incorporated in the said thesis.

Signature: ________________________________

Name of Supervisor: **Asst Prof. Dr Fawad Khan**

Date:____________________________________

Signature (HOD): __________________________

Date:____________________________________

Signature (Dean/ Principal) _________________

Date: ___________________________________

DECLARATION

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and acknowledgments.

Noshaba Naeem

October 2022

DEDICATION

“In the name of Allah, the most Beneficent, the most Merciful"

This research work is dedicated

to

MY PARENTS, TEACHERS AND SIBLINGS

for their love, endless support, and encouragement

ACKNOWLEDGEMENTS

I am grateful to Allah Almighty for giving me strength to keep going on with this thesis, irrespective of many challenges and troubles. All praises for HIM and HIM alone.

I am very grateful to my Thesis Supervisor Asst Prof Dr. Fawad Khan, Co-Supervisor Asst Prof Dr Shahzaib Tahir and GEC members who supervised the thesis / research in a very encouraging and helpful manner. They always guided me with their profound and valuable support that have helped me in achieving my research aims.

I would like to extend my feelings of gratitude towards my father Naeem Qaisar, and Mother Rashida Bashir for their care, love, and endless support through my times of stress and excitement.

Finally, I would like to express my appreciation to all the people who have provided valuable support to my study and whose names I could not bring to memory.

ABSTRACT

Smart Grid is an electrical power supply infrastructure that exploits communication technology to detect and react to changes in demand and supply. A smart grid's main objective is to maximize the use of electrical power by utilizing Realtime interaction between the user side and the generation side. Smart Meters (SM) are an essential component of smart grids, giving residential customers the ability to track and manage their energy costs. A SM today is capable of collection of real-time information on household electricity use. The immense volume of data generated by SM’s can be monitored and controlled in real time by utility companies to achieve operational accuracy. Therefore, the data collected by SMs meets utility-privacy tradeoff. On one side, utility providers require customer data in order to precisely and flexibly control household energy, i.e., the Control Center (CC) can provide electric power during the peak periods of electric use and can and control the charging of storage devices during periods of low demand. With fine-grained data, CC can also identify illegal users and can predict the electrical load. In addition, certain service providers may require customer data in order to enable smart home automation. SMs, on the other hand, lack security features that safeguard the confidentiality, integrity, authenticity, and privacy of user data. They collect fine grained energy usage data, which can compromise users' privacy, especially because the data is collected on a much larger scale, more frequently, and in a detailed manner. The fine-grained metering data could be used by an intruder to learn the consumer's identity and track his/her daily activities. Using fine grained consumption-data, malicious attackers can infer human activity inside a house. Hence, the tradeoff between user privacy and data usability becomes a crucial issue. One of the biggest challenges in smart grid’s research is to maintain user privacy while maximizing Data Utility along with proper Billing, as sharing user data can enable internal and external adversaries to learn about user habits and behaviors. In this research, we have proposed a secure, privacy-preserving mechanism that fulfils various security requirements, ensures maximum utility and accuracy in billing. Utilizing low communication and computation costs, the scheme guarantees the user’s privacy, while protecting the network on the customer's side from multiple types of attacks.

Keywords: Smart Grid, User Privacy, Digital Signature, Blockchain, Data Utility, Billing.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ECDSA	Elliptic Curve Digital Signature Algorithm
Pai	Paillier Cryptosystem
SM	Smart Meter
AG	Aggregator
TCA	Trusted Certification Authority
HLF	Hyperledger Fabric
HAN	Home-Area-Network
BAN	Building-Area-Network
NAN	Neighbor-Area-Network
SP	Service Provider
KGC	Key Generation Center
N	Total number of SMs in residential area
STA	Semi Trusted Authority
CC	Control Center
SM_{ID}	Smart Meter’s Identifier
AG_{ID}	Aggregator Identifier
X_{i,τ_j}	Plain Meter Reading of ith smart meter at jth time interval
X_{AG}	Plain Aggregated Data
C_{i,τ_j}	Encrypted Reading of ith smart meter at jth time interval
C_{z,τ_j}	Aggregated data for Load Monitoring
C_{T,τ_j}	Aggregated data for Billing

TS	Current Timestamp (date+time)
τ	Current Time period
τ_j	jth Time period
a	Starting queried time interval for Billing
b	Ending queried time interval for Billing
p_A, q_A	Large prime numbers
K	Security Parameter
S_{SM-AG}	Communication Cost SM to AG
$S_{z_{AG-CC}}$	Communication Cost AG to CC for Load Monitoring
$S_{\tau_{AG-CC}}$	Communication Cost AG to CC for Billing
PK_{SM}	Smart Meter Public Key
SK_{SM}	Smart Meter Secret Key
PK_{AG}	Aggregator Public Key
SK_{AG}	Aggregator Secret Key
PK_{CC}	Control Center Public Key
SK_{CC}	Control Center Secret Key
σ_{sm_i}	i Smart meter’s Signature
σ_{AG}	Aggregator Signature

Chapter 1

INTRODUCTION

1. Overview

Electric Power and Electronic Communication had a significant role in the twentieth century's fast expansion of civilization. Historically, each country's electrical grid has been a 'broadcast' system. A few central power stations generate electricity to satisfy the country's or region's demand, then distribute it using a large network of cables and transformers as is depicted in Figure 1.

The traditional electrical grid is an electricity distribution network that connects distributed electric energy customers to a few central generators. It employs a demand-driven strategy centered on forecasting consumption and reacting to any residual gaps between forecasted and actual consumption [24].

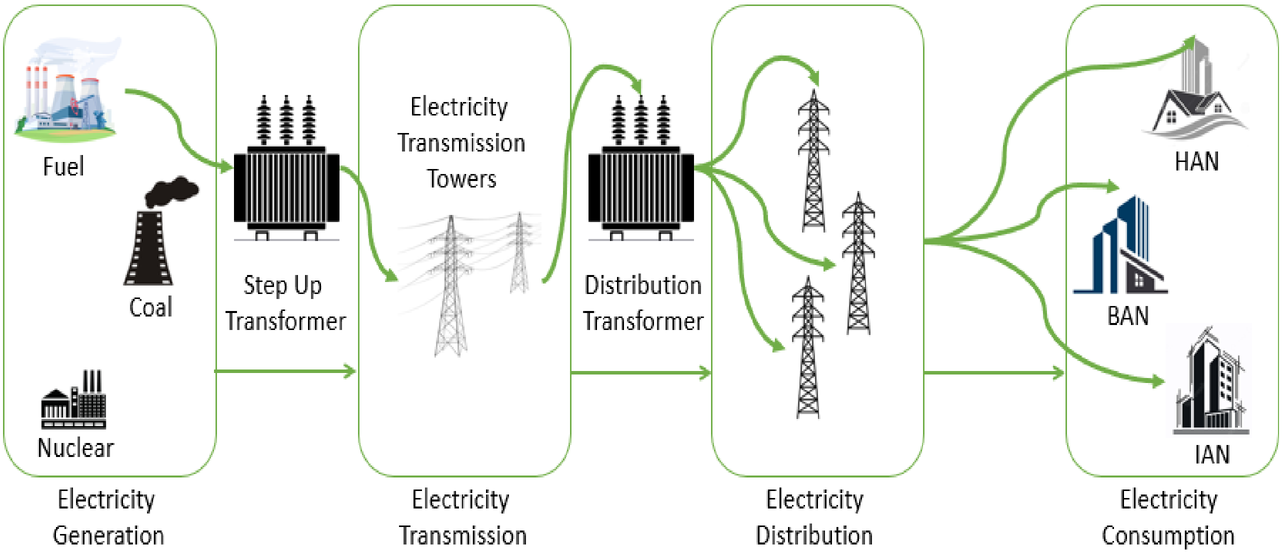


Figure 1 Traditional Electric Grid

While this paradigm has worked effectively for the past century or more, there is an increasing need to change the electric power business, both to solve ageing infrastructure and to handle new societal and environmental concerns to meet the demands of digital age society [25]. The main drivers of gradual transformation in power systems towards the SG paradigms are climate change, new power market trends, energy efficiency awareness, obsolescence of the existing power models, and increasing shift of consumer profiles to prosumer profiles.

The Smart Grid, however, is a modernization of 20th-century power grids that is envisioned as the next phase in the evolution of power supply networks. It's an electricity network that can integrate the actions of all users connected to it, including generators, consumers, and those who do both – to efficiently offer long-term, cost-effective, and secure electricity supplies [26]. Figure 2 provides an illustration of the Smart Grid idea.

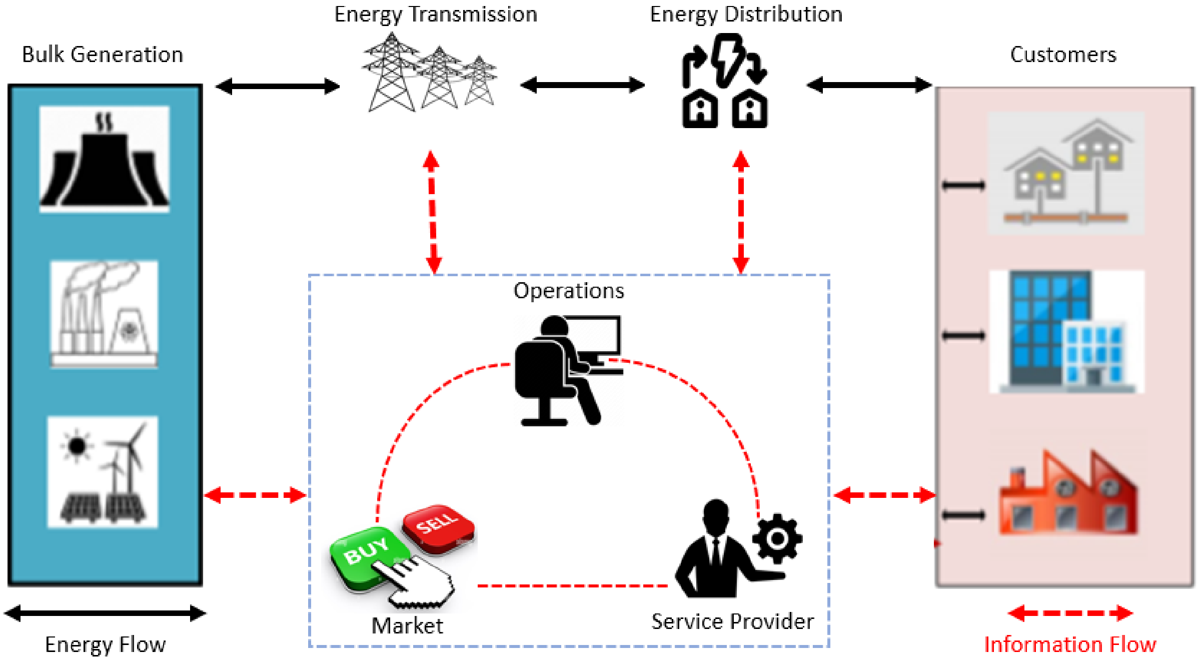


Figure 2 Conceptual Model of Smart Grid

As illustrated in Figure 2, SG network is comprised of several domains, such as bulk generation, energy transmission and distribution, clients, operations, market, and service providers. Two-way energy flow connects the upper domains, such as bulk generation, energy transmission, energy distribution, and customers (illustrated with black lines). The underlying domains, namely operation, market, and service provider, manage and control these top domains via two-way information flow (illustrated with red dotted lines) [27]. In the SG, this two-way flow of energy and data allows for additional capabilities between consumers and utilities. However, the introduction of several distributed generating and storage resources, as well as the use of renewable energy, emphasizes the need for Smart Metering systems capable of regulating and controlling such distributed resources.

A Smart Metering system entails the deployment of a heterogeneous infrastructure, comprising of metering devices, data gathering and processing systems and communication networks, as well as the installation and administration responsibilities that go along with it. The four major pillars of a smart metering system are: Smart Meter (SM), Data Aggregator (DA), and a Control Center (CC).

Smart metering device, SM collect data of each household's energy consumption at regular intervals. Data gathering helps manage the distributed resources in addition to assessing the condition of the power system. As a result, utility companies may regulate and manage the SG using smart metering networks that are connected with Sophisticated Sensors and Information and Communication Technology (ICT [28]). Despite smart metering networks' control and administration capabilities, the collected usage data is exploited by various automated and intelligent systems including Distributed Generation and Distributed Storage, Billing, Load Monitoring and Control System [29].

However, extensive use of smart metering network technology and ICT creates a number of security vulnerabilities, particularly when utility firms combine many automated applications. A deliberate attack on the Smart Grid’s Metering Network might cause power grid systems to slow down or shut down, crippling utility delivery systems. Individual users, as well as infrastructure such as control centers and substations, could be harmed if weaknesses in the SG metering network are exploited. Furthermore, a threat is not confined to the security of the SG metering network; it can also cause a slew of privacy issues for the end-users. SM, for example, typically delivers consumption reports every 15-30 minutes via wireless communications [31],[33]. Such reports can be intercepted by an eavesdropper in order to invade consumers' privacy, such as whether the property is vacant or occupied. People's private life routines can therefore be deduced from them or used against them for illegal purposes.

Following on from the aforementioned problems, privacy and security have recently been the focus of substantial investigation due to the importance of energy networks to the public's safety, security, and economic well-being.

1.1.Motivation

The biggest advantage of smart grids over traditional energy grids is their ability to remotely read fine grained measurements from each SM, which allows grid operators to efficiently balance the load and offer adapted time dependent rates. The privacy of residents is, however, seriously threatened by the acquisition of fine-grained data. Therefore, it is crucial that privacy rights are upheld without disrupting smart grid’s services such as data aggregation for Load Monitoring and Billing.

Many studies propose a secure aggregation strategy to solve privacy concerns. A service provider can only acquire meter reading’s aggregated result, using secure aggregation procedures, while individual meter readings remain private. Previous efforts have used public key homomorphic encryption techniques, commitment systems, or a trusted third-party to aggregate meter readings securely in the billing application. For the load monitoring application previous researches has used public key homomorphic encryption algorithms, secret sharing approaches, or distributed random noise generation to securely aggregate meter readings in an area [32]. In order to exchange messages efficiently and securely, several methods have been proposed, however, there is no single solution to solve all the problems associated with the smart grid at the same time. Either these studies weaken security assumptions, or they just aggregate data from a specific region for load monitoring purposes. In many studies, only data aggregation is catered for efficient usage of data by CC without considering the cost of performing the tasks. Others do not cater billing along with operations. The proposed scheme will be able to provide security and privacy to the customer data while maximizing the data utility, without burdening SM or CC. Furthermore, Blockchain technology is employed to improve transparency and help guard against data content alteration, as well as to ensure proof of existence of a certain content.

1.2.Research Objectives

The main objectives of this thesis are:

- To attain single-user data privacy, so that no private information about a user's behavioral habits is revealed to any harmful/malicious entity.
- To provide data security by preventing unauthorized entities from accessing or manipulating any user's consumption data (Confidentiality, Integrity).
- To ensure maximum data utility (load monitoring) and accurate billing at Smart Grid Control Center.
- The creation of a scheme that is efficient (in terms of communication and computation).
- Finally, utilizing the Python programming language to implement the technique and assess its effectiveness.

1.3.Contribution

The Aggregation Scheme will contribute in the following ways:

- A robust, privacy-preserving and secure blockchain-based data aggregation method that aggregates metering data without breaching user privacy has been proposed.
- The proposed scheme supports protection against various insider and outsider attacks like false data injection (FDI) and replays attacks. Various aggregation schemes do not

fulfil the security requirements. They just focus on aggregating the data for its usage for load monitoring purposes. The proposed scheme fulfils most of the security and privacy requirements during exchange of data between SM, AG and CC.

- To reduce computing costs and communication overhead, techniques such as the Paillier, Homomorphic Cryptosystem, Homomorphic Aggregation Method, and Authentication Mechanism- EllipticCurve Digital-Signature Algorithm (ECDSA) are combined. Blockchain, on the other hand, is used to assure immutability and offer a read-only database for the system.
- The approach will simultaneously provide load monitoring and billing based on exchanged data while incurring minimal computational and communication costs.
- Before using data, each entity verifies if it came from a legitimate source or not. In this way it will act as proactive in preventing any internal or external attack.

1.4.Thesis Outline

The research work has been organized and distributed in the following chapters:

- Chapter 1: A brief introduction is given, a problem statement is highlighted, followed by the motivation behind the research, and research objectives are enumerated. Furthermore, the contributions made through this research are highlighted.
- Chapter 2: This chapter includes an overview of existing aggregation schemes, followed up by pros and cons of each technique.
- Chapter 3: An overview of the preliminary cryptographic primitives employed to design the proposed scheme is provided in this chapter.
- Chapter 4: In this chapter, a system model, the assumed adversarial model under which the scheme will function is discussed. Main design goals of the proposed scheme are also discussed in this chapter. The chapter then includes the proposed scheme’s Flow diagram and each step of the technique is explained in detail.

- Chapter 5: The privacy and security analysis while discussing various design goals is discussed in this chapter. Moreover, chapter includes the implementation of the scheme using python language. Results of scheme are shown in terms of communication and computation cost.
- Chapter 6: The recommendation, conclusion and future work is covered by this chapter.

Chapter 2

LITERATURE REVIEW

2. Introduction

The world, including Pakistan, is becoming increasingly dependent on electric power as we move further into the digital age of the 21st century. Even though Pakistan has a vast amount of energy resources, it still suffers from energy shortages. The present energy supply is insufficient to even meet the current needs. In addition, Pakistan is also plagued with numerous power outages. Load shedding and blackouts have always been part of Pakistan's everyday life. We can only tackle such a massive problem collectively by implementing a smart grid infrastructure.

The smart grid (SG) has a lot of potential advantages, but it also introduces a lot of new security and privacy issues. The smart grid network automates the traditional electricity-network between utility companies and users using Information and Communication Technology_ICT. On one hand, the SG configuration facilitates the seamless flow of consumption data, demand_change messages, and price_change messages between users and the grid; on the other hand, it poses a number of cyber-security concerns.

Smart Meters (SM) are installed to report real-time consumption data of users. Electric consumption data from SM's is reported to utility service provider periodically, which allows them to adjust the supplement based on users' consumptions, thereby providing fine-grained energy supply. However, Realtime electricity consumption information can reveal information about the residents also. It can, for example, determine whether the occupant is at home or not, whether the television or any other machinery is operational, and so on. So, there is a reluctance among people to disclose this kind of personal-private information. Therefore, a suitable privacy preservation technique must be used to communicate these usage messages, in order to prevent them from being intercepted and modified by third parties. The electric system may be shut down as a result of security attacks if an effective security mechanism for the smart grid infrastructure is not developed, and clients may experience power disruptions.

In this section, we analyzed the privacy protection mechanisms that have been established for SG. The realization processes of these mechanisms for SG are based on four parameters, namely

(a) System Model

(b) Goal of the Research

(c) Limitations/Weaknesses

(d) Trust Model.

2.1.Data Aggregation Schemes

Yining et al. [1] present a privacy-preserving aggregation method that does not require the use of a Trusted third-party. To safeguard a single user's data, the technique encrypts consumption data using the EC-ElGamal cryptosystem and creates a virtual aggregation area instead of a physical one. In the work [2] by Zhitao Guan et al., EFFECT is proposed to achieve both the source authentication and aggregation using the Pailier cryptographic scheme and Secret Sharing Scheme. While ensuring individual privacy, the scheme also guarantees fault-tolerance. Wang et al. [3] use the Paillier cryptosystem in their fault-tolerant multi-subset aggregation approach. The entire consumption value is aggregated in this scheme, and the number of customers and total consumption are calculated in various numerical intervals without the use of any Trusted Third Party (TTP). In contrast, the study [5] proposes an aggregation mechanism based on Fog Computing. In this work, the Fog devices functions as a conduit between SM and the Control Center (CC), collecting real-time usage data, saving and aggregating it before sending it to the CC. Le Chen [6] presented MuDA, a Multifunctional Data Aggregation technique for privacy-preserving smart grid communications. The smart grid CC can employ MuDA to compute multiple statistical functions of the customers' data while maintaining their privacy in order to provide a variety of services. Researchers [7] presented a P2DA technique that uses Boneh Goh-

Nissim cryptography to protect against internal adversaries. Despite having a better potential security level, the suggested technique requires a big n to obtain convincing_ security, which increases communication costs, whereas Elliptic-Curve Cryptography-ECC delivers the same level of security with a smaller key-size. A lightweight scheme for aggregating electricity consumption that utilizes lightweight lattice-based homomorphic cryptosystems is proposed in [10]. In this scheme, Smart appliances, rather than smart metres, aggregate their readings. Study [39] provides a methodology for geographically aggregating data for load monitoring using simple cryptographic primitives as XOR operations and one-way hash functions.

Currently, several solutions provide privacy and security protections throughout the aggregation process, but they do not solve privacy concerns in billing since data aggregators can extract aggregated data from the aggregators. The billing system, on the other hand, uses both the feedback and billing systems to balance the bulk generation and consumption data for smart grid metering systems. This prevents the above-mentioned methods from being used for data aggregation and billing.

The studies [4], [8], [9], and [12], achieve both privacy preserving billing and aggregation without sacrificing security, but either they make very weak security assumptions or are computationally very expensive.

Whereas block-chain has emerged as a solution to centralization issues and the trusted third party, because of its decentralized characteristics. A number of studies are using blockchains as privacy preserving methods for aggregation. In studies [14]-[18], blockchain is used to efficiently and securely collect data from SM. However, the consensus mechanism in these studies places extra computational burden on SMs, which are resource constrained devices.

A detailed tabular analysis of schemes employed by different researchers for privacy preserving data aggregation along with the weakness and limitations, is presented in in the following Table 1, where SM, AG, CC represents Smart Meters, Aggregator and Control Center, respectively.

Ref.	System model	Goals	Limitations/ Weaknesses	Cryptographic Primitive	Trust Model
Yining [1] 2019	• SM • AG • CC	• Virtual Aggregation Area • Data’s Utility • Detection of lazy users to ensure practicability and light weight • Data Privacy	• When there are large number of malfunctioning SM in Residential area, anonymization of data in aggregation process is unlikely to attain, putting system's reliability at risk. • Billing not catered	• Lifted EC-ElGamal Cryptosystem • Signature Scheme C-L* (Based on elliptic curve)	• CC is honest but is also curious. • AG can't be trusted. • The active attack will not be launched by SM.
Zhitao [2] 2019	• CC • TCA • AG • SM	• Data Privacy • Fault-tolerance • Authentication • Integrity verification	• Billing not catered	• Paillier cryptosystem • Secret Sharing Scheme	• CC is trustful • SM’s are honest • AG is honest-but-curious • Active Adversary
Xiaodi [3] 2021	• SM • AG • CC	• Data privacy • Fault-tolerance • Dynamic entry and exit • Insider attack resiliency	• Billing not catered	• Paillier cryptosystem • Bilinear pairing	• Authorized entities have a secure communication channel. • Every entity has potential to be a privacy invade

Ref.	System model	Goals	Limitations/ Weaknesses	Cryptographic Primitive	Trust Model
Xin [4] 2021	• SM • TCA • CC • AG	• Data Privacy • Authenticity • Confidentiality • Integrity • Collusion-Attack Freeness • Avoiding Replay Attacks	• Processing is slowed by large cipher-text sizes and keys. • No trust model defined	• Batch RSA • Plus-Type Equations Homomorphic Aggregated Signatures, based on Batch RSA	● Not defined
Hayat [5] 2021	• TCA • CC • AG • SM	• Data Privacy • Fault Tolerance • Data Integrity • Authentication • Avoiding FDI and replay attacks.	• Extra Communication Costs are incurred when SMs fail. • Billing not catered	• BGN-Aggregation Scheme • ECDSA Authentication Mechanism	• CC and AG are honest-but-curious • SMs are honest. • Communication channel not secure
Chen [6] 2014	• TCA • CC • AG • SM	• Data Privacy • Grid communication with multi-functional aggregations • Resist differential attacks • Differential-privacy protection for multi-functional aggregations with low noise	• Not computationally efficient • Weak assumptions-Fully Trusted Entities • Billing not catered	• Composite order groups bilinear map • Cryptosystem Boneh-Goh-Nissim.	• CC and AG are both trustable • SMs are honest • Malicious Adversary

Ref.	System model	Goals	Limitations/ Weaknesses	Cryptographic Primitive	Trust Model
Debiao [7] 2017	• TCA • SM • AG	• Consumer' Privacy • Authentication • Integrity	• To provide convincing security, a big key size is employed, which results in greater transmission costs. • The ECC may accomplish the same level of security with a considerably smaller key size. • Billing not catered	• Boneh–Goh–Nissim	• Active adversary
Asma [8] 2017	• HAN • BAN • NAN • CC • TCA	• Consumer' Privacy • Confidentiality • Messages Integrity • Availability • Prevention of DOS attacks	• BAN is considered honest but curious. All the consumption data of each HAN is processed by BAN directly. • Malicious BAN can impact customer-side networks.	• Lattice-based scheme NTRU	• CC and BANs are honest but curious • Active adversary
Kaiping [9] 2018	• SM • AG • CC • TCA	• Single user data's privacy • Multi subset data aggregation • High efficiency	• No authentication Mechanism • Computationally costly	• Variant of Paillier homomorphic cryptosystem	• AG and CC are honest-but-curious • Malicious Adversary

Ref.	System model	Goals	Limitations/ Weaknesses	Cryptographic Primitive	Trust Model
Xuemin [10] 2018	• SM • TCA • CC • AG	• Consumers' privacy • Lightweight • Authenticity and Data Integrity	• Billing not catered	• Lightweight lattice-based homomorphic cryptosystem	• CC, AG, and SM's are honest but curious. • Active adversary
Yuwen [11] 2019	• SM • AG • CC • KGC	• Consumers' privacy • Multiple Data Reporting	• No threat model defined • Billing not catered	• Variant of Paillier Cryptosystem	• Not defined
Aarti [12] 2019	• CC • NAN • BAN • HAN • TCA	• Secure communications between HAN, BAN and CC • Dynamic Message Exchange Phase • Fixed Messages Exchange Phase	• Weak Assumption- Each Entity is trusted and is communicating via secured network.	• Lightweight R-LWE lattice based cryptography scheme	• NAN and CC communicate via a wired and secure connection • Trusted BAN
An Braeken [13] 2018	• CC • AG • SM • TCA	• Confidentiality • Integrity • Authentication • Efficiency • Dynamic Billing	• No Trust Model defined	• EllipticCurve Cryptography-ECC • Symetric Encryption • 1-Way Hash Operations	• Active adversary

Ref.	System model	Goals	Limitations/ Weaknesses	Cryptographic Primitive	Trust Model
Ozgur [14] 2020	• CC • Ledger • SM • SP	• User Anonymity • Data Integrity • Dynamic pricing • Billing • Data privacy • Billing Verification	• Only Billing is done	• Schorr signature scheme • Hash Functions • Permissioned ledgers for Outsourcing Usage Data, Data Aggregation, Billing, Dynamic Pricing	• CC is fully trusted • Ledger is fully trusted, operated by CC • SP is malicious entity and is operated by a private corporation. • Active Adversary
Rouba [15] 2021	• CC • SM	• Immutability • Privacy of users' energy measurements • The solution is built to withstand a variety of attacks, such as data forging, data injection, and replay attacks.	• No analysis done • Billing not catered	• Smart Contract • Additive Homomorphic Encryption • Blockchain	• Active Adversary
Lu [16] 2021	• CC • TCA • AG • SM	• Confidentiality • Integrity and Authentication • Robustness • Data Privacy • Efficiency • resist network attacks, forgery data	• Billing not catered	• Homomorphic Paillier-Cryptosystem • 1-Way Hash Chain Technique • DVAC consensus mechanism	• External Attackers

Ref.	System model	Goals	Limitations/ Weaknesses	Cryptographic Primitive	Trust Model
Yifan [17] 2021	• SM • AG • CC	• Security and privacy • Efficiency • Secure communication	• Billing not catered	• Private blockchain • Shared blockchain • Identity-Based • proxy re-encryption strategy- Bilinear pairing, Homomorphic • Bilinear pair-based signature scheme	• SMs, AG, And CC Are Honest but Curious • Malicious Attacker
Hongbin Fan [18] 2020	• CC • SM	• Privacy-preservation • Decentralizing • Data unforgeability and nonrepudiation • Data-Security • Confidentiality	• Billing not catered	• Blockchain (Merkle Tree- SHA-256- leader election algorithm) • Bilinear-Pairing • BLS Signature • Paillier-encryption	CC is not trusted. • SM is honest-but-curious.

Table 1 Summary of Existing Studies- Aggregation Schemes

The SG network sends data of customers' electricity consumption to various communication devices. However, managing a user's personal information poses privacy problems. Personal information protection risks include identity theft, determining individual behavioral patterns, revealing activities via residual data, identifying usage of specific household items, conducting real-time monitoring, profiling, monitoring consumers' behavior, and unwanted publicity and embarrassment. The security and privacy of consumers' electricity usage data must be protected while it is moved over the network. Various strategies have been evolved in smart metering systems for aggregating data in a secure and private manner. Some of these research focus on privacy-preserving data aggregation concerns utilizing standard network architecture, but they don't focus at billing mechanisms, hence they don't enable billing applications. Most systems are either vulnerable to hostile aggregator or internal attacks or are computationally expensive for resource-constrained SMs. Some studies focus primarily on data collection, neglecting to include the billing process. Meanwhile, the vast majority of approaches fall short of achieving user anonymity.

There is no single solution that can address all of the smart grid's issues at the same time. These studies either compromise security assumptions or just collect data from a single region for monitoring purposes. However, this research addresses the security and privacy concerns with minimal computational and communication cost, with an anonymous data exchange that can be used for billing and metering purposes on the grid's side without burdening SM or CCs.

Chapter 3

PRELIMINARIES

3. Introduction

The encryption system, signature scheme, and blockchain used to create an efficient technique are covered in this chapter. In order to efficiently aggregate the data coming from Smart Meters (SM) to Control Center (CC), we suggested a four-step aggregation approach. The plaintext data collected by SM from each house is encrypted using the Paillier Encryption technique. The data is signed using the ECDSA Signature Scheme so that it can be verified by the receiving entity. Furthermore, the scheme employs a private permissioned Hyperledger fabric (HLF) for data storage.

3.1.Paillier Homomorphic Cryptosystem

Homomorphic encryption enables users to operate on its encrypted data without first decrypting it. A probabilistic cryptographic scheme, Paillier cryptosystem was introduced by Paillier in 1999 based on the composite residuosity problem. Several privacy protection applications rely on the Paillier cryptosystem [19] to achieve homomorphic additive encryption, which is one of the most widely used methods. This system uses an asymmetric encryption approach to create homomorphic characteristics more efficiently than existing homomorphic algorithms. A Paillier encryption scheme can be proven to be secure against the chosen plaintext attack. The correctness and security of this algorithm was demonstrated in [19]. In this cryptosystem, the relationship between plaintext and ciphertext exists when a and b, two integers, are encrypted as $E_k(a)$ and $E_k(b)$ with a single key ‘k’, such that

$$E_k(a) \cdot E_k(b) = E_k(a\ +\ b).$$

The cryptosystem consists of three parts: Key Generation, Encryption and Decryption.

- **Key Generation**

A key generation algorithm generates two primes p_A, q_A given a security parameter k, where $|p_A| = |q_A| = k/2$ ($|p_A|$ and $|q_A|$ is the length of primes p_A and q_A). After that, another substantial number is calculated as

$$n = p_A . q_A$$

and

$$\lambda = lcm\,(p_A - 1, q_A - 1)$$

After that, define a function;

$$L(u) = \frac{u-1}{n}.$$

Following the selection of a generator $g \in Z^*{}_{n^2}$,,

$$\mu = (L(g^{\lambda} \bmod n^2))^{-1}$$

is calculated. A system's Public Key is $pk = (n, g)$ and Private Key is $sk = (\lambda, \mu)$.

- **Encryption**

A random number $r \in Z^*{}_n$ is selected for a given message $m \in Z_n$, then the cipher-text can be calculated as.

$$C = E(m) = g^m \cdot r^n \, mod\, n^2.$$

- **Decryption:**

Given a ciphertext $C = g^m \cdot r^n \, mod \, n^2$. the original plain message 'm' can be recovered using the secret key $sk = (\lambda, \mu)$ as

$$m = D(c) = L(c^{\lambda} mod\, n^2) \cdot \mu \, mod\, n.$$

3.2.Elliptic Curve Digital Signature Algorithm – ECDSA

Digital Signatures [21] are a popular and useful tool in information security. Integrity, Authentication and Non-Repudiation are the features that a digital signature provides. The elliptic curve digital signature algorithm (ECDSA) [20] is the elliptic curve equivalent of the Digital Signature Algorithm and performs the identical functions of Key Generation, Generation and Verification of Signatures. The complexity of the elliptic curve discrete logarithm problem and the elliptic curve cyclic groups over finite fields are the foundations of ECDSA. For the same level of security, ECDSA keys and signatures are substantially shorter than RSA's. Security-wise, a 3072-bit RSA signature is identical to a 256bits ECDSA signature.

- **Key Generation**

ECDSA signature scheme's private and public keys are generated via the key generation algorithm. A random integer in the range $[1..n-\ 1]$ is used to create the private key. The generating point P is multiplied by the private key to produce the public key, which is a point on the elliptic curve. The major steps in the generating process are as follows:

- Select an Elliptic Curve-E over Finite Field-F_p with a large prime n, that divides the number of points in E(F_p).
- Choose an n^{th} order base point, P, such that $P\ \in\ E(F_p)$.
- In the interval $[1, n-1]$, choose an unpredictable and unique integer, d.
- Equation $Q\ =\ dP$ is used to calculate Public Key Q
- The Private Key is d, and Q is the public key.

- **Signature Generation**

A message ‘M’ and a private key ‘d’ are the two inputs needed for the ECDSA signing algorithm, which produces a signature made up of two integers, {r, s}. The processes for creating a signature for message ‘M’ are as follows:

- Calculate the hash of message as $h\ =\ H(M)$, by using a cryptographic hash algorithm such as SHA-256.
- Generate k in the $[1,2\ \ldots, n-1]$ range using a secure random number generator.

    - Calculate $R\ (x1, y1)$ as a random curve point using $R\ =\ kP$
    - Calculate r using $x_i\ mod\ n$.
    - Calculate $\mathrm{s\ =\ k^{-1}(h\ +\ d\,r)\ \ mod\,n}$
    - Return signature: (r, s).

- **Signature Verification**

The signature $\{r, s\}$ and the signed message M, created by the signing algorithm and the public key, which corresponds to the signer's private key, are fed into the ECDSA signature verification procedure. The result is a Boolean value indicating whether the signature is legitimate or invalid. The steps for signature verification are as follows:

    - Obtain public key Q of signatory A
    - Check that the values $\{r, s\}$ are in $[1..n - 1]$ range.
    - Track down signature proof's inverse modular as, $w\ =\ s^{-1}\ mod\ p$
    - Calculate $h\ = H(M)$, where the secure hash algorithm 'H' is the same one that was employed to create the signature.
    - Calculate $u_1 =\ h\ *\ w\ mod\ n$
    - Calculate $u_2 =\ r\ *\ w\ mod\ n$
    - Using $u_1 *\ P\ +\ u_2 *\ Q$, recover random point (x_0, y_0)
    - Determine $v\ =\ x_0\ mod\ n$

Only if $v\ =\ r$ the signature for message 'M' is verified.

3.3.Blockchain

During the last lustrum, the scientific community was very interested in blockchain development and applications. Despite the lack of specific definition, Blockchain can be characterized as a Peer to Peer (P2P) Distributed System network technology in which all nodes in the network share a data store called a ledger. It has developed into a highly effective decentralized security mechanism because the majority of the network's nodes confirm the transactions on the ledger, maintaining confidence amongst all participants [34]. Thus, Blockchain is a Distributed Ledger Technology (DLT). Blockchain has a linked list-like data structure, with each new block linking to the one before it and so on until the genesis block is reached. The process creates a chain system, with the chain links created using encryption and hash functions. Because hash functions are involved, tampering with the data included in new blocks is extremely difficult; interfering with blocks necessitates modifying all prior blocks. This property is called Immutability, that guarantees data integrity. Another important feature of blockchain is its high availability. For example, information may be retrieved from a single node in the blockchain network because all nodes in the blockchain store all data transactions. Another key characteristic of blockchain is its transparency since it can be auditable. To put it another way, all transactions can be traced back and verified by anyone. A block is a collection of transactions from various network nodes. The network nodes distribute the transactions to the other network nodes for verification, agreement, and validation at the appropriate moment. Each node has a simple structure, consisting of a unique identifier, a hash that ties it to the preceding block (null if it is the genesis block), and the set of transactions [35]. Before the transaction is recorded in the block, all parties involved must sign and confirm it.

3.3.1. Private/Permissioned Blockchain

Private/permissioned blockchains have emerged as a viable alternative to public blockchains for deploying the technology within a defined group of users. Write permissions are allowed only to authorized nodes in a private (permissioned) blockchain architecture. All nodes or the public may have full or limited read permission, or it may be completely prohibited. Private blockchains may be appealing for some commercial applications that require a particular level of secrecy, auditability, and governance. A private blockchain’s participant can all be identifiable, but they don't have to trust each other. Limited information may or may not be available to the general public. In contrast to public blockchains, any defined authority can change the blockchain's rule set. Private blockchains can have a considerably simpler consensus mechanism, with new blocks being validated by a single node or a group of nodes.

- **Hyperledger Fabric**

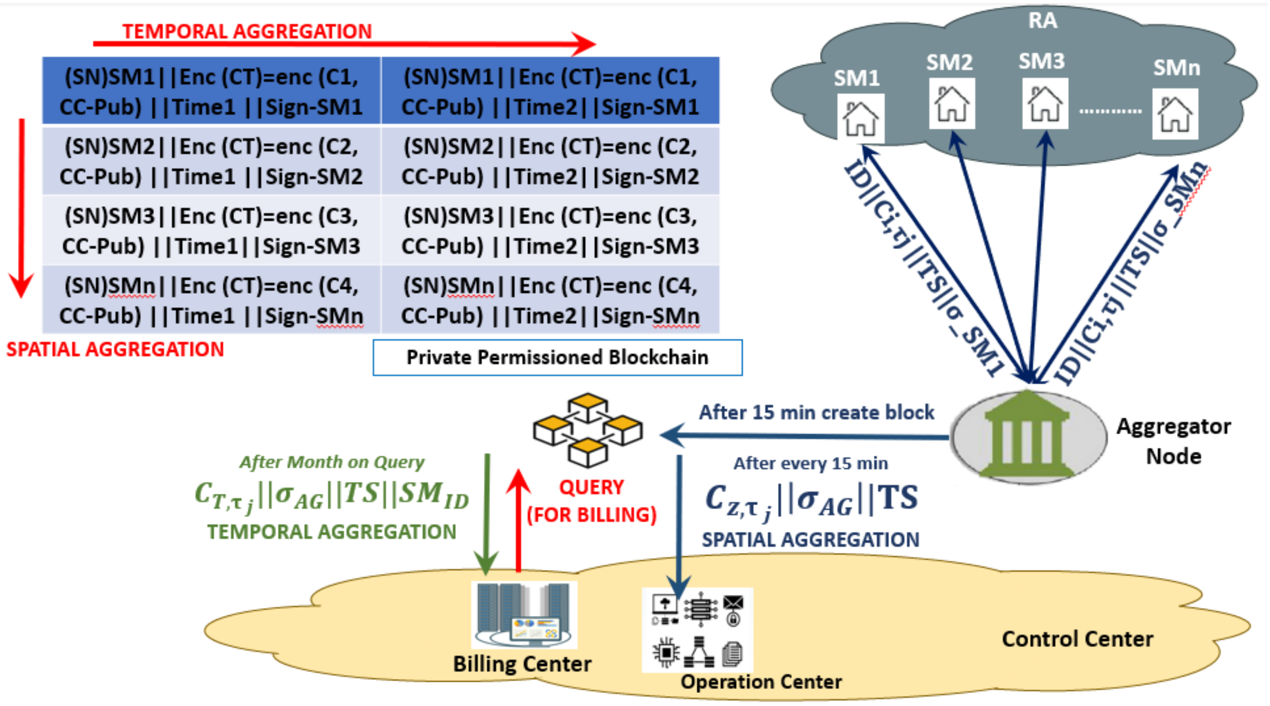
The Linux Foundation's Hyperledger fabric (HLF) is a private, permissioned blockchain. The Fabric network was launched in 2015 as a high-security network that allows members to track, exchange, and engage with digital assets. Fabric is the first distributed ledger to facilitate the execution of distributed applications written in standard programming languages, which are known as chain codes and are placed on the network of HLF nodes. The following components make up the HLF architecture: Client apps, endorsing nodes, Ordering nodes, Peer nodes. HLF's innovative transaction design, known as the Execute-Order-Validate architecture [36], sets it apart from competing platforms. This new architecture replaces all of the old platforms' traditional order-execute architecture. The consensus mechanism is used to order transactions in order-execute architecture. The execution phase follows, in which each peer processes transactions in the same order. Peers aren't restricted to a specific function. For some transactions, a peer may be an endorser, whereas for others, they may only be a committer. The work performed by peer and ordering nodes is similar to that performed by miners in other blockchain architectures [37] [38].

PROPOSED PRIVACY PRESERVING USER DATA-AGGREGATION AND BILLING SCHEME

4. Introduction

In this chapter, methodology adopted to conduct detailed research on the topic is discussed. We proposed a privacy preserving scheme that is compromising of four steps: System Initialization/Key Generation, Ciphertext and Signature Generation, Data Processing by Aggregator (AG), and Control Center (CC) Data Analysis, to efficiently aggregate the data for utility and billing. The flow of the technique with explanation of different entities involved is explained in detail. Moreover, an adversarial model and design goals this study aims to achieve are discussed in this chapter. Different notations used in this chapter are listed in LIST Of ABBREVIATIONS.

4.1.System Model



<table>
<tr><td>(SN)SM1||Enc (CT)=enc (C1, CC-Pub) ||Time1 ||Sign-SM1</td><td>(SN)SM1||Enc (CT)=enc (C1, CC-Pub) ||Time2||Sign-SM1</td></tr>
<tr><td>(SN)SM2||Enc (CT)=enc (C2, CC-Pub) ||Time1 ||Sign-SM2</td><td>(SN)SM2||Enc (CT)=enc (C2, CC-Pub) ||Time2||Sign-SM2</td></tr>
<tr><td>(SN)SM3||Enc (CT)=enc (C3, CC-Pub) ||Time1||Sign-SM3</td><td>(SN)SM3||Enc (CT)=enc (C3, CC-Pub) ||Time2||Sign-SM3</td></tr>
<tr><td>(SN)SMn||Enc (CT)=enc (C4, CC-Pub) ||Time1 ||Sign-SMn</td><td>(SN)SMn||Enc (CT)=enc (C4, CC-Pub) ||Time2||Sign-SMn</td></tr>
</table>

Figure 3 System Model

System model of the proposed technique is presented in Figure 3 and consists of following entries.

- **Smart Meter (SM):**

Every user has a SM, which collects the associated user's electricity usage data and reports it to AG on a regular basis, i.e. every 15 minutes. The suggested approach takes into account n-SMs in a residential area.

- **Aggregating Node (AG):**

AG is responsible for providing interface for blockchain to the SMs. It gathers the data from SMs after regular intervals and creates a block containing all the data on the blockchain, so that both temporal and spatial aggregation can be done on the data for operations and billing purposes by employing blockchain and digital signatures. AG acts as an ordering node, validated by the CC. After completing transactions, AG broadcasts them to a specified list of validated SMs in that area, allowing them to verify their validity. The ledger stores user IDs, encrypted consumption data, a timestamp and the signature for every transaction.

- **Control Centre (CC):**

Control Centre entity includes different centers for processing the data that is periodically gathered by SMs, including an Operation Centre and a Billing Centre. CC is responsible for collection, processing and analysis of data that is reported by AG after regular intervals. With this gathered data, in addition to providing accurate billing, CC can adjust electricity generation, transmission, and distribution to meet varying demands. CC has strong computing capacity to carry out all the processes efficiently.

4.2.Adversary Model

1. AG, and CC are considered to be honest but curious or semi honest.
2. Due to imposed security properties, CC is unable to read specific user data after every 15 minutes as it only gets aggregated data for both load monitoring and billing, yet it is regarded

interested to access user's private data. However, CC can only read temporal aggregated data provided by AG upon request for billing purposes.

3. SM is considered as a tamper-proof device.
4. The communication channel is not secure.
5. It is assumed that no entity AG and CC will collude to get single user’s data.
6. Any attacker, internal or external, tries to figure out how much electricity each user consumes.
7. An attacker may be able to penetrate CC, but only aggregated data will be revealed.
8. To unbalance the load, an external attacker creates a bogus SM and transmits bogus consumption data to the CC.
9. An adversary can initiate replay attacks.
10. Internal adversary can try to alter the saved consumption data.

4.3.Design Goals:

The design goals for resolving the challenges listed above are as follows:

1. Customer’s Privacy

The scheme's major goal is to meet the network's security requirements, which means it must ensure user privacy while maximizing data utility and accurate billing.

2. Security Properties and Attack Mitigation

A further objective of proposed scheme is to satisfy different security features. The security features includes:

- **Data Confidentiality:** To prevent unauthorized disclosure of sensitive information.
- **Integrity, Non Repudiation:** To prevent any illegal data manipulation.
- **Immutability:** Making it impossible for any entity to alter, modify, or fabricate data stored on the network by employing blockchain.

By authenticating each SG entity involved in the process before it carries out any operations on the data, the proposed technique also intends to mitigate the following attacks.

- **External Attacker's False Data Injection**
- **Replay Attacks**

3. Supports Load Monitoring along with proper Billing
4. Communication and Computation Efficiency.

4.4.Proposed Privacy Preserving Scheme

4.4.1. Overview

As illustrated in Figure 4, the main workflow overview of the scheme is as follows.

SM, AG, and CC are the main stakeholders in the proposed scheme. As a start, CC generates the public and private key pair using the paillier key generation algorithm in order to encrypt and decrypt data. Then CC shares its public key with the SM, so that SM can encrypt the data with it, which could then be decrypted by CC with its private key. In a similar way, SM’s and AG generates their private and public keys using Digital Signature ECDSA key generation method and make their public keys publicly available to all entities of system. In this way, AG, using SM's public key, can authenticate SM, while CC can authenticate AG using its public key.

SMs are responsible for encryption of meter reading data and generating a Signature against the readings. With CC's public key, SMs encrypts the data using PAILIER cryptosystem, while SMs private key is used to generate a signature using ECDSA signature generation algorithm. By encrypting metering data and digitally signing it, privacy and confidentiality are guaranteed. After every 15 minutes, the SM sends C_{i,τ_j} concatenated with Timestamp TS, Smart Meter Identifier $\mathrm{SM_{ID}}$, and Digital Signature $\sigma_{\mathrm{SM_i}}$ $(C_{i,\tau_j}||\mathrm{SM_{ID}}||\mathrm{TS}||\sigma_{\mathrm{SM_i}})$ to AG.

AG acts as an intermediary node between SM and CC, processing both operations and billing data while running blockchain. As a primary responsibility, it collects metering data securely from all SMs, verifies their signatures and aggregates them by homomorphic addition of pailliar ciphertexts. AG also grants the read only access of blockchain to SM users, so that they can verify their bills.

Spatial aggregation is done for Supporting Load Monitoring Applications, i.e., data from each household is aggregated every 15 minutes and saved on blockchain. The data from all SM_i is combined by AG, which also creates a Digital Signature for the aggregated data before sending it to CC for decryption. After receiving the data from AG, CC validates their signature, decrypts the aggregated data, and gets the area's usage statistics for specific time frame τ_j. After every 15 minutes, the process is repeated. The steps 1-5 in Figure 4 shows Load Monitoring process.

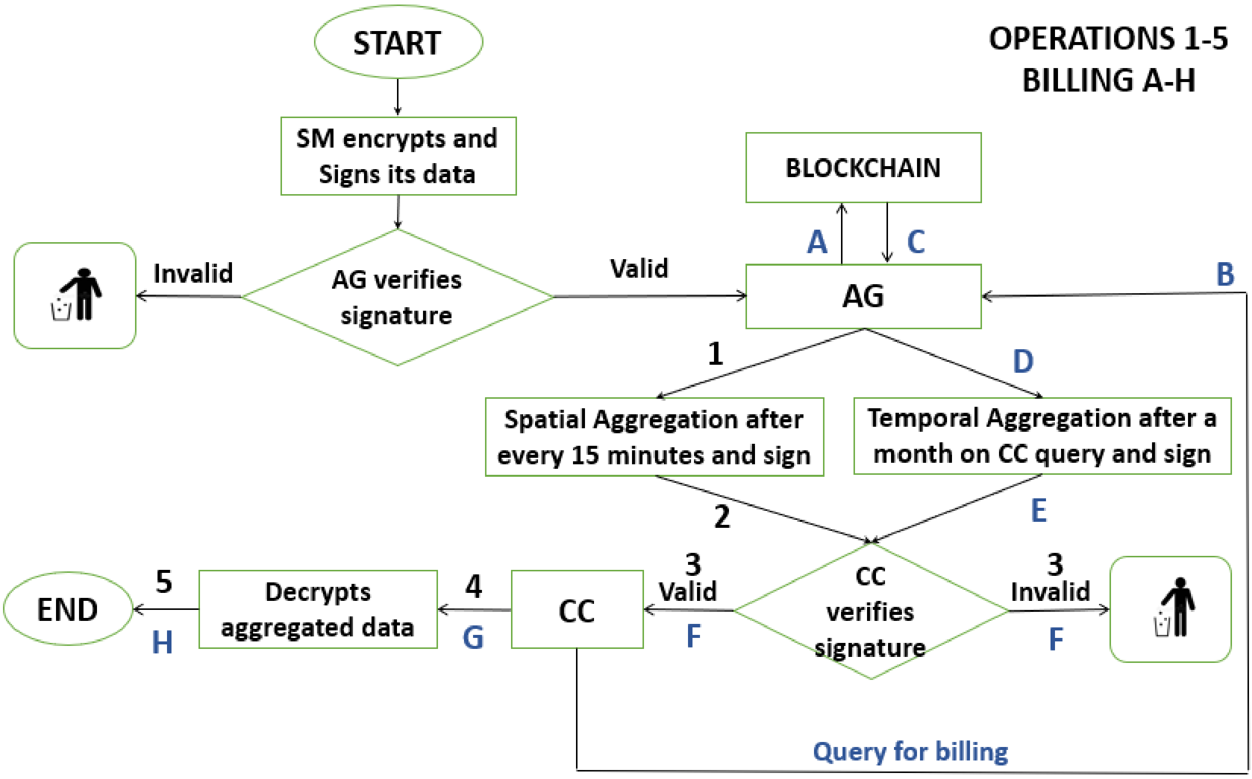


Figure 4 Workflow

AG also performs an additional task in the proposed scheme, which is temporal aggregation on CCs queries, for billing purposes. For this case, AG, after checking the validity of the data, also adds it to the ledger. After every transaction, a block is created containing Encrypted Consumption Data, Encrypted Aggregated Data (Load Monitoring), SM IDs, Timestamps, and Signatures $C_{i,\tau_j}||SM_{ID}||TS||\sigma$. After a period of month, on CCs query for a sum of meter readings over a time period τ_j, AG sums up the individual Encrypted Meter Readings over time τ_j and send the encrypted aggregated consumption of each SM along with signature padded with ID to CC (billing centre).

CC verifies the data before processing it and decrypts it with its private key. CC will only get the aggregated consumption of a household. In this way proper billing is done. CC also shares the aggregated ciphered data it received from AG together with the bill. Since each user has access to the ledger, which contains the ciphered usage data of each SM separately, every user with SM can check its bill by aggregating the data stored in the ledger and comparing it with the aggregated usage data provided by CC. In this way every user is able to verify their own bills.

4.4.2. Initialization/Key Generation

The proposed system initializes with all the entities generating their public and private key pairs for both encryption and signing process. For encryption purposes, CC generates its public and private key pair using paillier key generation algorithm and shares its public key with SM. The paillier public key and private key are required for homomorphic data encryption and decryption, respectively. Similarly, SM and AG generate key pairs for signing process using ECDSA algorithm. A private key is needed to generate a signature, and a public key is required for confirming that signature. SM_i shares their public keys padded with their ID with AG and AG with CC so that they can be authenticated. During the deployment process, both CC and AG receive Smart Meter ID lists (SM serial numbers) for a particular area. When SMs generate their public keys for signature verification, they share a public key padded with the SM ID with AG. AG verifies if the key was generated by legitimate SM by comparing the last 32 bits of the message to the list. It is assumed that if the ID exists in the list then the public key received is from a legitimate SM, otherwise AG discards this key.

Figure 5 depicts the initialization process in detail; the processes are as follows.

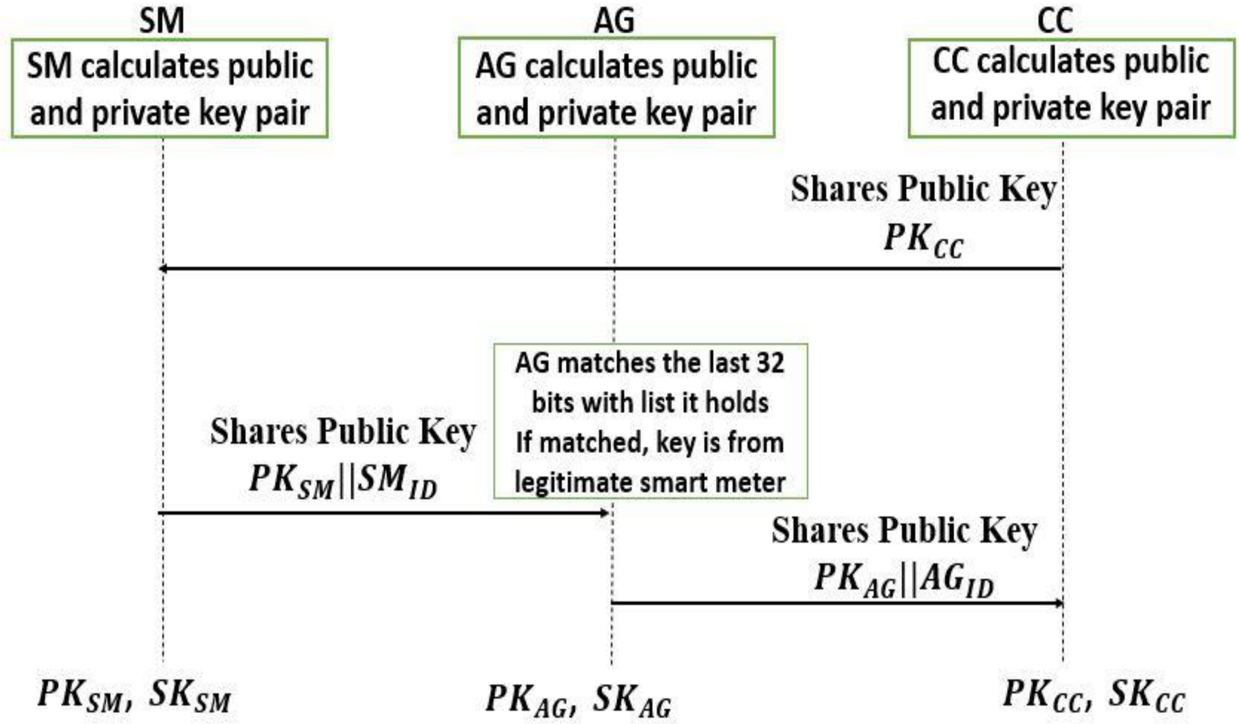

SM
SM calculates public and private key pair
AG
AG calculates public and private key pair
CC
CC calculates public and private key pair
Shares Public Key PK_{CC}
AG matches the last 32 bits with list it holds If matched, key is from legitimate smart meter
Shares Public Key $PK_{SM}||SM_{ID}$
Shares Public Key $PK_{AG}||AG_{ID}$
PK_{SM}, SK_{SM}
PK_{AG}, SK_{AG}
PK_{CC}, SK_{CC}

Figure 5 Initialization

STEP 1 Paillier Parameter Generation

For Encryption and decryption of SM_i usage data, CC chooses two large prime numbers p_A, q_A, where $|p_A| = |q_A| = k/2$, and generates

$$n, p_A, q_A, \lambda, \mu, g$$

as described in Section III, using security parameter k of 1024 bit prime number and n^2 will be approximately 2048-bit number. The associated private key and the public key are

$$PK_{CC} = (\mathrm{n}, \mathrm{g}), SK_{CC} = (\lambda, \mu).$$

CC shares the Public Key $\{n, g\}$ with SM_i in the system and keeps its Private Key (λ, μ) secret [19].

STEP 2 ECDSA Parameter Generation

For signature generation and verification SM_i and AG generates keys: A private key $(SK_{SM_i}, SK_{\mathrm{AG}})$ and public key $(\mathrm{P}K_{SM_i}, \mathrm{P}K_{\mathrm{AG}})$, by selecting a unique integer 'd' from the range $[1, n-1]$ and n^{th} order base point P, such that $P \in E(F_p)$ as described in Section III. Both AG and SM_i shares their Public Key with other entities of the system and keeps their Private Keys secret for Signature Generation [20].

After the key creation activity is completed, the keys are distributed to each system entity for authentication and decryption. The Algorithm # 1. contains the steps for the task.

KEY GENERATION ALGORITHM 1

```
1. Start
2. Input: Prime numbers $\mathbf{p}_A$ and $\mathbf{q}_A$ with k=1024 bits for pailier encryption and decryption
3. Generation: $\mathbf{n} = \mathbf{p_A}.\mathbf{q_A}$
   $\boldsymbol{\lambda} = \mathbf{lcm}\ (\mathbf{p_A} - \mathbf{1}, \mathbf{q_A} - \mathbf{1}$
   $\mathbf{g} \in \mathbf{Z}^*_{\mathbf{n}^2}$ ,   $\mu = (\mathbf{L}(\mathbf{g}^{\boldsymbol{\lambda}} \mathbf{mod}\ \mathbf{n}^2))^{-1}$
4. Output:   $PK_{CC} = (\mathbf{n}, \mathbf{g})$, $SK_{CC} = (\boldsymbol{\lambda}, \boldsymbol{\mu})$  for Paillier Encryption and Decryption
5. Generation $\mathbf{PK}_{SM_i}$, $\mathbf{SK}_{SM_i}$, $\mathbf{PK}_{AG}$, $\mathbf{SK}_{\mathbf{AG}}$  for Signature Generation and Verification
6. Return $PK_{CC}$, $SK_{CC}$, $\mathbf{PK}_{SM_i}$, $\mathbf{SK}_{SM_i}$, $\mathbf{PK}_{\mathbf{AG}}$, $\mathbf{SK}_{\mathbf{AG}}$
7. End
```

4.4.3. Smart Meter Data Processing

Since the user's consumption data is submitted to AG on a regular basis, such as every 15 minutes, SM must encrypt these sensitive data to preserve the user's privacy. To encrypt the data SM_i uses following steps:

1. SM_i (i=1…. N) collects electricity consumption data, X_{i,τ_j} at time-stamp TS
2. To encrypt these data, SM_i picks a random number $r \in Z^*{}_n$ and computes the ciphertext as $C_{i,\tau_j} = g^{X_{i,\tau_j}} \cdot r_{i,\tau_j}{}^n \, mod\, n^2$
3. SM_i computes a signature σ_{SM_i} as discussed in section III, as $\sigma_{SM_i} =$ $SK_{SM_i}(H(C_{i,\tau_j}$ || TS) where TS is the current time-stamp to avoid replay attacks.

   In signing process, Random point R (represented by x-coordinate) is encrypted into a number 's' using elliptic-curve transformations with the private key SK_{SM_i} and the hash h (H(C_{i,τ_j} || TS) into '***s'***, which serves as evidence that the person signing the message is aware of the private key S_{SMi}. Due to the ECDLP hardness problem, the signature{r, s} cannot reveal the private key.
4. SM_ithen sends C_{i,τ_j}||σ_{SM_i}|| TS || SM_{ID} to AG.

Algorithm#2 contains the steps for doing the task.

SMART METER DATA PROCESSING ALGORITHM 2

```
1. Start
2. Input: Message $X_{i,\tau_j}$ , Timestamp TS, $SM_{ID}$
   Generation: $C_{i,\tau_j} = E\left(X_{i,\tau_j}\right) = g^{X_{i,\tau_j}} \cdot r_{i,\tau_j}{}^n \, mod\, n^2$
      $\sigma_{SM_i} = SK_{SM_i}(H(C_{i,\tau_j}$ || TS)
3. Output:    Cipher $C_{i,\tau_j}$ and Signature $\sigma_{sm_i}$
4. Return $C_{i,\tau_j}, \sigma_{SM_i}$, TS, $SM_{ID}$
5. End
```

4.4.4. Aggregator Data Processing

After receiving SM_i report $C_{i,\tau_j}, \sigma_{SM_i}, \mathrm{TS}, SM_{ID}$ (i =1,...,N) AG performs following steps for checking the legitimacy of received report:

1. AG checks the timestamp TS
2. AG computes the signature $\sigma'_{SM_i} = \mathrm{PK}_{SM_i}(\mathrm{H}(C_{i,\tau_j} \,||\, \mathrm{TS})$, using public key and the hash. The signature verification process decodes back the proof 's' from signature to revert it back to the original point 'R'. A comparison is made between the recovered '***R'*'s** x-coordinate and the '***r'*** value from the signature; if they are equal, the report is accepted by the AG; otherwise, it is not processed.

The verification process is as follows:

$$\mathrm{h} = \mathrm{H}(C_{i,\tau_j} \,||\, \mathrm{TS})$$

$$\mathrm{R}' = (\mathrm{h} * w * P) + \left(r * w * \ \mathrm{PK}_{SM_i}\right)$$

as

$$\mathrm{PK}_{SM_i} = \ \mathrm{SK}_{SM_i} * P$$

$$\mathrm{R}' = (\mathrm{h} * w * P) + \left(r * w * \ (\mathrm{SK}_{SM_i} * P)\right)$$

$$\mathrm{R}' = w * P\left(h + r * \ \mathrm{SK}_{SM_i}\right)$$

as

$$\mathrm{s} \ = \ \mathrm{k}^{-1}\left(\mathrm{h} \ + \ \ \mathrm{SK}_{SM_i} * \mathrm{r}\right)\mathrm{mod\ n}\,;$$

and
$$w \ = \ s^{-1}\, mod\, n$$

$$w = (\mathrm{k}^{-1} * \left(\mathrm{h} \ + \ \ \mathrm{SK}_{SM_i} * \mathrm{r})\right)^{-1} mod\, n$$

$$w = \mathrm{k} * \ \left(\mathrm{h} \ + \ \ \mathrm{SK}_{SM_i} * \mathrm{r}\right)^{-1} mod\, n$$

so

$$\mathrm{R}' = \mathrm{k} * \left(\mathrm{h} \ + \ \ \mathrm{SK}_{SM_i} * \mathrm{r}\right)^{-1} * P\left(h + \mathrm{SK}_{SM_i} * r\right)$$

$$\mathrm{R}' = \mathrm{k} * P$$

According to ECDSA signature verification algorithm, r' is the x coordinate of R, so if r'=r, then signature is valid.

When AG receives all the reports from SMs, after verifying them, AG do two tasks: Spatial Aggregation for Load Monitoring and Temporal Aggregation for Billing purpose.

- **Spatial Aggregation for Supporting Load Monitoring Applications**

After every 15 minutes, once each AG completes data validation, each AG aggregates the encrypted data of N SMs at τ_j time through homomorphic paillier addition using the following formula:

$$\boldsymbol{C_{z,\tau_j} = \prod_{i=1}^{N} C_{i,\tau_j}}$$

where τ_j is the current time period.

$$C_{z,\tau_j} = g^{\sum_{i=1}^{N} X_{i,\tau_j}} . \sum_{i=1}^{N} r_{i,\tau_j}{}^{n} \ mod\ n^2$$

$$C_{z,\tau_j} = g^{X_{1,\tau_j} + X_{2,\tau_j} + \dots + X_{N,\tau_j}} . (r_{1,\tau_j} + r_{2,\tau_j} + \cdots .. + r_{N,\tau_j})^n \ mod\ n^2$$

AG then using its Private Key, generates the signature $\sigma_{AG} = \mathrm{SK_{AG}}(\mathrm{H}(C_{z,\tau_j} || \mathrm{TS})$ and sends the report $C_{z,\tau_j} || \sigma_{AG} ||$TS to CC.

- **Block Creation and Temporal Aggregation**

After data validation, AG keeps all the reports containing $C_{i,\tau_j}, \sigma_{SM_i}$, TS, SM_{ID} in ledger as shown in Figure 6. The ciphered aggregated data is also stored in the ledger. Read only permission is granted to SM_i of that particular area so that they can cross check their usage information at the end of each month to validate their bills.

For billing purposes, CC is allowed to query AG for sum of readings over a time period τ. I.e,

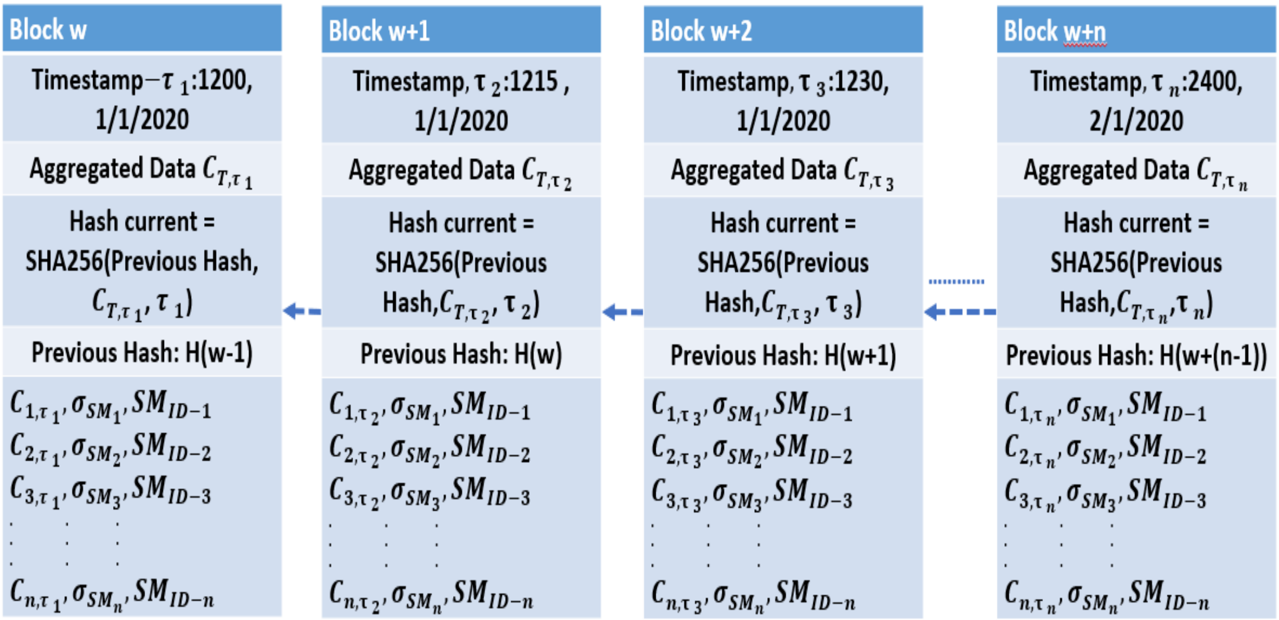
$$\prod_{\tau_j = a}^{b} C_{i,\tau_j}$$

After every month, CC queries AG for usage data of each SM for particular time period $\tau_j(\tau_{a,}\tau_{a+1},\tau_{a+2} \ldots .\tau_b)$,(1 month =31*24*4=2976 data values per SM). AG aggregates the encrypted data of SM_i for queried time interval through homomorphic paillier addition using the following formula:

$$C_{T,\tau_j} = \prod_{j=a}^{b} C_{i,\tau_j}$$

$$C_{T,\tau_j} = \ g^{\sum_{j=a}^{b} X_{i,\tau_j}} . \sum_{j=a}^{b} r_{i,\tau_j}{}^{n} \ mod \ n^2$$

$$C_{T,\tau_j} = \ g^{X_{i,\tau_a}+X_{i,\tau_{a+1}}+\cdots+X_{N,\tau_b}} . (r_{i,\tau_a} + r_{i,\tau_{a+1}} + \cdots .. + r_{N,\tau_b})^n \ mod \ n^2$$



Block w
Timestamp$-\tau_1$:1200, 1/1/2020
Aggregated Data C_{T,τ_1}
Hash current = SHA256(Previous Hash, C_{T,τ_1}, τ_1)
Previous Hash: H(w-1)
$C_{1,\tau_1}, \sigma_{SM_1}, SM_{ID-1}$
$C_{2,\tau_1}, \sigma_{SM_2}, SM_{ID-2}$
$C_{3,\tau_1}, \sigma_{SM_3}, SM_{ID-3}$
$C_{n,\tau_1}, \sigma_{SM_n}, SM_{ID-n}$
Block w+1
Timestamp, τ_2:1215 , 1/1/2020
Aggregated Data C_{T,τ_2}
Hash current = SHA256(Previous Hash,C_{T,τ_2}, τ_2)
Previous Hash: H(w)
$C_{1,\tau_2}, \sigma_{SM_1}, SM_{ID-1}$
$C_{2,\tau_2}, \sigma_{SM_2}, SM_{ID-2}$
$C_{3,\tau_2}, \sigma_{SM_3}, SM_{ID-3}$
$C_{n,\tau_2}, \sigma_{SM_n}, SM_{ID-n}$
Block w+2
Timestamp, τ_3:1230, 1/1/2020
Aggregated Data C_{T,τ_3}
Hash current = SHA256(Previous Hash,C_{T,τ_3}, τ_3)
Previous Hash: H(w+1)
$C_{1,\tau_3}, \sigma_{SM_1}, SM_{ID-1}$
$C_{2,\tau_3}, \sigma_{SM_2}, SM_{ID-2}$
$C_{3,\tau_3}, \sigma_{SM_3}, SM_{ID-3}$
$C_{n,\tau_3}, \sigma_{SM_n}, SM_{ID-n}$
Block w+n
Timestamp, τ_n:2400, 2/1/2020
Aggregated Data C_{T,τ_n}
Hash current = SHA256(Previous Hash,C_{T,τ_n}, τ_n)
Previous Hash: H(w+(n-1))
$C_{1,\tau_n}, \sigma_{SM_1}, SM_{ID-1}$
$C_{2,\tau_n}, \sigma_{SM_2}, SM_{ID-2}$
$C_{3,\tau_n}, \sigma_{SM_3}, SM_{ID-3}$
$C_{n,\tau_n}, \sigma_{SM_n}, SM_{ID-n}$

Figure 6 Blockchain data structure implemented by AG. Transactions per day per SM

AG then using its private key, generates the signature $\sigma_{AG} = \ \mathrm{SK}_{AG}(\mathrm{H}(C_{T,\tau_j}||\ \mathrm{TS})$ and sends the report that contains $C_{T,\tau_j}||\sigma_{AG}||\mathrm{TS}||SM_{ID}$ to CC. In this way CC will only get the aggregated meter readings for billing purposes and cannot access individual meter reading of a particular user/SM at a particular instant of time, which is privacy requirement with the context of user/SM that CC should not be able to know.

Additionally, CC provides aggregated ciphered data together with bills for each user to verify the validity of the bill.

AGGREGATOR DATA PROCESSING ALGORITHM 3

```
Start
Input: Message $C_{i,\tau_j}$, $\sigma_{SM_i}$, $SM_{ID}$
Signature Verification
$\sigma'_{SM_i} = PK_{SM_i}(H(C_{i,\tau_j} \,||\, TS)$
If $\sigma_{SM_i} = \sigma'_{SM_i}$
    Verified $C_{i,\tau_j}$
Accepted Consumption Encrypted Report
//Spatial Aggregation for Load Monitoring
  for i=1; i<=N; i++
      { $C_{Z,\tau_j} = \prod_{i=1}^{N} C_{i,\tau_j}$ }
  $\sigma_{AG} = SK_{AG}(H(C_{Z,\tau_j} \,||\, TS))$
Output:    Cipher $C_{Z,\tau_j}$ and Signature $\sigma_{AG}$
Return $C_{Z,\tau_j}$, $\sigma_{AG}$, TS to CC
//Temporal Aggregation for Billing
Input: CC Query $\prod_{\tau_j=a}^{b} C_{i,\tau_j}$
for j=a; j<=b; j++
    { $C_{T,\tau_j} = \prod_{j=a}^{b} C_{i,\tau_j}$}
  $\sigma_{AG} = SK_{AG}(H(C_{T,\tau_j} \,||\, TS)$
Output:    Cipher $C_{T,\tau_j}$ and Signature $\sigma_{AG}$
```

```
    Return $C_{T,\tau_j}, \sigma_{AG}, \mathrm{TS}, SM_{ID}$
Else
    Rejected $C_{i,\tau_j}$
End
```

4.4.5. Control Center Data Analysis and Decryption

After receiving report for both temporal $C_{i,\tau_j}, \sigma_{AG}, \mathrm{TS}, SM_{ID}$ or spatial $C_{Z,\tau_j}, \sigma_{AG}, \mathrm{TS}$ aggregation from AG, CC performs following steps for checking the legitimacy of received report:

1. CC checks the timestamp TS to avoid replay attacks and message freshness.
2. For billing aggregated data, CC computes the signature $\sigma'_{\mathrm{AG}} = \mathrm{PK}_{\boldsymbol{AG}}(\mathrm{H}(C_{T,\tau_j} \,||\, \mathrm{TS})$, if $\sigma'_{\mathrm{AG}}$and σ_{AG} are equal, CC accepts the report otherwise it does not process the report.
3. Similarly, for spatially aggregated data, CC computes the signature $\sigma'_{\mathrm{AG}} = \mathrm{PK}_{\boldsymbol{AG}}(\mathrm{H}(C_{z,\tau_j} \,||\, \mathrm{TS})$, if $\sigma'_{\mathrm{AG}}$and σ_{AG} are equal, CC accepts the report otherwise it does not process the report.

After checking legitimacy of received report, CC decrypts the aggregated ciphertext based on pailier cryptosystem [19], using its private key $\mathrm{SK}_{\boldsymbol{CC}} = (\lambda, \mu)$ as:

$$X_{AGG} = D(C_{z,\tau_j}) = L({C_{z,\tau_j}}^{\lambda} mod\; n^2) \cdot \mu\; mod\; n$$

as

$$\mu = (L(g^{\lambda} \bmod n^2))^{-1}$$

so

$$X_{AGG} = D(C_{z,\tau_j}) = \frac{L({C_{z,\tau_j}}^{\lambda} mod\; n^2)}{L(g^{\lambda} \bmod n^2)} mod\; n$$

This provides CC with simple aggregated readings of N SMs for a specific time period τ_j, which aids CC in performing utility operations. Whereas, for plain aggregated reading of particular SM, CC computes

$$X_{AGG} = D(C_{T,\tau_j}) = L({C_{T,\tau_j}}^{\lambda} mod\, n^2) \cdot \mu\, mod\, n$$

$$X_{AGG} = D(C_{T,\tau_j}) = \frac{L({C_{T,\tau_j}}^{\lambda} mod\, n^2)}{L(g^{\lambda} \bmod n^2)} mod\, n$$

And do proper billing on the basis of this data.

Control Centre Data Analysis and Decryption ALGORITHM 4

```
Start
Input: Message $C_{T,\tau_j}$ , $\sigma_{AG}$, $SM_{ID}$,TS ; Message $C_{Z,\tau_j}$ , $\sigma_{AG}$, $SM_{ID}$,TS
Algorithm followed for spatial aggregated data
$\sigma'_{AG} = PK_{AG}(H(C_{Z,\tau_j} || TS)$
If $\sigma_{AG}=\sigma'_{AG}$
    Verified $C_{Z,\tau_j}$ , Accepted
    $X_{AGG} = D(C_{Z,\tau_j}) = L({C_{Z,\tau_j}}^{\lambda} mod\, n^2) \cdot \mu\, mod\, n$
Else
    Rejected $C_{Z,\tau_j}$
Output: Cipher $C_{z,\tau_j}$ and Signature $\sigma_{AG}$
Return $C_{Z,\tau_j}$, $\sigma_{AG}$, TS to CC
Algorithm followed for temporal aggregated data
$\sigma'_{AG} = PK_{AG}(H(C_{T,\tau_j} || TS)$
If $\sigma_{AG}=\sigma'_{AG}$
    Verified $C_{T,\tau_j}$ , Accepted
    $X_{AGG} = D(C_{T,\tau_j}) = L({C_{T,\tau_j}}^{\lambda} mod\, n^2) \cdot \mu\, mod\, n$
Else
```

```
    Rejected $C_{T,\tau_j}$
Output:    Plaintext Reading $X_{AGG}$
End
```

Chapter 5

ANALYSIS OF PROPOSED AGGREGATION SCHEME

5. Introduction

This chapter focuses on in depth security and performance analysis of the proposed scheme. Firstly, a theoretical analysis of the scheme is carried out demonstrating that the proposed scheme fulfils various security requirements by considering different design goals assumptions. A comparison is made with different state of the art studies to compare our scheme. The proposed scheme is implemented using python 3.7 on PyCharm interpreter. In addition, in this chapter, we provide the results of the secure aggregation strategy implementation in the SG architecture. We studied the computational and communication costs to assess the performance and compared the results to existing systems ([3],[5],[9],[11]). The findings were obtained on a machine with an Intel® CoreTM m3-7Y30 CPU running at 1.61GHz, 8-GB of RAM, and Windows-10 installed. The results were derived using dataset from 15 prosumers' energy consumption and PV generation (July 15, 2021) [30]. The dataset contains consumption information for 10 SMs over a year. The python paillier [22] library is adopted for encryption and decryption purposes and fastecdsa [23] library is adopted for signature generation and verification. For pailier key generation, two big primes $|p_A|$ and $|q_A|$ of length 512 bits are utilized, as well as an ECDSA SHA-256 bit hash for signature generation. There are n number of SMs in the proposed model that create electricity usage data and communicate it to the Aggregator (AG). AG aggregates consumption data received from their associated SMs and sends only one aggregated reading to Control Center (CC).

5.1.Security and Privacy Analysis

As previously stated, our primary attention is on safeguarding the user's electricity usage data while guaranteeing that utility and billing processes are administered properly by the CC. In this section, we show that the that the proposed scheme preserves the data privacy of metering data, ensures source authentication, and prevents FDI and replay attacks by taking different considerations.

5.1.1. Design Goal 1: Consumer Privacy

Any form of privacy attack on the Smart Meter's (SM) data is prevented.

Assumption 1:

An external adversary may listen in on SMs and AG communications in order to collect electricity consumption data X_{i,τ_j}.

Proof:

$SM_{\boldsymbol{i}}$ reports ciphertext data $C_{i,\tau_j} = g^{X_{i,\tau_j}} \cdot r_{i,\tau_j}{}^{n} \, mod \, n^2$ to AG. An adversary will require the SM's private key in order to access the encrypted data C_{i,τ_j} in plain text. Because the external adversary does not have access to the Paillier cryptosystem's private key 'λ', it is unable to decipher the ciphertext C_{i,τ_j} in order to collect data on a particular user's power use. To preserve consumers' privacy, the power usage data of a single SM is not disclosed.

Assumption 2:

If set of SMs is compromised by an internal adversary to gain the power consumption X_{N,τ_j} of SM_N.

Proof:

If any internal adversary 'A' compromise N−1 SMs, i.e., SM_1, SM_2 , SM_3 ... SM_{N-1}, then, the adversary obtains ciphertext consumption data C_{1,τ_j}, C_{2,τ_j} , $C_{3,\tau_j}\cdots$, C_{N-1,τ_j}. To access X_{N,τ_j} of SM_N, adversary must have the secret key 'λ'. This means that the internal adversary will be unable to access X_{N,τ_j} without 'λ'. We may conclude that, regardless of how many SMs cooperate, the adversary cannot reveal the other users' electricity consumption information X_{i,τ_j}.

Assumption 3:

CC cannot read individual SM_N data, even if CC is compromised it cannot provide specific individual meter reading plain data.

Proof:

For load monitoring CC only gets aggregated data of N SMs as

$$C_{z,\tau_j} = g^{\sum_{i=1}^{N} x_{i,\tau_j}} . \sum_{i=1}^{N} {r_{i,\tau_j}}^{n} \ mod\, n^2$$

And for billing, CC gets aggregated data of SM_N for particular time interval as

$$C_{T,\tau_j} = g^{\sum_{j=a}^{b} x_{i,\tau_j}} . \sum_{j=a}^{b} {r_{i,\tau_j}}^{n} \ mod\, n^2$$

CC can only obtain aggregated result through decryption and cannot read individual data of any SM. Furthermore, if any servers within CC are hacked as a result of malfunction or internal attack, the aggregated data can only be revealed, and the plaintext of individual SM power consumption data at particular time period cannot be recovered by decryption of aggregated data, implying that the CC only received aggregated usage data from AG rather than individual metering data.

5.1.2. Design Goal 2: Security Properties and Attacks Mitigation

Security Property 1: Confidentiality

The consumption data of any household should not be disclosed to unauthorized entity.

Proof:

The electricity usage data includes user private information and corporate information. In SM's data generation phase, each reading is processed using Paillier encryption algorithm to get the ciphertext. Meanwhile, the AG utilize the additive homomorphic attributes to aggregate the ciphertext in the same area. After receiving the aggregated ciphertext of the smart metres in the residential area, only CC is able to decrypt the aggregated plaintext data. Even if the opponent intercepts the ciphertext over the public channel of the SMs in a particular area, the attacker will not be able to deduce any useful information about the usage data sent by the SM because the data is encrypted. As the Paillier Cryptosystem is provably secure against chosen-plaintext attack. The confidentiality of user power consumption data is guaranteed.

Security Property 2: Data Integrity and Non-Repudiation

The consumption data of any household should not be disclosed or manipulated by any unauthorized entity.

Proof:

- **From SM to AG**

A digital signature authentication method will be able to identify any attempts by an attacker to deliver fake data packets while hiding the identities of the actual meters. The AG will accept data packets if the signature and identity are validated, else they will be rejected.

$\sigma_{SM_i} = \ \mathrm{SK}_{SM_i}(\mathrm{H}(C_{i,\tau\ j}\ ||\ \mathrm{TS})$ generated by SM_i

$\sigma'_{SM_i} = \ \mathrm{PK}_{SM_i}(\mathrm{H}(C_{i,\tau\ j}\ ||\ \mathrm{TS})$ generated by AG

if ($\sigma_{SM_i} == \sigma'_{SM_i}$)

keep $C_{i,\tau\ j}$, Otherwise Reject

- **From AG to CC**

AG generates its own signature $\sigma_{AG} = \ \mathrm{SK}_{AG}(\mathrm{H}(C_{z,\tau\ j}||\ \mathrm{TS})$ and pad it with the aggregated encrypted reading and Timestamp. After receiving this data packet CC also verifies it through a Digital Signature authentication procedure as.

$$\sigma'_{\mathrm{AG}} = \ \mathrm{PK}_{\boldsymbol{AG}}(\mathrm{H}(C_{T,\tau\ j}\ ||\ \mathrm{TS})$$

if $\sigma'_{\mathrm{AG}}$and σ_{AG} are equal, CC accepts the report otherwise it does not process the report.

Hence, data integrity is protected from SM to AG and AG to CC. As each entity’s private key is kept by itself so the information it signs and sends cannot be denied.

Security Property 3: Immutability

Proof:

Each of the user data is stored by AG on a blockchain with the current timestamp padded with it. Each of the SM in a residential area also has a copy of it. As blockchain is an immutable database, no one can manipulate data that is already in the blockchain. Once the data is saved in the blockchain, it cannot be changed by any entity of the system.

Assumption Attack 1: Source Authentication and False Data Injection Attacks

Attackers are capable of sending altered data by impersonating as Smart Grid entities.

Proof:

In the initialization phase, SMs and AG shares their public key with the CC. Each SM's public key is available to AG and AGs public key is available to CC. The private key is kept private by each entity to itself. The signatures generated by each entity by their private keys are authenticated using their public keys. Any false message/transaction will be caught using this authentication before it is processed. This approach protects the system from malicious entities attempting to launch fake data injection attacks, i.e., Malicious parties are unable to unbalance the load by providing different consumption data than what a user consumes.

Assumption Attack 2: Replay Attacks

An eavesdropper has the ability to intercept and replay the ciphertext in order to alter the status of the estimation results.

Proof:

Timestamp mechanism is an effective method to resist replay attacks. Since all the data packets sent from SM to AG $C_{i,\tau_j} || \sigma_{SM_i} ||$ TS $\|$ $SM_{ID}$and AG to CC $C_{T,\tau_j} \big|\big| \sigma_{AG} \big|\big|TS||SM_{ID}$ are being time stamped, where data's TS can be verified. The data packet will be rejected if it is for a previous period. In addition, all signature information also contains timestamp information.

$$\sigma = S\mathrm{K}(\mathrm{H}(C || \mathrm{TS})$$

5.1.3. Design Goal 3: Load Monitoring and Billing

Assumption 1: Load Monitoring

The scheme can provide maximum data utility.

Proof:

After every 15 minutes, once each AG completes data validation using ECDSA signature verification procedure, AG aggregates the encrypted data of N SM of an area at τ_j time through homomorphic paillier addition using the following formula:

$$C_{z,\tau_j} = \prod_{i=1}^{N} C_{i,\tau j}$$

This aggregated data is sent to CC, where CC can decrypt it with private its private key for load monitoring of that particular area. The decryption is carried out as follows:

$$X_{AGG} = D(C_{z,\tau_j}) = \frac{L(C_{z,\tau_j}{}^{\lambda} mod\, n^2)}{L(g^{\lambda} \bmod n^2)} mod\, n$$

This provides CC with simple aggregated plain readings of N SMs for a specific time period τ_j, which aids CC in performing utility operations.

Assumption 2: Billing

The scheme can simultaneously provide data utility and proper billing.

Proof:

After data validation, AG also keeps all the reports containing $C_{i,\tau j}$, σ_{SM_i}, TS, SM_{ID} in ledger. When required, After every month, CC queries AG for usage data of each SM for particular time period τ_j ($\tau_a, \tau_{a+1}, \tau_{a+2} \ldots \tau_b$). AG aggregates the encrypted data of SM$_N$ for queried time interval through homomorphic paillier addition using the following formula:

$$C_{T,\tau j} = \prod_{j=a}^{b} C_{i,\tau_j}$$

This aggregated data is sent to CC, where CC can decrypt it with private its private key for billing of each household entity. The decryption is carried out as follows:

$$X_{AGG} \; = \; D(C_{T,\tau_j}) \; = \frac{L(C_{T\,,\tau_j}{}^{\lambda} mod\; n^2)}{L(g^{\lambda} \bmod n^2)} mod\; n$$

This provides CC with simple aggregated plain readings of each SM for a specific time period $\tau_{\,a}$ to $\tau_{\,b}$ (based on query), which aids CC in performing billing.

Security properties of DABS and design goals compared with existing schemes are as shown in Table 2, where the metric Trusted Third Party represents a fully trusted entity used in the system for generation and distribution of encryption and signature keys to every smart grid's entity.

	[3]	**[5]**	**[9]**	**[11]**	**[17]**	**DABS**
	June 2021	**October 2020**	**February 2018**	**May 2019**	**January 2021**	
Confidentiality	✓	✓	✓	✓	✓	✓
Integrity	❖	✓	❖	✓	✓	✓
Source Authentication	❖	✓	❖	✓	✓	✓
Non Repudiation	❖	✓	❖	✓	✓	✓
Privacy	✓	✓	✓	✓	✓	✓
Immutability	❖	❖	❖	❖	✓	✓
Replay Attack	❖	✓	❖	✓	✓	✓
False Data Injection attacks	❖	✓	❖	✓	✓	✓
Initialization without Trusted Third Party	✓	❖	❖	❖	❖	✓

Load Monitoring	✓	✓	✓	✓	✓	✓
Billing	❖	❖	✓	❖	❖	✓
Bill Verification	❖	❖	❖	❖	❖	✓

Table 2 Comparison between DABS and other Related Schemes

5.2.Performance Evaluation

Our scheme's goal is to provide security while still being lightweight in terms of communication and computation overheads. The overhead of communication is related to the message required between the multiple entities involved in the communication. The time required to execute the number of operations in the scheme is referred to as computation overhead. Explained below is the analysis of computation and communication overhead.

5.2.1. Computation Cost

Theoretical Analysis

The proposed scheme assumes that the area has n users (SMs) and that all the user data can be effectively collected. Data Encryption, Signature Generation and Verification, Aggregation, and Decryption are the four primary phases of the entire process. We begin by looking at how the report $C_{i,\tau\ j}$||σ_{SM_i}|| TS || SM_{ID} by SM_i is generated. Ciphertext $C_{i,\tau\ j}$ is generated using one multiplication operation and two times exponentiation operations; σ_{SM_i} is generated using two times multiplication operations, one modular inverse operation and one hash operation.

After AG receives the ciphertext from n SMs, AG initially authenticates the source and integrity of the received data via signature verification, which includes five times multiplication operations, one modular inverse operation and one hash operation. AG performs two type of aggregation for load monitoring and bill calculation. For both type of user’s data aggregation, computation complexity is n-multiplication operations. After aggregating the data, AG again generates the signature $\sigma_{\mathrm{AG}}$with its private key which also requires two times multiplication operations, one modular inverse operation and one hash operation.

Then AG sends report $C_{z,\tau_{j}}||\sigma_{AG}||$TS to CC after every 15 minutes for Load Monitoring. Upon receiving the report, CC checks the report's integrity and authenticates the sender's identity, which requires five times multiplication operations, one modular inverse operation and one hash operation. Following successful authentication, CC uses Paillier decryption to decrypt the aggregated report, which includes one exponentiation and two multiplication operations.

AG sends another report that contains $C_{T,\tau_{j}}||\sigma_{AG}||TS||SM_{ID}$, on CC query and sends it to CC on the basis of which CC calculates the bills. CC checks the report's integrity and authenticates the sender's identity after receiving it, which requires five times multiplication operations, one modular inverse operation and one hash operation. Following successful authentication, CC uses Paillier decryption to decrypt the aggregated report, which contains one exponentiation operation and two multiplication operations.

The computational time required by Data Encryption, Signature Generation, Signature Verification, Aggregation and Decryption operations, are denoted by $T_E + T_S + T_G + T_V + T_D$, respectively. Table 3 shows the computation overhead for the main system entities.

Computation Cost			
User		Encryption + Signature Generation	$T_E + T_S$
AG		Signature Verification	T_V
	Spatial	Aggregation + Signature Generation	$T_G + T_S$
	Temporal	Aggregation+ Signature Generation	$T_G + T_S$
CC	Spatial	Signature Verification + Decryption	$T_V + T_D$
	Temporal	Signature Verification + Decryption	$T_V + T_D$

Table 3 Computation Overhead on the major entities

To demonstrate the scheme's efficiency, the computing complexity is compared to schemes [3],[5],[9],[11]. In table 4, detailed costs of each step compared with other schemes are shown. Whereas, table 5 compares the overall overhead of each entity for Load Monitoring, where E, M, H, P and L represents Exponentiation, Multiplication, Hash, Bilinear Pairing and Discrete Logarithm respectively. For ESDSA signature generation we are only considering Hash and Multiplication Operations.

Scheme	Computation Cost	
[3]	SM	Encryption+ Blind factor (4E + 2M + H) + 3 × (2M + P + H)
	AG	Aggregation n·M
	CC	Decryption n + 1(M)
[5]	SM	Encryption + Sign Gen n*[(4M+3E) +(2M+H)]
	AG	Verification + Aggregation+ Sign Gen n*(5M+H) + (n*M) +(2M+H)
	CC	Verification+ Decryption 5M+H +L
[9]	SM	Encryption n*(5E+4M+H)
	AG	Aggregation n*M
	CC	Decryption E+ M

Scheme			**Computation Cost**
[11]	SM		Encryption + Sign Gen n*[((l+1) *M+ (l+1) *E) +2H+P]
	AG		Verification + Aggregation+ Sign Gen (2H+P) + (n[(l+1) M]) + (2H+P)
	CC		Verification+ Decryption (2H+P) +(E+M)
Our Scheme	SM		Encryption+ Sign Gen n*[(M+2E) +(2M+H)]
	AG	Spatial	Verification +Aggregation + Sign Gen n*(5M+H) + (n*M) +(2M+H)
		Temporal	Aggregation + Sign Gen (n*M) +(2M+H)
	CC	Spatial	Verification + Decryption 5M+H +(2M+E)
		Temporal	Verification + Decryption 5M+H +(2M+E)

Table 4 Comparing computation complexity between DABS and other schemes

Ref	Overhead SM	Overhead AG	Overhead CC
[3]	n ∗ [4E + 8M + 4H + 3P]	n · M	n + 1[M]
[5]	n ∗ [3E + 6M + H]	(6n + 2)M + (n + 1) H	5M + H + L
[9]	n ∗ (5E + 4M + H)	n ∗ M	E + M
[11]	n ∗ [(L + 1) ∗ E + (L + 1) ∗ M + 2H + P]	n(L + 1)M + 4H + 2P *L= types of data*	E + M + 2H+P
DABS	n ∗ [2E + 3M + H]	(6n + 2)M + (n + 1)H (n + 2)M+H 1 reading/ month /SM for billing	E + 7M + H

Table 5 Overall Computational Overhead of each entity for Load Monitoring

From table 4 and the comparative summary of security properties discussed in table 2, it can be clearly seen that DABS is catering much of the security properties along with proper billing and load monitoring with minimal computational overhead. On the other hand, schemes [3],[5], and [11] are not catering billing. Moreover, many of the security properties like integrity, authentication is not catered by schemes [3],[9] as discussed in previous section in table 2. These schemes are not using any authentication mechanism like signature scheme as used in DABS.

So, we can say that DABS is providing all the security properties along with load monitoring and billing with minimal computational cost as compared to other studies.

Simulations

Python is used to implement the proposed strategy (PyCharm IDE). The findings were obtained on a machine with Intel® CoreTM m3-7Y30, CPU running at 1.61GHz, 8-GB of RAM, and Windows-10 installed. The results are based on data from 15 prosumers' energy consumption and PV generation (July 15, 2021) [30]. The python paillier library is adopted for encryption and decryption purposes and fastecdsa library is adopted for signature generation and verification. Two large primes p_A, q_A of 512-bits are used for Paillier Key Generation and ECDSA SHA256 hash for generation of signature. Table 6 shows the results.

	Encryption/SM	**Signature Generation per SM**	**Signature Verification**	**Decryption**
User	2.2 ms	3.12 ms	-	-
AG	-	3.12 ms	2.53 ms	-
CC	-	-	2.53 ms	1 ms

Table 6 Computational Cost of Encryption, Signature Generation and Verification, Decryption

We conducted the experiments with electricity consumption value from the range of 20 to 100 Wh, and the results are shown in Figure 7. We can observe from the graph that the encryption cost of DABS does not change much as the amount of electricity consumed increases. The Paillier's encryption cost remains steady at around 2 milliseconds; thus, the proposed approach is overall optimal.

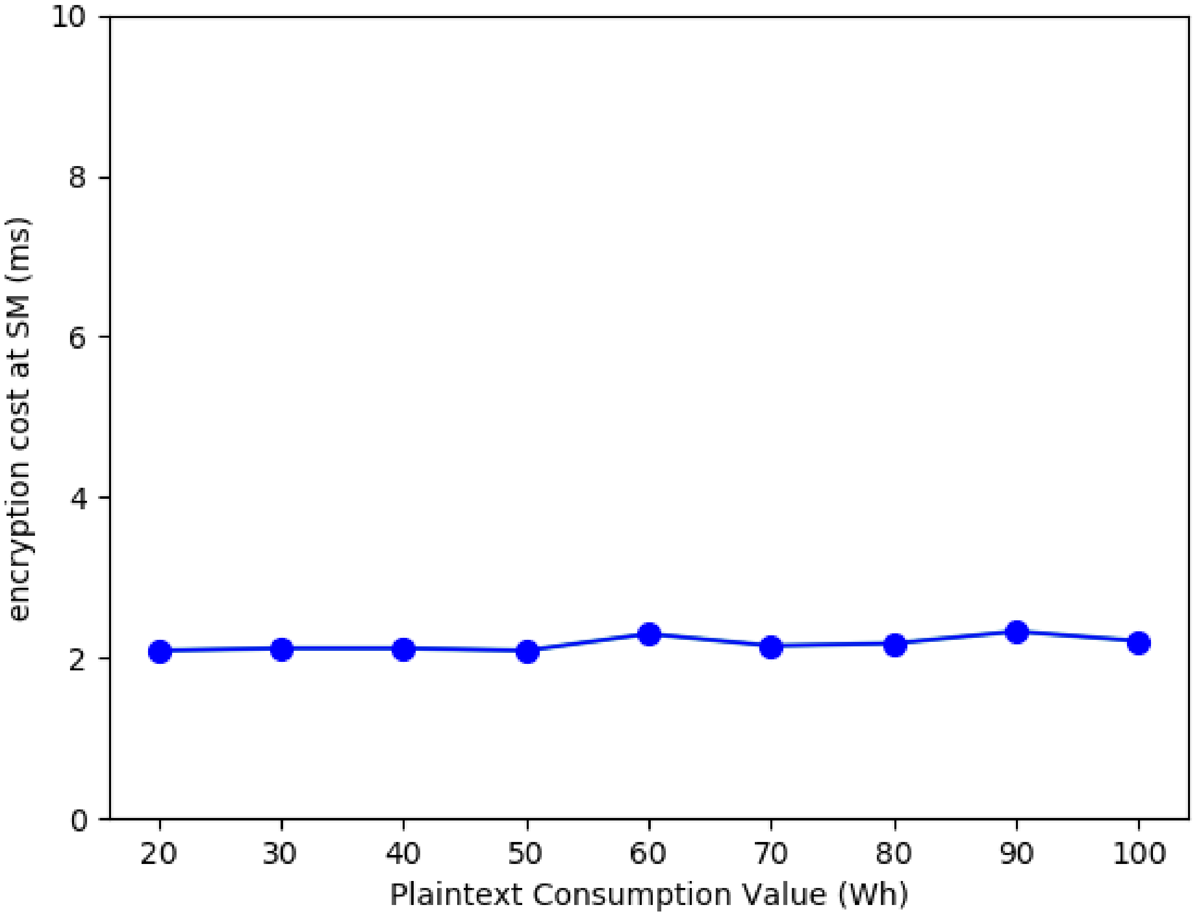


Figure 7 Encryption Cost at SM

Similarly, the DABS scheme's encryption cost is contrasted with the schemes [5], [9], and [11]. Figure 8 illustrates the cost of encryption with users varying from 100 to 1000. As the number of SMs rises, it is evident that DABS demonstrate more superiority. As an example, the overall encryption time for DABS is around 1680 milliseconds when the number of SMs is 800, but the encryption times for schemes [5], [9], and [11] are 2776, 4528, and 1824 milliseconds, respectively, when the number of SMs is 800. We can draw the conclusion that DABS exhibits excellent expansibility and can significantly lower computing costs.

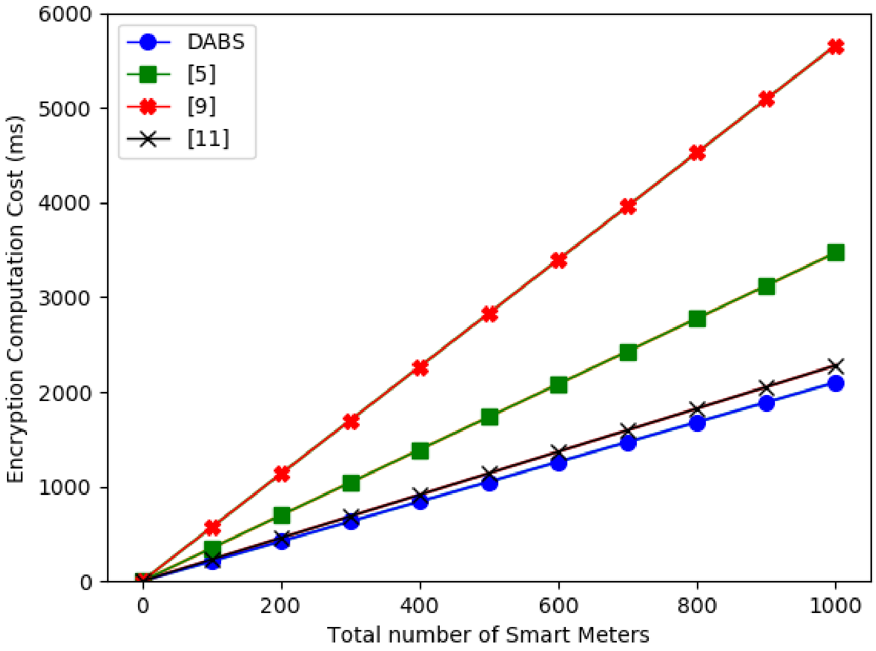


Figure 8 Comparison of Encryption Cost at SM

The cost of aggregation computation for our suggested approach compared with schemes [9] and [11] is shown in Figure 9, with the number of users ranging from 100 to 1000. When the total number of SMs reaches 1000, the aggregation process in our scheme takes about 50.06 milliseconds. Figure demonstrates that DABS has a lower cost when compared to scheme [11] and roughly the same aggregation cost as scheme [9].

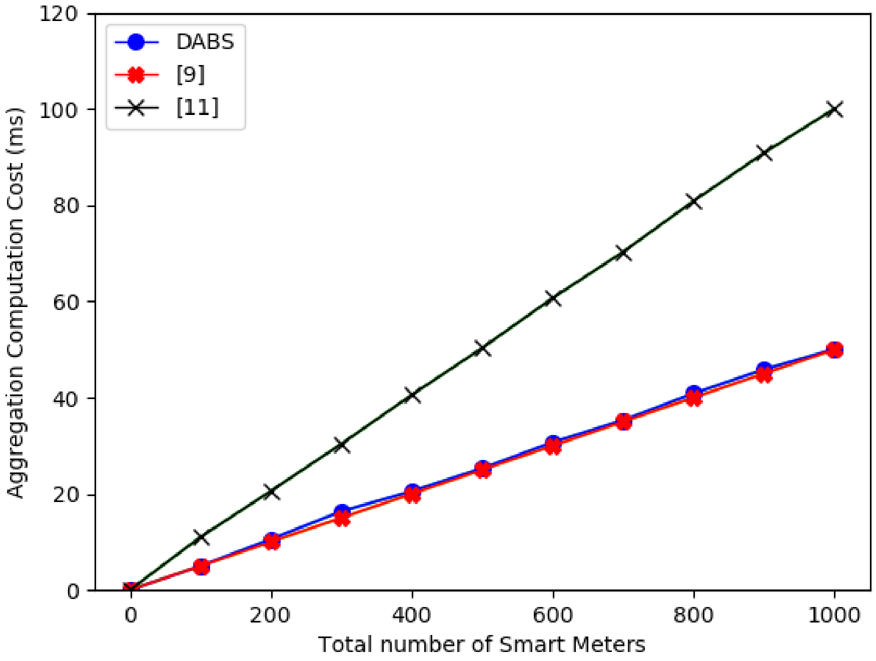


Figure 9 Comparison of Aggregation Cost

To investigate how the number of users influenced the decryption overhead, we implemented our suggested approach at CC with 100 to 1000 users and compared it with schemes [9] and [11]. Figure 10 displays the comparison results in terms of the decryption computation cost. The graph shows that the decryption time of our proposed solution stays around 1 millisecond regardless of the user count. Given that CC often handles a sizable amount of user data reported by numerous AGs each period, this is highly efficient and practical. The figure shows that our suggested approach has a lower decryption overhead than schemes [9], and [11].

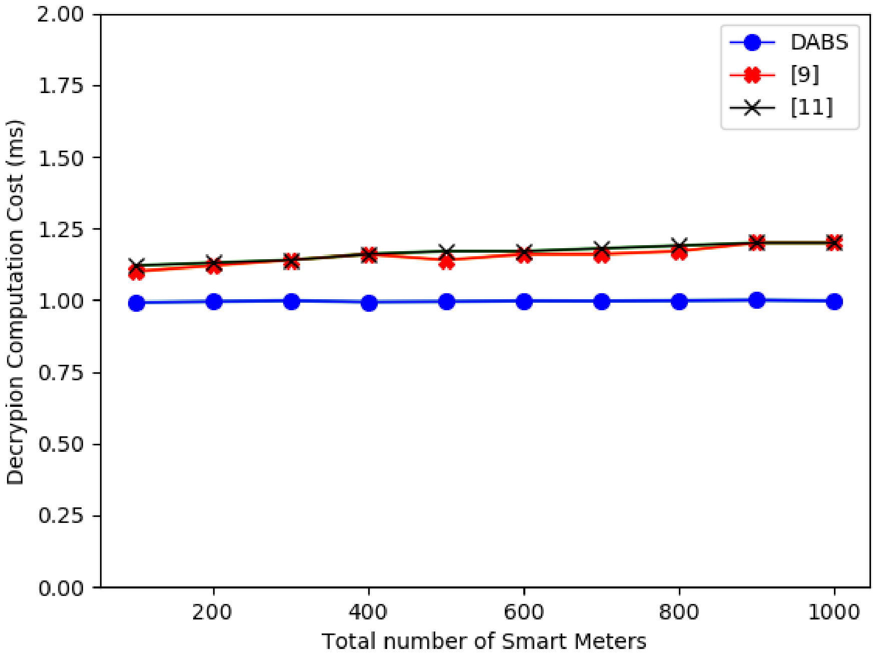


Figure 10 Comparison of Decryption Cost

Moreover, we compared the overall encryption efficiency of DABS for Load Monitoring with the schemes [9] and [11]. Figure 11 shows the comparison results. It is evident from the graph that DABS technique requires comparatively less cost to efficiently encrypt, aggregate, and decrypt the data than schemes [9] and [11].

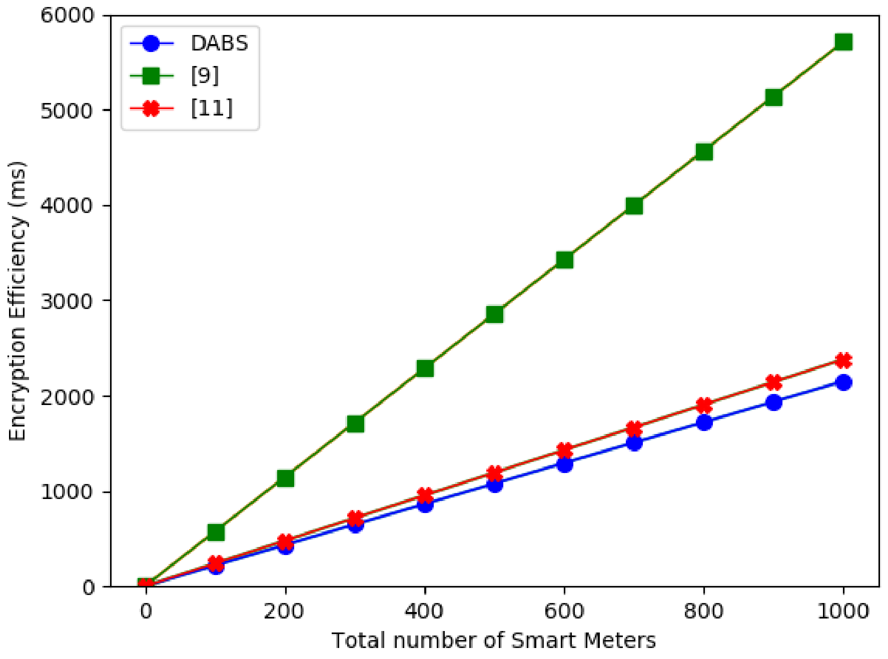


Figure 11 Comparison of Encryption Efficiency

5.2.2. Communication Cost

Communication overhead in the proposed system can be split up into two parts: the cost of communication from an SM to an AG, and the cost of communication from an AG to a CC; table 8 shows the comparison results. In the Paillier cryptosystem, the length of k is set to 1024bits, and size of n^2 is set to 2048bits. The SHA-256 hash produces a 256-bit output, which is utilized to generate the signature. A timestamp is 32 bits, and identity of a SM is 32 bits as well.

In first phase, the report $C_{i,\tau_j}||\sigma_{SM_i}||$ TS || SM_{ID} is created using the information gathered by SM and is sent to the AG. The size of the one user's report is S_{SM-AG} = $|C_{i,\tau_j}|$ + $|\sigma_{\mathrm{SM}_i}|$ + | TS | + $|SM_{ID}|$. The maximum communication overhead in this phase for a residential area with n users is $S_{SM-AG_{max}}$ = n × S_{SM-AG} = n × ($|C_{i,\tau_j}|$ + $|\sigma_{SM_i}|$ + | TS | + $|SM_{ID}|$)=2368n bits. The AG then collects reports from n users and sends the two combined reports to the CC in the next phase. One report is for load monitoring that is send every 15 minutes to CC. The report for load monitoring is generated as ($C_{z,\tau_j}||\sigma_{AG}||$TS), so the size of this report is $S_{Z_{AG-CC}}$ = $|C_{z,\tau_j}|$ + $|\sigma_{AG}|$ + | TS |=2336 bits. Second report is for billing, that is send to CC on CC's query. The report generated for billing is generated as $C_{T,\tau_j}||\sigma_{AG}||\mathrm{TS}||SM_{ID}$ and the size of report is $S_{T_{AG-CC}}$ = $|C_{T,\tau_j}|$ + $|\sigma_{AG}|$ + | TS |+| $SM_{ID}| = 2368$ bits.

The communication overheads in our scheme's SM-AG and AG-CC phases are then compared to those in [3], [5], [9], and [11], as shown in Table 8. The same length parameters (k= 1024 bits; p=q =512 bits) are used for each scheme to generate the reports.

Ref	SM to AG	AG to CC		Communication Cost (bits)
		Load Monitoring	**Billing**	
[3]	n $*$ (|Ci|)	|Cj |	-	2048n +2048
[5]	n $*$ (|Ci| + |TSi| + $\|\sigma_i\|$)	|Cj | + $\|\sigma_j\|$	-	2320n+2320
[9]	n $*$ (|Ci|)	|Cj|	|Cj|	2048n+2048(LM/15min) + 2048(billing/month)
[11]	n $*$ (|Ci| + |TSi| + $\|\sigma_i\|$) + |IDi|)	|Cj| + |Vj| + $\|\sigma_j\|$ + |IDj|	-	2368n+2368
DABS	n $*$(C_{i,τ_j}|+|σ_{sm_i}|+| TS | + |SM_{ID}|)	|C_{z,τ_j}| + |σ_{AG}| + | TS |	|C_{T,τ_j}| + |σ_{AG}| + | TS |+| SM_{ID}|	2368n+2336(LM/15min) + 2368(billing/month)

Table 7 Comparing communication complexity between DABS and other schemes

It is demonstrated that the suggested approach efficiently implements sender authentication and data integrity verification using ECDSA. The advantage of the digital signature is that it eliminates the need for key generation and sharing between SM and AG during each round of communication. It is expensive to produce session keys for each round for thousands of SMs. The method also accommodates load monitoring and billing aggregation functions with negligible additional communication overhead.

Chapter 6

CONCLUSION AND FUTURE WORK

6. Conclusion

For customer-side networks in smart grid, privacy and confidentiality are primary issues. A smart grid privacy-preserving data aggregation system based on blockchain is proposed in this research. The suggested approach ensures the privacy of electricity customers as well as the confidentiality and integrity of exchanged electricity usage messages. Smart Meters (SM) data is recorded on the blockchain by Aggregator (AG), a semi-trusted node. Permissioned blockchain creates a transparent framework that allows each entity to track transactions. Each user can calculate their own bill, ensuring that the billing data is accurate. The security and integrity of messages during transmission is ensured by the ECDSA signature and Paillier encryption. Security analysis shows that the technique meets the standards for SM privacy and security. Proposed privacy preserving scheme meets many privacy and security requirements, as well as data utility and billing, with minimal computational cost, according to the performance evaluation. Moreover, the approach also has a low communication overhead and does not necessitate the use of a trusted party, a trusted authority, or secure communication channels.

6.1.Future Work

The Paillier Cryptosystem will not be secure in the post-quantum era, hence a system based on a mechanism that will be secure in the post-quantum era needs to be devised. Moreover, the current work will be improved in the future to better optimize it and lower the cost of computation and communication.

BIBLIOGRAPHY

[1]. Liu, Y., Guo, W., Fan, C. I., Chang, L., & Cheng, C. (2018). A practical privacy-preserving data aggregation (3PDA) scheme for smart grid. IEEE Transactions on Industrial Informatics, 15(3), 1767-1774.

[2]. Guan, Z., Zhang, Y., Zhu, L., Wu, L., & Yu, S. (2019). EFFECT: An efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid. Science China Information Sciences, 62(3), 1-14.

[3]. Wang, X., Liu, Y., & Choo, K. K. R. (2020). Fault-tolerant multisubset aggregation scheme for smart grid. IEEE Transactions on Industrial Informatics, 17(6), 4065-4072.

[4]. Qian, J., Cao, Z., Lu, M., Chen, X., Shen, J., & Liu, J. (2021). The secure lattice-based data aggregation scheme in residential networks for smart grid. IEEE Internet of Things Journal, 9(3), 2153-2164.

[5]. Khan, H. M., Khan, A., Jabeen, F., & Rahman, A. U. (2021). Privacy preserving data aggregation with fault tolerance in fog-enabled smart grids. Sustainable Cities and Society, 64, 102522.

[6]. Chen, L., Lu, R., Cao, Z., AlHarbi, K., & Lin, X. (2015). MuDA: Multifunctional data aggregation in privacy-preserving smart grid communications. Peer-to-peer networking and applications, 8(5), 777-792.

[7]. He, D., Kumar, N., Zeadally, S., Vinel, A., & Yang, L. T. (2017). Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries. IEEE Transactions on Smart Grid, 8(5), 2411-2419.

[8]. Abdallah, A., & Shen, X. (2015). Lightweight security and privacy preserving scheme for smart grid customer-side networks. IEEE Transactions on Smart Grid, 8(3), 1064-1074.

[9]. Li, S., Xue, K., Yang, Q., & Hong, P. (2017). PPMA: Privacy-preserving multisubset data aggregation in smart grid. IEEE Transactions on Industrial Informatics, 14(2), 462-471.

[10]. Abdallah, A., & Shen, X. S. (2016). A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. IEEE Transactions on Smart Grid, 9(1), 396-405.

[11]. Chen, Y., Martínez-Ortega, J. F., Castillejo, P., & López, L. (2019). A homomorphic-based multiple data aggregation scheme for smart grid. IEEE Sensors Journal, 19(10), 3921-3929.

[12]. Agarkar, A. A., & Agrawal, H. (2019). Lightweight R-LWE-based privacy preservation scheme for smart grid network. International Journal of Information and Computer Security, 11(3), 233-254.

[13]. Braeken, A., Kumar, P., & Martin, A. (2018). Efficient and privacy-preserving data aggregation and dynamic billing in smart grid metering networks. Energies, 11(8), 2085.

[14]. Oksuz, O. (2020, January). Privacy Preserving Data Aggregation and Dynamic Billing System in Smart Grid Using Permissioned Blockchain. In CS & IT Conference Proceedings (Vol. 10, No. 1). CS & IT Conference Proceedings.

[15]. Badra, M., & Borghol, R. (2021, April). Privacy-Preserving and Efficient Aggregation for Smart Grid based on Blockchain. In 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS) (pp. 1-3). IEEE.

[16]. Lu, W., Ren, Z., Xu, J., & Chen, S. (2021). Edge blockchain assisted lightweight privacy-preserving data aggregation for smart grid. IEEE Transactions on Network and Service Management, 18(2), 1246-1259.

[17]. Li, K., Yang, Y., Wang, S., Shi, R., & Li, J. (2021). A lightweight privacy-preserving and sharing scheme with dual-blockchain for intelligent pricing system of smart grid. Computers & Security, 103, 102189.

[18]. Fan, H., Liu, Y., & Zeng, Z. (2020). Decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain. Sensors, 20(18), 5282.

[19]. Paillier, P. (1999, May). Public-key cryptosystems based on composite degree residuosity classes. In International conference on the theory and applications of cryptographic techniques (pp. 223-238). Springer, Berlin, Heidelberg.

[20]. Johnson, D., Menezes, A., & Vanstone, S. (2001). The elliptic curve digital signature algorithm (ECDSA). International journal of information security, 1(1), 36-63.

[21]. Kaur, R., & Kaur, A. (2012, September). Digital signature. In 2012 International Conference on Computing Sciences (pp. 295-301). IEEE.

[22]. https://github.com/data61/python-paillier

[23]. https://github.com/ShadowJonathan/fastecdsa-any

[24]. https://www.sageautomation.com/blog/traditional-grids-vs-smart-grids-why-were-making-the-shift

[25]. Gulich, O. (2010). Master's Thesis technological and Business Challenges of Smart Grids. Aggregator's Role in Current Electricity Market.

[26]. Official EU Commission Smart Grid Definition. https://s3platform-legacy.jrc.ec.europa.eu/smartgrids#:~:text=A%20Smart%20Grid%20is%20an,and%20security %20of%20supply%20and

[27]. Kumar, P., Lin, Y., Bai, G., Paverd, A., Dong, J. S., & Martin, A. (2019). Smart grid metering networks: A survey on security, privacy and open research issues. IEEE Communications Surveys & Tutorials, 21(3), 2886-2927.

[28]. Panajotovic, B., Jankovic, M., & Odadzic, B. (2011, October). ICT and smart grid. In 2011 10th International Conference on Telecommunication in Modern Satellite Cable and Broadcasting Services (TELSIKS) (Vol. 1, pp. 118-121). IEEE.

[29]. Uribe-Pérez, N., Hernández, L., De la Vega, D., & Angulo, I. (2016). State of the art and trends review of smart metering in electricity grids. Applied Sciences, 6(3), 68.

[30]. https://zenodo.org/record/5106455#.Yp-ZRXZBzIV

[31]. Hart, G. W. (1992). Nonintrusive appliance load monitoring. Proceedings of the IEEE, 80(12), 1870-1891.

[32]. Jawurek, M. (2013). Privacy in smart grids (Doctoral dissertation, Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU)). https://opus4.kobv.de/opus4-fau/frontdoor/index/index/docId/3645

[33]. Lin, H. Y., Tzeng, W. G., Shen, S. T., & Lin, B. S. P. (2012, June). A practical smart metering system supporting privacy preserving billing and load monitoring. In International Conference on Applied Cryptography and Network Security (pp. 544-560). Springer, Berlin, Heidelberg.

[34]. Clint, R. V. (2019). DLT/Blockchain as a building block for enterprise transformation. IEEE Engineering Management Review, 47(1), 24-27.

[35]. Fernández-Caramés, T. M., & Fraga-Lamas, P. (2018). A Review on the Use of Blockchain for the Internet of Things. Ieee Access, 6, 32979-33001.

[36]. Shalaby, S., Abdellatif, A. A., Al-Ali, A., Mohamed, A., Erbad, A., & Guizani, M. (2020, February). Performance evaluation of hyperledger fabric. In 2020 IEEE International Conference on Informatics, IoT, and Enabling Technologies (ICIoT) (pp. 608-613). IEEE.

[37]. Olivares-Rojas, J. C., Reyes-Archundia, E., Gutiérrez-Gnecchi, J. A., Cerda-Jacobo, J., & González-Murueta, J. W. (2019). A novel multitier blockchain architecture to protect data in smart metering systems. IEEE Transactions on Engineering Management, 67(4), 1271-1284.

[38]. Malik, H., Manzoor, A., Ylianttila, M., & Liyanage, M. (2019, December). Performance analysis of blockchain based smart grids with Ethereum and Hyperledger

implementations. In 2019 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) (pp. 1-5). IEEE.

[39]. Gope, P., & Sikdar, B. (2018). Lightweight and privacy-friendly spatial data aggregation for secure power supply and demand management in smart grids. IEEE Transactions on Information Forensics and Security, 14(6), 1554-1566.