MOBILE NETWORK AND CLOUD BASED PRIVACY-PRESERVING DATA AGGREGATION AND PROCESSING

MOHD RIZUAN BAHARON

A thesis submitted in partial fulfilment of the requirements of Liverpool John Moores University for the degree of Doctor of Philosophy

April 2017

TABLE OF CONTENTS

ACKNOWLEDGEMENTS

I would like to thank my director of study, Professor Qi Shi, for his support and guidance of this thesis. He has been providing invaluable suggestions and encouragement from the beginning of my research. This completed thesis would never be possible without his help and support.

Furthermore, I would like to thank my co-supervisors Dr. David Llewellyn-Jones and Professor Madjid Merabti for their continuous guidance in the process of conducting this research.

I am also grateful to the Department of Computer Science, Liverpool John Moores University for giving me an opportunity to further my PhD study and Universiti Teknikal Malaysia Melaka, Malaysia for their scholarship that helped me to achieve my goals.

Last but not least, I would like to thank my wife and my four daughters for their love, patience, understanding and encouragement in completing this thesis.

ABSTRACT

The emerging technology of mobile devices and cloud computing has brought a new and efficient way for data to be collected, processed and stored by mobile users. With improved specifications of mobile devices and various mobile applications provided by cloud servers, mobile users can enjoy tremendous advantages to manage their daily life through those applications instantaneously, conveniently and productively. However, using such applications may lead to the exposure of user data to unauthorised access when the data is outsourced for processing and storing purposes. Furthermore, such a setting raises the privacy breach and security issue to mobile users. As a result, mobile users would be reluctant to accept those applications without any guarantee on the safety of their data.

The recent breakthrough of Fully Homomorphic Encryption (FHE) has brought a new solution for data processing in a secure motion. Several variants and improvements on the existing methods have been developed due to efficiency problems. Experience of such problems has led us to explore two areas of studies, Mobile Sensing Systems (MSS) and Mobile Cloud Computing (MCC). In MSS, the functionality of smartphones has been extended to sense and aggregate surrounding data for processing by an Aggregation Server (AS) that may be operated by a Cloud Service Provider (CSP). On the other hand, MCC allows resource-constraint devices like smartphones to fully leverage services provided by powerful and massive servers of CSPs for data processing.

To support the above two application scenarios, this thesis proposes two novel schemes: an Accountable Privacy-preserving Data Aggregation (APDA) scheme and a Lightweight Homomorphic Encryption (LHE) scheme.

MSS is a kind of WSNs, which implements a data aggregation approach for saving the battery lifetime of mobile devices. Furthermore, such an approach could improve the security of the outsourced data by mixing the data prior to be transmitted to an AS, so as to prevent the collusion between mobile users and the AS (or its CSP). The exposure of users' data to other mobile users leads to a privacy breach and existing methods on preserving users' privacy only provide an integrity check on the aggregated data without being able to identify any misbehaved nodes once the integrity check has failed. Thus, to

overcome such problems, our first scheme APDA is proposed to efficiently preserve privacy and support accountability of mobile users during the data aggregation. Furthermore, APDA is designed with three versions to provide balanced solutions in terms of misbehaved node detection and data aggregation efficiency for different application scenarios.

In addition, the successfully aggregated data also needs to be accompanied by some summary information based on necessary additive and non-additive functions. To preserve the privacy of mobile users, such summary could be executed by implementing existing privacy-preserving data aggregation techniques. Nevertheless, those techniques have limitations in terms of applicability, efficiency and functionality. Thus, our APDA has been extended to allow maximal value finding to be computed on the ciphertext data so as to preserve user privacy with good efficiency. Furthermore, such a solution could also be developed for other comparative operations like Average, Percentile and Histogram. Three versions of Maximal value finding (Max) are introduced and analysed in order to differentiate their efficiency and capability to determine the maximum value in a privacy-preserving manner. Moreover, the formal security proof and extensive performance evaluation of our proposed schemes demonstrate that APDA and its extended version can achieve stronger security with an optimised efficiency advantage over the state-of-the-art in terms of both computational and communication overheads.

In the MCC environment, the new LHE scheme is proposed with a significant difference so as to allow arbitrary functions to be executed on ciphertext data. Such a scheme will enable rich-mobile applications provided by CSPs to be leveraged by resource-constraint devices in a privacy-preserving manner. The scheme works well as long as noise (a random number attached to the plaintext for security reasons) is less than the encryption key, which makes it flexible. The flexibility of the key size enables the scheme to incorporate with any computation functions in order to produce an accurate result. In addition, this scheme encrypts integers rather than individual bits so as to improve the scheme's efficiency. With a proposed process that allows three or more parties to communicate securely, this scheme is suited to the MCC environment due to its lightweight property and strong security. Furthermore, the efficacy and efficiency of this scheme are thoroughly evaluated and compared with other schemes. The result shows that this scheme can achieve stronger security under a reasonable cost.

Keywords

Fully Homomorphic Encryption, Partially Homomorphic Encryption, Mobile Cloud Computing, Mobile Sensing System, Privacy-preserving Data Aggregation, Data Security and Integrity, Users Accountability, Approximate-Greatest Common Divisor Problem, Discrete Logarithm Problem, Modular Arithmetic, Additive and Non-Additive Statistical Functions, Maximal Value Finding.

PUBLICATIONS RESULTING FROM THIS THESIS

JOURNAL

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Secure Remote Data Processing in Cloud Computing", *International Journal of Computer Theory and Engineering, Vol. 5, No. 6, Dec 2013*.

CONFERENCE PROCEEDINGS

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Enhancing Security of Data and Their Related Processing in Cloud Computing", *Proceeding of the 13th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking and Broadcasting*, pp. 22-26, 2012. Liverpool, United Kingdom.

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Secure Remote Data Processing in Cloud Computing", *3rd International Conference on Computer and Communication Devices* (ICCCD 2013), June 2013. Kuala Lumpur, Malaysia.

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Efficient and Secure Remote Data Storing and Processing", *12th European Conference on Information Warfare and Security* ECIW-2013, 11 – 12 July 2013. Jyväskylä, Finland.

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Secure Rendering Process in Cloud Computing", *Eleventh Annual Conference on Privacy, Security and Trust* PST2013, 10 July 2013 – 12 July 2013 Tarragona, Spain.

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Secure Video Transcoding in Cloud Computing", *2nd International Conference in Cloud Security Management*, 23 – 24 October 2014, Reading, United Kingdom.

Baharon, M., R., Shi, Q., Llewellyn-Jones, D., "A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing", *15th IEEE International Conference in Computer and Information Technology (CIT 2015)*, 26 – 28 October 2015, Liverpool, United Kingdom

POSTER

Baharon, M. R., Shi, Q., Llewellyn-Jones, D., Merabti, M., "Efficient and Secure Remote Data Storing and Processing", *12th European Conference on Information Warfare and Security* ECIW-2013, 11 – 12 July 2013. Jyväskylä, Finland.

LIST OF SYMBOLS AND ACRONYMS

LHE	Lightweight Homomorphic Encryption
APDA	Accountable Privacy-preserving Data Aggregation
FHE	Fully Homomorphic Encryption
HE	Homomorphic Encryption
FE	Functional Encryption
FFHE	Fully Functional Homomorphic Encryption
MCC	Mobile Cloud Computing
MSS	Mobile Sensing System
WSN	Wireless Sensor Network
PC-USSs	People Centric-Urban Sensing Systems
λ	Security parameter
$\mathbb{Z}_q$	Finite field of integers of modulo q
$\langle a \rangle$	A generator
$\|\|$	Concatenation
$\lceil n \rceil$	Round-up to the nearest integer greater than or equal to n
GCD	Greatest Common Divisor
CSP/CS	Cloud Service Provider or Cloud Server
AS	Aggregation Server
DC or u_i	Data Contributors or Mobile users
DU	Data Users
DA	Data Authority
Max^C	Count based Maximal Value Finding
Max^B	Bit-based Maximal Value Finding

Max^{H}	Hybrid version of both Count and Bit-based Maximal Value Finding
$\log_2(x)$	Logarithm of *x* with base 2
$KeyGen(p, r)$	Key generation algorithm based on a prime p and a random key r
$Enc(c_i, e_i)$	A symmetric encryption algorithm of input e_i using a secret key c_i
$Dec(c_i, e_i)$	A symmetric decryption algorithm of input e_i using a secret key c_i
n	Number of mobile users
τ	Length of data item d_i
Sk_f	A functional secret key
m	A master key

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 FOREWORD

The role of society has emerged to depend on a number of key sensing systems due to rapid technology advance over recent decades. In sensing systems, mobile users are becoming the main contributors to the data creation as they are actively contributing to data generating and processing via the use of mobile devices such as smartphones and tablets. With the emerging technology of mobile devices, more sensors can be embedded in the devices to collect, process and distribute data among people. As data is shared and distributed among people and processed by arbitrary applications, there is a serious risk of unattended leakage of personal sensitive information such as personal health information, user locations and sensitive personal images and videos. Such a scenario has led to the rise of concerns about user privacy and data security in mobile applications [1], [2].

To overcome such concerns, secure communication is required to protect user privacy and data security. One solution to this concern is to use an encryption scheme that enables data to be protected against unauthorised users and also allows the encrypted data to be processed by arbitrary applications without any need for decryption. Thus, in this chapter,

MSS and MCC are introduced, along with a discussion on the motivation behind the research in this thesis. The contributions to knowledge are also presented, in addition to the aims and objectives of the work, the methodology taken and the thesis structure.

1.2 MOBILE COMPUTING

Nowadays, mobile devices like smartphones and tablets are the most effective and convenient communication tools that are not bounded by time and place. Mobile devices are quickly gaining popularity due to the support for a wide range of applications like gaming, image processing, video processing and online social network services. According to International Data Corporation (IDC)'s statistics, the worldwide smartphones market grew 13% year over year in 2015, with totalling encompassing 341.5 million shipments of these devices. Furthermore, the revenue of mobile app stores was near $40 billion in 2015 and expected to reach $100 billion in 2020 [2]–[6].

Mobile users can enjoy various services from mobile applications like iPhone and Google apps. Such rapid progress of the mobile computing is achieved via the great advance of the smartphones in terms of computing resources in the past few years. The latest smartphones are equipped with high performance processors, RAM, sensors and data storage. According to [7], Sony Xperia S comes with a 1.5GHz Dual Core processor, 1GB RAM, 32GB data storage support and 1750mAh battery. Similarly, HTC One X has a 1.5GHz Quad-core processor, 1GB RAM, 32GB data storage support and 1800mAh battery.

However, according to [8], though mobile devices have significantly increased their capabilities, those devices still have serious limitations to execute complicated computations. This limitation can be shown by comparing mobile devices to desktop machines. Devices like iPhones, Android Series and Mobile Window Series have slightly lower processing capabilities and significantly reduced memory capacities. Furthermore, their storage capacities are at 5 to 10 times smaller as well as their network bandwidth. In some cases, mobile services or applications require extensive computing power and massive storage spaces to run on mobile devices [9]. Thus, the mobile devices with low resources compared to desktop machines will suffer from a negative effect on the service operation and quality [5].

1.2.1 MOBILE SENSING SYSTEMS

MSS has received a great deal of attention due to the rapid development in mobile phones via extending their capabilities to sense and process urban information. MSS may be composed of thousands of mobile nodes like smartphones and an aggregator like a CSP that are connected through the wireless medium. In MSS, mobile devices are embedded with sensors to sense surrounding data like health information in a distributed manner. The sensing data is collected and transmitted using a technique called data aggregation to increase the battery lifetime of mobile devices. Such an aggregation technique also increases the network lifetime by reducing the bandwidth size of data to be transmitted and battery energy usage [1], [10]–[12]. The collected data is transmitted to the aggregator for further process in order to generate some useful information, which benefits to the mobile users.

1.2.2 MOBILE CLOUD COMPUTING

The emerging technology in mobile devices mainly in terms of computing resources and storage spaces has led MCC to receive a great deal of attention for further exploration and enhancement. On the one hand, such enhancements allow initial processing on the data such as encryption and decryption by mobile devices before outsourcing the data to the CSPs. Those processing are essential to protect the data and preserve the privacy of mobile users against the CSPs. On the other hand, as cloud computing services have emerged to suit the mobile devices implementation, rich-mobile applications have been developed to attract billions mobile users to leverage those services in a very convenient way with very minimal efforts through wireless connections and mobile devices [13]. The widespread Internet infrastructure like Wi-Fi and high-speed mobile Internet access like 4G allow those applications provided by the CSPs (e.g. Amazon, Google and Microsoft Azure) to be widely available and reliable to be accessed and leveraged by mobile users [14] .

In the next section, we describe data outsourcing techniques that allow data to be gathered and processed by the AS like a CSP. Those techniques allow mobile devices, which have limited capabilities mainly in term of battery lifetime, to reduce their battery consumption by outsourcing the data and their related processing to the AS.

1.3 DATA OUTSOURCING

Data outsourcing is one of the best solutions, which allows data to be processed remotely in an efficient and secure manner. This data outsourcing is crucial particularly for resource-constrained devices like smartphones and tablets, by leveraging tremendous benefits provided by powerful and massive servers of CSPs. Furthermore, such a technique could improve the battery lifetime of mobile devices as they can avoid intensive data processing by themselves. In this thesis, two techniques of data outsourcing will be exploited, namely, data offloading and data aggregation.

1.3.1 DATA AGGREGATION TECHNIQUE

In MSS, data aggregation is a good technique for allowing useful data, which has been generated or sensed by mobile devices, to be gathered and processed efficiently by an AS for statistical analysis purposes. Such a technique is proposed to increase the battery lifetime of mobile nodes as the aggregated data reduces the number of bits transmitted, so as to reduce the bandwidth usage in the data transmission [15]–[17]. Reducing the bandwidth in the data transmission increases the battery lifetime of mobile devices [18]. Furthermore, this technique could improve users or data contributors' privacy and security against a malicious AS due to this technique mixing the collected data prior to the transmission to the AS so that it is difficult for the AS to identify the data from any particular individual contributor [11], [19], [20]. However, the technique does not preserve privacy among the data contributors as the data could be distributed from one distributor to another in a plaintext form [21]. Such an issue has received a great attention from researchers to propose privacy-preserving data aggregation techniques based on homomorphic properties [18], [22]–[25]. Such properties allow data to be processed in its ciphertext form without the need for decryption, so as to hide personal information of the data owner from the processor. Having a scheme with lightweight properties (i.e. demanding less resources from mobile devices for the encryption) would enable the concern to be addressed efficiently and securely.

1.3.2 OFFLOADING TECHNIQUE

In MCC, the battery lifetime of mobile devices can be extended by offloading their large tasks like complex mathematical calculations, search functions, image processing,

gaming applications, downloading files and security applications to powerful and massive servers [26]. The work in [5] shows that portable computers that execute their large tasks remotely can save significant amounts of battery power by up to 51%. Furthermore, a computation offloading technique avoids taking a long application execution time on mobile devices, which requires a large amount of power consumption [5], [27]. To apply this offloading technique, some previous works have been carried out to evaluate the effectiveness of the technique through experiments. The results demonstrate that the remote application execution can save energy significantly [28]. However, offloading may not be always the best option as some factors like the network connectivity, communication bandwidth, utilisation cost and mobile device energy have to be considered. Furthermore, security is one of the most prominent bottlenecks in the adoption of the offloading technique [7].

1.4 MOTIVATIONS

1.4.1 USER PRIVACY AND DATA SECURITY

The data shifted to other parties for storing and processing purposes can have implications on its privacy and security [29], [30]. Since the data is processed by the other parties, all the information about the data is disclosed to them. Such a factor makes it evident that all the data cannot be outsourced without considering the privacy and security implications [31]–[34]. One possible solution to outsourcing data is to transform it into an encrypted form. This solution can prevent unauthorized access to the data. However, if the data is encrypted, then it normally needs to be decrypted by a third party in order to perform operations on the data [34]–[40]. Such decryption allows the third party to gain some or all of the information about the data [41]–[43]. To overcome this problem, Fully Homomorphic Encryption (FHE) that supports arbitrary functions on ciphertext data seems to be one of the best solutions because of its properties for encrypted data to be processed without decryption [44], [45], [46]–[49]. Thus, a new lightweight HE scheme needs to be proposed to allow ciphertext data to be processed securely and efficiently by CSPs.

1.4.2 Problem on Homomorphic Encryption

A lot of FHE schemes have been proposed and improved upon but efficiency is still a big obstacle for their implementation. This efficiency is due to the fact that the complexity of the FHE schemes is normally high, which makes them not suitable to be implemented especially on small devices like smartphones, tablets, etc. [36], [37], [46], [48], [49]. Furthermore, performing data encryption before outsourcing the data requires additional processing by the mobile device and hence consumes extra energy [50]. Also the resource limitation of mobile devises means that security algorithms proposed for desktop machine environments may not directly work on the mobile devices. Therefore, a scheme with lower complexity is really needed to execute the encryption process on small devices without much resource degradation [51].

1.4.3 Limitation on Mobile Devices

Mobile devices nowadays have evolved in their specifications and technology. However, they still have a limited amount of processing power compared to desktop machines. To encrypt using a FHE scheme, the devices require a large amount of processing power [25], [52]. Furthermore, mobile devices need to be charged regularly as higher computation in generating the ciphertext can quickly decrease the lifetime of the battery. Moreover, a large ciphertext size produced by the encryption scheme requires a large bandwidth for transmitting the data that leads to decreasing the battery lifetime even further [1], [4], [7], [12], [50], [53]. Therefore, a lightweight scheme is demanded to allow ciphertext data to be generated by mobile devices resource-efficiently and can be processed by other parties without sacrificing the privacy of user data and disclosing its information [52], [53].

1.4.4 Accountability

As the privacy of mobile users is becoming a major concern in data outsourcing, preserving their privacy is really crucial. However, this privacy-preserving leads to another difficulty of supporting the accountability of the users' behaviour in order to deter some users from misbehaving, who could misuse the privacy preservation property to hide users' identities for malicious purposes. The accountability here is referring to holding mobile users involved in collecting and processing the data in a network

responsible for their behaviour. This accountability is important because the data collected must be genuine and processed correctly to ensure the integrity of the final result.

In mobile networks, participating users could consist of both well-behaved and misbehaved users [54]. The misbehaved users are those who have a specific interest outside their responsibility [55] for providing the requested data to an AS. They might want to discover other users' sensor data and use it for their own purposes. Furthermore, the misbehaved users may contribute fake data [56] that leads to a wrong result at the end of the process [57]. Identifying such users in the data aggregation process is very challenging. An appropriate technique needs to be proposed to detect and hold the misbehaved users accountable for their misbehaviour, while minimising device performance degradation.

1.5 RESEARCH QUESTIONS

The motivations stated above have led us to list some research questions that need appropriate solutions to be figured out from this research work. Those questions are listed as below:

1.5.1 In MSS, the integrity of aggregated data received from mobile users could be verified if the data is in a plaintext form. However, such a form of data reveals the data content as well as the privacy of the data contributors to the aggregator. Thus, how can a scheme be provided to verify the integrity of the aggregated data without revealing the content and the privacy of the data contributors as well as reducing energy consumption of mobile devices?

1.5.2 In MSS, data contributors may consist of both well-behaved and misbehaved nodes. Tracking such misbehaved nodes in a privacy-preserving environment is crucial and challenging due to the inherent limitations of mobile devices. Thus, how can we track down those misbehaved nodes in such an environment without increasing the energy consumption of mobile devices?

1.5.3 In MSS, the aggregated data needs further processing to produce some meaningful results to the data users. Thus, how can a scheme be provided to support additive

and non-additive functions without revealing the data content and increasing the energy consumption of mobile devices?

1.5.4 In MCC, the generated data is transmitted directly by mobile users to a Cloud Server (CS) for data processing. As MCC uses a different structure and computational functions compared to MSS, a different encryption scheme needs to be implemented. Thus, how can we provide a scheme that allows a CS to process the data arbitrarily without revealing the data content, compromising the privacy of mobile users and increasing the energy consumption of mobile devices?

1.5.5 All the aforementioned schemes must be secured against potential threats and attacks in both MSS and MCC environments. Thus, how can it be proved that the proposed schemes are secured against those threats and attacks?

1.6 AIM AND OBJECTIVES

The research questions stated above have led us to propose the development of two novel schemes for the MSS and MCC application scenarios. The key aim of the schemes is to enable data to be collected, stored and processed in its ciphertext form efficiently and securely by mobile devices. This needs to provide strong data security and preserve user privacy while holding the users accountable for their behaviour.

This research aim is achieved through the completion of multiple research objectives, including:

1.6.1 To develop a novel scheme, which can verify the integrity of aggregated data in MSS while preserving mobile users' privacy. This objective has been achieved by using an extensive literature survey on privacy-preserving data aggregation techniques to provide the proposed novel APDA, which is able to verify the integrity of aggregated data without disclosing any information related to mobile users to the CSPs as to be described in Chapter 2 and 4.

1.6.2 To propose an innovative scheme that can detect any misbehaved nodes in a data aggregation process in MSS without compromising individual nodes' data privacy. Such an objective has been achieved by using the proposed APDA with various versions to detect any misbehaved nodes in the aggregation process

certainly and efficiently as to be described in Chapter 4. In addition, we have evaluated those versions to fulfil this objective as to be described in Chapter 5.

1.6.3 To extend the functionality of the above proposed scheme to support additive and non-additive statistical functions over encrypted data certainly and efficiently. We have developed an extended APDA by introducing three versions of a maximal value function in Chapter 4 and evaluated those versions in Chapter 5 to fulfil this objective.

1.6.4 To design a new FHE scheme, which has a lightweight property applicable to resource-constrained devices in MCC. This objective has been achieved by the proposed new LHE scheme that supports both addition and multiplication in an encrypted form as to be described in Chapter 6. Furthermore, we have evaluated this scheme to fulfil this objective and described it at end of this chapter.

1.6.5 To design secure schemes without scarifying the privacy of mobile users during data outsourcing. This objective has been fulfilled by the extensive security analyses provided on the proposed schemes against several attacks launched by a curious AS or malicious devices as well as attacks on the schemes themselves such as key recovery attacks and a many time pad attack. We have shown in Chapter 5 and 6 that our schemes are secured against those attacks.

1.7 NOVEL CONTRIBUTIONS

This thesis offers two kinds of homomorphic encryption schemes, which have some essential characteristics that are desired by many applications on processing data in an encrypted form. The main novel contributions of our research work are as follows:

1.7.1 In MSS, mobile users or their devices may consist of both well-behaved and misbehaved ones [21], [54], [58]–[65]. Holding a device accountable for its behaviour in such an environment is a very challenging task as the data is encrypted to preserve the users' privacy. Therefore, our first contribution is to propose a novel privacy-preserving data aggregation technique called APDA. This scheme allows data to be aggregated in a privacy-preserving manner with an improved functionality that enables misbehaved nodes to be detected by an AS. Three approaches are introduced with varying capabilities. APDA^{W} is proposed

to efficiently determine misbehaved nodes with less certainty. It offers lower computation and communication as there is no digital signature included in the ciphertext data. Conversely, $APDA^S$ is proposed to certainly pinpoint which nodes have misbehaved, but its efficiency is lower. It requires high computation and communication as signatures are involved to detect misbehaved nodes with certainty. To take advantage of the strengths of these two methods and rectify their weaknesses, $APDA^H$ is proposed by mixing both versions to certainly pinpoint the misbehaved nodes with better efficiency.

1.7.2 In data aggregation, the purpose of having aggregated data is to generate additive and non-additive statistical results like Count, Sum, Average, Min/Max and Percentile. A number of researchers have proposed schemes to allow such functions to be executed on ciphertext data. Nevertheless, their limitations on efficiency and functionality are still the major challenges for their implementation [19], [21], [23], [66]–[73]. Thus, our second contribution is to propose an extended version of ADPA to support maximal value finding with better security and efficiency. Such an extended version could be developed for other comparative operations but is excluded from this thesis. Three modes of the scheme for determining a maximum value with different capabilities are proposed. Max^C is designed to accurately determine the maximum value, but its efficiency is lower. Conversely, Max^B is developed to efficiently determine the maximum value with less certainty. To take advantage of both methods, Max^H is proposed as a mixed version of the methods to more efficiently determine the maximum value with certainty.

1.7.3 The existing FHE schemes cannot be directly implemented by resource-constrained devices as complexity and efficiency are the big obstacles for their implementation [34], [37], [44], [47], [48], [58], [74], [75]. The main reason of such complexity and inefficiency is that those schemes encrypt every individual bit and require a large public key size for each encryption [35], [76]. These conditions rapidly reduce the battery lifetime of mobile devices during encrypting the data. Thus, implementing such schemes requires extra computing resources to encrypt and decrypt the data, more bandwidth for transmitting the data and more

storage spaces for storing the data [1], [11], [38], [46], [59]. Therefore, our third contribution is to propose a new LHE scheme with less complexity but remains strong security to be implemented in MCC, so as to extend the battery lifetime of mobile devices. This scheme supports arbitrary functions on ciphertext data and it works well as long as the size of noise is less than that of the encryption key. This flexibility of the key size selection enables the scheme to incorporate any computation functions with better efficiency. As a result, such a scheme allows mobile devices, which have limited computing resources and storage spaces, to leverage rich mobile applications provided by CSPs in an efficient and secure manner.

1.8 THESIS STRUCTURE

The research detailed in this thesis is divided into six (6) chapters. An overview for each of the remaining chapters is provided as below.

Chapter 2 – Mobile Sensing Systems and Mobile Cloud Computing: This chapter investigates associated research into processing data in an encrypted form. Furthermore, we define what MSS are, how they function, what weaknesses they tend to have and how they are currently protected against threats using data aggregation and partially HE schemes. Moreover, MCC is also explored to give an overview of how mobile applications could benefit from the emerging technology of cloud computing.

Chapter 3 – Mathematical Foundation: This chapter will draw some of the mathematical concepts that have been used and utilised throughout this thesis. Furthermore, this chapter will give an overview into the fundamental concepts and the development of FHE schemes. In addition, research methodology will be presented at the end of this chapter.

Chapter 4 – Accountable Privacy-preserving Data Aggregation: In this chapter, the design of APDA and its extended version to support maximal value finding on aggregated data will be presented. It will entail an outline of its specification, a detailed discussion on its architecture and an explanation of its operation.

Chapter 5 – Security Analysis and Performance Evaluation of APDA and Max Methods. This chapter provides security analysis and performance evaluation on the proposed schemes. This chapter also describes a detailed analysis of the data aggregation results.

A visual interpretation of the data aggregation process and a justification of the results obtained will also be put forward in this chapter.

Chapter 6 – A New Lightweight Homomorphic Encryption Scheme: In this chapter, we propose a new LHE scheme that allows mobile users to leverage rich mobile applications provided by CSPs. The scheme will be compared with other schemes to demonstrate its efficiency. Furthermore, an in-depth security analysis will be provided to show that although the scheme has lightweight features, it can still provide stronger security to protect the encrypted data.

Chapter 7 – Conclusions and Future Work: In the final chapter of this thesis, the research presented here will be concluded by discussing the accomplishments of this research work. This chapter will also highlight how future work can be built on for research purposes.

1.9 SUMMARY

This chapter has provided an overview of the thesis related to MSS and MCC. The issues of user privacy and accountability as well as data security and integrity, which are becoming major concerns in both MSS and MCC, have motivated us to conduct the research presented in this thesis for better solutions. To achieve this, five project objectives have been set out and the accomplishment of these objectives has resulted in three novel research contributions, which are the APDA scheme, its extended version and the LHE scheme. APDA is proposed to support the accountability of users' behaviours, while its extended version is devised to allow some statistical calculations to be executed efficiently in a privacy-preserving manner. Furthermore, the new LHE scheme is developed to suit MCC needs on lightweight and strong security properties. Also the thesis structure is provided at the end of this chapter to briefly describe the following chapters in this thesis. The next chapter will thoroughly describe how both MSS and MCC are implemented and figure out all benefits and limitations inherent in both paradigms.

CHAPTER 2

MOBILE SENSING SYSTEMS AND MOBILE CLOUD COMPUTING

2.1 INTRODUCTION

The emerging technology of computing paradigms has opened up a new way for data to be shared and benefited among mobile users. Support from cloud computing has also signified the rise of MSS and MCC to mobile users as all burdens of heavy computations are taken and managed by the cloud server. In addition, the wide spread of network communication links such as Wi-Fi access and high-speed technology like 3G/4G have contributed to this achievement. Nevertheless, as data is user-generated content, it may consist of sensitive information related to mobile users. Sharing and disclosing such data to other parties like other mobile users or a cloud server can lead to a disaster for the data contributors as such parties are assumed to be untrusted. They may misuse the data for their own benefits. Experiencing security issues such as privacy, confidentiality, accountability and integrity of the shared data has prevented mobile users from contributing their data into such systems as there is no guarantee how their data will be protected. Thus, this chapter introduces and explores two areas of studies, which are MSS and MCC.

MSS is gaining more popularity as more sensors can be embedded in the mobile devices so that the devices can provide better sensing functionality than wireless sensor nodes [1], [23], [67], [71]. Sensors that are built into mobile phones on the one hand, are more powerful than traditional sensor nodes as they can be conveniently recharged. Furthermore, such sensing systems can not only sense a surrounding environment like road congestion and crowd in an event but also information related to mobile users like health information and locations. In an urban area, the sensed data can be collected on a massive scale by leveraging thousands of individual mobile phones and a near-pervasive wireless-network infrastructure like Wi-Fi access [56].

On the other hand, some mobile applications like gaming and complex mathematical algorithms require extensive computing resources and massive storage spaces to be executed by mobile devices. Running such applications on mobile devices is not practical as it reduces the battery lifetime rapidly. Thus, MCC is introduced to mobile devices in order to leverage the tremendous advantages provided by cloud computing to resolve such limitations. MCC is one of the interesting research areas in cloud computing. This is due to the main objectives of cloud computing, which are to facilitate access to ample computing resources by small and resource-constrained devices like smartphones and tablets. Furthermore, mobile users can experience with rich mobile applications through the widespread third/fourth generation (3G/4G) mobile networks and Wi-Fi access. According to [53], MCC can be defined as "*a rich mobile computing technology that leverages unified elastic resources of varied clouds and network technologies toward unrestricted functionality, storage and mobility to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle*".

In the following sub-sections, service models, benefits and limitations inherent within both MSS and MCC will be explained. Such explanations are essential to understand the issues in both areas of studies in order to provide solutions to their existing problems.

2.2 MOBILE SENSING SYSTEMS

MSSs differ significantly from traditional WSNs that focus on urban environments for sensing and collecting data. In MSSs, sensing devices belong to individuals with diverse

interests, while in WSNs, the sensor nodes are owned and managed by a single authority. Furthermore, mobile devices have better computing resources and storage spaces than sensor nodes and can be recharged regularly. Moreover, MSSs feature dynamic node mobility, while WSNs mostly consist of static sensing elements, which are either physically placed or randomly distributed across a target area of interest [77].

On the other hand, mobile sensing data is more related to personal information rather than environmental, agricultural or industrial monitoring data in WSNs. In MSSs, humans act as data contributors, while in WSNs, sensor nodes are responsible for sensing and generating data to be collected and processed by an AS [78]. Though both MSSs and WSNs have significant differences, both types of application may gather sensed data using the same approach called data aggregation as it can increase the overall lifetime of the applications. The reason is that, such an approach requires less bandwidth for transmitting the aggregated data and uses less battery energy.

2.2.1 System Approach

A MSS leverages a data aggregation approach in order to increase its overall network lifetime. For example, in hierarchical networks, special nodes are used to aggregate the sensed data in order to decrease the number of data packets transmitted to the root node like a CSP. As a result, the energy efficiency of the whole network can be improved as less bandwidth requires for data transmission. Such networks consist of several categories like cluster-based, chain-based and tree-based networks. In brief, each category of hierarchical networks can be described as follows.

Cluster-based networks consist of a cluster head, gateway nodes and ordinary sensor nodes as shown in Figure 1. In this network, sensor nodes are divided into clusters and the data that is sensed by sensor nodes is sent to a high end sensor called a cluster head in each cluster. A cluster head aggregates all the data and transmits the aggregated data via a tree structure to the base station. In addition, the gateway nodes link adjacent clusters. On the other hand, chain-based networks comprise of sensor nodes, leader heads and a sink node as illustrated in Figure 2. In such networks, each sensor node communicates and transmits the sensed data only to the closer neighbour to form a chain using an algorithm like a greedy algorithm [79]. This formed chain will lead to the leader

node, which collects and submits the data to the sink node [80]. Such a setting reduces the distance of data transmission so as to decrease the energy dissipation per round [81].

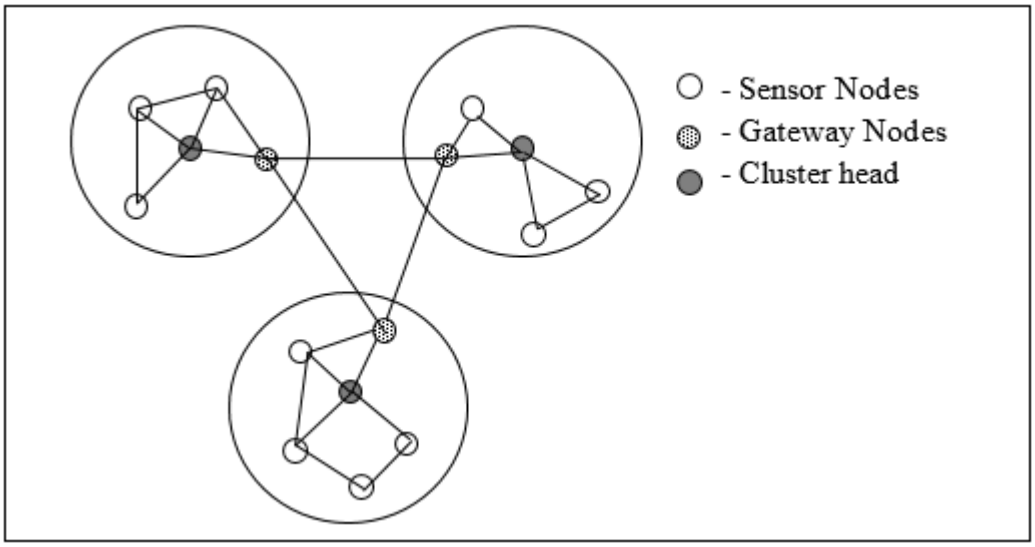


Figure 1: Cluster-based Networks

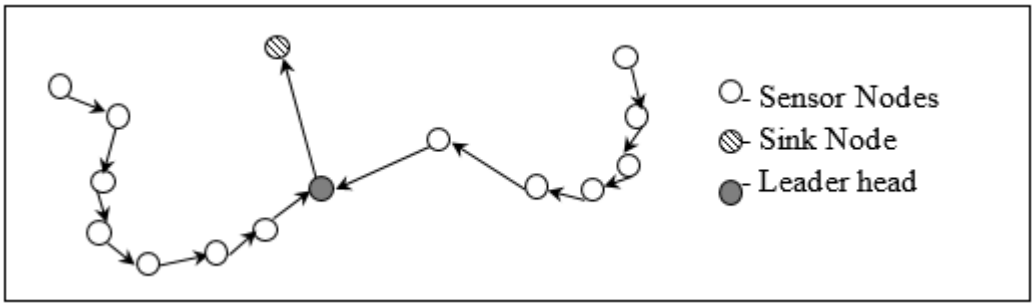


Figure 2: Chain-based Networks

In tree-based networks, all nodes are organized into a tree or in a hierarchical form as shown in Figure 3. The intermediary nodes help to perform data aggregation and transmit the aggregated data upwards towards the root node [21].

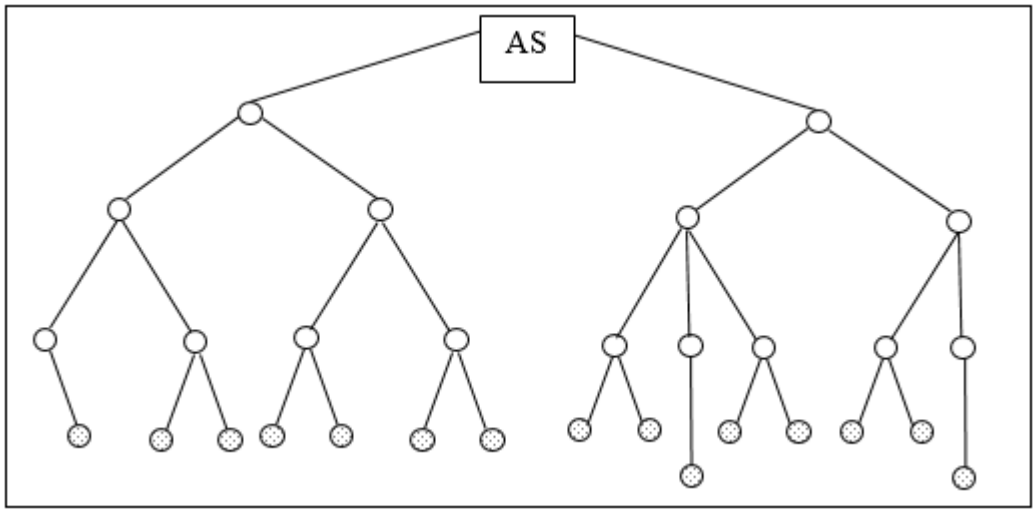


Figure 3: Tree-based Networks

- Leaf nodes (children): resource-limited sensor nodes/data collecting sensors.

- Intermediary nodes (parent/grandparent): act as intermediary data aggregators, aggregating the data received from their child nodes and forwarding the aggregated results to their parents.

AS - Tree root: a resource-enhanced server.

2.2.2 BENEFITS

A MSS implements data aggregation to gather useful data and expresses the result in a summary form such as additive and non-additive statistical calculations. It has become one of the fundamental processing procedures for saving the energy in WSNs. The main goal of data aggregation algorithms is to gather and aggregate data in an energy efficient manner so that the network lifetime will be prolonged and reduce the latency of receiving data by the sink. The overall network lifetime is increased because the data aggregation implementation reduces redundancy in the transmitted data. Furthermore, such a technique can solve the implosion and overlap problems in data centric routing.

Data aggregation also reduces the number of data transmissions and retransmissions. As a result, this technique will directly decrease energy consumption and hence increase the overall network lifetime [131]. Moreover, this technique reduces the number of bits

transmitted, thereby decreasing the bandwidth usage. In addition to reducing the number of data transmissions and retransmissions, data aggregation will directly reduce the number of collisions and the wastage of time. Finally, the data aggregation increases robustness and accuracy of information obtained from the network [16]. Nevertheless, such an approach still has some major issues related to the aspects of data security and integrity as well as user privacy and accountability.

2.2.3 CHALLENGES AND LIMITATIONS

Data aggregation offers great advantages to prolong a network lifetime especially in WSNs and MSS. Nevertheless, this technique also presents significant challenges to the data and mobile users. One of the major challenges is the security of the sensed data itself as the data is tied to specific individuals. For instance, an adversary wishes to discover the data in order to reveal some information that can be used to identify a user's location.

Furthermore, the integrity of the sensed data is also important. The data that has been aggregated must be genuine to ensure that some statistical results could be correctly generated. CarTel [8] and VTrack [3] are the examples of data integrity breaches that use traffic statistics such as an average speed as an indicator of congestion to help system users to do route planning. A malicious driver may manipulate the aggregate result to prevent other users from choosing its current road.

Moreover, in a MSS, mobile data is shared among users to allow data to be aggregated and processed by an AS (or a CSP). Nevertheless, users are reluctant to share their data as their privacy like personal and location information could be disclosed to the system. In [77], a study of the relationship between air quality and public health is one of the examples, where the study required some information related to personal health data such as heart rates, blood pressure levels and weights at different sections of an urban area. Mobile users were reluctant to reveal their personal data if their data would be used to invade their privacy. In addition, as such data is very crucial to determine the health condition of the data owner, the data should be genuine and reliable. Such examples highlight the necessity for verifiable privacy-preserving data aggregation techniques that can ensure strong user privacy and also aggregation integrity [23], [56].

Finally, the system may consist of well-behaved and misbehaved users with an easy access to wireless networks that may have a weak security control. A node could misbehave by refusing to forward necessary data packets in order to save its energy (selfishness) or simply degrade network performance (maliciousness). Furthermore, this node may modify their devices in terms of software or hardware to increase their individual network benefit while expending less resources. To prevent such concerns, every network device can be equipped with a tamper-proof hardware. However, this accessory requires extra costs as such a hardware is not cheap [8]. Hence, the network must be able to detect instances of misbehaviour, identify the misbehaving nodes and revoke them from the aggregation process [55]. Such challenges provide a new research area of study to propose solutions to the stated problems.

2.2.4 Existing Solutions and Limitations

Some contributions have been made through several projects like MetroTrack [82] and SensorPlanet [12] to accelerate the rise of MSSs. For example, SensorPlanet is a Nokia initiated global research framework for mobile device-centric WSNs [12]. These projects have attracted several universities and companies to work together to facilitate research on data analysis and mining, visualization and machine learning.

The UCLA Urban Sensing initiative is proposed to compose a sensor-based recording of users' experiences and environments [12]. Furthermore, the Intel-sponsored Urban Atmospheres project is proposed to explore the human condition by using embedded sensors in mobile devices. Also the Massachusetts Institute of Technology Cartel project provides a mobile communications infrastructure based on car-mounted communication platforms exploiting open Wi-Fi access points in a city, and it provides urban-sensing information such as traffic conditions [23]. Moreover, the CitySense project involving Harvard University, the city of Cambridge (Massachusetts, USA) and BBN Technologies provides a static sensor mesh offering similar types of urban sensing data feeds [31].

Though such projects have facilitated the rise of MSSs, the key issue of how to preserve the privacy of mobile users is still a big challenge on the urban sensing implementation. Thus, a number of solutions have been proposed to deal with the issue as described in Sub-section 2.2.3. AnonySense is a privacy-aware architecture for realising pervasive

applications that is based on collaborative and opportunistic sensing by personal devices. This solution has focused on protecting the privacy of participants of sensing data while allowing their devices to reliably contribute high-quality data to the large-scale applications. This scheme incorporates new privacy-aware techniques into secure tasking and reporting, and demonstrates that their solution consumes minimum device resources. The scheme can also protect privacy against outside eavesdroppers and other mobile nodes [59]. However, according to Ziling, E., et al., such a solution does not protect privacy against other ASs. The scheme also does not support additive and non-additive aggregation functions on ciphertext data. Furthermore, the scheme does not support node failure and data loss resilience, which refers to whether the AS can compute a correct aggregate result when a few mobile nodes become offline or a few messages get lost [67].

To overcome the disadvantages of AnonySense, PriSense has been proposed to provide a novel solution to privacy-preserving data aggregation in People-Centric Urban Sensing Systems (PC-USSs) [77]. This scheme is based on the concept of data slicing and mixing to preserve the privacy of mobile users against any curious ASs. It also supports a wide range of statistical additive and non-additive aggregation functions on ciphertext data. Furthermore, the scheme protects privacy against outside eavesdroppers, other mobile nodes or the ASs. Nevertheless, according to Ziling, E., et al., this scheme does not support node failure and data loss resilience [67].

To improve the weaknesses of PriSense, PDA has been proposed in [77]. It is a novel privacy-preserving robust data aggregation scheme in PC-USSs. The scheme is designed based on K-anonymity, homomorphic encryption and secret sharing. This scheme also supports a wide range of statistical additive functions without leaking individual sensed data. Furthermore, PDA is robust to node failure and data loss. Moreover, the scheme protects privacy against outside eavesdroppers, other mobile nodes or the ASs [67]. However, such a scheme does not support non-additive functions and provides an integrity check to the aggregated data, thus preventing the non-genuine data to be aggregated.

In another work, VPA has been proposed to provide a novel peer-to-peer based solution to verifiable privacy-preserving data aggregation in PC-USSs [23]. It achieves strong user privacy by letting each user exchange random shares of its data with other peers, while

ensuring data integrity through a combination of Trusted Platform Module and homomorphic encryption Message Authentication Code (MAC). VPA leverages the same technique as in PDA, which is the slicing technique to provide data privacy in the aggregation process. Furthermore, VPA can support a wide range of statistical additives using VPA^{+} and non-additive aggregation results by implementing VPA$^{\oplus}$.

However, VPA has to accept that the Max/Min aggregation functions naturally disclose some information about a user's data: any user's data will be smaller than or equal to the maximum value d_{max} and larger than or equal to the minimum value d_{min}. The reason why the VPA cannot prevent this kind of privacy breach is due to the aggregate functions themselves. As a result, VPA ignores such a natural privacy breach and focuses on the loss of privacy occurring in the query process [23]. On the other hand, VPA implements a signature for each data encryption and thus requires more computing resources to compute the signature, i.e. hash values. Furthermore, VPA also provides an integrity check on the aggregated data as in PDA. However, if the integrity check has failed, the scheme is not able to detect which nodes have misbehaved, and this limitation does not allow the scheme to revoke such misbehaved nodes from a further aggregation process.

As a conclusion, though all the aforementioned schemes guarantee mobile users' privacy while some can securely support statistical analyses on the aggregated data, their scheme efficiency still can be improved to prolong the network lifetime of MSS. Furthermore, the most essential flaw of the proposed schemes is that, they cannot support the accountability of the participating users for their behaviour. Such a limitation prevents the system from detecting the misbehaviour instantly, identifying the misbehaving nodes and revoking them from a further aggregation procedure. These reasons have led us to propose a novel technique that can support participating users' accountability in a privacy-preserving manner.

On the one hand, the emerging technology of mobile devices has allowed rich-mobile applications provided by CSPs to be accessed and leveraged by mobile users via wireless links. On the other hand, such improvements allow security elements like encryption and decryption processes to be executed by mobile devices to improve the security of the outsourcing data. Thus, the next section describes the implementation of mobile devices in MCC and how such devices could be benefited by cloud technology to facilitate mobile

users to utilise all services provided by CSPs in a privacy-prserving manner with less resources degradation.

2.3 MOBILE CLOUD COMPUTING

Mobile devices are becoming the major computing component nowadays thanks to their prevalence of ubiquitous connection via the high-speed network technology such as 3G/4G and Wi-Fi access. Nevertheless, limitations on computing resources, storage spaces and battery lifetime lead to the need for external support architecture like cloud computing as it can provide elastic resources to applications on those devices. The implementation of cloud computing by mobile devices has led to the emergence of several computing paradigms such as Mobile Cloud Computing (MCC) [26], Fog Computing [83]–[88] and Mobile-edge Computing (MEC) [89]–[93]. Such paradigms share the same concepts and promise better solutions to mobile devices by alleviating all the burdens of heavy computations and complex tasks to the cloud servers. Nevertheless, those paradigms have specific structures that may only suitable for particular services requested by mobile users. Moreover, such paradigms may face other limitations such as security issues that need further improvements.

For instance, Fog computing and MEC provide some advantages mainly in providing low latency and improving Quality-of-Service (QoS) for streaming and real-time applications as they provide processing and storage resources to lower layers of the computing architecture [84], [88], [92]. However, such paradigms introduce other drawbacks in term of reliability, as the lower layer devices are not reliable compared to a cloud server [83], [93]. In addition, those devices' connectivity cannot be guaranteed, which implies the requested service cannot be fulfilled even its computational hardware is working [83], [92]. Furthermore, the edge nodes usually consist of end devices, which are smaller and less powerful than a massive and powerful cloud server [85], [94], [95]. Thus, applications demanding heavy computations need to be processed by cloud servers. Such requirements lead to the need for MCC implementation and its detail is provided in the sub-sequent sub-section.

2.3.1 ARCHITECTURE

MCC is designed on a standard cloud service model. Gartner [33] has defined cloud computing as *a style of computing where massively scalable IT-enabled capabilities are delivered 'as a service' to external customers using Internet technologies*. According to National Institute of Standards and Technology (NIST) USA, cloud computing is *a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources like networks, servers, storage, applications and services that can be rapidly provisioned and released with minimal management effort or service provider interaction* [53].

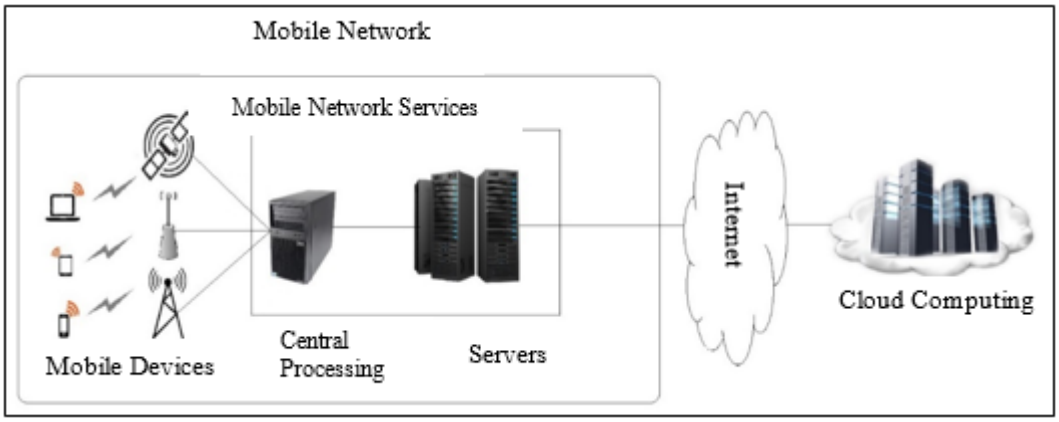


Figure 4: Mobile Cloud Computing Architecture

In MCC, mobile devices obtain services from a centralized cloud like Amazon Web Services (AWS) and Google [72]. The cloud is located in a remote centralized cloud infrastructure as illustrated in Figure 4. Mobile devices can access the data centre resources through a near-pervasive network infrastructure such as Wi-Fi Access Points (APs) or 3G/4G cellular networks. In this setting, the cloud represents a third party to provide services to the original content providers and mobile devices. By using the cloud, a mobile device can offload part or all of its workload to the cloud in order to process its data by utilising the massive cloud resources [14].

2.3.2 BENEFITS

(i) **An Improvement on computing power and battery resources**

Even though the capability of mobile devices has been improved, certain facilities like wireless communication appear to be highly power-consumptive. Several techniques in [96] have been introduced to reduce the power consumption and save energy during communication. Furthermore, the battery lifetime can be extended by offloading large tasks for remote processing. The work in [96] shows that the portable computers executing their large tasks remotely can save up to 51% of battery power. Dinh, H., T., et al. [27] proposed a computation offloading technique to outsource large computations and complex processing from resource-limited devices to massive resource machines like servers in clouds. Such a technique prevents taking long application execution time on mobile devices that requires a considerable amount of power consumption on the devices.

(ii) **An improvement on a capacity of data storage**

Mobile devices suffer from limitations on storage capacity. Thus, MCC is developed to enable mobile users to store or access a large amount of data on the cloud through wireless networks. The Amazon Simple Storage is one of the services provided to facilitate the data storage of mobile users. Another example is the Image Exchange which utilizes the large storage space in clouds to serve mobile users. By using the cloud storage as a service, the users can save significantly their mobile devices' energy and storage space due to all data being stored and processed in the clouds [27].

(iii) **Cost saving**

In MCC, the costs like time, energy and fees need to be considered carefully prior to offloading mobile data to the clouds. This silent commitment can ensure that mobile users enjoy all services provided by the clouds under a reasonable cost without degrading their mobile device performance. Furthermore, Walker et al. [4] have proposed a model that highlights the advantages of leveraging storage from the cloud rather than to buy extra computing resources and spend more on the software [9].

2.3.3 CHALLENGES AND LIMITATIONS

In MCC, mobile users leverage all services provided by the clouds through web applications. The data that has been outsourced to the cloud can have implications on privacy and security as the data is going out of users' control into the public domain. Furthermore, as the data is stored and managed in the cloud, security and privacy settings

depend on the IT management provided by the cloud. On the other hand, the CSP typically works with many third party vendors, so there is no guarantee how these vendors may safeguard the data. Moreover, data on the cloud may be stored at multiple locations across different states and countries. Thus, the security of the data depends on its location, and regulations for accessing the data might differ from one country to another [50]. As a result, the implications on security and privacy of the data need to be considered prior to the data being outsourced to the cloud.

To remedy such concerns, an encryption technique is one of the possible solutions to protect data in the cloud. This technique prevents unauthorized users from access to meaningful data even when the storage is breached at the cloud end. However, if the data is encrypted, then it has to be decrypted at the CSP because of the need to perform operations on the data. On the other hand, performing the encryption before sending the data to the cloud requires some additional processing on the mobile device and consumes extra energy [50]. Furthermore, as MCC is a cloud-based environment, all security weaknesses of mobile systems are inherited in MCC. Such weaknesses require a different approach to be implemented due to resource limitations on the mobile devices. [52].

2.3.4 Existing Solutions and Their Limitations

Normally, providing strong security and high privacy means requiring more computing resources and energy consumption. Furthermore, increasing data security decreases the functionality that can be executed on the data [97], [98]. Due to those reasons, a lot of research has been conducted to provide security and improve the privacy of outsourced data with the consideration of energy consumption and solution efficiency. To ensure the integrity of users' data stored in cloud servers with improved energy efficiency, Itani et al. [99] have proposed an energy efficient framework for mobile devices using the concept of incremental cryptography and trusted computing. Furthermore, Jia et al. [43] have introduced a secure data service that outsources data and security management overhead to a cloud in a trusted mode. Also proxy re-encryption and identity-based encryption have been implemented to provide data privacy and fine-grained access control with small cost and communication overheads. Moreover, Shukla et al. [100] have proposed a scheme for smartphones to ensure the security and integrity of mobile users' files stored in cloud servers with low energy consumption.

Even though the aforementioned schemes have some advantages to enhance the battery lifetime of mobile devices, such schemes are based on trust, which is contrary to our MCC setting, where we assume that CSPs are not trusted. Thus, those schemes are not suitable for implementation in our setting or environment. To overcome such a problem, privacy-preserving encryption schemes [60], [101], [102] have been proposed and are summarised below.

(i) **Data encryption and related problems**

To ensure data integrity in cloud-based applications, data needs to be protected using strong encryption algorithms like RSA. This security measure can also be done through any common Public Key Infrastructure (PKI). This form of encrypted data is good for storage in cloud but rather costly to process [33]. This argument is also supported by other researchers who attempt to avoid the use of primitive encryption schemes to protect data [33], [102]–[106].

According to Gentry [107], encrypting one's data seems to nullify the benefits of such data storage servers like clouds. Data encrypted with ordinary encryption schemes makes it virtually impossible for someone to manipulate the underlying data in any useful way without being able to decrypt them. Furthermore, from the clouds' point of view, encrypting data in the cloud storage prevents the data itself from being processed by SaaS or PaaS applications such as Salesforce.com or Google Apps. Moreover, encryption may not be suitable since this technique prevents indexing or searching on the data [33], [103]. Though such problems have received serious attentions from [108]–[116] to work on searchable encryption, the proposed solutions still have limitations and need further improvements in order to be implemented in clouds [116]–[118].

(ii) **Alternative methods and related problems**

The difficulty of processing data in its encrypted form has led reseachers to come out with other techniques like a technique based on interaction [32], [119]–[123], programme obfuscation [124], [125], data and file fragmentation [126], [127] and data concealment [105]. Sahai [124] proposed a solution based on how interaction and secure hardware can help to compute encrypted data. Indeed, in his joint work with Barak, Goldreich, Impaggliazzo, Rudich, Vadhan, and Yang, they show that encrypted programs (program obfuscation) are in general impossible [124].

Mahmood [103] has proposed a method for data fragmentation and suggested limiting the amount of data that should be decrypted for processing in clouds. Similarly, Tian et al. [126] have conducted research to improve the assurance and scalability of a heterogeneous distributed system by dividing storage servers into different server groups and developing a file fragmentation and allocation approach. However, this approach has significant drawbacks in terms of the time cost to reconstruct a file from its fragments, which is obviously greater than non-fragmentation storage methods. In addition, through this approach, performance degradation becomes inevitable. In another research work, Delettre et al. [105] have proposed a data concealment component to solve the issue of the confidentiality of data stored in cloud databases. Although the poposed solution is efficient, it is necessary to improve the data marking method to avoid concatenation of the mark with the data. As alternative methods have such significant drawbacks, other techniques are required to efficiently protect data and their related processing in clouds.

(iii) **Homomorphic Encryption and related problems**

A significant amount of current research has been conducted by utilizing homomorphism [18], [34], [39], [45], [46], [73], [128], [129]. Homomorphism based schemes hold a great promise that allows data to be processed in its ciphertext form without decryption. Nevertheless, as discussed in chapter 2, efficiency is still the major challenge to implementing those existing schemes in practical applications. Thus, further improvements on such schemes are needed to ensure that they can be implemented efficiently without degrading the performance of the processor, especially for mobile devices. Futhermore, there is a need for a lightweight secure framework that provides security with minimum communication and processing overheads on mobile devices [24].

2.4 SUMMARY

This chapter has overviewed two major areas, MSS and MCC. They demand heavy computation in an encrypted form. Both types of system have been investigated in detail to understand their architectures, benefits and limitations in order to propose appropriate solutions to their problems. Furthermore, the relevant existing solutions have been studied and compared to see how they work and identify their weaknesses in terms of implementation, particularly their functionality and efficiency. Based on the weaknesses identified, further research work will be conducted and described in the next few chapters

to propose better solutions that are more applicable to the MSS and MCC settings. In the next chapter, some basics mathematical materials are introduced and exploited to develop our new schemes, which utilise homomorphic properties to preserve mobile users' privacy in both MSS and MCC environments.

CHAPTER 3

MATHEMATICAL FOUNDATIONS

3.1 INTRODUCTION

This chapter discusses some basic mathematical materials including number theory and abstract algebra as well as some fundamental statements on computational theory, homomorphism and its application in cryptography. Here, we will provide definitions and theorems on the fundamental mathematics that will be used throughout this thesis. Such definitions and theorems are essential for deeply understanding the concept of FHE schemes that will be exploited to allow data to be processed in its ciphertext form without sacrificing the privacy of the data users or contributors. At the end of this chapter, we will describe our research methodology for this research work. We will also provide a brief explanation and necessary validation techniques of our proposed methods.

3.2 ALGEBRAIC NUMBER THEORY

A scheme that is based on homomorphism utilises fundamental concepts in algebraic number theory such as fields. Thus, in this section, we describe some related number theory concepts and algorithms like groups, modulo arithmetic, prime numbers, and Euclidean algorithm.

3.2.1 GROUPS

Definition 3.1. A group $(G,*)$ is a nonempty set G together with a binary operation $*: G \times G \to G$ and $(a,b) \to a * b$ such that the following conditions hold:

(i) Associativity:

$$\forall a,b,c \in G: a * (b * c) = (a * b) * c$$

(ii) Existence of an identity element e:

$$\exists e \in G \; \forall a \in G: a * e = e * a = a$$

(iii) Existence of an inverse element a':

$$\forall a \in G \; \exists a' \in G: a * a' = a' * a = e$$

Furthermore, there are some basic concepts in groups can be described as below:

(i) A group $(G,*)$ is said to be an abelian group if the operation $a * b$ is commutative, that is $a * b = b * a$.

(ii) A finite group G is a group which (as a set) is finite. The order of a finite group (denoted as $|G|$) is the number of elements in it.

(iii) Let $a \in G$. If there exists a positive integer n such that $a^n = e$, then a is said to have a finite order and the smallest such positive integer n is called the order of a, denoted $|a| = n$.

(iv) A subset H of a group G is said to be a subgroup if with the same operation $(*)$ and identity element (e), $(H,*)$ is a group [130].

Moreover, a group G is said to be a cyclic group if there exists an element $a \in G$ such that a generates G, i.e. $G = \langle a \rangle$ or a is the generator of G, then G is called a cyclic group [130]. A group G is said to be a finite group, if its elements is finite. In addition, a group $(\boldsymbol{G},*,+)$ is called a ring if the following properties are satisfied:

(i) It is an additive finite abelian group with its multiplication being associative.

(ii) It has an identity element on each of the addition and multiplication operations.

(iii) The addition and multiplication operations are linked via the distributive law:

$$a \cdot (b + c) = a \cdot b + a \cdot c = (b + c) \cdot a \tag{3.1}$$

Furthermore, a ring is called a commutative ring if its multiplication operation is commutative [131]. Next, the definition of modulo arithmetic is given prior to providing the definition of a field.

3.2.2 MODULO ARITHMETIC

Informally, modulo arithmetic can be defined as two integers having the same remainder when they are divided by a positive integer m. The formal definition is given as below.

Definition 3.2. If a and b are integers and m is a positive integer, then a is congruent to b modulo m if m divides $a - b$. The notation $a \equiv b \ (mod\ m)$ is used to indicate that a congruent to b modulo m. If a and b are not congruent modulo m, then the notation $a \not\equiv b \ (mod\ m)$ is used [132].

A field that is generated by modulo a prime number plays an important role in cryptography due to its finite elements. Such a field is called a prime field. This finite field is used to represent each byte of data as a vector, so that encryption and decryption can be manipulated easily. In the next sub-section, a formal description of a field is given.

3.2.3 PRIME FIELDS

For a given prime p, the finite field of order p, F_p, is defined as the set of integers $\{0,1,\ldots,p-1\}$, together with the arithmetic operations modulo p. Furthermore, the set $F_p = \{0,1,\ldots,p-1\}$ together with the arithmetic operations modulo p is said to be a commutative ring. Moreover, any positive integer in F_p has a multiplicative inverse if and only if that integer is relatively prime to p. Therefore, if p is prime, then every integer in F_p has a multiplicative inverse [133].

The arithmetic operations in a finite field are addition, multiplication and inversion. These operations can be described as follows:

(i) Addition: If $a, b \in F_p$, then $a + b = r$, where r is the remainder when $a + b$ is divided by p and $r < p - 1$. This is known as addition modulo p.

(ii) Multiplication: If $a, b \in F_p$, then $a \cdot b = s$, where s is the remainder when $a \cdot b$ is divided by p and $s < p - 1$. This is known as multiplication modulo p.

(iii) Inversion: If a is a non-zero element in F_p, the inverse of a modulo p, denoted a^{-1}, is the unique integer $c \in, F_p$ for which $a \cdot c = 1$

3.2.4 PRIMES

In the field of cryptography, prime numbers play an important role to provide stronger security to the data. For instance, in the RSA encryption scheme, its security is depending on the prime factorization of a public key, which is a multiplication of two distinct primes. Thus, this sub-section provides a formal definition of prime numbers and other several definitions associated with them.

Definition 3.3. (*Prime Numbers*) A prime is a positive integer greater than 1, which is divisible by no positive integers other than 1 and itself [134].

Prime numbers can be determined by using a primality test such as Fermat's little theorem and Euler's theorem tests. Such tests are used to determine whether a given integer greater than 1 is a probable prime or a composite number [130].

Definition 3.4. (*Relatively Prime*)

If a and b are integers, i.e. $a, b \in \mathbb{Z}$ and the Greatest Common Divisor, $\mathrm{GCD}(a, b) = 1$, then a and b are said to be relatively prime or coprime. Sometimes the phrase 'a is prime to b' is also used [134]. A formal definition of GCD is given as below.

Definition 3.5. (*Greatest Common Divisors*) The GCD of two integers a and b, that are not both zero, is the largest integer, which divides both a and b without a remainder. The GCD of a and b is written as $\mathrm{GCD}(a, b)$ [134], [135].

In addition, GCD can be determined by using an algorithm called the Euclidean Algorithm. This algorithm can be described as below.

3.2.5 EUCLIDEAN ALGORITHM

The GCD can be computed using the division algorithm. The systematic use of the division algorithm will produce a successive algorithm called the Euclidean algorithm. This algorithm can efficiently be used to determine the GCD.

Theorem 3.1. Let a and b be integers ($a, b \in \mathbb{Z}$) such that $a \geq b > 0$, and set $a = r_{-1}$ and $b = r_0$. By repeatedly applying the Division Algorithm, we get $r_{j-1} = r_j q_{j+1} + r_{j+1}$ such that q_{j+1} and r_{j+1} are the quotient and the remainder at $j + 1$ respectively, with

$0 < r_{j+1} < r_j$ for all $0 \leq j < n$, where n is the least nonnegative number such that $r_{n+1} = 0$ in which case $\mathrm{GCD}(a, b) = r_n$ [134].

Several elements like a one-way function and Discrete Logarithm Problem (DLP) are used to provide various tools in cryptography. For instance, a hash function is a kind of one way functions, while the ElGamal algorithm is one of the schemes that the security is based on DLP [136]. On the other hand, a scheme that utilises homomorphism is also implementing such elements and can be described as below.

3.2.6 ONE-WAY FUNCTION

In brief, a one-way function f can be explained as a function that is "easy to compute" (i.e. f can be computed by a probabilistic polynomial time (PPT) algorithm) but "difficult to invert" (i.e. attempting to invert will succeed with low probability using a PPT algorithm) on the average. A hash function is one of the examples of the one-way function. In brief, a hash function can be described as a function that compresses an input of arbitrary length to an output of a fixed-size. Both one-way function and a hash function are given in the Definition 3.6 and the definition 3.7 respectively.

Definition 3.6. A function $f : \{0,1\}^* \rightarrow \{0,1\}^*$ is a one-way function if f is:

1. *Easy to compute*: There is a deterministic PPT f such that for all sufficiently large $k \in \mathbb{Z}^+$ in the input function and all $x \in \{0,1\}^k$, f can be computed in polynomial time.

2. *Difficult to invert*: For any PPT algorithm A, there is a negligible function v_A:

$$\Pr[\, x \leftarrow \{0,1\}^k ; y \leftarrow f(x); z \leftarrow A(1^k, y): f(z) = y \,] \leq v_A(k)$$

The PPT algorithm can be defined as follows. An algorithm A has PPT if it uses randomness and its running time is bounded by some polynomial in the input size [69], [138].

Definition 3.7. A *hash function* is a computationally efficient function mapping binary strings of arbitrary length to binary strings of some fixed length, called *hash-values* [139].

3.2.7 DISCRETE LOGARITHM PROBLEM

Definition 3.8. Let $\mathbb{Z}_p^*$ be a multiplicative group modulo p such that p is a large prime, and $g \in \mathbb{Z}_p^*$ be a generator. According to [140], the Discrete Logarithm Problem (DLP) for the group $\mathbb{Z}_p^*$ can be stated as follows:

Given prime p, generator g of $\mathbb{Z}_p^*$, and an element $c \in \mathbb{Z}_p^*$, find the unique integer exponent e with $0 \leq e \leq p-2$ such that

$$c \equiv g^e (mod\ p) \tag{3.2}$$

3.2.8 HOMOMORPHISM

A group homomorphism has been widely used in the field of cryptography as its properties provide a great promise to allow data to be processed in its ciphertext form without the need for decryption [38], [141], [142]. A formal definition of a group homomorphism is given as below [135].

Definition 3.9. Let $*_G$ and $*_H$ be arbitrary operations in groups G and H respectively. A function $f: G \to H$ from group G to group H is a (group) homomorphism if the group operation is preserved in the sense that:

$$f(g_1 *_G g_2) = f(g_1) *_H f(g_2) \tag{3.3}$$

For all $g_1, g_2 \in G$, let e_G be the identity in G and e_H the identity in H. A group homomorphism f maps e_G to e_H: $f(e_G) = f(e_H)$.

Note that f must preserve the inverse map due to:

$f(g)f(g^{-1}) = f(gg^{-1}) = f(e_G)$, therefore: $f(g^{-1}) = f(g)^{-1}$.

3.3 HOMOMORPHIC ENCRYPTION SCHEMES

In this section, we review the state-of-the-art of the work in the area of HE schemes, its implementation issues related to its complexity and the improvement that has been made to enable such schemes to be implemented efficiently.

3.3.1 INTRODUCTION

An encryption scheme is called a partially HE scheme if it supports either addition or multiplication operations on encrypted data but not both operations at the same time [39]. For example, the unpadded RSA is multiplicatively homomorphic due to such a reason; given $E(m_1) = {m_1}^e \bmod n$ and $E(m_2) = {m_2}^e \bmod n$ (i.e. the ciphertexts of plaintexts m_1 and m_2 respectively). The product of $E(m_1)$ and $E(m_2)$ is given below:

$$
\begin{aligned}
& E(m_1) \cdot E(m_2) \\
& \quad = ({m_1}^e \bmod n \cdot {m_2}^e \bmod n) \bmod n \\
& \quad = ({m_1}^e \cdot {m_2}^e) \bmod n \\
& \quad = (m_1 \cdot m_2)^e \bmod n \\
& \quad = E(m_1 \cdot m_2)
\end{aligned}
$$

As $\big(E(m_1) \cdot E(m_2)\big) = E(m_1 \cdot m_2)$, then the unpadded RSA is multiplicatively homomorphic [143]. Furthermore, the Paillier cryptosystem is additively homomorphic due to such a reason; given $E(m_1, r_1) = g^{m_1}{r_1}^n \bmod n^2$ and $E(m_2, r_2) = g^{m_2}{r_2}^n \bmod n^2$ as the ciphertexts of m_1 and m_2 respectively. The product of the ciphertexts is given below:

$$
\begin{aligned}
& E(m_1, r_1) \cdot E(m_2, r_2) \\
& \quad = (g^{m_1}{r_1}^n \bmod n^2 \cdot g^{m_2}{r_2}^n \bmod n^2) \bmod n^2 \\
& \quad = (g^{m_1}{r_1}^n \cdot g^{m_2}{r_2}^n) \bmod n^2 \\
& \quad = g^{m_1+m_2}\tilde{r}^n \bmod n^2 \\
& \quad = E(m_1 + m_2).
\end{aligned}
$$

As $E(m_1, r_1) \cdot E(m_2, r_2) = E(m_1 + m_2)$, then the Paillier cryptosystem is additively homomorphic [144]. In addition, if an encryption scheme can support both addition and multiplication at the same time on ciphertext data for any degree of polynomial, it is turned to be a FHE scheme [145].

3.3.2 LATTICE BASED SCHEMES AND RELATED PROBLEMS

The idea of FHE was first posed by Rivest et. al. almost 30 years ago. Even though a lot of work has been proposed to achieve double homomoprhism, all of them still have limitations. For example, the ElGamal and Pallier schemes only support a single homomorphic operation [39], while Boneh, Goh and Nissim support arbitrary additions but only one multiplication [146]. Nevertheless, in 2009, Gentry has constructed the first FHE scheme based on an ideal lattice. Since then, a significant amount of current research work has been conducted by utilizing homomorphic properties [44], [45], [48], [49], [75], [128], [147]–[149] due to its special class of encryption functions which allow encrypted data to be operated on directly without the need for any knowledge about the decryption function [150].

This salient feature has made an enormous contribution to the solution of the problems identified earlier. Gentry [107] has defined 'fully' as having no limitation on what manipulations can be performed on ciphertext. The scheme should enable data to be kept secret but allow users with no decryption keys to compute any result of the data, although the function is very complex. In a cloud setting, for example, a CSP never sees any unencrypted data or details about what users are searching for [107]. Started with Gentry's original construction scheme, there are now a number of such schemes in the literature [45], [48], [74], [128], [147], [149], [151]–[153], which are all based on lattices. In this sub-section, we provide an overview of the initial developments on Lattice-based FHE schemes starting with the Gentry's scheme.

Lattices have been widely used in cryptography. They hold a great promise for post-quantum cryptography while enjoying very strong security proofs based on worst-case hardness. Post-quantum cryptography is cryptographic algorithms that secure against threats by quantum computing. Quantum computing becomes a threat to the existing algorithms such as RSA, Diffie-Hellman (DH) and Elliptic Curve Diffie-Hellman (ECDH) key exchange, which are designed respectively based on integer factorisation, discrete logarithm and elliptic curve discrete logarithm problems. Quantum computing is able to break the security of those algorithms much more quickly than any classical computers as it provides massive computing power [154]. Lattice-based cryptography provides relatively efficient implementations as well as great simplicity. Informally, a

lattice can be defined as a set of points in an n-dimensional space with a periodic structure, such as the one illustrated in Figure 5. This figure shows a lattice with two possible bases of basis vectors (b_1, b_2) and (c_1, c_2). The figure also shows that a lattice has many bases.

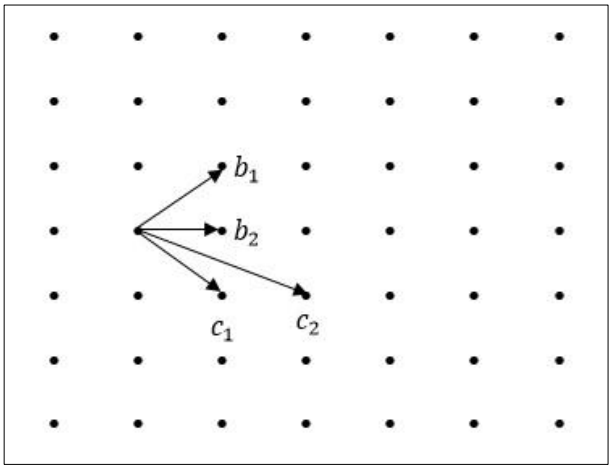


Figure 5: A two-dimensional lattice and two possible bases

A lattice is defined as the set of all integer combinations:

$$\mathcal{L}(\boldsymbol{b}_1, \ldots \boldsymbol{b}_n) = \{\sum_{i=1}^{n} x_i \boldsymbol{b}_i : x_i \in \mathbb{Z} \text{ for } 1 \leq i \leq n\} \quad (3.4)$$

of a n linearly independent vectors $\boldsymbol{b}_1, \ldots \boldsymbol{b}_n$ in a real number set, $\mathbb{R}^n$. The set of vectors $\boldsymbol{b}_1, \ldots \boldsymbol{b}_n$is called a basis for the lattice. A basis can be represented by the matrix $B = [\boldsymbol{b}_1, \ldots \boldsymbol{b}_n] \in \mathbb{R}^{n \times n}$ having the basis vectors as columns. Using the matrix notation, the lattice generated by the matrix $B \in \mathbb{R}^{n \times n}$ can be defined as $\mathcal{L}(B) = \{Bx : x \in \mathbb{Z}^n\}$, where Bx is the usual matrix multiplication [155].

In the earlier works on FHE, there were three proposed FHE schemes based on lattices as outlined below. Those schemes are designed with bootstrapping, without bootstrapping and without modulus switching.

(i) **Fully Homomorphic Encryption with Bootstrapping**

Gentry [145] has proposed a FHE scheme based on lattices that enable the encryption scheme to have a decryption algorithm with low circuit complexity. Furthermore, ideal lattices have been used in order to make the scheme bootstrappable. This can be achieved as lattices offer an additive structure while an ideal lattice algorithm has a multiplicative structure that enables the evaluation of deep arithmetic circuits. The security of the

scheme is based on the hardness of the lattice problem which is Boundary Distance Decoding (BDD). BDD is a variant of the Closest Vector Problem (CVP) in which the target is guaranteed to be close to the lattice, relative to minimum distance λ of the Lattice. CVP can be explained as, given a lattice L represented by some basis and a target point y, find the lattice point closest to y.

From this breakthrough, all existing schemes have common traits as of Gentry's work and they add a small "noise" component during encryption. Such encryption techniques made the scheme not really practical as efficiency is a big challenge for their implementation. There have been several attempts [148], [156]–[159] at implementing most of the schemes mentioned, but none of them comes even close to being practical. For instance, the work in [157] manages to execute one AES encryption homomorphically in eight days using a massive amount (tens of GBs) of RAM memory. Therefore, a lot of further work on the scheme has been made in order to achieve simplicity and improve the efficiency while securing the data.

(ii) **Fully Homomorphic Encryption without Bootstrapping**

The schemes proposed in [128] and [160] are based on Leaning With Error (LWE), where a vector in Z_q is used to represent the ciphertext. LWE is a problem to recover a secret $s \in \mathbb{Z}_q^n$ given a sequence of 'approximate' random linear equations on s [161]. Moreover, the security of the schemes is based on lattice problems with quasi-polynomial approximation factors. In such schemes, noise is introduced for every ciphertext generated for security reason. Noise magnitude $B$is doubled during homomorphic addition, while nearly squared during homomorphic multiplication. By using the scheme in [128], the noise increases from an initial magnitude of B to B^{2^L}after L levels of multiplication. Thus, this requires a very large q, i.e. $q \approx B^{2^L}$, to be used in order to allow for a correct result to be computed from such levels of homomorphic operation on ciphertext data. As a result, such a setting has affected the efficiency of the scheme due to a large size of ciphertext to be transmitted so as to hire more bandwidth to be leveraged for transmitting the data. Furthermore, such a setting also reduces the security of the scheme as the scheme security is proportional to the ratio B/q.

On the other hand, the scheme in [128] allows such a limitation to be remedied by proposing a noise management technique without any bootstrapping approach. By using such a technique, the ciphertext vector will be reduced by a factor of w using a scale down approach called a "modulus switching" technique. This approach also reduces q to q/w as well as reduces the noise B to B/w. Thus, computing L levels of homomorphic multiplication will reduce the value to $^{q}/_{B^{L}}$ while the noise B will remain the same. Such an approach reduces the value of $q \approx B^{L+1}$ to be used, so as to improve the scheme's efficiency compared to the previous one. However, according to [149] such an approach requires a complicated process on the ciphertext data to be evaluated homomorphically.

(iii) **Fully Homomorphic Encryption without Modulus Switching**

Brakerski had presented a scale invariant scheme [149], where the ciphertext is scaled down by a factor of q so that the noise magnitude turns to B/q. This setting leads to the homomorphic multiplication only requiring the noise magnitude multiplied by a polynomial factor $p(n)$. Thus, to compute L levels of homomorphic multiplication, it only requires $q = B \cdot p(n)L$ due to the noise growth from B/q to $B/q \cdot p(n)L$. This choice of q improves the scheme's efficiency compared to the previous two schemes.

The advantages of the scheme are that it only uses the same modulus throughout the evaluation process with no need for "modulus switching", and this modulus can take an arbitrary form. Furthermore, in all previous work, the ciphertext noise grows quadratically ($B \rightarrow B^2 \cdot poly(n)$) with every multiplication (before "refreshing"), while the noise in this scheme only grows linearly ($B \rightarrow B \cdot poly(n)$) [149], [162]. However, according to Lauter et al., [156] even though such constructions of FHE schemes have been proposed and improved upon, they are still far from practical as they are suffering from poor efficiency. Thus, all known FHE schemes seem to have a long way to go before they can be used in practice.

For easy reference, the main weaknesses of the existing FHE schemes together with potential solutions to the problems are summarised below:

- Existing FHE schemes are designed with high complexity to provide strong security [22], [49], [159], [163]. Such complexity prevents resource-constrained devices like smartphones to efficiently execute those schemes in order to protect

mobile data [25], [52]. Thus, a scheme with less complexity but strong security is needed to allow the aforementioned devices to leverage rich-mobile applications provided by the clouds [34], [51], [156], [164].

- Existing FHE schemes are encrypting individual bits so as to make them easier to achieve fully homomorphic properties. Nevertheless, such an encrypting technique requires a huge bandwidth and storage spaces that prevent their applicability to resource-constrained devices. Especially they would shorten these devices' battery lifetime due to the need to continuously transmit huge amounts of data [44], [47], [48], [74], [163], [165]. Thus, a scheme, which encrypts integers rather than individual bits, could help to achieve a lower bandwidth in transmitting the data and require less storage space.

3.3.3 Integer Based Schemes and Related Problems

A scheme that is based on simplicity such as over the integers is demanded to reduce the complexity so as to improve efficiency. Several works that have proposed schemes based on the integers can be found in [44], [48], [147]. Such works have provided encryption schemes with bigger message spaces as all operations on ciphertext data are executed over the integers. For instance, the scheme proposed by M. Djik et al. is based on simple integer arithmetic like addition and multiplication. The scheme achieves fully homomorphic properties by leveraging the bootstrappable approach that was proposed by Gentry. However, the scheme also suffers from efficiency issues due to its use of a large public key size $O(\lambda^{10})$, where λ represents the scheme's security parameter [35]. Several attempts [44], [48], [166] have successfully improved the size of public keys. For example, the work in [167] has further reduced the size of the public keys to $O(\lambda^{3.5})$. Nevertheless, such an improved key size is still considered to be high for being implemented in resources-constrained devices like smartphones and tablets [35], [44], [46].

In the works [99] and [102], an efficient symmetric FHE scheme has been proposed to allow cloud-based applications to process data in its ciphertext form. Both schemes are believed to be simple and can achieve fully homomorphic properties as they are based on matrix operations. Both schemes have supported homomorphic addition and multiplication with linear complexity and nearly linear complexity respectively.

However, such schemes require more bandwidth for transmitting data due to the ciphertext data being in the form of matrices. Therefore, some enhancements to the scheme are needed prior to be implemented by resources-constraint mobile devices like smartphones and tablets. In chapter four, we will further describe the schemes' efficiency based on several experimental results.

3.4 Research Methodology

In this research work, we have selected two computing paradigms, which are MSS and MCC, to be investigated in order to propose our new methods. The reason is that both computing paradigms provide a very promising solution to data sharing and processing effectively and efficiently. Nevertheless, experiencing security issues such as data confidentiality and user's privacy as well as inherent limitations on mobile devices in both computing paradigms has led us to leverage some existing methods described in Chapter 2 and mathematical foundation concepts with new enhancements to reflect to our research questions listed in Chapter 1. Those methods will be described in the sub-sequent section, while the new enhancements will be explained in Sub-section 3.4.2. Finally, we provide a validation process to our proposed schemes prior to summarising this chapter.

3.4.1 The Rationale of Selected Methods

(i) **Privacy-preserving Data Aggregation (DA) and Data Offloading (DO) techniques**

In the MSS setting, we assume all parties are untrusted. Thus, a technique such as privacy-preserving data aggregation is good for implementation in MSS [1], [11], [23], [71], [98]. This is because, such a technique can protect user privacy by mixing the data prior to submitting it to a cloud server. In addition, this technique improves the battery lifetime of mobile devices by reducing the data to be transmitted. To implement this technique, we adapt a hierarchical network structure proposed in [23] to locate the nodes in a way that the lowest level of this structure consists of leaf nodes without child or children. The upper layer of this structure will consists of parent, grandparent and root nodes before reaching the cloud server, which are all assumed as untrusted nodes. On the other hand, data offloading is good for implementation in MCC as such a technique allows data to be transmitted directly to the cloud server [4], [5], [13], [26], [168], [169]. This technique

also reduces battery consumption as all computation burdens are taken by the cloud server. The aim of both techniques is to allow the outsourced data to be processed without disclosing the content of the data and scarifying the privacy of mobile users. To fulfil this aim, homomorphic properties are the best options as such properties allow the transmitted data to be processed in its ciphertext form without decryption [22], [39], [49], [163], [170].

(ii) **Homomorphic Encryption Schemes**

The concept of homomorphism was introduced almost 40 years ago by Rivest et. al [171]. Since then, many encryption schemes have been designed based on homomorphic properties. A partially homomorphic encryption scheme is a scheme which supports ciphertext computation either addition or multiplication at a time. Conversely, an FHE is a scheme that allows ciphertext data to be computed arbitrarily without any limitations. In MSS, a partial homomorphic scheme is good for implementation as data is aggregated using the addition operation via the hierarchical structure. In the meanwhile, a scheme that supports fully homomorphic properties is demanded in MCC to process arbitrary functions on ciphertext data. Both partially and fully homomorphic properties require a broad understanding of mathematical foundations as it covers fundamental mathematical concepts in number and group theory. For instance, the recent FHE schemes have been proposed based on Lattices and integers [35]. There are distinct advantages of such schemes as described in the previous section. For our setting, we have designed our homomorphic encryption schemes based on integers as such a selection could achieve simplicity, improve efficiency and reduce the communication costs.

(iii) **Homomorphic MAC**

In MSS, the aggregated data must be genuine in order to further process such data. Otherwise, the aggregated data will be wasted as it will not provide a correct result to benefit other mobile users and the system. Thus, a homomorphic MAC is good to be implemented as such a technique could verify the aggregated data [23]. Such a technique is important to validate the aggregated data in order to allow a correct result to be obtained from the aggregated result.

All the aforementioned methods are good to be implemented in both computing paradigms. The reason is that, such methods with further enhancements could provide

solutions to the research questions mentioned in Section 1.5. A direct mapping of those methods and the research questions is provided in Table 1 below.

Table 1: Research Methodology versus Research Questions

Research Methodology	Research Question 1	Research Question 2	Research Question 3	Research Question 4	Research Question 5
Privacy-preserving DA and DO techniques	√	√	√	√	√
Homomorphic Encryption Scheme	√	√	√	√	√
Homomorphic MAC	√				√

3.4.2 THE PROPOSED METHODS

In MSS, we propose APDA schemes that consist of weak, strong and hybrid versions. Those versions are signified as $APDA^{W}$, $APDA^{S}$ and $APDA^{H}$, respectively. APDA allows data to be aggregated in its ciphertext form so as to protect data confidentiality and preserve the privacy of mobile users. This is because APDA leverages homomorphic properties with better efficiency. In addition, APDA allows nodes, which misbehaved in the aggregation process to be detected certainly and efficiently. Briefly, $APDA^{W}$ is designed without involving a signature so that it can achieve high efficiency but lower misbehaviour detection accuracy. In contrast, $APDA^{S}$ is designed with signature involved so that it can achieve high misbehaviour detection accuracy but lower efficiency. To balance both schemes, $APDA^{H}$ is proposed by combining the two schemes so as to achieve high detection accuracy with better efficiency.

Generally, each version of APDA consists of research processes as shown in Figure 6. Those processes begin with the commitments collection process. This process determines the willingness of mobile users to contribute their data in the aggregation process. In addition, each mobile user will leverage this commitment as an encryption key to encrypt their data as this value is unique and only known by them and the AS. The next process involves aggregated data encryption from the lowest layer of children nodes to the highest layer of root node prior to submitting the final aggregated data to an AS. In this process,

APDA^{W} is implemented. This weak version will consists of no signature to accelerate misbehaved node tracking. If the integrity checking has failed, APDA^{W} can only detect the misbehaved nodes with certainty if the node has the leaf or child node status. Otherwise, APDA^{W} can only assume the nodes as suspicious. In contrast, By using APDA^{S}, the AS generates and assigns appropriate keys to each participating mobile user for generating the signature of each piece of generated data. This signature will be used by the aggregator to track down any misbehaved nodes if the integrity checking has failed. In contrast, if the integrity checking is positive, then the AS will recover the result of the aggregated data. Embedding a signature to each piece of generated data guarantees any misbehaved nodes to be detected with certainty. Nonetheless, such a setting lowers the scheme efficiency. Thus, to certainly detect the misbehaved nodes with better efficiency, APDA^{W} and APDA^{S} are combined in such a way that, APDA^{W} is executed first to determine the suspicious nodes. When the suspicious nodes are detected, then only those nodes need to execute APDA^{S}. This can improve the efficiency of the misbehaved nodes tracking process while able to determine the misbehaved nodes certainly. This approach is called as APDA^{H}.

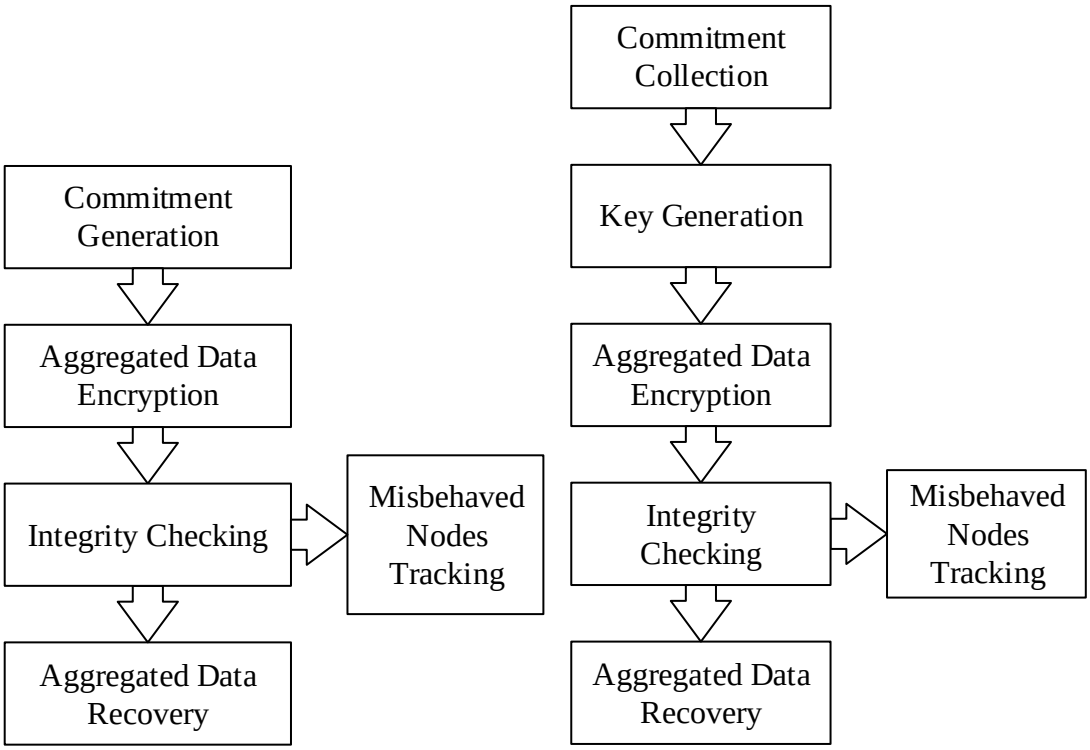


Figure 6: APDA Research Processes in MSS

The results of aggregated data need further processes such as additive and non-additive statistical functions to benefit other mobile users in MSS. Thus, an extended version of APDA is proposed to determine a maximal value of the aggregated data. Other statistical functions also use the same concept of maximal value finding and will not be described in this thesis. The maximal value finding determines a maximum value without revealing any contents of mobile data.

In another mobile computing paradigm i.e. MCC, a new LHE scheme is proposed to allow the data generated by mobile devices to be processed arbitrarily by a cloud server without decryption. The LHE scheme leverages integers to achieve good simplicity and better efficiency. This scheme consists of three research processes, which are for key generation, data encryption and data processing and recovery as shown in Figure 7. Briefly, the Data User (DU) uses two primary keys to generate a secret key in the key generation process. DU then sends this secret key to Data Contributors (DCs) securely. DCs use such a key for encrypting their data in the data encryption process prior to submitting the ciphertext data to CS. Having received the ciphertexts data from DCs, CS then processes the ciphertexts data without decryption in the data processing process and transmit the result to DU. Finally, DU decrypts the received result to recover its plaintext data in the data recovery process. For simplicity, Figure 7 illustrates the flow of the aforementioned processes to implement the LHE scheme.

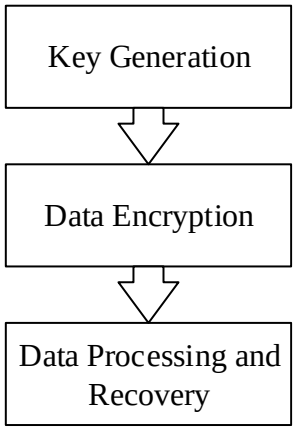


Figure 7: LHE Research Process in MCC

3.4.3 VALIDATION TECHNIQUES

In this research work, we have selected several parameters to measure data confidentiality, data integrity, users' privacy, users' accountability and scheme

efficiency. Those measurement parameters are crucial to validate whether our proposed schemes can achieve all the objectives listed in Chapter 1. In brief, we use a suitable length of keys to measure data confidentiality and users' accountability whereas a homomorphic MAC is used to measure data integrity. In addition, we use time to measure the scheme's efficiency. We will prove that our proposed schemes can be validated by using all the measurement parameters theoretically and will be detailed in Chapter 5 and 6. Furthermore, our scheme's efficiency will be proved by using a simulation approach that will be conducted to validate that our scheme can provide better efficiency compared to the chosen scheme.

3.5 SUMMARY

This chapter has covered some fundamental concepts of mathematics like prime numbers, modulo arithmetic and GCD algorithms that have been exploited to develop FHE encryption schemes. Each concept is defined for understanding and clarification purposes. Furthermore, we have briefly introduced some background of FHE schemes. The improvement on such schemes provides a great promise for encrypted data to be processed without decryption, thus allowing resource-limited devices like mobile phones to enjoy huge computing resources and massive storage spaces provided by CSPs in MSS and MCC. By using such a scheme, existing approaches like data offloading and data aggregation can be tailored to run on mobile devices in an efficient and secure manner while reducing their energy consumption. At the end of this chapter, we have described our research methodology to highlight the selected network settings to be investigated. In addition, we have also provided the reason for selecting some existing methods to be enhanced prior to proposing our schemes. Finally, we provide a brief explanation on validation techniques to validate that our proposed scheme can achieve all the objectives listed in Chapter 1. In the next chapter, we will provide our first solution to be proposed in MSS.

CHAPTER 4

ACCOUNTABLE PRIVACY-PRESERVING DATA AGGREGATION IN MSSS

4.1 INTRODUCTION

Data aggregation is a good technique for implementation in MSS as this technique offers great advantages to prolong the network lifetime of such a system. Nonetheless, this MSS setting could consist of untrusted parties like a curious AS and malicious MNs together with untrusted intermediary network devices like wireless routers. Thus, the data outsourced to each of these parties or devices needs to be protected as it may contain sensitive information, which relates to the data owner such as location and health information. As a result, this MSS setting demands a suitable technique like a privacy-preserving data aggregation to be implemented to preserve users' privacy during the aggregation process. This technique allows an AS to collect and process sensor data in a way that the data should be kept secret and the aggregator or other nodes learn nothing from the intermediary or the final aggregated data.

A lot of privacy-preserving data aggregation techniques have been proposed and improved upon [21], [23], [25], [36], [67]. Those techniques have worked well and preserved the privacy of data users by leveraging a homomorphic encryption scheme [16],

[18], [23], [172], [173]. Nevertheless, such techniques cannot be directly implemented by mobile nodes like smartphones and tablets in MSSs as sensor nodes in WSNs. The main reason is that, the sensor nodes in WSNs are statically distributed in an observation area for monitoring and data collection purposes. In such a distribution, each node can be assigned to a pre-shared key for a secure communication. In contrast, mobile nodes in MSSs are moving in and out of the networks, which makes assigning a key for each node beforehand is a very challenging task [23], [77]. In addition, those devices belong to individuals (with diverse interests), who contribute to and involve in aggregating the data [1], [23], [77]. Furthermore, an aggregated data received by an AS needs to be verified in order to get a genuine result without revealing any information of individual data to the AS [11], [15], [18], [22], [23], [172], [174].

Even though the existing techniques support a full verification on privacy-preserving data aggregation, their efficiency in terms of computation complexity and execution time still can be improved. At the same time, a better scheme could be proposed to increase the performance of mobile devices, which have limited computing resources [23]. By using those methods, a new verification technique on the aggregated data by an AS provides two important results, which are positive verification (the aggregated data is genuine) and negative verification (the aggregated data consists of fake data). On the one hand, the former result allows the AS to process the aggregated data to retrieve some important statistical results like Count, Average, Min/Max and Percentile. Existing schemes [1], [23], [67], [71], [77] allow those results to be computed in an encrypted form, thus preserving the privacy of the participating mobile users. Nevertheless, the aforementioned schemes suffer from efficiency issues due to their high complexity. On the other hand, the negative verification on a wrong aggregated result could be caused by the following reasons:

(i) Either child or parent nodes submit wrong data to their ancestor.
(ii) Both child and parent nodes submit wrong data to their ancestor.
(iii) Parent nodes aggregate data wrongly.

The negative verification requires further actions to be taken by the aggregator to determine any nodes, which have misbehaved by submitting wrong data or aggregate the data wrongly. Whenever the misbehaved nodes are detected, they have to re-submit the

correct data to allow the aggregator to generate a correct statistical result. This determination process of such nodes is called misbehaviour nodes detection.

To the best of our knowledge, none of the existing schemes is able to support mobile users' accountability in MSSs and provide an efficient verification on the aggregated data in a privacy-preserving manner. Within limited resources on small devices, it is a very challenging task to efficiently identify any nodes, which have misbehaved from the aggregated data. At the same time, it is even more challenging to cater for all security aspects such as data security, users' privacy and functionality on a ciphertext. Based on such reasons, an exploration on partially HE schemes has received a great deal of attention. A low complexity scheme is called for to support statistical functions like additive and non-additive functions on ciphertext data [23]. Thus, such requirements have demanded an improved partially HE scheme to be proposed.

As a result, in this chapter, we propose a novel Accountable Privacy-preserving Data Aggregation (APDA) method that allows mobile data to be aggregated in a secure manner without compromising the privacy of the users to an AS. Our method is essential to achieve our first, second and third objectives as listed in the introduction chapter. This method operates in three operational modes:

(i) Weak accountability, which enables to track down suspicious or misbehaved nodes although some suspicious nodes might not have misbehaved.
(ii) Strong accountability, which enables to pinpoint exactly which nodes have misbehaved.
(iii) Hybrid accountability, which is a combination of the above two modes to precisely identify misbehaved nodes among suspicious ones.

The above three modes will be referred to $APDA^W$, $APDA^S$ and $APDA^H$, respectively. As will be made clear in the next few subsequent sections, although $APDA^W$ may not be able to determine which nodes have certainly misbehaved in an aggregation, this method involves less computation and communication, which make it more efficient. Conversely, $APDA^S$ can accurately decide which nodes have misbehaved with lower efficiency. To take advantage of both strengths from the two modes and rectify both schemes' weaknesses, $APDA^H$ provides a mixed solution by running $APDA^S$ only on the suspicious nodes identified by $APDA^W$ in the previous round of aggregation and applying $APDA^W$

to the rest. Evidently $APDA^H$ has two special cases of $APDA^H$. That is, $APDA^H$ becomes $APDA^W$ when there is no suspicious node, and it is the same as $APDA^S$ when every node is suspicious. This promises the hybrid combination will provide the certainty of misbehaviour identification among the suspicious nodes while offering balanced efficiency (i.e. it is between those of $APDA^S$ and $APDA^W$ with its scale depending on the number of suspicious nodes).

In the next section, we will describe a threat model in MSSs to be a guideline for designing our new solution scheme.

4.2 THREAT MODEL

In MSSs, an AS and Mobile Nodes (MNs) communicate with one another to aggregate sensing data via a hierarchical data aggregation technique so as to allow additive and non-additive statistical functions to be retrieved from the aggregated data. By using this technique, the AS sends a request to MNs via wireless devices like wireless router or network provider infrastructures to collect information from MNs such as personal data (e.g. health, monthly salary or home address information) or sensed data about the surrounding area (e.g. crowd in an event, road congestion or local weather information). After receiving the request and having the related information, MNs will response to the AS by submitting the data they have possessed. However, as such information is closely related to MNs (e.g. their locations and health conditions), submitting the data via the aforementioned technique leads to the exposure of the data to a curious AS, malicious MNs, network infrastructures or an intermediary party like an adversary that observes the aggregation process. In order to give a better description about the threats, we divide them into internal and external threats, which are elaborated separately below.

4.2.1 INTERNAL THREAT

In a hierarchical data aggregation setting, an AS and MNs are normally assumed to be trusted parties. Nevertheless, they may be curious or malicious and have an intention to reveal the content of the final or intermediate aggregated data for its curiosity or other purposes. To reveal such information, they may launch several attacks that can be described as follows:

(i) At the top of our hierarchical structure, the AS receives data to be processed in its ciphertext form. However, the curious AS may want to discover the content of the data to reveal the privacy of MNs.

(ii) In our hierarchical data aggregation, each MN transmits the sensed data in the ciphertext form to its parent node for aggregation purposes. However, a malicious parent node may want to reveal the contents of the received data to invade the privacy of a specific MN. In addition, as MNs belong to individuals with diverse interests, it is difficult to ensure all MNs are reliable in MSSs.

4.2.2 EXTERNAL THREATS

In our proposed system, all the communications among an AS and MNs are executed via open wireless devices like Wi-Fi routers. Such devices are not secure and reliable as they are made public and thus vulnerable to external attacks like data interception and Denial of Service (DoS) attacks [175]. Those attacks can be orchestrated by an adversary in the following scenarios considered in this chapter:

(i) An adversary node that observes the aggregation process and may want to eavesdrop on some communication devices between the AS and MNs.

(ii) An adversary or malicious node, which intends to join the aggregation by submitting or aggregating false or fake data to prevent the AS from generating the correct result.

(iii) An adversary node that attempts to compromise and manipulate some MNs to reveal other nodes' data and thus invade their privacy.

(iv) Both AS and mobile devices communicate wirelessly through network infrastructures like wireless routers. Such infrastructures may vulnerable to physical attacks. Nevertheless, such attacks will not be considered in this thesis and will be highlighted in future works.

4.3 DESIGN GOAL

Based on some related aspects and the description of the above threats, we design our scheme to fulfil the following requirements:

(i) *Scheme Applicability*: Most of the existing verifiable privacy-preserving data aggregation schemes are designed with different capabilities such as for WSNs with

static topologies. In contrast, MSSs consist of mobile nodes that are dynamically moving in and out of the networks. Thus, we need to design a scheme that allows mobile nodes with a mobile topology to sense and aggregate data under a reasonable cost.

(ii) *Data Security and User Privacy*: In MSSs, an AS could be curious while MNs could be malicious. Thus, we need to design a scheme that allows data to be aggregated hierarchically and processed in its ciphertext form so as to prevent an AS and MNs from disclosing the aggregated data. Such a scheme preserves the privacy of individual mobile users.

(iii) *Data Integrity*: In MSSs, an AS and MNs communicate via public wireless devices. As such a communication device is not secure and reliable, we need to design a scheme that allows the aggregated data to be verified by the AS in a privacy-preserving manner, so as to protect the integrity of the aggregated data.

(iv) *Scheme Functionality*: In both WSNs and MSSs, retrieving some essential information on the aggregated data is the main purpose of acquiring the aggregated data. Thus, we need to design a scheme that allows maximal value finding to be computed over the aggregated data in a privacy-preserving manner. Furthermore, such a method could be developed further to compute other comparative operations like Average, Percentile and Histogram in the similar way.

(v) *User Accountability*: In MSSs, the aggregation process could involve misbehaving nodes, which contribute fake data to the aggregation process. Detecting such a node with certainty and efficiency in such a system is a very challenging task as communicated data is encrypted to preserve users' privacy. Thus, we need to design a scheme that allows misbehaved nodes to be detected with certainty in a privacy-preserving manner and at a reasonable cost.

In the following section, we provide our first solution, i.e. APDA^{W}. We describe all the processes executed by each MN and an AS, including commitment generation, aggregated data encryption, aggregated data recovery, misbehaved nodes tracking and privacy improvement.

4.4 PRIVACY-PRESERVING DATA AGGREGATION WITH WEAK ACCOUNTABILITY (APDA^{W})

For implementing APDA in an aggregation process in MSS, several parameters need to be set beforehand. Suppose that there are n mobile devices participating in data aggregation, each of which is denoted as nodes u_i $(1 \leq i \leq n)$ and there is an AS for collecting aggregated data from each of nodes u_i. A hierarchical structure is used to represent the mobile devices as nodes and their connections as links. All nodes in the network are connected wirelessly. Furthermore, we assume that every two connected nodes have already authenticated each other and established a secure communication channel between them if necessary.

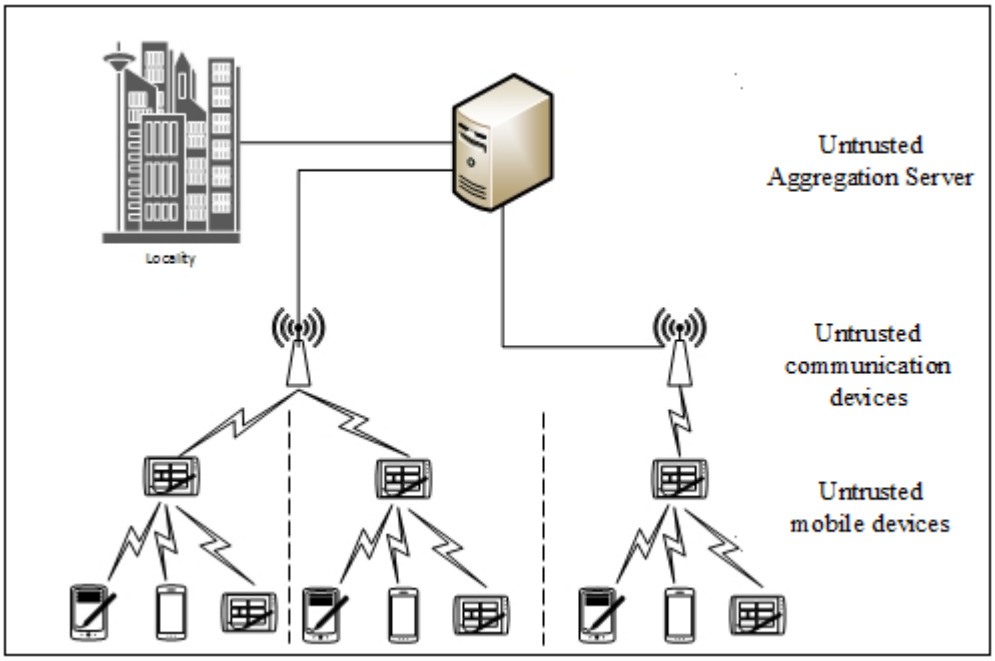


Figure 8: A MSS setting

Figure 8 provides an overview of MSS setting and inherent security flaws of MSS. Such flaws could be remedied by appropriate techniques such as privacy-preserving methods proposed in [21], [23], [25], [36], [67]. In our setting, each network consists of four different types of nodes namely, leaf nodes, parent nodes, a root node and an AS. Each node has different roles and capacities. Leaf nodes act as data contributors to their parent nodes. They consist of mobile devices, which have computing and storing limitations like smartphones or tablets. Parent nodes act as data aggregators for their children in addition to the role of data contributors. They aggregate their own data with their children's data

and send the results to their parents. This aggregation and forwarding process is repeated until the root node receives the data from its children, which acts as a special parent node for the final data aggregation and transfers the aggregated result to the AS. Finally, the AS collects the aggregated data, extracts the information needed, and checks its integrity.

4.4.1 AGGREGATED DATA ENCRYPTION

This sub-section describes the APDA^{W} process on an aggregated data encryption via the hierarchical structure described earlier. Similar to the work in [23], each round of data aggregation has two phases, commitment collection and data aggregation. The commitment collection phase allows an AS to collect commitments from all the nodes, which will collectively enable the AS to check the integrity of the aggregated result to be detailed later. The commitment of each node should be bound to its value to be aggregated, but the AS cannot infer the value from the commitment in order to preserve the node's data privacy. The data aggregation phase allows all the nodes to contribute their values to the aggregation in an encrypted form in a Bottom-Up (BU) way along the hierarchical node structure until the root node completes its data aggregation and passes the result to the AS. The AS can then decrypt the result to reveal the aggregated value without knowing the individual values added by different nodes and examine its match to the commitments collected to ensure its integrity. The processes of collecting the commitments and aggregating ciphertext data by each node using APDA^{W} can be illustrated in Figure 9. In addition, Figure 10 shows the process executed by the AS for collecting the commitment, recovering the aggregated data and integrity checking based on the commitments collected.

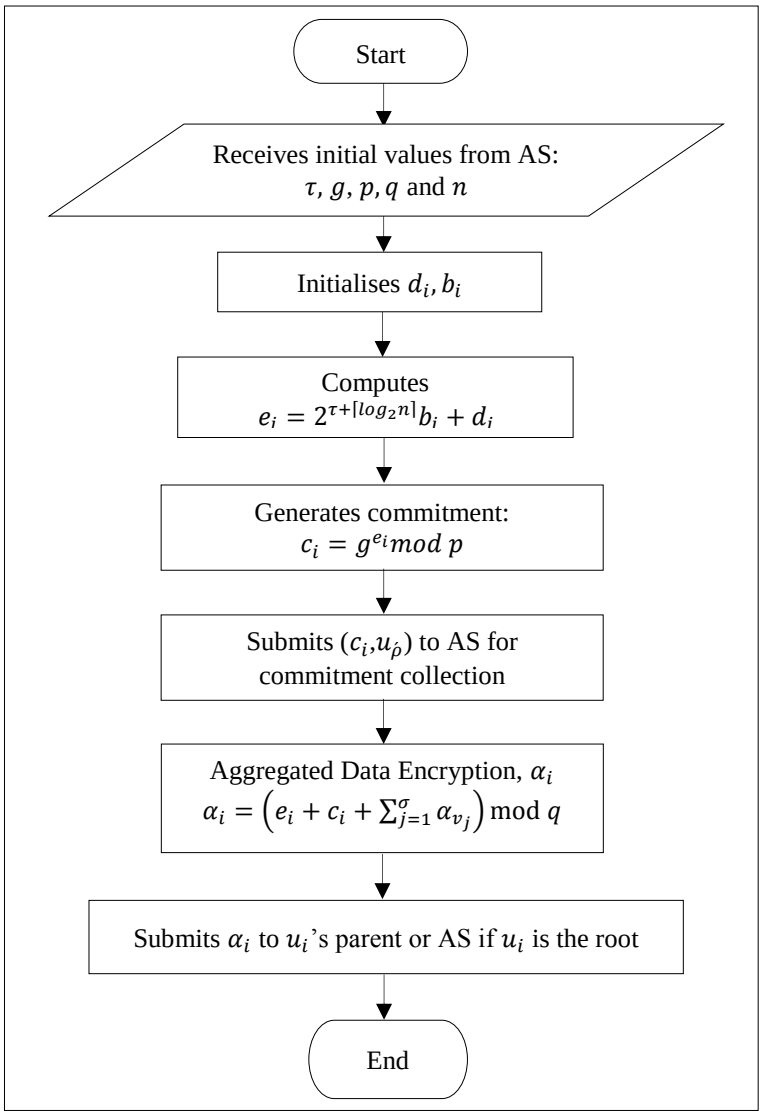


Figure 9: Commitment Generation and Aggregated Data Encryption by each node using APDA^{W}

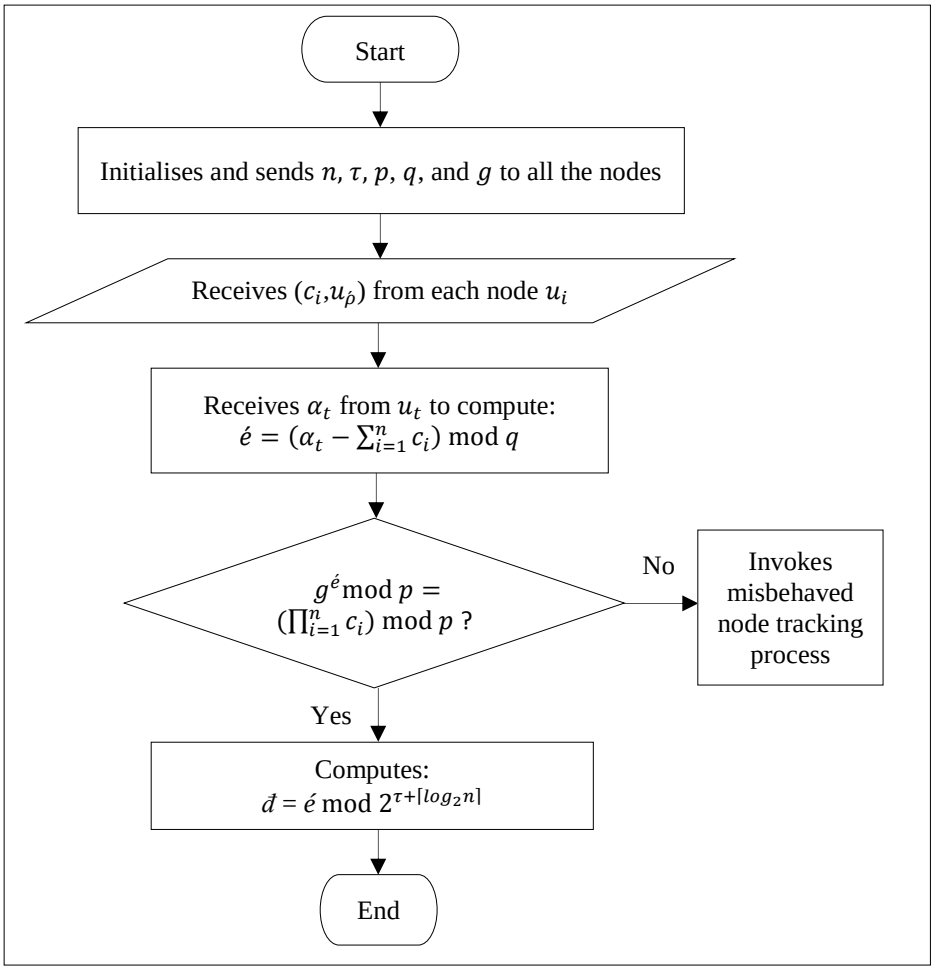


Figure 10: Commitment Collection, Aggregated Data Recovery and Integrity Checking by AS using APDA$^{\mathbf{W}}$

As described earlier, each round of data aggregation consists of two phases and the details are presented as follows. To create the commitments in the first phase, we adopt the Schnorr signature scheme's parameters [176]. Namely, p is a large prime, q is a prime factor of $p-1$, and g $(1 < g < p)$ is a generator such that $g^q \bmod p = 1$.

Let d_i be a value of a node u_i $(1 \leq i \leq n)$, which u_i will contribute to the data aggregation. Since d_i may be within a small range, it is necessary to randomise d_i to enlarge its range before the commitment generation to make the discovery of d_i from its associated commitment harder. The randomisation of d_i is the same as that defined in [23] and repeated below:

$$e_i = 2^{\tau+\lceil log_2 n \rceil} b_i + d_i \text{, for every } i \ (1 \leq i \leq n) \qquad (4.1)$$

Here, τ is the maximal bit length of value d_i to be aggregated, and b_i is a random number

picked up by u_i, of which the bit size l_b is determined in terms of the difficulty of exhaustive search for decryption. $2^{\tau+\lceil log_2 n \rceil}$ in e_i implies $2^{\tau+\lceil log_2 n \rceil} > \sum_{i=1}^{n} d_i$, which is needed for the AS to recover the aggregated result as to be detailed in Sub-section 4.4.2. From the definition of e_i, it is clear that e_i's bit size is $l_e = \tau + \lceil log_2 n \rceil + l_b$. Additionally, we require $\sum_{i=1}^{n} e_i < q$ that can always be satisfied by selecting a suitable size of q. This condition is needed to allow the AS decrypting the aggregated data in Sub-section 4.4.2.

In order to generate its commitment, u_i computes:

$$c_i = g^{e_i} \bmod p \tag{4.2}$$

and sends $(c_i, u_{\acute{\rho}})$ to the AS securely. Here, $u_{\acute{\rho}}$ is the identity of u_i's parent, and this information enables the AS to build the hierarchical relationships among all the nodes, which are required for misbehaved node tracking to be presented in Sub-section 4.4.3. Note that each commitment c_i submitted to and successfully authenticated by the AS is assumed to contain correct e_i [23].

After the successful completion of the above phase, the data aggregation phase begins with the leaf nodes with no children. Each of these nodes sends its encrypted value to be defined below to its parent. Every parent node then aggregates all the received values with its own encrypted one and passes the result to its parent. This process is repeated until the root node finishes its data aggregation and sends the result to the AS. Finally, the AS decrypts the received result to recover the aggregated value and checks its integrity based on the commitments collected in the first phase. In addition, we have extended this phase to include a process for identifying possible misbehaved nodes if the aggregated value does not match the commitments, so that certain actions can be taken to further investigate such nodes in subsequent aggregations. The detail of this process will be specified in Sub-section 4.4.3.

We now present how a node u_i generates its aggregated data to be sent to its parent, which is different from the method given in [23]. Let $\{\alpha_{v_1}, \ldots, \alpha_{v_\sigma}\}$ denote a set of values received by u_i from its σ children. Here we assume that there exists a secure communication channel between u_i and each of its children for encrypted and authenticated data transmissions, and any unauthorised nodes and even the AS cannot

decrypt the encrypted data.

To generate its aggregated value, u_i performs the following calculation:

$$\alpha_i = (e_i + c_i + \sum_{j=1}^{\sigma} \alpha_{v_j}) \bmod q \quad (4.3)$$

Here, $e_i + \sum_{j=1}^{\sigma} \alpha_{v_j}$ is an aggregation of all the received values α_{v_j} together with u_i's own value e_i, where $\sum_{j=1}^{\sigma} \alpha_{v_j} = 0$ when u_i is a leaf node. α_i is the encryption of the aggregation with u_i's commitment c_i as a secret key. This encryption is different from the one in [23] that uses a hash yielded from a secret key shared with the AS. Since u_i submitted c_i to the AS securely, c_i can be treated as a shared secret key between them. This avoids the hash calculation used in [23] so as to improve the efficiency of data encryption by each node and decryption by the AS.

After the generation of α_i, u_i sends α_i to its parent $u_{\acute{\rho}}$ securely. Note that $u_{\acute{\rho}}$ cannot decrypt α_i without knowing c_i. Although the AS can decrypt α_i with stored c_i, the AS is unable to get α_i as α_i is sent to $u_{\acute{\rho}}$ securely.

Having received and authenticated the data from all the children, $u_{\acute{\rho}}$ yields its aggregated data in the same way as u_i did, and sends the result to its parent. This process is repeated until the root node passes its aggregated data to the AS. Then, the AS will recover the plaintext result by executing a data recovery process that will be described in the following sub-section.

4.4.2 AGGREGATED DATA RECOVERY

In this sub-section, the process of recovering the aggregated data from its ciphertext form is described as follows. Upon the reception of result α_t from root node u_t, the AS confirms the authenticity of α_t. If the confirmation is positive, the AS decrypts α_t with collected commitments c_i to recover the aggregated randomised value *é* by computing:

$$é = (\alpha_t - \sum_{i=1}^{n} c_i) \bmod q \quad (4.4)$$

This is due to the following relationships where $\sum_{i=1}^{n} e_i < q$ as stated in Sub-section 4.4.1:

$$\acute{e} = (\alpha_t - \sum_{i=1}^{n} c_i) \bmod q$$
$$= (e_t + c_t + \sum_{j=1}^{\sigma} \alpha_{v_j} - \sum_{i=1}^{n} c_i) \bmod q$$
$$= (\sum_{i=1}^{n} e_i + \sum_{i=1}^{n} c_i - \sum_{i=1}^{n} c_i) \bmod q$$
$$= \sum_{i=1}^{n} e_i .$$

The above relationship is true because of $\left(\sum_{j=1}^{\sigma} \alpha_{v_j}\right) \bmod q = \sum_{j=1}^{\sigma} \left(e_{v_j} + c_{v_j}\right) \bmod q$, leading to:

$$\left(e_t + c_t + \sum_{j=1}^{\sigma} \alpha_{v_j}\right) \bmod q = \left(e_t + c_t + \sum_{j=1}^{\sigma} \left(e_{v_j} + c_{v_j}\right)\right) \bmod q$$
$$= (\sum_{i=1}^{n} e_i + \sum_{i=1}^{n} c_i) \bmod q.$$

After the recovery of *é*, the AS checks its integrity based on commitments c_i by confirming:

$$g^{\acute{e}} \bmod p = (\prod_{i=1}^{n} c_i) \bmod p \tag{4.5}$$

This relation is because of:

$$g^{\acute{e}} \bmod p = g^{\sum_{i=1}^{n} e_i} \bmod p = (\prod_{i=1}^{n} g^{e_i}) \bmod p = (\prod_{i=1}^{n} c_i) \bmod p.$$

If the above check is successful, the AS completes the aggregation process by calculating the aggregated value:

$$\acute{d} = \acute{e} \bmod 2^{\tau + \lceil log_2 n \rceil} \tag{4.6}$$

Based on the condition $2^{\tau + \lceil log_2 n \rceil} > \sum_{i=1}^{n} d_i$ given in Sub-section 4.4.1, the above calculation results in:

$$\acute{d} = \acute{e} \bmod 2^{\tau + \lceil log_2 n \rceil} = (\sum_{i=1}^{n} e_i) \bmod 2^{\tau + \lceil log_2 n \rceil} = (\sum_{i=1}^{n} (2^{\tau + \lceil log_2 n \rceil} b_i + d_i))$$
$$\bmod 2^{\tau + \lceil log_2 n \rceil} = \sum_{i=1}^{n} d_i.$$

However, if the integrity check fails, it means that some node(s) misbehaved by sending out incorrect data, so the AS initiates the tracking process to be introduced in the next sub-section to identify possible nodes, which have misbehaved based on the AS's collected node commitments.

4.4.3 MISBEHAVED NODE TRACKING

This sub-section describes a tracking process of misbehaving nodes if the integrity check has failed based on Equation 4.5. This process should enable the identification of the

nodes without compromising individual nodes' data privacy. The tracking process consists of two stages:

(a) The AS will ask each node u_i to send $z_i = (g^{\alpha_i} \bmod p) \bmod q$ to the AS securely. Note that although the AS knows each commitment c_i included in α_i as a key, the AS cannot derive any value e_i from z_i (see Section 5.2 for a detailed analysis).

(b) Having received and authenticated such values z_i from all the nodes, the AS then constructs its own value y_i for every node u_i from the stored commitments in a BU way (a top-down (TD) checking method will be discussed later in this sub-section) along the hierarchical node structure and compares y_i with z_i received in the first stage. A mismatch between them (i.e. $y_i \neq z_i$) indicates a misbehaved node, as the aggregated result associated with z_i is different from the committed one corresponding to y_i. In this case, one of the following scenarios must have led to the mismatch:

   (i) u_i misbehaved for wrong z_i, although all its children sent their correct data to u_i.

   (ii) Some of the children misbehaved by sending incorrect data to u_i, although u_i behaved honestly to yield α_i and z_i.

   (iii) Both u_i and some of its children misbehaved for the wrong data generations.

   To pinpoint the misbehaved node(s), the AS needs to verify whether each child u_j submitted correct α_j to u_i. As assumed in Sub-section 4.4.1, there is a secure communication channel between u_i and u_j for message encryption and authentication, so the verification can be done based on the authentication information. However, a main problem with such a verification is that the AS has to get hold of α_j for the verification, which enables AS to decrypt α_j in the way defined in Equation 4.4 for the discovery of the corresponding intermediary aggregated result. This weakens the data privacy. Hence, the scheme proposed in this section does not perform such verification. Instead it simply identifies u_i and its children as suspicious nodes for the misbehaviour. This scheme can work with the one to be defined in Section 4.5 to determine exactly which node is guilty of the misbehaviour in successive aggregations.

   There is a special case for u_i when u_i is a leaf node. The mismatch between y_i and z_i

(i.e. $y_i \neq z_i$) indicates that u_i certainly misbehaved because in this situation u_i has no children to blame for the generation of its incorrect z_i (or α_i). The AS needs to take certain actions against such misbehaviour, e.g. expelling u_i from further aggregations.

After the above identification of the misbehaved or suspicious nodes, the AS can carry on its examination of other nodes. To continue, the AS first requests u_i's parent $u_{\acute{\rho}}$ to send $z_i' = g^{\alpha_i} \bmod p$ to the AS securely if u_i is not the root, where $u_{\acute{\rho}}$ got α_i from u_i during the data aggregation. The AS then assigns received z_i' to y_i (i.e. $y_i = z_i'$) so that the AS can proceed with its value construction and inspection for $u_{\acute{\rho}}$. This tracking process continues until the root node has been examined.

For simplicity, we provide a flow chart as in Figure 11 to illustrate the process involves for detecting misbehaved nodes using APDA^{W}.

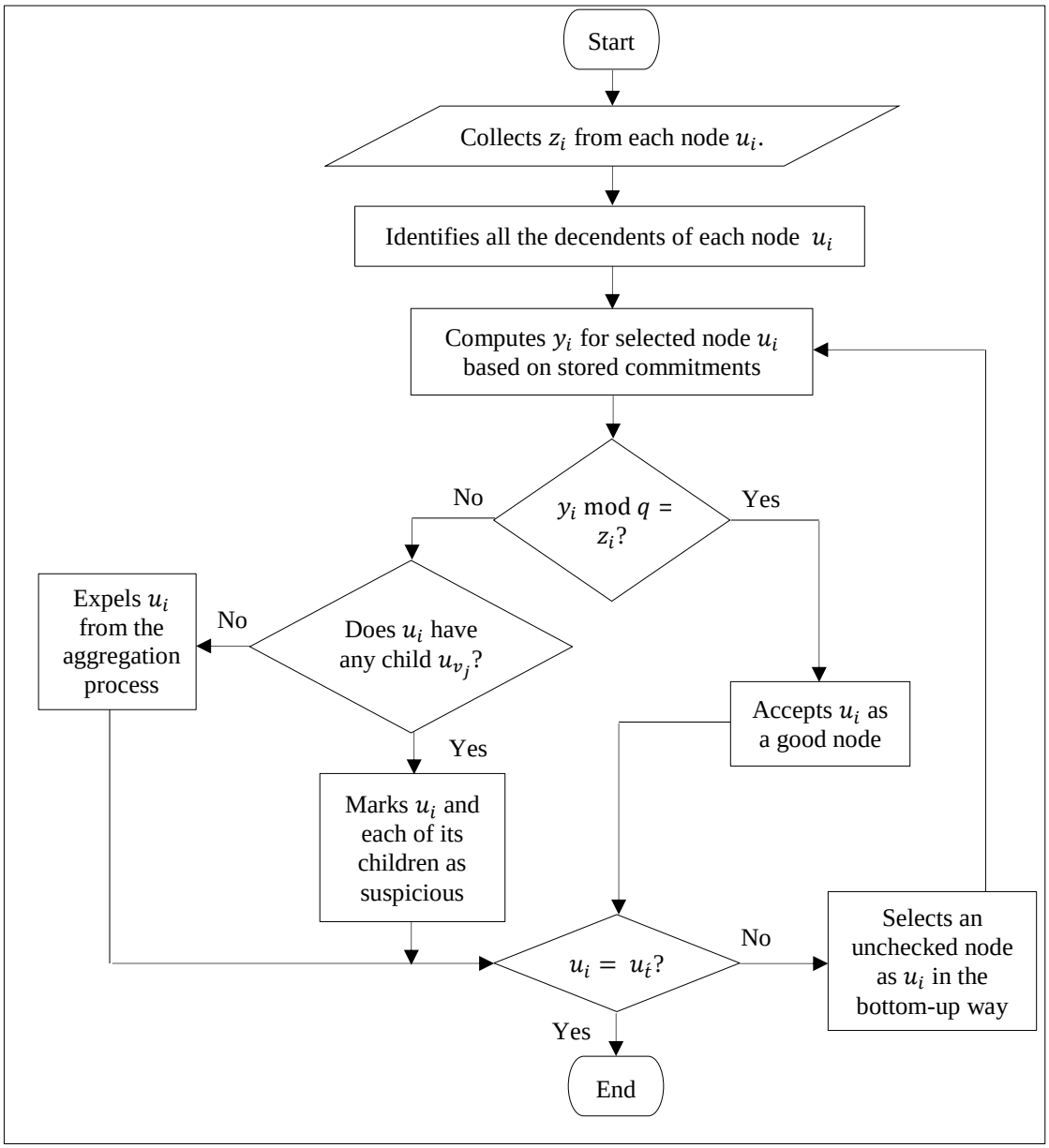


Figure 11: The Misbehaviour Nodes Tracking Process by AS using APDA^{W}

We now present the above tracking method in detail. Having received and authenticated values z_i from all the nodes in the first stage, the AS starts the construction of its values y_i in the second stage. It first builds the hierarchical node relationships from the parent node information stored with the commitments. For example, to find all the children of a given node u_i, the AS searches the commitments to spot those with u_i as the parent (see Sub-section 4.4.1) so that the nodes associated with those commitments are the children of u_i, and if no such commitments are found, u_i is a leaf node without any child.

The AS then begins with the leaf nodes for the value construction. For each leaf node u_i, the AS applies its stored commitment information (c_i, $u_{\acute{\rho}}$) with $c_i = g^{e_i} \bmod p$ defined in Equation 4.2 to perform:

$$y_i = c_i g^{c_i} \bmod p \tag{4.7}$$

The AS then compares y_i with z_i received from u_i. If $y_i \bmod q \neq z_i$, u_i certainly misbehaved due to its leaf node status, as discussed earlier. This check is based on the fact below for correct y_i and z_i (see Sub-section 4.4.1):

$$y_i \bmod q = (c_i g^{c_i} \bmod p) \bmod q = (g^{e_i + c_i} \bmod p) \bmod q = (g^{\alpha_i} \bmod p) \bmod q = z_i.$$

The AS continues the check for all the leaf nodes. Suppose that the value z_i received from every leaf node u_i is valid. The AS then moves to construct the value $y_{\acute{\rho}}$ of each parent node $u_{\acute{\rho}}$ of some leaf nodes with stored information ($c_{\acute{\rho}}$, $u_{\acute{g}}$) and generated values $\{y_{v_1}, \dots, y_{v_\sigma}\}$ for all the σ chidren of $u_{\acute{\rho}}$. This requires the following calculation:

$$y_{\acute{\rho}} = (c_{\acute{\rho}} g^{c_{\acute{\rho}}} \prod_{j=1}^{\sigma} y_{v_j}) \bmod p \tag{4.8}$$

Similar to the comparison for the leaf nodes, the AS compares $y_{\acute{\rho}}$ with $z_{\acute{\rho}}$ received from $u_{\acute{\rho}}$ to confirm $y_{\acute{\rho}} \bmod q = z_{\acute{\rho}}$, which is based on the following relationships:

$$\begin{aligned} y_{\acute{\rho}} \bmod q &= \left(c_{\acute{\rho}} g^{c_{\acute{\rho}}} \left(\prod_{j=1}^{\sigma} y_{v_j}\right) \bmod p\right) \bmod q \\ &= \left(g^{e_{\acute{\rho}} + c_{\acute{\rho}}} \left(\prod_{j=1}^{\sigma} g^{\alpha_{v_j}}\right) \bmod p\right) \bmod q \\ &= \left(g^{e_{\acute{\rho}} + c_{\acute{\rho}} + \sum_{j=1}^{\sigma} \alpha_{v_j}} \bmod p\right) \bmod q \\ &= (g^{\alpha_{\acute{\rho}}} \bmod p) \bmod q \\ &= z_{\acute{\rho}}. \end{aligned}$$

If $y_{\acute{\rho}} \bmod q \neq z_{\acute{\rho}}$ (i.e. $z_{\acute{\rho}}$ is incorrect), $u_{\acute{\rho}}$ and its σ children are deemed as suspicious nodes for the misbehaviour of generating wrong $z_{\acute{\rho}}$.

The above result for $u_{\acute{\rho}}$ can be extended to any other nodes including the root. Thus, the AS repeats such value construction and comparison for every parent. Afterwards, the next upper level of parent nodes will be examined in the same way. This process continues until the root node is checked.

The aforementioned tracking method follows a BU process to examine every node for the misbehaviour identification. However, if a node u_i and all its descendants behaved correctly, the inspection of u_i's value z_i alone by the AS is sufficient to confirm the correctness of all its descendants' values. In other words, correct z_i assures that all the descendants of u_i behaved correctly without checking them, so as to accelerate the tracking process. This is because if any descendant u_j misbehaved, its incorrect α_j would be eventually aggregated into u_i's output α_i, resulting in wrong α_i and hence invalid z_i that would fail the AS's inspection. Note that u_i could misbehave by sending correct z_i to the AS but a wrong α_i' to its parent in order to avoid being detected by the AS. However, such misbehaviour can be spotted when the parent node is examined. This misbehaved node tracking follows a Top-Down (TD) process.

The above discussion also implies that incorrect α_i or z_i itself is insufficient to judge whether u_i misbehaved, as its result α_i (or z_i) depends on those from its descendants. In this case, it is necessary to examine the behaviours of u_i's descendants as well. One way of the examination is to use the BU tracking method described earlier in this sub-section. Alternatively, a TD tracking process can be employed, but after some descendants have been recognised as misbehaved, the BU process is also required to work upwards to confirm whether their ancestors also misbehaved. Evidently, running both the TD and BU examinations slows down the tracking process. Thus, we only use the TD examination to identify correctly behaved nodes, rather than chasing misbehaved nodes, at a higher level so as to exclude the inspection of their descendants for better efficiency.

Another issue of the TD checking is about how many levels of nodes should be checked. It can only gain better efficiency when the nodes checked are fewer than those confirmed as good nodes and excluded from the checking. Otherwise, it would be simpler to run the BU tracking process defined earlier. Nevertheless, determining the number of such levels is challenging, as it relies on several factors such as how many nodes may misbehave and where they could locate in the aggregation tree or hierarchy, which are unknown beforehand. Therefore, we only check all the children of the root node in this work as to be detailed below, and further checking can be applied at other levels. For example, if a child u_i of the root fails the examination and has multiple levels of descendants, then the nodes at the middle level are inspected to see whether the nodes below the middle level

are good and can be ruled out from the inspection.

To check z_i received from each child u_i of the root node, the AS first determines the set of all the descendants of u_i, $\{u_{v_1}, \dots, u_{v_\sigma}\}$, and collects the set of values, $\{c_{v_1}, \dots, c_{v_\sigma}\}$, from their commitments. The AS then computes:

$$\check{c}_i = (c_i + \sum_{j=1}^{\sigma} c_{v_j}) \bmod q, \text{ and } y_i = c_i g^{\check{c}_i} (\prod_{j=1}^{\sigma} c_{v_j}) \bmod p \tag{4.9}$$

If $y_i \bmod q = z_i$ (i.e. z_i is correct), all the descendants of u_i are deemed as good nodes and hence excluded from the further checking. This improves the efficiency of the misbehaviour tracking process, particularly when the number of the descendants is large.

The above verification is based on the following relationships:

$$\begin{aligned} y_i \bmod q &= \left(c_i g^{\check{c}_i} (\prod_{j=1}^{\sigma} c_{v_j}) \bmod p\right) \bmod q \\ &= \left(g^{e_i} g^{c_i + \sum_{j=1}^{\sigma} c_{v_j}} \left(\prod_{j=1}^{\sigma} g^{e_{v_j}}\right) \bmod p\right) \bmod q \\ &= \left(g^{e_i + c_i + \sum_{j=1}^{\sigma} (e_{v_j} + c_{v_j})} \bmod p\right) \bmod q \\ &= (g^{\alpha_i} \bmod p) \bmod q \\ &= z_i. \end{aligned}$$

Having examined all the children of the root node and excluded the good nodes for further checking, the AS initiates the BU tracking process defined earlier to inspect all the other nodes for the misbehaviour identification. Note that the AS makes use of the good child nodes' already computed values y_i, together with those constructed in the BU way for the suspicious child nodes, to yield the root node's value y_t.

4.4.4 PRIVACY IMPROVEMENTS

The APDA^{W} approach presented in the previous sub-sections works well under the assumption that there is no collusion between the AS and any nodes involved in the aggregation. Otherwise, it is vulnerable if the AS colludes with a node u_i to obtain and decrypt its data such as α_i for the intermediary aggregated result or even individual node values, resulting in a breach of the data privacy. To counter this threat, the work in [23] proposes ideas of data slicing and mixing to enhance the protection of individual node values so that the AS would have to compromise multiple nodes in order to obtain a

node's value, which is much harder to achieve.

Two methods are presented in [23] to implement the above ideas. The first one is to divide a node u_i's value e_i defined in Equation 4.1 into $\beta + 1$ random slices such that:

$$e_i = \sum_{j=1}^{\beta+1} \grave{s}_{i,j} \bmod q \tag{4.10}$$

u_i sends each slice $\grave{s}_{i,j}$ to a different parent $u_{\acute{\rho}}$ securely for aggregation. This means that if u_i is secure, the AS would have to compromise or collude with at least these $\beta + 1$ parents to get e_i (e.g. when u_i is a leaf node). The larger β is, the more difficult the AS can gain e_i.

Our APDA^{W} approach can accommodate this slicing method by expanding u_i's commitment into $\{(c_{i,1}, u_{\acute{\rho}_1}), \ldots, (c_{i,\beta+1}, u_{\acute{\rho}_{\beta+1}})\}$, where each slice $\grave{s}_{i,j}$ has a tuple $(c_{i,j}, u_{\acute{\rho}_j})$ with $c_{i,j} = g^{\grave{s}_{i,j}} \bmod p$ and $u_{\acute{\rho}_j}$ as a parent node receiving the slice from u_i. Then u_i transfers each non-aggregated slice $\grave{s}_{i,j}$ $(1 \leq j < \beta + 1)$ to parent $u_{\acute{\rho}_j}$ separately and securely, i.e. $\grave{s}_{i,j}$ is encrypted with a key shared solely between u_i and $u_{\acute{\rho}_j}$[23] and $u_{\acute{\rho}_j}$ then decrypts the encryption received to recover $\grave{s}_{i,j}$ for its data aggregation. u_i only aggregates $\grave{s}_{i,\beta+1}$ with any received data, including non-aggregated slices from some of its children, to yield α_i as defined in Equation 4.3. This indicates that each node u_i applies only $c_{i,\beta+1}$ as a key for the encryption of its aggregated data. Thus, the AS uses $\sum_{i=1}^{n} c_{i,\beta+1}$ to decrypt the final aggregated data received (see Equation 4.4).

During the tracking process, each node u_i only sends $z_i = (g^{\alpha_i} \bmod p) \bmod q$ to the AS. To construct y_i to check the correctness of each z_i, the AS divides u_i's children into two groups, $\{u_{v_1}, \ldots, u_{v_\sigma}\}$ with each node u_{v_j} linked to stored $(c_{v_j,\varphi}, u_i)$ where $1 \leq \varphi < \beta + 1$, and $\{u_{\acute{v}_1}, \ldots, u_{\acute{v}_{\acute{\sigma}}}\}$ with each node $u_{\acute{v}_j}$ associated to $(c_{\acute{v}_j,\beta+1}, u_i)$. Here, the two groups are empty if u_i is a leaf node. This case means that in the data aggregation stage, u_i received a non-aggregated slice from each u_{v_j} in the first group and the aggregated slice from each $u_{\acute{v}_j}$ in the second group, and these slices were then aggregated with u_i's slice $v_{i,\beta+1}$ to create α_i. Hence, the AS performs the following calculation to yield y_i:

$$y_i = (c_{i,\beta+1} g^{c_{i,\beta+1}} (\prod_{j=1\ (1\leq\varphi<\beta+1)}^{\sigma} c_{v_j,\varphi})(\prod_{j=1}^{\acute{\sigma}} y_{\acute{v}_j})) \bmod p \tag{4.11}$$

The AS applies y_i to check z_i received from u_i, in the way described in Sub-section 4.4.3.

This checking mechanism is based on the value of y_i, where:

$$
\begin{aligned}
y_i \bmod q &= \left(c_{i,\beta+1} g^{c_{i,\beta+1}} \left(\prod_{j=1\ (1\le\varphi<\beta+1)}^{\sigma} c_{v_j,\varphi}\right)\left(\prod_{j=1}^{\acute{\sigma}} y_{\acute{v}_j}\right) \bmod p\right) \bmod q \\
&= \left(g^{\grave{s}_{i,\beta+1}+c_{i,\beta+1}} \left(\prod_{j=1\ (1\le\varphi<\beta+1)}^{\sigma} g^{\grave{s}_{v_j,\varphi}}\right)\left(\prod_{j=1}^{\acute{\sigma}} g^{\alpha_{\acute{v}_j}}\right) \bmod p\right) \bmod q \\
&= \left(g^{\grave{s}_{i,\beta+1}+c_{i,\beta+1}+\sum_{j=1\ (1\le\varphi<\beta+1)}^{\sigma} \grave{s}_{v_j,\varphi}+\sum_{j=1}^{\acute{\sigma}} \alpha_{\acute{v}_j}} \bmod p\right) \bmod q \\
&= (g^{\alpha_i} \bmod p) \bmod q \\
&= z_i.
\end{aligned}
$$

Evidently, APDA^{W} can also offer the enhanced privacy while retaining the accountability.

The second method given in [23] does not slice u_i's value e_i but instead lets u_i collaborate with a number of other nodes to encrypt e_i with the keys that are each shared only between u_i and one of the collaborative nodes. The encryption is designed in such a way that when the root passes the final aggregated result to the AS, encrypted e_i is already decrypted during the aggregation process. Thus there is no additional decryption needed from the AS. Evidently, to obtain e_i, the AS has to gain access to these collaborative nodes' shared keys for the decryption, which is more difficult to accomplish. For further details, please refer to paper [23].

Our APDA^{W} approach can also incorporate the above method by simply replacing e_i with its encrypted version and then performing the data aggregation and recovery as well as misbehaviour node tracking in the normal way described in Sub-sections 4.4.1-4.4.3. This enhances APDA^{W} with stronger privacy protection while still offering the accountability. This method will be further exploited in Section 4.6.

4.5 Accountability Enhancement

This section describes the enhancement versions of APDA^{W} to allow misbehaved nodes to be detected certainly during an aggregation process. The two improved versions of APDA are APDA^{S} and APDA^{H}, and the details of both versions are presented below.

4.5.1 APDA^{S} for Strong Accountability

This sub-section describes a strong version of our privacy-preserving data aggregation. As discussed in Sub-section 4.4.3, to surely identify a misbehaved node, a signature

created by the node on its aggregated data is required to hold the node accountable for its behaviour. In this sub-section, we propose a scheme to embed a built-in signature of each node in its aggregated data to accomplish the strong accountability for APDA^S. The main idea used for the scheme is to extend the aggregated data item α_i (defined in Equation 4.3) of each node u_i by yielding a signature of u_i on α_i with its private key k_i and then combining α_i and the signature to form a new aggregated data item $\bar{\alpha}_i$. As a result, $\bar{\alpha}_i$ serves as not only the encryption of u_i's data but also its signature on the data so that u_i cannot falsely deny the generation of $\bar{\alpha}_i$.

The key benefit from the above built-in signature solution is two-fold. First, it allows nodes to generate and verify such a signature more efficiently than a separate signature. Secondly, the solution also enables the AS to examine these built-in signatures more efficiently for tracking down misbehaved nodes, without overloading the AS for the decryption of the final aggregated data. These advantages will be discussed further in the data verification and misbehaviour tracking process for APDA^S. Prior to describing the process of APDA^S in detail, we provide Figure 12 to illustrate the process for the commitment generation and aggregated data encryption performed by each MN using APDA^S. In addition, Figure 13 shows the process of the commitments collection from all MNs, aggregated data recovery and integrity checking executed by the AS.

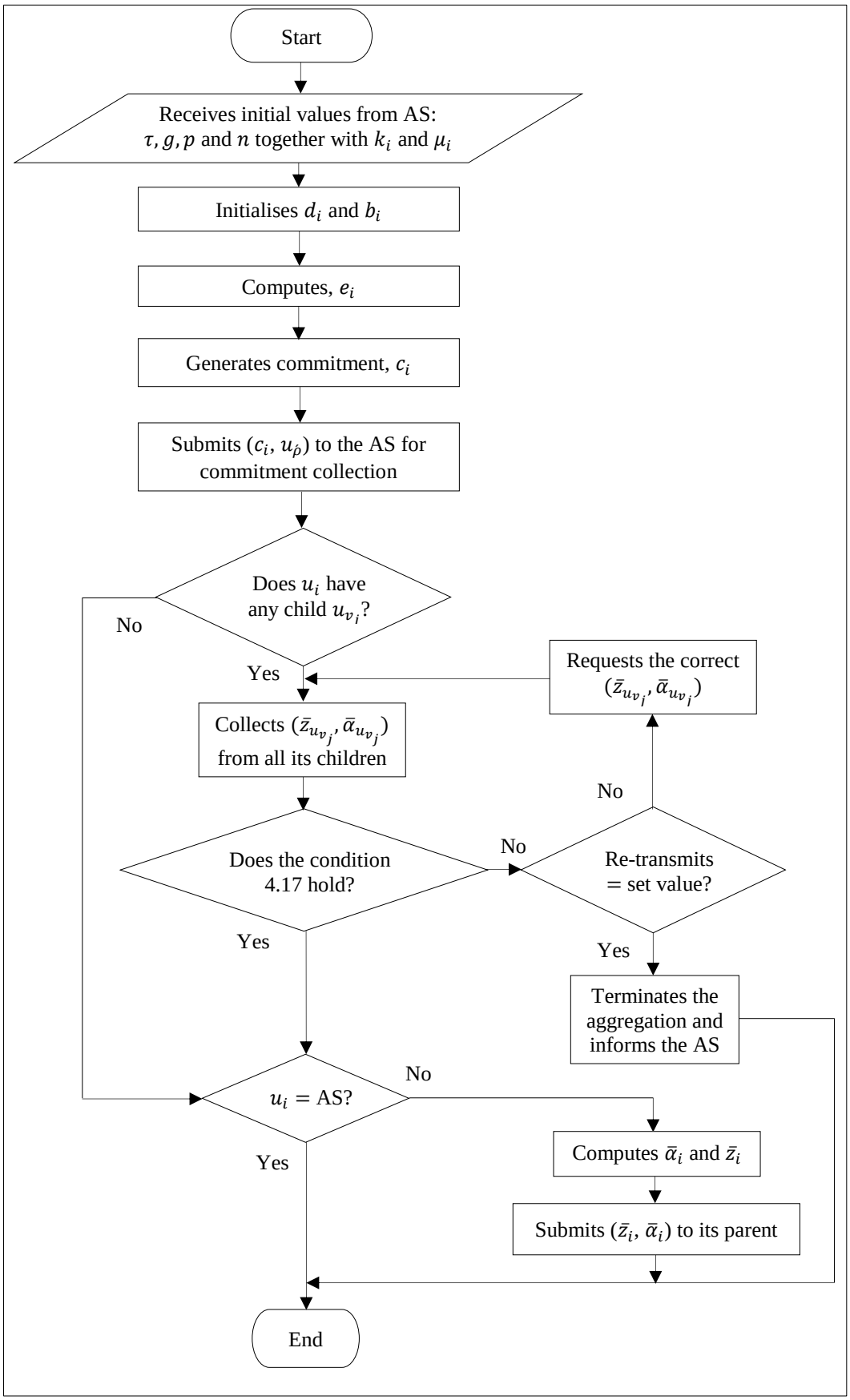

Start
Receives initial values from AS: τ, g, p and n together with k_i and μ_i
Initialises d_i and b_i
Computes, e_i
Generates commitment, c_i
Submits (c_i, $u_{\hat{\rho}}$) to the AS for commitment collection
Does u_i have any child u_{v_i}?
No
Yes
Requests the correct $(\bar{z}_{u_{v_j}}, \bar{\alpha}_{u_{v_j}})$
Collects $(\bar{z}_{u_{v_j}}, \bar{\alpha}_{u_{v_j}})$ from all its children
No
Does the condition 4.17 hold?
No
Re-transmits = set value?
Yes
Yes
Terminates the aggregation and informs the AS
u_i = AS?
No
Yes
Computes $\bar{\alpha}_i$ and $\bar{z}_i$
Submits ($\bar{z}_i$, $\bar{\alpha}_i$) to its parent
End

Figure 12: Commitment Generation, Aggregated Data Encryption and Integrity Checking by each node using APDA^S

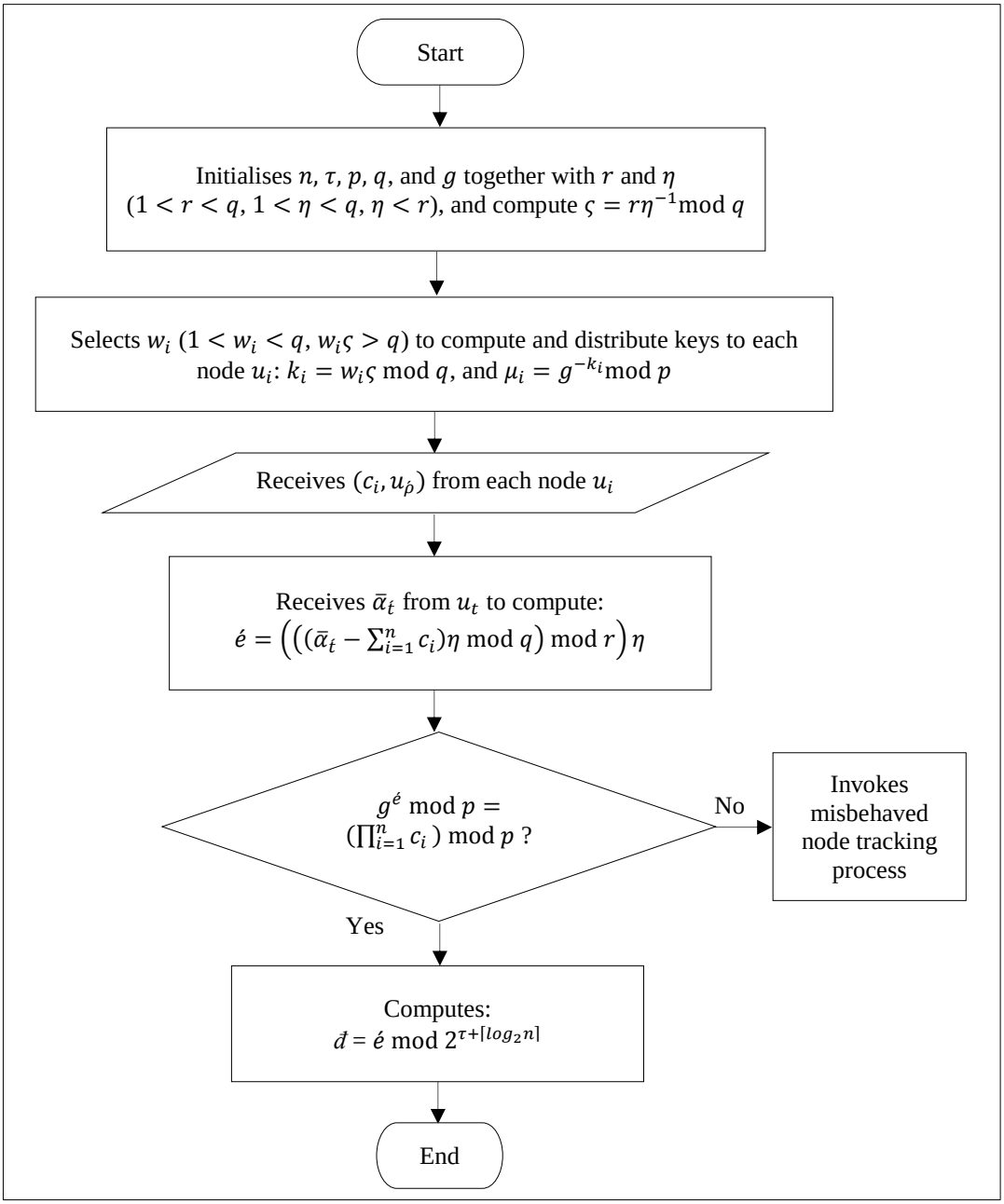


Figure 13: Commitment Collection, Aggregated Data Recovery and Integrity Checking by AS using APDA^S

We now elaborate the above idea to extend APDA^W to incorporate the built-in signatures. To create a signature, the AS needs to assign a pair of private and public keys, k_i and μ_i, to each node u_i ($1 \leq i \leq n$) based on parameters p, q and g defined in Sub-section 4.4.1.

The AS begins with selecting two random primes r and η ($1 < r < q$, $1 < \eta < q$ and $\eta < r$) as its secret keys to compute $\varsigma = r\eta^{-1} \bmod q$, where η^{-1} is the modular multiplicative inverse of η modulo q. For each node u_i, the AS chooses a number w_i ($1 < w_i < q$) such that $w_i\varsigma > q$, to define u_i's private and public keys:

$$k_i = w_i\varsigma \bmod q, \text{ and } \mu_i = g^{-k_i} \bmod p \tag{4.12}$$

The bit lengths of r, η and w_i in each private key need to meet the following conditions:

$$\text{(a) } l_e + \lceil log_2 n + log_2 \eta \rceil < \lfloor log_2 r \rfloor \tag{4.13a}$$

$$\text{(b) } l_h + l_w + \lceil log_2 n + log_2 r \rceil < \lfloor log_2 q \rfloor \tag{4.13b}$$

Here, l_e defined in Sub-section 4.4.1 is the maximum bit length of e_i, l_h expresses the bit length of hash values to be used for signatures, and l_w denotes the maximum bit length of w_i.

The condition (4.13a) above implies $\eta \sum_{i=1}^{n} e_i < 2^{l_e} \eta n \leq 2^{l_e + \lceil log_2 n + log_2 \eta \rceil} < r$, and the condition (4.13b) indicates $hr \sum_{i=1}^{n} w_i < 2^{l_h + l_w} nr \leq 2^{l_h + l_w + \lceil log_2 n + log_2 r \rceil} < q$, where h is a hash value. These two conditions are needed in order to allow the AS to decrypt the final aggregated data while providing built-in signatures for accountability, which will be explained further. After the key generation, the AS sends each pair of keys k_i and μ_i to the corresponding node u_i securely and stores them for the accountability checking to be presented later.

Additionally, the public keys need to be distributed to relevant nodes for signature verifications. One way for the key distribution is to ask the AS to issue a certificate to each node for its public key, and then to let the nodes pass their certificates to others that need to verify their signatures. Alternatively, an identity based scheme such as the one given in [177], [178] can be employed to allow the AS to create a pair of private and public keys for each node from its identity so that others can generate the node's public key from its identity. This avoids the above public key certification. For simplicity, we do not specify any particular means for public key distribution in this thesis. We simply assume that when a node needs to verify a signature, it has already got the valid public key for the verification.

We now present how to incorporate signatures into the data aggregation specified in Section 4.4. The commitment phase described in Sub-section 4.4.1 remains unchanged.

The extension begins with the data aggregation phase and is defined below in four parts for aggregated data encryption, data verification, aggregated data recovery and misbehaviour nodes tracking. The details of each part of the extension are described as follows.

(i) Aggregated Data Encryption

The aggregated data item α_i of every node u_i in Equation 4.3 is extended as follows:

$$x_i = (e_i + c_i + \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j}) \bmod q,\ \bar{z}_i = g^{x_i} \bmod p, \text{ and}$$

$$\bar{\alpha}_i = (x_i + h(s \,\|u_i\|\bar{z}_i \bmod q)k_i) \bmod q \quad (4.14)$$

Here, $\{\bar{\alpha}_{v_1}, \ldots, \bar{\alpha}_{v_\sigma}\}$ denotes a set of aggregated values received by u_i from its σ children, assuming that the correctness of each value has been verified by u_i as to be detailed later. s is the session number for the current aggregation. x_i represents the aggregation of all the received values $\bar{\alpha}_{v_j}$ $(1 \leq j \leq \sigma)$ together with u_i's own data $e_i + c_i$. $\bar{z}_i$ is intended to allow u_i's signature on x_i to be verifiable by another node with no need to know x_i as to be detailed below. $h(s \,\|u_i\|\bar{z}_i \bmod q)$ is a hash of s, node identity u_i and $\bar{z}_i$ concatenated by the operator '||', which signifies the aggregated data processed by u_i in the current session. Evidently, $\bar{\alpha}_i$ serves as an encryption of $e_i + \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j}$ via $c_i + h(s \,\|u_i\|\bar{z}_i \bmod q)k_i$ and a signature on $\bar{z}_i$ (or $h(s \,\|u_i\|\bar{z}_i \bmod q)$) signed with u_i's private key k_i. It is easy to see from the formations of α_i and $\bar{\alpha}_i$ that α_i is extended to include the embedded signature part $h(s \,\|u_i\|\bar{z}_i \bmod q)k_i$ to form $\bar{\alpha}_i$.

After the generation of $\bar{\alpha}_i$, u_i sends $(\bar{z}_i, \bar{\alpha}_i)$ to its parent $u_{\acute{\rho}}$ for verification prior to be aggregated with its parent's data.

(ii) Aggregated Data Verification

To verify the correctness of $(\bar{z}_i, \bar{\alpha}_i)$ received, $u_{\acute{\rho}}$ checks the following condition, where $u_{\acute{\rho}}$ knows u_i's public key μ_i as assumed earlier:

$$g^{\bar{\alpha}_i} \mu_i^{h(s\|u_i\|\bar{z}_i \bmod q)} \bmod p = \bar{z}_i \quad (4.15)$$

If the above condition holds, $u_{\acute{\rho}}$ accepts $(\bar{z}_i, \bar{\alpha}_i)$ because no other node could produce signature $\bar{\alpha}_i$ on $\bar{z}_i$ without knowing private key k_i. Otherwise, $u_{\acute{\rho}}$ rejects $(\bar{z}_i, \bar{\alpha}_i)$ and requests u_i to send correct ones.

The above condition is based on the following fact:

$$g^{\bar{\alpha}_i}\mu_i{}^{h(s\|u_i\|\bar{z}_i \bmod q)} \bmod p$$

$$= g^{x_i+h(s\|u_i\|\bar{z}_i \bmod q)k_i} g^{-h(s\|u_i\|\bar{z}_i \bmod q)k_i} \bmod p$$

$$= g^{x_i} \bmod p$$

$$= \bar{z}_i.$$

In the case where $u_{\acute{\rho}}$ has multiple children, it can perform the above signature verification for all the children's signatures together rather than separately to reduce the number of exponentiations involved, which are the most costly computations. Let $\{(\bar{z}_{v_1}, \bar{\alpha}_{v_1}), \ldots, (\bar{z}_{v_\sigma}, \bar{\alpha}_{v_\sigma})\}$ denote the data received by $u_{\acute{\rho}}$ from its σ children. For the joint signature verification, $u_{\acute{\rho}}$ calculates:

$$\hat{x}_i = \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j} \bmod q, \text{ and } f_i = g^{\hat{x}_i} \bmod p \tag{4.16}$$

and then confirms the following condition:

$$f_i \prod_{j=1}^{\sigma} \mu_{v_j}{}^{h\left(s\|u_{v_j}\|\bar{z}_{v_j} \bmod q\right)} \bmod p = \prod_{j=1}^{\sigma} \bar{z}_{v_j} \bmod p \tag{4.17}$$

This relation is due to:

$$f_i \prod_{j=1}^{\sigma} \mu_{v_j}{}^{h\left(s\|u_{v_j}\|\bar{z}_{v_j} \bmod q\right)} \bmod p$$

$$= g^{\sum_{j=1}^{\sigma} \bar{\alpha}_{v_j}} \prod_{j=1}^{\sigma} \mu_{v_j}{}^{h\left(s\|u_{v_j}\|\bar{z}_{v_j} \bmod q\right)} \bmod p$$

$$= \prod_{j=1}^{\sigma} g^{\bar{\alpha}_{v_j}} \mu_{v_j}{}^{h\left(s\|u_{v_j}\|\bar{z}_{v_j} \bmod q\right)} \bmod p$$

$$= \prod_{j=1}^{\sigma} \bar{z}_{v_j} \bmod p.$$

Clearly, the joint signature verification incurs $\sigma + 1$ exponentiations, whereas the separate verification of each signature requires a total of 2σ exponentiations. Hence the former is computationally more efficient.

(iii) Data Recovery

The above aggregated data generation and verification are repeated by each node until root node u_t passes its aggregated data ($\bar{z}_t$, $\bar{\alpha}_t$) to the AS. Similarly, the AS needs to inspect the validity of ($\bar{z}_t$, $\bar{\alpha}_t$) and then decrypts $\bar{\alpha}_t$ for the recovery of the aggregated randomised value *é* by:

$$\acute{e} = \left(\left((\bar{\alpha}_t - \sum_{i=1}^{n} c_i)\eta \bmod q\right) \bmod r\right)/\eta \tag{4.18}$$

The above relation is due to the following relationships:

$$\acute{e} = \left(\left(\left(\bar{\alpha}_t - \sum_{i=1}^{n} c_i\right)\eta \bmod q\right) \bmod r\right)/\eta$$

$$= \left(\left(\left(x_t + h(s\|u_t\|\bar{z}_t \bmod q)k_t - \sum_{i=1}^{n} c_i\right)\eta \bmod q\right) \bmod r\right)/\eta$$

$$= \left(\left(\left(e_t + c_t + \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j} + h(s\|u_t\|\bar{z}_t \bmod q)k_t - \sum_{i=1}^{n} c_i\right)\eta \bmod q\right) \bmod r\right)/\eta$$

$$= \left(\left(\left(\sum_{i=1}^{n} e_i + \sum_{i=1}^{n} c_i + \sum_{i=1}^{n} h(s\|u_i\|\bar{z}_i \bmod q)k_i - \sum_{i=1}^{n} c_i\right)\eta \bmod q\right) \bmod r\right)/\eta$$

$$= \left(\left(\left(\sum_{i=1}^{n} e_i + \sum_{i=1}^{n} h(s\|u_i\|\bar{z}_i \bmod q)w_i r\eta^{-1}\right)\eta \bmod q\right) \bmod r\right)/\eta$$

$$= \left(\left(\left(\eta\sum_{i=1}^{n} e_i + r\sum_{i=1}^{n} h(s\|u_i\|\bar{z}_i \bmod q)w_i\right) \bmod q\right) \bmod r\right)/\eta$$

$$= \left(\left(\eta\sum_{i=1}^{n} e_i + r\sum_{i=1}^{n} h(s\,\|u_i\|\bar{z}_i \bmod q)w_i\right) \bmod r\right)/\eta$$

$$= \left(\eta\sum_{i=1}^{n} e_i\right)/\eta$$

$$= \sum_{i=1}^{n} e_i\,.$$

The above relationships make use of the conditions (4.13a) and (4.13b) stated earlier, namely

$$\eta\sum_{i=1}^{n} e_i < r, \text{ and } \eta\sum_{i=1}^{n} e_i + r\sum_{i=1}^{n} h(s\,\|u_i\|\bar{z}_i \bmod q)w_i < q.$$

After the recovery of *é*, the AS checks its integrity and derives result *đ* using Equations 4.5 and 4.6 respectively.

(iv) Misbehaviour Tracking

The combined TD and BU tracking processes presented in Sub-section 4.4.3 are also applicable here after extensions. Prior to describing the tracking process using APDA^{S} in detail, we provide a flow chart as in Figure 14.

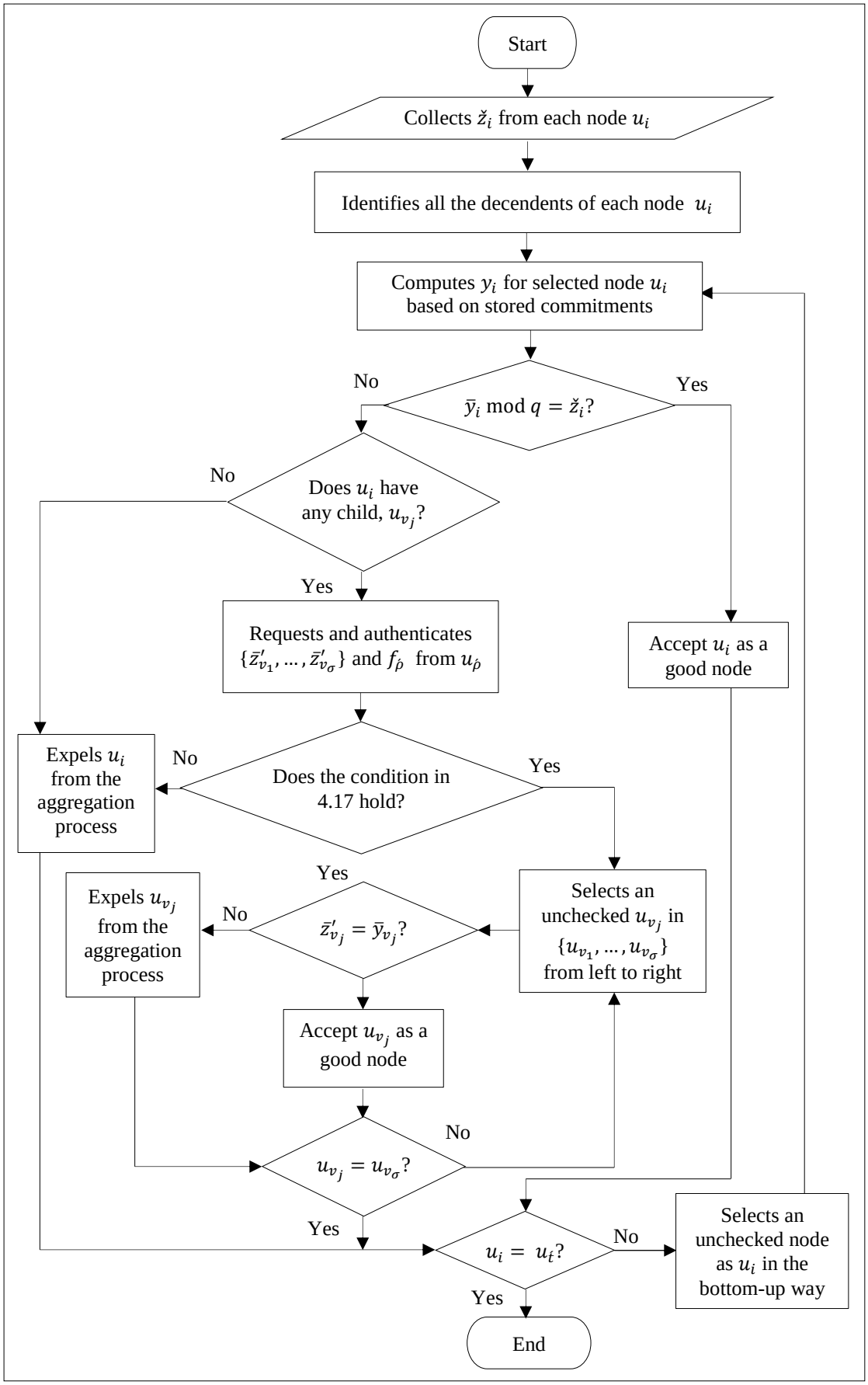

Start
Collects $\check{z}_i$ from each node u_i
Identifies all the decendents of each node u_i
Computes y_i for selected node u_i based on stored commitments
No
$\bar{y}_i \bmod q = \check{z}_i$?
Yes
No
Does u_i have any child, u_{v_j}?
Yes
Requests and authenticates $\{\bar{z}'_{v_1}, \dots, \bar{z}'_{v_\sigma}\}$ and $f_{\hat{\rho}}$ from $u_{\hat{\rho}}$
Accept u_i as a good node
Expels u_i from the aggregation process
No
Does the condition in 4.17 hold?
Yes
Yes
Expels u_{v_j} from the aggregation process
No
$\bar{z}'_{v_j} = \bar{y}_{v_j}$?
Selects an unchecked u_{v_j} in $\{u_{v_1}, \dots, u_{v_\sigma}\}$ from left to right
Accept u_{v_j} as a good node
No
$u_{v_j} = u_{v_\sigma}$?
Yes
$u_i = u_t$?
No
Selects an unchecked node as u_i in the bottom-up way
Yes
End

Figure 14: The Misbehaviour Nodes Tracking Process by AS using APDA^S

We first extend the BU process to accurately identify misbehaved nodes. In the first stage of the process, each node u_i sends $\check{z}_i = \bar{z}_i \bmod q$ to the AS securely. In the second stage, the AS constructs the following value for each leaf node u_i as defined in Equation 4.7:

$$\bar{y}_i = c_i g^{c_i} \bmod p$$

and confirms $\bar{y}_i \bmod q = \check{z}_i$, where

$\check{z}_i = \bar{z}_i \bmod q = (g^{x_i} \bmod p) \bmod q = (g^{e_i + c_i} \bmod p) \bmod q =$
$(c_i g^{c_i} \bmod p) \bmod q.$

Suppose that every $\bar{y}_i$ matches received $\check{z}_i$ (i.e. $\bar{y}_i \bmod q = \check{z}_i$). The AS then proceeds to build a value $\bar{y}_{\acute{\rho}}$ for each parent node $u_{\acute{\rho}}$ of some leaf nodes with its stored commitment information ($c_{\acute{\rho}}$, $u_{\acute{g}}$). Let $\{\bar{y}_{v_1}, \dots, \bar{y}_{v_\sigma}\}$ be the set of values constructed by the AS for the σ chidren of $u_{\acute{\rho}}$. The AS applies them to yield $\bar{y}_{\acute{\rho}}$ as follows:

$$\bar{\varepsilon}_{\acute{\rho}} = \left(c_{\acute{\rho}} + \sum_{j=1}^{\sigma} h\left(s || u_{v_j} || \bar{y}_{v_j} \bmod q\right) k_{v_j}\right) \bmod q, \text{ and}$$

$$\bar{y}_{\acute{\rho}} = c_{\acute{\rho}} g^{\bar{\varepsilon}_{\acute{\rho}}} \left(\prod_{j=1}^{\sigma} \bar{y}_{v_j}\right) \bmod p \tag{4.19}$$

This leads to the following relationships:

$$\begin{aligned}
\bar{y}_{\acute{\rho}} &= c_{\acute{\rho}} g^{\bar{\varepsilon}_{\acute{\rho}}} \left(\prod_{j=1}^{\sigma} \bar{y}_{v_j}\right) \bmod p \\
&= g^{e_{\acute{\rho}}} g^{c_{\acute{\rho}} + \sum_{j=1}^{\sigma} h\left(s \,\|\, u_{v_j} \,\|\, \bar{y}_{v_j} \bmod q\right) k_{v_j}} g^{\sum_{j=1}^{\sigma} x_{v_j}} \bmod p \\
&= g^{e_{\acute{\rho}} + c_{\acute{\rho}} + \sum_{j=1}^{\sigma} \left(x_{v_j} + h\left(s \,\|\, u_{v_j} \,\|\, \bar{y}_{v_j} \bmod q\right) k_{v_j}\right)} \bmod p \\
&= g^{e_{\acute{\rho}} + c_{\acute{\rho}} + \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j}} \bmod p \\
&= g^{x_{\acute{\rho}}} \bmod p \\
&= \bar{z}_{\acute{\rho}}.
\end{aligned}$$

Hence, the AS compares between constructed $\bar{y}_{\acute{\rho}}$ and received $\check{z}_{\acute{\rho}}$. If $\bar{y}_{\acute{\rho}} \bmod q = \check{z}_{\acute{\rho}}$ (i.e. both $u_{\acute{\rho}}$ and its children behaved properly), the AS then continues its value construction and comparison for other nodes. Otherwise, some of $u_{\acute{\rho}}$ and its children misbehaved. In this case, the AS needs a further investigation by verifying the signatures of $u_{\acute{\rho}}$'s children

in the joint way defined earlier to see whether some of them submitted wrong data to $u_{\acute{\rho}}$. For this verification purpose, the AS requests $u_{\acute{\rho}}$ to send its possessed $\{\bar{z}'_{v_1}, \ldots, \bar{z}'_{v_\sigma}\}$ and $f_{\acute{\rho}}$ to the AS. Note that the AS knows the public key μ_{v_j} of every child u_{v_j}. If the joint signature verification is negative (i.e. $f_{\acute{\rho}} \prod_{j=1}^{\sigma} \mu_{v_j}^{h\left(s \| u_{v_j} \| \bar{z}'_{v_j} \bmod q\right)} \bmod p \neq \prod_{j=1}^{\sigma} \bar{z}'_{v_j} \bmod p$), the AS concludes that $u_{\acute{\rho}}$ misbehaved because $u_{\acute{\rho}}$ failed to properly carry out the same verification. Here, although the AS could further inspect each child's signature to confirm whether some of them are guilty of wrong doing as well, this would impose extra resource consumptions on $u_{\acute{\rho}}$ for providing the signatures to the AS, and misbehaved $u_{\acute{\rho}}$ could refuse to cooperate with the AS for the inspection. Thus we do not check the children's behaviour in this case.

Otherwise, the positive verification outcome means that each child u_{v_j} cannot deny the provision of its $\bar{z}'_{v_j}$ to $u_{\acute{\rho}}$. The AS then examines the correctness of every $\bar{z}'_{v_j}$ by comparing it with constructed $\bar{y}_{v_j}$. If each pair of $\bar{z}'_{v_j}$ and $\bar{y}_{v_j}$ match (i.e. $\bar{z}'_{v_j} = \bar{y}_{v_j}$, meaning that $\bar{z}'_{v_j}$ is correct), $u_{\acute{\rho}}$ surely misbehaved as invalid $\check{z}_{\acute{\rho}}$ can only be caused by $u_{\acute{\rho}}$'s own data. Conversely, when a pair of $\bar{z}'_{v_j}$ and $\bar{y}_{v_j}$ do not match (i.e. $\bar{z}'_{v_j} \neq \bar{y}_{v_j}$), u_{v_j} is guilty of supplying incorrect $\bar{z}'_{v_j}$ to $u_{\acute{\rho}}$, which is evidenced by its verified signature (i.e. u_{v_j} correctly signed incorrect $\bar{z}'_{v_j}$). After the identification of the misbehaved nodes among all the children, the AS also checks whether $u_{\acute{\rho}}$ misbehaved, by constructing $\bar{y}'_{\acute{\rho}}$ from $f_{\acute{\rho}}$ received from $u_{\acute{\rho}}$ instead of correct $\{\bar{y}'_{v_1}, \ldots, \bar{y}'_{v_\sigma}\}$ built by the AS:

$$\bar{y}'_{\acute{\rho}} = c_{\acute{\rho}} g^{c_{\acute{\rho}}} f_{\acute{\rho}} \bmod p \tag{4.20}$$

and confirming $\bar{y}'_{\acute{\rho}} \bmod q = \check{z}_{\acute{\rho}}$ due to the relation below:

$$\bar{y}'_{\acute{\rho}} \bmod q = \left(g^{e_{\acute{\rho}} + c_{\acute{\rho}}} f_{\acute{\rho}} \bmod p\right) \bmod q = \left(g^{e_{\acute{\rho}} + c_{\acute{\rho}} + \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j}} \bmod p\right) \bmod q =$$
$$\left(g^{x_{\acute{\rho}}} \bmod p\right) \bmod q = \check{z}_{\acute{\rho}}.$$

The purpose for the above check is to see whether $u_{\acute{\rho}}$ correctly yielded its $\check{z}_{\acute{\rho}}$ from $f_{\acute{\rho}}$. If $\bar{y}'_{\acute{\rho}} \bmod q \neq \check{z}_{\acute{\rho}}$, $u_{\acute{\rho}}$ misbehaved as well. Otherwise, $\bar{y}'_{\acute{\rho}} \bmod q = \bar{z}_{\acute{\rho}}$ would be the result.

After the completion of the above examination, the AS uses the same way to carry on the tracking process. Similar to the tracking in Sub-section 4.4.3, in the case of incorrect $\check{z}_{\acute{\rho}}$,

the AS asks $u_{\acute{\rho}}$'s parent $u_{\acute{g}}$ to send its possessed $\bar{z}_{\acute{\rho}}$ to the AS securely if $u_{\acute{\rho}}$ is not the root. The AS then assigns received $\bar{z}_{\acute{\rho}}$ to $\bar{y}_{\acute{\rho}}$ (i.e. $\bar{y}_{\acute{\rho}} = \bar{z}_{\acute{\rho}}$) so as to enable the subsequent construction of $\bar{y}_{\acute{g}}$ for $u_{\acute{g}}$ to proceed. This tracking process continues until the root node is checked.

To extend the TD tracking process discussed in Sub-section 4.4.3, let $u_{\acute{\rho}}$ be a child of the root node. The AS then finds out all the descendants of $u_{\acute{\rho}}$, expressed as $\{u_{v_1}, \ldots, u_{v_\pi}\}$, and utilises the values in $\{\check{z}_{v_1}, \ldots, \check{z}_{v_\pi}\}$ received from these descendants, rather than those constructed by the AS, to compute:

$$\check{\varepsilon}_{\acute{\rho}} = \left(c_{\acute{\rho}} + \sum_{j=1}^{\pi} c_{v_j} + \sum_{j=1}^{\pi} h\left(s \left\| u_{v_j} \right\| \check{z}_{v_j}\right) k_{v_j}\right) \bmod q, \text{ and}$$

$$\check{y}_{\acute{\rho}} = c_{\acute{\rho}} g^{\check{\varepsilon}_{\acute{\rho}}} \prod_{j=1}^{\pi} c_{v_j} \bmod p \tag{4.21}$$

This relation is due to:

$$\begin{aligned}
\check{y}_{\acute{\rho}} &= c_{\acute{\rho}} g^{\check{\varepsilon}_{\acute{\rho}}} \prod_{j=1}^{\pi} c_{v_j} \bmod p \\
&= g^{e_{\acute{\rho}}} g^{c_{\acute{\rho}} + \sum_{j=1}^{\pi} c_{v_j} + \sum_{j=1}^{\pi} h\left(s \left\| u_{v_j} \right\| \check{z}_{v_j}\right) k_{v_j}} \prod_{j=1}^{\pi} g^{e_{v_j}} \bmod p \\
&= g^{e_{\acute{\rho}} + c_{\acute{\rho}} + \sum_{j=1}^{\pi} \left(e_{v_j} + c_{v_j} + h\left(s \left\| u_{v_j} \right\| \check{z}_{v_j}\right) k_{v_j}\right)} \bmod p \,.
\end{aligned}$$

As each descendant u_{v_j} contributes $e_{v_j} + c_{v_j} + h\left(s||u_{v_j}||\check{z}_{v_j}\right) k_{v_j}$ to the aggregation, it is clear that $\check{y}_{\acute{\rho}}$ represents the aggregation of all the contributions of $u_{\acute{\rho}}$'s descendants together with $u_{\acute{\rho}}$'s own contribution $e_{v_j} + c_{v_j}$. If $\check{y}_{\acute{\rho}} \bmod q = \check{z}_{\acute{\rho}}$ (i.e. $\check{z}_{\acute{\rho}}$ is correct), the AS can exclude $u_{\acute{\rho}}$'s descendants from further checking.

4.5.2 APDA^{H} FOR HYBRID ACCOUNTABILITY

Comparing APDA^{W} presented in Section 4.4 with APDA^{S} in Sub-section 4.5.1, it is clear that APDA^{W} offers better efficiency as APDA^{S} requires additional exponentiations for data generation and verification as well as extra data transmission. However, the higher computational costs incurred by APDA^{S} bring the benefit of stronger accountability to pinpoint misbehaved nodes, whereas APDA^{W} can identify a group of suspicious nodes without the certainty on which of them actually committed the misbehaviour.

One way to balance the strengths and weaknesses of the two methods is to apply them in a combined manner, represented as APDA^{H}. That is, APDA^{W} operates first. If the

aggregated result fails the integrity check by Equation 4.5, the tracking process of APDA^{W} will enable the AS to find a group of misbehaved or suspicious nodes. In the next round of data aggregation, the AS specifically targets those suspicious nodes by demanding them to run APDA^{S} and letting the rest continue the operation with APDA^{W}. If some suspicious nodes misbehave again, APDA^{S} will certainly spot them. It is also possible that a repeated failure of the integrity check leads to the identification of new suspicious nodes, so that APDA^{S} needs to expand its operation to cover these new nodes in the subsequent round of aggregation. In the worst case, every node eventually executes APDA^{S}. On the other hand, when there is no misbehaved node, only APDA^{W} is implemented on all the nodes.

There are two main differences in running APDA^{W} and APDA^{S} separately. First, any node $u_{\dot{\rho}}$ with a mixture of normal and suspicious children only needs to verify the signatures received from those suspicious children. Secondly, in the tracking process, $u_{\dot{\rho}}$ only passes the signature information of its suspicious children to the AS for verification in case a further investigation is required and the signature part of $\bar{\varepsilon}_{\dot{\rho}}$ constructed for $u_{\dot{\rho}}$ by the AS only includes the signatures of the suspicious children (see Sub-section 4.5.1). In other words, reducing the number of suspicious children to 0 turns the value construction and verification for $u_{\dot{\rho}}$ by APDA^{S} into that by APDA^{W}.

Hence the above differences do not affect the application of the signature verification and misbehaved node tracking methods of APDA^{S} to APDA^{H}. Additionally, the final aggregated value $\bar{\alpha}_t$ received by the AS may contain signatures only from some nodes. Similarly, the recovery method of APDA^{S} remains valid for the AS to recover the aggregated result $\vec{d}$ from $\bar{\alpha}_t$. Even when there is no signature included in $\bar{\alpha}_t$, the recovery method still works.

Evidently APDA^{H} achieves strong accountability for the behaviour of the targeted suspicious nodes, while offering a level of efficiency between those of APDA^{W} and APDA^{S}, depending on the number of suspicious nodes. The fewer suspicious nodes an aggregation involves, the more efficient APDA^{H} is.

To make APDA^{H} more efficient, we take two measures to reduce the amounts of computation and communication. The first is to let the current round of data aggregation

skip the commitment collection phase and start directly from the data aggregation phase, after suspicious nodes have been identified using APDA^{W} in the previous round. In other words, the current round is only to re-aggregate the data collected in the previous round, assuming that the data is still valid.

The second measure is for the current round to request only the suspicious nodes to generate new data using APDA^{S}, and to ask the non-suspicious nodes to do nothing or reuse their own data from the previous round for the new data re-aggregation through APDA^{W} as to be detailed below. Evidently the two measures avoid commitment re-generation and re-communication, exclude some non-suspicious nodes from the data re-aggregation, and minimise the aggregation processing for the other non-suspicious nodes. These result in significantly reduced computations and communications for better efficiency.

We now detail how to fulfil the above second measure. To begin with the current round of data aggregation, the AS should first notify each node u_i of what to perform. We employ 2-bit codes for the notification, where the first bit of a code indicates whether u_i is a suspicious node with 0 = no and 1 = yes, and the second bit signifies whether u_i has any suspicious descendant also with 0 = no and 1 = yes. This leads to the following combinations accompanied with the actions to be taken by u_i:

00: u_i does nothing in the current round.

01: u_i uses APDA^{W} to aggregate its own data together with those received from its children.

10: u_i initiates the current round by using APDA^{S} to aggregate its own data together with those received from its children in the previous round, or just yield its own data if u_i is a leaf node.

11: u_i applies APDA^{S} to aggregate its own data together with those received from its children.

These codes are further interpreted below:

(a) The first code is for the case where neither u_i nor any of its descendants are suspicious nodes, meaning that its aggregated data passed to its ancestors in the previous round is still valid and hence there is no need for u_i to do anything in the

current round. This case is also valid when u_i is a leaf node with no descendant.

(b) The second code is used when u_i is not a suspicious node but at least one of its descendants is a suspicious node. In this case, u_i utilises APDA^{W} to re-aggregate its data, which was produced in the previous round, with the data received in the previous round from each child with the 00 code plus the data newly generated in the current round from each child with a non-00 code. This means that u_i's re-aggregation reuses the previous data from each child of u_i, which is not a suspicious node and has no suspicious descendant, together with the new data from each child that is a suspicious node or has at least one suspicious descendant.

(c) The third code is for the situation where u_i is a suspicious node but has no suspicious descendants. That is, if u_i is not a leaf node, each of its descendants has code 00 and does nothing in the current round. Hence, u_i makes a start via APDA^{S} to yield its new data if u_i is a leaf node, and yield and aggregate its new data with those received from its children in the previous round otherwise.

(d) The fourth code applies when u_i is a suspicious node and has at least one suspicious descendant. The only difference from the second case is that u_i utilises APDA^{S} for the data re-aggregation.

Based on these 2-bit codes, the AS forms a 2*n*-bit string $\check{s}$ to define what each of the *n* nodes should do in the current round of data aggregation, and multicasts $\check{s}$ to all the nodes that can then check their own codes. If a node u_i has code 00 or 10, it either does nothing or makes a start with no need to get new data from its children. Otherwise (i.e. the code is 01 or 11), u_i has to wait for new data from some of its children for the aggregation, and it can find out which children will send it the new data, by checking each child's code in $\check{s}$. If the code is not 00, u_i needs to receive new data from this child, and otherwise (i.e. the child does nothing in the current round), u_i simply reuses the data gathered from this child in the previous round for its aggregation.

In addition, to maintain the data security, we slightly amend the formation of x_i defined for APDA^{S} in Equation 4.14 by replacing c_i in x_i with a new secret $\acute{c}_i$ to ensure that each re-aggregation via APDA^{S} uses a new secret for the data encryption. To generate $\acute{c}_i$, node u_i gets its c_i and the current session number s' to calculate a hash $\acute{h} = h(c_i \| s')$, concatenates $\acute{h}$ multiple times (i.e. $\tilde{h} = \acute{h} \| ... \| \acute{h}$) until the total bit length of $\tilde{h}$ is at least

$\lfloor log_2 q \rfloor - 1$, and then takes the first $\lfloor log_2 q \rfloor - 1$ bits of $\tilde{h}$ as $\acute{c}_i$. Here we require that different aggregation rounds use different session numbers. The amended x_i is given below:

$$\acute{x}_i = (e_i + \acute{c}_i + \sum_{j=1}^{\sigma} \bar{\alpha}_{v_j}) \bmod q \tag{4.22}$$

Note that APDA$^{\mathrm{W}}$ is applied only to a node u_i with code 01 and u_i simply reuses its data yielded in the previous round to aggregate new data from its descendants in the current round. Namely, u_i does not generate new data in the current round, so there is no need to amend APDA$^{\mathrm{W}}$.

From the above discussion, it is clear that the aggregation tree is divided into two parts. The upper part consists of the nodes with codes 01, 10 and 11, which produce new aggregated data, whereas the lower part comprises those nodes with code 00, which do nothing in the current round of aggregation. Obviously, the bigger the lower part is, the less computation and communication are needed, and thus the more efficient the re-aggregation is.

Similarly, the misbehaviour tracking process also benefits from the above measure for better efficiency. This is because the AS only needs to gather necessary data from the nodes in the upper part and re-construct their corresponding values for comparison. The nodes in the lower part do not contribute to the new aggregation, so the AS simply reuses their values constructed in the previous round for the current misbehaviour tracking. Obviously, the bigger the lower part is, the more efficient the tracking process is.

4.6 Other Privacy-Preserving Operations

Our APDA$^{\mathrm{M}}$ (M $\in$ {W, S, H}) approach proposed in Sections 4.4 and 4.5 can be extended to other privacy-preserving aggregations including a group of comparative operations such as the determination of a maximum, minimum, median or percentile for a given data set. As an example, in this section we illustrate a new way to apply APDA$^{\mathrm{M}}$ for the AS to find the maximal value among those contributed by the *n* nodes, and also present a novel and more efficient method for the determination of a maximum. These solutions can be adapted to the other comparative operations.

4.6.1 MAXIMAL VALUE FINDING

The main idea to be explored for such finding is to partition the range of the nodes' data into a predefined number γ of equal size non-overlapped sub-ranges. For each sub-range, every node u_i sets a number as 0 if its value d_i falls outside the sub-range and 1 otherwise. u_i then concatenates its γ numbers into a single value d_i' in the ascending order from the lowest sub-range to the highest one. Clearly, d_i' contains at most one 1 with the rest being 0s. Afterwards, such values d_i' are aggregated into a value *đ'* for the AS using the APDA^{M} approach. It is easy to see that each sub-range has a number in *đ'*, which counts the nodes with their data in the sub-range, and the total of these numbers is not more than *n*. The AS then picks up the highest sub-range with a nonzero number (i.e. the maximal value is in this sub-range) to repeat the above process for another round of range partition and data aggregation only within the selected sub-range. This continues until the number of values in a sub-range falls to 1, meaning that the value in the highest such sub-range with a non-zero number is the maximum.

We now spell out the above Count-based Maximal value finding method, denoted as Max^{C}. As stated in Sub-section 4.4.1, the maximal bit length of a node value d_i is τ, namely, d_i is in the range $[0, 2^{\tau} - 1]$. Let γ be set as $\gamma = 2^{\theta}$ representing the number of sub-ranges resulting from the partition of a given range with $\theta > 0$. ϵ is the number of decimal digits needed to represent the largest number of nodes falling in a particular sub-range, and τ' is the maximal bit length of a concatenated value d_i', where $\tau' = \lceil log_2(10^{\epsilon} - 1) \rceil \gamma$ with $\lceil log_2(10^{\epsilon} - 1) \rceil$ being the maximal bit length of the number of nodes in any sub-range. This indicates that the number $\acute{n}$ of rounds for range partition and data aggregation to determine the maximal value is $\acute{n} = \lceil \tau/\theta \rceil$. Note that the last partition yields sub-ranges with only one value in each, and may result in fewer sub-ranges than γ, in which case each node u_i randomly sets 0 or 1 in concatenated value d_i' for any unused sub-range.

Suppose that the $\acute{s}$th sub-range $R_{\beta-1,\acute{s}} = [\upsilon_{\beta-1,\acute{s}}, \upsilon'_{\beta-1,\acute{s}}]$ from the previous round $\beta - 1$ of range partition ($1 \leq \beta \leq \acute{n}$) is selected for the next round β of range partition, where $R_{\beta-1,\acute{s}}$ is the original data range when $\beta = 1$, i.e. $R_{0,1} = [0, 2^{\tau} - 1]$. The *j*th sub-range in the βth round is defined as:

$$R_{\beta,j} = [\upsilon_{\beta-1,\acute{s}} + (j-1)\lceil 2^{\tau-\beta\theta}\rceil, \upsilon_{\beta-1,\acute{s}} + j\lceil 2^{\tau-\beta\theta}\rceil - 1] \quad (4.23)$$

Here, j is within $1 \leq j \leq \gamma$ for $1 \leq \beta < \acute{n}$ or $\beta = \acute{n}$ and $\tau \bmod \theta = 0$, meaning that each partition before the last one produces γ sub-ranges or the last partition also yields γ sub-ranges when there is no unused sub-range, i.e. $\tau \bmod \theta = 0$. Otherwise, j falls in $1 \leq j \leq 2^{\tau \bmod \theta}$ for the case of $\beta = \acute{n}$ and $\tau \bmod \theta \neq 0$, indicating that the number of sub-ranges created from the last partition is fewer than γ.

For example, given $\tau = 8$ and $\theta = 3$, the number of rounds needed to find the maximum is $\acute{n} = \lceil \tau/\theta \rceil = \lceil 8/3 \rceil = 3$, and the original data range is $R_{0,1} = [0, 255]$. The number of sub-ranges produced from each of the first two rounds is $\gamma = 2^{\theta} = 2^3 = 8$, and the number for the last round is $2^{\tau \bmod \theta} = 2^{8 \bmod 3} = 2^2 = 4$. The sub-ranges for each round are listed below:

Round1: $R_{1,1} = [0, 31]$, $R_{1,2} = [32, 63]$, $R_{1,3} = [64, 95]$, $R_{1,4} = [96, 127]$, $R_{1,5} = [128, 159]$,

$R_{1,6} = [160, 191]$, $R_{1,7} = [192, 223]$, $\boldsymbol{R_{1,8} = [224, 255]}$(selected sub-range).

Round 2: $R_{2,1} = [224, 227]$, $R_{2,2} = [228, 231]$, $R_{2,3} = [232, 235]$, $R_{2,4} = [236, 239]$, $R_{2,5} = [240, 243]$,

$R_{2,6} = [244, 247]$, $\boldsymbol{R_{2,7} = [248, 251]}$ (selected sub-range), $R_{2,8} = [252, 255]$.

Round 3: $R_{3,1} = [248, 248]$, $\boldsymbol{R_{3,2} = [249, 249]}$ (with the max), $R_{3,3} = [250, 250]$, $R_{3,4} = [251, 251]$.

Here, we suppose that the number of nodes in $R_{1,8}$ after the first round of data aggregation is greater than 0 (i.e. the maximal value is in this sub-range), so it is partitioned into another 8 sub-ranges for the second round. Assuming that the numbers of nodes in $R_{2,8}$ and $R_{2,7}$ after the second round of data aggregation are 0 and over 0 respectively (i.e. the maximum is in $R_{2,7}$), $R_{2,7}$ is divided into 4 sub-ranges for the third round with each sub-range containing only one value, meaning that it cannot be further partitioned. Let the numbers of nodes in $R_{3,4}$ and $R_{3,3}$ be both 0, and the number for $R_{3,2}$ be 1. As $R_{3,2}$ contains

only one value, the maximum is 249.

Moreover, assume that u_i's value d_i is equal to maximum 249, and the number of nodes falling in any sub-range does not exceed 2 decimal digits, i.e. $\epsilon = 2$. The number d_i' constructed by u_i for each of the above three rounds is $d_i' = 10^{14}$ for round 1, $d_i' = 10^{12}$ for round 2 and $d_i' = 10^2$ for round 3. The maximal bit length τ' of d_i' is $\tau' = \lceil log_2(10^\epsilon - 1) \rceil \gamma = 7 \times 8 = 56$.

The process for finding the maximum begins with the AS specifying τ based on the bit size of data d_i possessed by the individual nodes as well as θ and ϵ. To start the first round of data aggregation, the AS sends τ, θ and ϵ to all the nodes. Each node u_i then calculates the number of rounds, $\acute{n} = \lceil \tau/\theta \rceil$, and the number of sub-ranges from each partition, $\gamma = 2^\theta$ or $\gamma = 2^{\tau \bmod \theta}$ for the last round.

For each round β of range partition and data aggregation, every node u_i defines all the sub-ranges based on Equation 4.23 for round β using τ and θ, and constructs d_i' with bit length $\tau' = \lceil log_2(10^\epsilon - 1) \rceil \gamma$ for data aggregation in relation to its d_i and the defined sub-ranges, in the way described earlier. Afterwards, u_i applies the agreed APDA^M to aggregate its d_i' with those received from its children if u_i is not a leaf node, and sends the result to its parent or the AS if u_i is the root.

Upon receipt of the aggregated result for the current round, the AS decides the sub-range $R_{\beta,j}$ containing the maximum, and informs all the nodes of j to start partitioning $R_{\beta,j}$ for the next round of data aggregation. This continues until the last round successfully finishes, so that the AS obtains the maximal value. Evidently the AS gets the maximum without knowing which node has the value. Hence the data privacy is preserved.

4.6.2 Efficient Maximal Value Finding

In this sub-section, we explore the idea described in Sub-section 4.6.1 in a new way. Instead of getting the number of nodes for each sub-range that contains the nodes' values, we use a single bit to indicate whether there is a node with its value in the sub-range. This Bit-based Maximal value finding method, signified as Max^B, offers two benefits. The first is to increase the number of sub-ranges yielded from each round of partition to $\gamma = \tau'$, namely, every value d_i' becomes a bit string, and each of its τ' bits represents a sub-range.

This reduces the number of rounds needed to find the maximum so as to gain better efficiency. The second is to simplify the aggregation process using simpler and more efficient bitwise operations to replace the numerical ones, leading to a one-stage process for aggregation instead of the two-stage one presented in Sub-section 4.4.1. This has a positive impact on the efficiency and implementation of the method.

We now present the new method in detail. It follows the same definitions of τ, $\acute{n}$, γ, θ and τ' given in Sub-section 4.6.1, where $\gamma = \tau'$. Also the construction of each string d_i' remains similar, i.e. for each sub-range j ($1 \leq j \leq \tau'$), if node u_i's value d_i falls in the sub-range, the corresponding bit of d_i' , expressed as $d_i'[j]$, is assigned $d_i'[j] = 1$, and otherwise $d_i'[j] = 0$.

However, the aggregation of strings d_i' from all the n nodes is different. For each round of aggregation, every node u_i directly sends its encrypted d_i', which will be defined later, to the AS in such a way that the AS cannot decrypt it to obtain d_i' but can get the following decrypted aggregation with $\oplus$ denoting the bitwise exclusive-or operation:

$$\bar{d} = \bigoplus_{i=1}^{n} d_i', \text{ with } \bar{d}[j] = \bigoplus_{i=1}^{n} d_i'[j] \text{ for every } j\ (1 \leq j \leq \tau') \qquad 4.24$$

If $\bar{d}[j] = 1$, there is at least one node with its value in the jth sub-range for the current round. It also means that there are an odd number of 1s in the jth bits among the n strings contributed by all the nodes, in order to yield $\bar{d}[j] = 1$. Nevertheless, when $\bar{d}[j] = 0$, it is possible that there are an even number of 1s in the jth bits. In other words, we cannot rule out that the maximum is not in a sub-range with 0 for its corresponding bit in $\bar{d}$.

To rectify the above problem, we replace each string d_i' with ϑ (> 0) strings, $d_{i,1}', \ldots, d_{i,\vartheta}'$, which have the same length τ'. To form these strings, u_i checks every sub-range against its value d_i. If d_i falls in the jth sub-range, u_i randomly selects $d_{i,\acute{s}}'[j] = 1$ or $d_{i,\acute{s}}'[j] = 0$ for every string $d_{i,\acute{s}}'$ ($1 \leq \acute{s} \leq \vartheta$). For any other bit, u_i simply sets $d_{i,\acute{s}}'[\bar{\jmath}] = 0$ ($\bar{\jmath} \neq j$). The purpose for the use of these multiple strings and randomisation is to provide a high probability that in the case of at least one node with its value in the jth sub-range, there is a $\acute{s}$ such that the n bits $d_{i,\acute{s}}'[j]$ ($1 \leq i \leq n$) contain an odd number of 1s. Hence the result of aggregating these bits $d_{i,\acute{s}}'[j]$ via $\oplus$ is 1, denoted as $\bar{d}_{\acute{s}}[j] = 1$, which indicates that some nodes have their values in the jth sub-range.

After the above string construction, each node u_i sends its encrypted ϑ strings, $d'_{i,1}$, …, $d'_{i,\vartheta}$, to the AS for aggregations. Similarly the AS can compute the following decrypted aggregation without obtaining individual strings $d'_{i,\acute{s}}$:

$$\bar{d} = \bigvee_{\acute{s}=1}^{\vartheta} \bar{d}_{\acute{s}} = \bigvee_{\acute{s}=1}^{\vartheta} (\bigoplus_{i=1}^{n} d'_{i,\acute{s}}) \quad 4.25$$

Here, the bitwise-or operation signified as $\bigvee$ implies that $\bar{d}[j]$ is 1 if at least one of the ϑ bits $\bar{d}_{\acute{s}}[j]$ is 1.

Number ϑ can be determined based on the required probability of letting the AS get $\bar{d}[j] = 1$ when there is at least one node with its value in the *j*th sub-range. This probability is calculated as:

$$\bar{p} = \frac{2^{\vartheta}-1}{2^{\vartheta}} \quad 4.26$$

For example, given $\bar{p} = 0.999$, $\vartheta = 10$ is needed as the number of strings for each round of data aggregation. To generate 64 sub-ranges from a partition, each string (e.g. $d'_{i,\acute{s}}$) has 64 bits, and each node needs a total of 640 bits for its 10 strings.

Similar to the Max^{C} method in Sub-section 4.6.1, the AS selects the highest sub-range with 1 in its associated bit in $\bar{d}$ (i.e. the maximum is highly likely in the sub-range) for the next round of range partition and data aggregation. This is repeated until the $\acute{n}$ rounds are completed.

To prevent the AS from gaining a node's strings, we propose an encryption scheme based on the data mixing idea [23] discussed in Sub-section 4.4.4 to allow each node to collaborate with others for its string encryption. More specifically, suppose that a node u_i has agreed with σ_i (> 0) other nodes in $\{u_{v_1}, \ldots, u_{v_{\sigma_i}}\}$ for the collaboration, and u_i shares a different key $\acute{k}_{i,v_j}$ with each node u_{v_j} ($1 \le j \le \sigma_i$) where $\acute{k}_{i,v_j} = \acute{k}_{v_j,i}$. Note that the collaboration is a mutual relationship in the sense that if u_{v_j} is in u_i's collaborator set, u_i also appears in u_{v_j}'s collaborator set. Also we suppose that two different nodes very likely have different sets of collaborative nodes. The details on how to find the collaborative nodes and establish the shared keys are given in [23].

For the βth round of aggregation with a session number s, u_i applies all the σ_i shared keys to encrypt its $\acute{s}$th string $d'_{i,\acute{s}}$ for every $\acute{s}$ ($1 \le \acute{s} \le \vartheta$):

$$\bar{e}_{i,\acute{s}} = d'_{i,\acute{s}} \oplus (\oplus_{j=1}^{\sigma_i} \acute{h}(s \,\|\beta\, \|\acute{s}\|u_i\|u_{v_j}||\acute{k}_{i,v_j})) \quad 4.27$$

Here, $\acute{h}(\cdot)$ is a one-way hash function with the bit size of its output equal to the size τ' of $d'_{i,\acute{s}}$, where the output serves as a secret for the encryption of $d'_{i,\acute{s}}$. Note that the output size of a standard hash function such as SHA [179], [180] may be different from τ'. If the size is longer, the first τ' bits are taken as the result of $\acute{h}(\cdot)$. Otherwise, multiple hashes (e.g. with each being the hash of the previous one apart from the first one) can be concatenated with a total length of at least τ', and then the first τ' bits are used as the output of $\acute{h}(\cdot)$. Additionally, for '$u_i\|u_{v_j}$' in $\acute{h}(\cdot)$, we presume $i < v_j$, and otherwise '$u_{v_j}||u_i$' should replace '$u_i\|u_{v_j}$'. This ensures that both u_i and u_{v_j} generate the same hash value for their encryptions respectively.

After the encryption, u_i directly transfers the ϑ encrypted strings, $\bar{e}_{i,1}$, …, $\bar{e}_{i,\vartheta}$, to the AS securely. Here we follow the same assumption in Sub-section 4.4.1 that the strings sent directly from each node to the AS are correct after they have been authenticated by the AS using an agreed scheme that is not part of the work proposed in this thesis.

Once the AS has collected and successfully authenticated the strings from all the nodes, the AS aggregates them to produce:

$$\bar{d} = \mathsf{V}_{\acute{s}=1}^{\vartheta}(\oplus_{i=1}^{n} \bar{e}_{i,\acute{s}}) \quad 4.28$$

This leads to the following decrypted result, showing the aggregation of all the strings contributed by the *n* nodes:

$$\bar{d} = \mathsf{V}_{\acute{s}=1}^{\vartheta}(\oplus_{i=1}^{n} d'_{i,\acute{s}}) \quad 4.29$$

It is easy to see this outcome because each secret hash $\acute{h}(s \,\|\beta\, \|\acute{s}\|u_i\|u_{v_j}||\acute{k}_{i,v_j})$ in $\bar{e}_{i,\acute{s}}$ also appears in $\bar{e}_{v_j,\acute{s}}$ as a secret, namely, the two identical secrets connected by the $\oplus$ operation cancel each other and hence contribute to the decryption of both $\bar{e}_{i,\acute{s}}$ and $\bar{e}_{v_j,\acute{s}}$. This is why there is no additional decryption needed from the AS.

Based on the above description, it is evident that the AS cannot recover an original string $d'_{i,\acute{s}}$ from $\bar{e}_{i,\acute{s}}$ without knowing the associated secret hashes. Although the AS could make use of received $\bar{e}_{i,\acute{s}}$ and $\bar{e}_{v_j,\acute{s}}$ to try $\bar{e}_{i,\acute{s}} \oplus \bar{e}_{v_j,\acute{s}}$, it is unlikely that the AS can get $d'_{i,\acute{s}} \oplus$

$d'_{v_j,\acute{s}}$ as u_i and u_{v_j} very likely have different sets of collaborative nodes, namely, $\bar{e}_{i,\acute{s}}$ and $\bar{e}_{v_j,\acute{s}}$ are very likely encrypted using different sets of secret hashes by u_i and u_{v_j}, respectively.

4.6.3 HYBRID APPROACH

The two methods Max^{C} and Max^{B} proposed in the previous two sub-sections can be combined to take advantage of Max^{B} for more quickly narrowing down the range containing the maximum and Max^{C} for more accurately determining the maximum. This hybrid approach is represented as Max^{H}.

One way to implement Max^{H} is to utilise Max^{B} to more efficiently find the maximum $\acute{m}$, as it offers a much larger number of sub-ranges for each partition and hence fewer rounds of simpler data aggregation. However, there is a small probability that $\acute{m}$ is not really the maximum, as discussed in Sub-section 4.6.2. Hence, after the identification of $\acute{m}$ by Max^{B}, Max^{C} is employed to count how many nodes have their values in the range $[\acute{m} + 1, 2^{\tau} - 1]$ with τ being the bit length of data d_i, as Max^{C} can certainly find the correct maximum but with lower efficiency. If the number is 0, $\acute{m}$ is surely the maximum. Otherwise, Max^{C} continues its operation by partitioning $[\acute{m} + 1, 2^{\tau} - 1]$ until a new maximum is found. This combined method assures the correctness of the maximum while showing a performance advantage.

4.7 SUMMARY

In this chapter, we have detailed the development of APDA for misbehaved nodes detection and its extensions for finding the maximal value among aggregated data. We have proposed APDA^{W}, APDA^{S} and APDA^{H} to allow data to be aggregated securely in a privacy-preserving manner. These methods also provide a novel approach to supporting the accountability of mobile users in the data aggregation process. In addition, the extended versions of APDA offer new ways for the maximum value finding. We have proposed Max^{C}, Max^{B} and Max^{H} to allow a maximum value to be determine without disclosing mobile users' privacy to the CS. In the next chapter, our analysis on the schemes' security and performance evaluation will be provided and discussed to support

our claim that all of the proposed schemes are efficient and secured to suit in MSS implementation.

CHAPTER 5

SECURITY ANALYSIS AND PERFORMANCE EVALUATION OF APDA AND MAX METHODS

5.1 INTRODUCTION

In this chapter, we analyse the security of all the versions of APDA and Max proposed in the previous chapter. This chapter begins with a security analysis against both internal and external threats described in Chapter 4. Then, this chapter provides a comparison with existing methods to select the best one to be compared quantitatively with our scheme. Finally, our schemes' experimental results and necessary discussions will be presented, which provides meaningful evidence to support the conclusions that we will make at the end of this chapter.

5.2 SECURITY ANALYSIS

By considering both internal and external threats described in Section 4.2, our scheme may face several attacks that intend to compromise the scheme's security. Those attacks include:

(i) A curious AS, which intends to discover particular data that is related to a specific individual. Disclosing the data to the AS unavoidably breaches the privacy of the data owner.

(ii) Malicious MNs, which intend to reveal the content of the sensed data from other nodes, which also unavoidably invades the privacy of the data owners.

(iii) Communication link devices like Wi-Fi routers, which are public and not secure and reliable.

(iv) An adversary, which prevents the AS from aggregating a correct result by submitting fake data to the aggregation process.

Therefore, our scheme has been designed to prevent the aforementioned security attacks and can be analysed as below.

5.2.1 A CURIOUS AS

In MSSs, an AS is supposed to be a reliable party in the aggregation process. Nevertheless, it could be curious and want to discover the content of individual data it receives from each MN. This simple threat could lead to a disaster to the MNs as revealing their individual data to the AS would unavoidably disclose the privacy of the data owners. Our security analysis on such a threat can be further detailed as follows.

(i) Commitment collection

By using APDA, the AS receives the commitments sent by all MNs in the form of $c_i = g^{e_i} \bmod p$, which serves as a one-way function, for each node u_i. Any attempt to discover data content e_i from commitment c_i by the curious AS is well known to be hard due to the one-way feature of c_i. In detail, e_i is the randomised data in a plaintext form, and the best known technique to derive e_i from c_i is by using the Number Field Sieve (NFS) [181], [182]. The success of solving this problem is dependent on the length of the modulus p. The complexity of the NFS method increases as the length of p increases.

To obtain e_i from c_i, the AS may compute the discrete logarithm of c_i, which is e_i in a multiplicative group modulo p (i.e. $\mathbb{Z}_p^*$ and $e_i \in \mathbb{Z}_p^*$) as below:

$$\log_g c_i = e_i \tag{5.1}$$

If the AS can compute the above discrete logarithm, then it can retrieve u_i's plaintext e_i. However, the computation is infeasible even when using the most efficient technique

known as NFS, as the size of p is typically 1024 bits or larger [182]. Based on this reason, we can say that our proposed scheme does not allow the curious AS to retrieve any information about plaintext e_i from commitment c_i passed to the AS.

(ii) Aggregated Data Encryption

By using APDA^{W} on the one hand, the data is transmitted securely via the hierarchical structure to prevent an individual data item from being disclosed to the AS. The data is aggregated using Equation 4.3 as below:

$$\alpha_i = (e_i + c_i + \sum_{j=1}^{\sigma} \alpha_{v_j}) \bmod q.$$

The AS only receives the final aggregated data α_t and then decrypts it as follows:

$$\acute{e} = (\alpha_t - \sum_{i=1}^{n} c_i) \bmod q.$$

In this case, the AS is able to recover the final aggregated data, but not the data of any individual node as the AS does not receive any intermediately aggregated data. Hence our scheme preserves the data privacy of individual mobile users against the curious AS.

Nevertheless, the intermediate aggregated data $\alpha_i = (e_i + c_i) \bmod q$ might face a brute force attack by a curious node as α_i is transmitted to its parent for aggregation. Since e_i is encrypted using the commitment c_i only known by the node u_i and the AS, having an α_i by the curious node would not allow it to recover e_i.

On the other hand, the security of $\bar{\alpha}_i$ that is encrypted using APDA^{S} is guaranteed as the scheme's security level has been increased by embedding a signature signed with a private key k_i (i.e. $k_i = w_i\zeta \bmod q$) in the ciphertext data. Having ciphertext $\bar{\alpha}_i$ would not allow any curious node to reveal its content due to the plaintext being encrypted via both the commitment and signature. As long as key k_i and commitment c_i are not revealed to any curious node, the ciphertext is secured against the brute force attack on the ciphertext.

Furthermore, as the AS distributes secret key k_i to each node u_i for encryption, a curious node might want to discover the information in k_i by launching a brute force attack on the key itself. As described in Sub-section 4.5.1, each secret key k_i contains r (part of the master key for decryption), i.e. $k_i = w_i\varsigma \bmod q$ with $\varsigma = r\eta^{-1} \bmod q$. By having k_i, a curious node could try to gain some related information such as the size of η as we have set $\eta < r$. Nevertheless, having $k_i$will not allow the curious node to reveal any

information on r as long as ς is not exposed to any node in the network and there is no known plaintext attack.

(iii) Aggregated Data Verification and Misbehaved Node Detection

In MSSs, data verification is an essential procedure to undergo before recovering the aggregated data in the plaintext form. The reason is that the data aggregation could come from misbehaving nodes that intend to prevent the aggregator from getting the correct result by submitting fake data during the aggregation process. By using APDA, the aggregated data can be verified based on the AS's collected node commitments. Such a verification procedure guarantees the integrity of the aggregated data. The reason is that the AS can compare the collected node commitments with the received aggregated data based on Equation 4.5 as follows:

$$g^{\acute{e}} \bmod p = (\prod_{i=1}^{n} c_i) \bmod p.$$

As no information will be leaked in this verification process, the privacy of each MN is preserved.

If the integrity check has failed, the AS can detect the misbehaved nodes without affecting the privacy of the mobile users. Such misbehaved nodes can be tracked down by using APDA^{W} because all the nodes including the misbehaved ones have to submit z_i directly to the AS. Then, the AS will compare the received z_i from every node, u_i with the constructed value y_i based on its stored commitments. If both values are different, then the AS concludes node u_i certainly misbehaved if u_i is a leaf node with no children. In this case, the AS needs to take certain actions such as expelling u_i from further aggregations as a result of this misbehaviour. Otherwise, the AS simply identifies node u_i and its children as suspicious nodes. In such misbehaviour node tracking, each z_i consists of individual ciphertext data α_i. Revealing α_i allows the curious AS to decrypt it due to the AS possess the related individual key for the decryption. Nevertheless, this attack is intractable as z_i is generated using the similar way to the commitment generation. Therefore, the data privacy of each individual MN is preserved against the curious AS.

However, as described in Section 4.4, APDA^{W} is not able to track down a misbehaved node with certainty if the node has children. The reason is that there is no verification

process executed by the node to verify its children data. In contrast, APDA^S provides such verification so as to allow the misbehaved node to be tracked down at every level of the aggregation with certainty by the AS. In this verification process, each child has to submit its encrypted data α_i, embedded with a signature signed with its secret key, together with z_i. However, α_i is submitted to the AS in the form similar to that of the commitment generation (see Sub-section 4.5.1), namely the AS can verify but cannot obtain α_i.Thus, the data privacy of each individual MN is preserved against the curious AS.

In addition, to allow the verification process by each parent node, public keys μ_i need to be distributed to relevant nodes. Based on the same security analysis described in Sub-section 5.2.1 (i), disclosing the public keys μ_i will not allow any parties to retrieve any information about their associated secret keys k_i.

We now describe the security analysis of this verification process as follows. The distribution of a public key μ_i using APDA^S attracts a curious node to recover the secret key k_i from μ_i where:

$$k_i = w_i\zeta \bmod q \text{ and } \mu_i = g^{-k_i} \bmod p$$

For this purpose, the curious node computes the discrete logarithm of μ_i:

$$\text{Log}_g\, \mu_i = -k_i \tag{5.2}$$

Retrieving the corresponding secret key enables the curious node to generate a correct signature on the corresponding ciphertext. However, according to [183], for large p (i.e. $p = 1024$ bits), the best known algorithm for solving this problem is Pollard's rho method, which takes about $\sqrt{\pi q/2}$ steps. In our scheme setting, since our choice of $q \approx 2^{186}$, then the expression (5.2) represents an infeasible amount of computation, so the public key is secure against the DLP attack.

(iv) Maximal Value Finding

In MSSs, an aggregator needs to communicate with all MNs to retrieve the maximum value within the aggregated data. In this thesis, we have proposed two methods Max^C and Max^B for maximal value finding on the data contributed by n nodes. By using the former method, MNs respond to the AS's request via the hierarchical data aggregation and the

response is delivered in an encrypted form based on the APDA approach. Thus, the security of the former method is inherited from the security of APDA.

In contrast, the later method uses a different approach from APDA. By using this method, each MN directly transmits a number of encrypted strings $\bar{e}_{i,ś}$ (encrypted using the bitwise exclusive OR operation) to the AS. The AS then collects and aggregates them to receive the response in a plaintext form as it's already decrypted during the aggregation process as described in Sub-section 4.6.2. By using this scheme, the AS would not be able to retrieve any individual node's original strings as we implement the idea of data mixing so that each node collaborates with others for its string encryption and decryption. Furthermore, each string is encrypted using a one-way hash function on a set of keys shared between a set of collaborative nodes. Without knowing the associated secret hashes, the AS would not be able to retrieve or decrypt any individual string $d'_{i,ś}$ from $\bar{e}_{i,ś}$. If the AS picked any $\bar{e}_i$ and $\bar{e}_{v_{i,ś}}$ to perform $\bar{e}_i \oplus \bar{e}_{v_{i,ś}}$, then the AS is unlikely to yield $\bar{d}_i \oplus \bar{d}_{v_{i,ś}}$ due to it is very likely that u_i and $u_{v_{i,ś}}$ have different sets of collaborative nodes. This can be proven by using a simple probability test as follows.

For this test, we let u_i and u_{v_j} be randomly distributed in a network consisting of 50 nodes. We assume $\bar{e}_{i,ś}$ and $\bar{e}_{v_{j,ś}}$ are the data received from u_i and u_{v_j} respectively by the AS. We know that $\bar{e}_{i,ś}$ and $\bar{e}_{v_{j,ś}}$ are encrypted by using a set of keys shared between n collaborative nodes and the probability of u_i or $u_{v_j}$to pick one node to be one of its collaborative nodes is $1/48$. Thus, the probability of the two nodes to pick the same node to be one of their collaborative nodes is:

$$P(X \cap Y) = P(X) \times P(Y) = 1/48 \times 1/48 = 1/2304 = 0.0004$$

As the probability is too small, we conclude that u_i and u_{v_j} are very likely to have different sets of collaborative nodes.

5.2.2 MALICIOUS MNS

By using APDA, each MN submits the sensed data to its parent node for the aggregation process. The parent node aggregates the received data with its own data hierarchically. In the situation where a malicious MN aggregates the data received from its children, it may want to discover the content of the data, which unavoidably breaches the data privacy of

its children. Nevertheless, since the sensed data is encrypted using the homomorphic encryption, the malicious MN would not be able to disclose any information without having the secret key that is only known by the data owner and the AS. In the case of the AS and MNs colluding to discover any contents of the intermediate aggregated data, our APDA still can protect the data via two types of methods, which are the slicing and mixing technique and encryption using a shared key between collaborative nodes. The former technique prevents the malicious MNs to discover any intermediate aggregated data as the data has been sliced prior to the encryption and aggregation process as defined in Equation 4.10, i.e.

$$e_i = \sum_{j=1}^{\beta+1} \grave{s}_{i,j} \bmod q.$$

The malicious MNs need to collaborate with $\beta + 1$ parent nodes to discover the information that relates to the targeted MN. When the number of collaborative nodes is high, discovering the content of the data would be hard.

In the second technique, data e_i is encrypted using the secret keys shared between u_i and each of its collaborative nodes. Such a key is created and kept secret by u_i and one of its collaborative nodes. Therefore, the malicious MNs need to obtain the keys from all the collaborative nodes in order to decrypt encrypted e_i. This is hard to accomplish when the number of collaborative nodes is high.

5.2.3 PUBLIC COMMUNICATION DEVICES

In MSSs, all communication and data transmission are executed via open public wireless devices like Wi-Fi routers. As such devices are not reliable and vulnerable to several attacks like data interception and modification [175], the adversaries could attempt to intercept the communication devices to disclose information contained in the aggregated data. By using APDA, such attacks can be prevented as the data is transmitted in its ciphertext form. Attempting to discover the content of the ciphertext data will be intractable without a valid key for the decryption. Therefore, the adversary will be exhausted to invade the data privacy of mobile users.

5.3 A Comparison with Existing Solutions

Our proposed APDA scheme has considered all essential requirements as stated in Section 4.3 to allow efficient data processing in its ciphertext form. In this section, we select some encryption schemes that can be compared directly to our scheme in terms of these requirements.

5.3.1 Scheme Applicability

The existing encryption schemes have been designed based on specific target users and environments. Different users and environments require different security features with respect to different application scenarios. For instance, the schemes like MAI, RPDA and VerDP are designed for WSNs, cloud environments or smart metering applications [36], [57], [69], [184] with static environments. In contrast, schemes like PDA, PPSense, PLAM and PPDM and VPA [23], [25], [66], [67], [70] are designed for MSSs, which consider mobility, data integrity and users' privacy as their main requirements. Even though the later schemes are suitable to be implemented in MSSs, they are lack of other aspects like scheme functionality and users' accountability. Furthermore, those schemes' complexity needs to be reduced to prolong the battery lifetime of mobile devices. To remedy such limitations on those schemes, we have proposed a scheme that can be implemented in MSS as this scheme considers the issues of complexity, mobility, functionality and user accountability.

5.3.2 Data Security and Users' Privacy

A scheme that provides data security and supports users' privacy in data aggregation is really demanded to prevent a curious AS and malicious MNs from disclosing the data content and invading individual users' data privacy for illegitimate interests. Thus, a homomorphic encryption scheme is the best option to encrypt the data in an aggregation process. This is due to the fact that, in these encryption schemes, all the computations on the aggregated data are done in its ciphertext form without decryption [23], [25], [36], [57], [66], [67], [69], [70], [72], [184]. Nevertheless, implementing those techniques prevents the aggregated data from being verified as such schemes do not provide any further avenues for aggregated data verification. Consequently, the genuine aggregated

data cannot be ensured by the aggregator for further processing to retrieve some statistical results on the aggregated data.

5.3.3 DATA INTEGRITY

The integrity of aggregated data is the main crucial aspect in data aggregation. The reason is that such data will be used for generating further important information to be referred by other users. Thus, schemes that could verify the integrity of the aggregated data have been proposed and designed as in [23], [25], [36]. Those schemes support such a requirement through a process called integrity checking. By using such a process, the aggregated data will be compared with the information collected at the beginning of the process to determine the genuineness of the final results. However, those schemes' complexities are too high to implement in MSSs.

The reason of such inefficiency is that each ciphertext involves many hash values and the computation of these values by each node consumes the energy of the node's battery. In contrast, we have proposed a scheme, i.e. APDA^{W}, which allows the aggregated data to be verified in a privacy-preserving manner without the need of hash functions. Such a design improves the efficiency of data to be aggregated and decrypted. Furthermore, by using the existing schemes, the AS has to create a lot of keys that need to be assigned to each node. As a result, such schemes require heavy computation especially on exponentiations. Furthermore, those schemes demand heavy communication costs for the AS to send a secret key to every node in the network. In contrast, our proposed scheme APDA^{W} only uses a node's commitment as its encryption key. Thus, the AS does not need to generate a unique key for every node so as to improve the efficiency of the whole system.

5.3.4 SCHEME'S FUNCTIONALITY

Another requirement for this comparison is the capability of a scheme to support statistical functions on the aggregated data. The proposed schemes in [23], [36], [67], [69] are able to support additive and non-additive functions on ciphertext data inefficiently. The main reason is that those schemes require heavy computation and communication costs to generate such statistical results. For instance, the work in [23] has proposed a technique for determining the maximal value of the aggregated data. This technique

requires more conversations to determine the maximum value of the aggregated data due to each conversation only dividing the range into half and this division leading to a long step by step process to complete. In contrast, the Max proposed in our scheme increases the range for every round so as to reduce the number of conversations for maximal value finding. Such a method increases the scheme efficiency as well as prolongs the network lifetime by reducing the communication costs on the whole network. Furthermore, in the improved version of Max^B, we have introduced a simpler and more efficient way for data to be processed by the AS without the need for decryption. Such an improvement leads to a one-stage process for aggregation instead of the two-stage process for Max^C.

5.3.5 USERS' ACCOUNTABILITY

Finally, a scheme that supports accountability on mobile users is really demanded in the aggregation process. The reason is that, in data aggregation, some nodes may be curious, malicious or both. To the best of our knowledge, MAI [57] is the only one that addresses node accountability in WSNs. However, our APDA is significantly different from [57] in the following aspects.

First, the scheme proposed in [57] targets only nodes in WSNs with static topologies, while APDA can support privacy-preserving data aggregation in a MSS in a mobile topology. In addition, our extended APDA schemes support maximal value finding without scarifying the mobile users' privacy. However, MAI did not consider any family of data aggregation to be performed on the aggregated data.

Secondly, the scheme proposed in [57] has been designed to allow any malicious aggregator in WSNs to be detected by its child nodes. By using such a scheme, each node has to re-calculate the aggregation result using its own data with those received from its sibling. Then, each node has to compare with the aggregation result received from the grandparent nodes (sent by its parent node). The comparison of these two results is used to determine the behaviour of the node's parent. In contrast, APDA requires only the aggregators, which sense as well, aggregate the sensed data while others just need to sense and forward the data to the aggregators. Such a setting could reduce the energy consumption of leaf nodes for computing and transmitting data so as to prolong their battery lifetime. Furthermore, APDA can surely detect any misbehaved nodes without

compromising the privacy of other well-behaved nodes during the data aggregation and misbehaviour tracking processes.

For a direct comparison, we summarise all the above requirements that can be supported by the existing schemes as listed in Table 2. Based on this comparison, we can conclude that the work in [23] is the only one that almost fulfils the selected requirements except the accountability. Therefore, we select the work in [23] and further investigate on its efficiency. We conduct several experiments and compare the results with our proposed scheme to determine the best scheme to implement in MSSs. The simulation and evaluation of our work will be given in the subsequent sub-sections.

Table 2: The Compared Scheme based on the selected Criteria

The Comparing Schemes	The comparing Criteria					
	Applicability	Users' privacy	Data Integrity	Accountability (Misbehaviour Nodes Detection)	Additive Aggregation functions	Non-additive Aggregation Functions
PPDM [25]	MSS	Yes	No	No	No	No
MAI [57]	WSNs	Yes	Yes	Yes	No	No
PPSense [66]	MSS	Yes	No	No	No	No
PDA [67]	MSS	Yes	No	No	Yes	Yes
VPA [23]	MSS	Yes	Yes	No	Yes	Yes
PLAM [70]	MSS	Yes	Yes	No	No	No
RPDA [36]	WSNs	Yes	No	No	Yes	No
VerDP [184]	CC	Yes	Yes	No	No	No
APDA	MSS	Yes	Yes	Yes	Yes	Yes

5.4 PERFORMANCE EVALUATION

This section elaborates the performance evaluation of our proposed scheme by providing the experimental results and analysis. The experimental results are essential to validate our aim and objectives stated in Section 1.6. Prior to providing the experimental results, we will describe application or experimental settings that are required by two distinct software packages. These are the Matlab version 15a and OPNET version 14.5.A. The purpose of the former software is for data computation, while the latter is for simulating data transmission. Finally, our experimental results and their analysis will be presented, which provides meaningful evidence to support the conclusions that we will make at the end of this section.

5.4.1 EXPERIMENTAL SETUPS

Our simulation for data transmission among nodes in a network is set up using the powerful simulation software OPNET version 14.5.A. There are a lot of commercial and freeware network simulators in the market like OPNET, NS, OMNeT++ and SIMULINK, which are popular and widely used. Among them, OPNET offers more user friendliness, flexibility and portability. OPNET stands for Optimised Network Engineering Tools and was created by OPNET Technologies, Inc., which was founded in 1986. OPNET is a powerful network simulation tool set and can create and test large network environments via software.

For our experimental setups, we consider seven scenarios which consist of 103 nodes, 154 nodes, 205 nodes, 256 nodes, 307 nodes, 358 nodes and 409 nodes respectively. They have been modelled to compare the efficiency of the verifiable privacy-preserving data aggregation between versions of our APDA and the chosen scheme VPA [23]. Furthermore, those settings have been implemented to investigate the performance of the whole process using the versions of our APDA. This whole process covers the commitment generation, aggregated data encryption, data recovery, integrity checking and misbehaviour nodes detection.

Furthermore, we investigate the rate of increase in misbehaving nodes with respect to an increasing size of networks. For simulation purposes, each network is divided into the same size of cells. Each cell has 25 nodes consisting of one parent node and 24 children, i.e. each child is connected to its parent within the same cell. To construct a hierarchical structure, each parent node is linked to a node, named a grandparent node, at the upper layer of the structure, and each grandparent is then connected to the root node. Finally, this root node is linked to the AS for aggregated data submission. For example, a network of 100 nodes is divided into 4 cells with 25 nodes for each, in addition to the AS, root node and two grandparent nodes with each connected to two parent nodes. Such a network is illustrated in Figure 15. Moreover, each node has a dedicated destination address to avoid collusion and reduce any delay in data transmission.

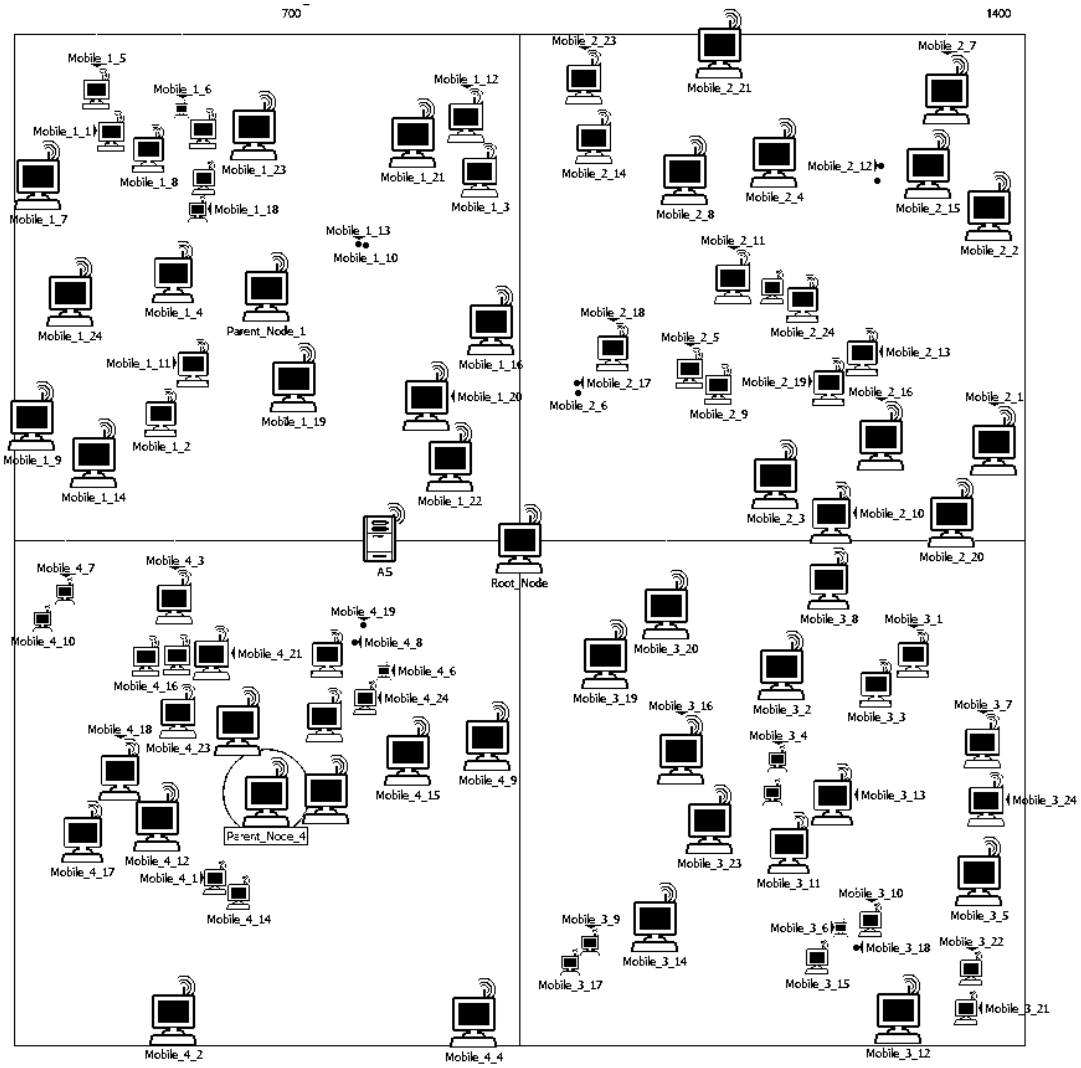


Figure 15: The simulation setting of a network that consists of 103 nodes

Furthermore, Figure 16 shows the OPNET 14.5.A interface for setting up our parameters before running the simulation. As shown in the figure, there are several attributes that need to be inserted before running the simulation. The first two attributes are for node identification. Furthermore, the third attribute is for the start time of the simulation and the packet size that needs to be transmitted by each node in a network. Finally, the last attribute is the setting up of a wireless connection between nodes that includes the wireless LAN MAC address, BSS identifier, data rate etc. For the simulations using OPNET 14.5.A, we run each of the simulations for 50 runs to achieve accuracy and consistency [23]. An input of the simulations is taken from the results (the execution time) generated by Matlab. Those results are converted into bytes before simulating the data aggregation using OPNET.

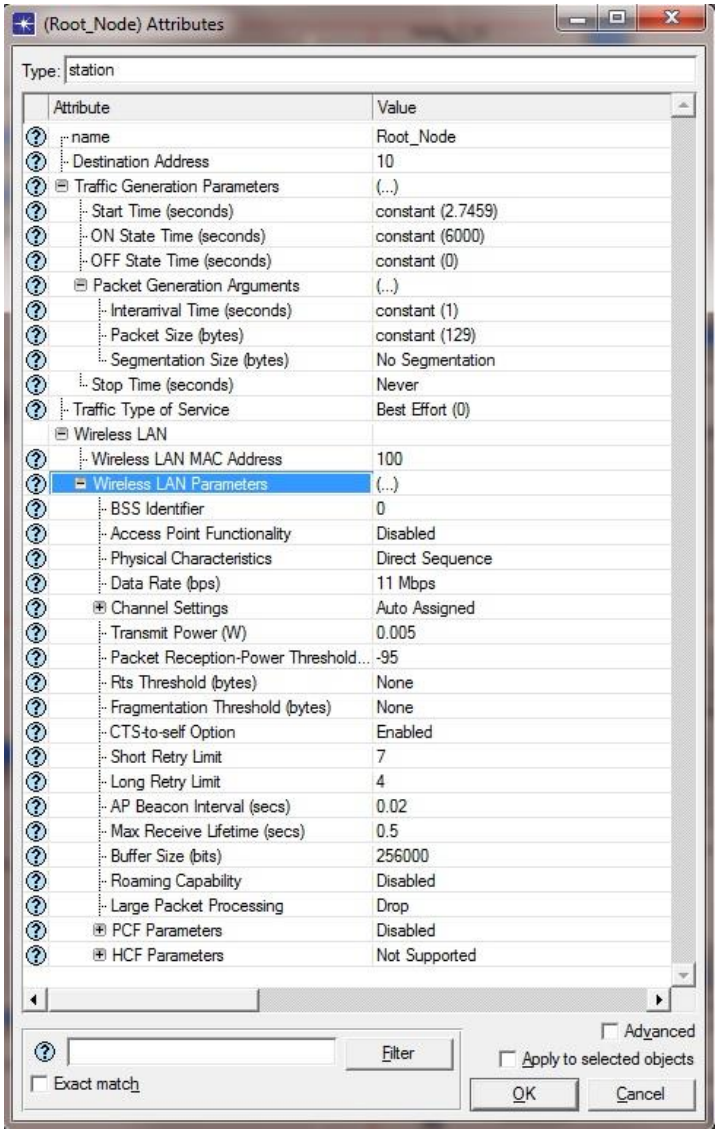


Figure 16: Simulation attributes settings

For our experimental purposes, we have designed three different behaviour scenarios to investigate the efficiency of the proposed scheme on different sizes of networks. Those scenarios are described as below.

Scenario 1: There is at least one misbehaved node distributed randomly in a single cell of the networks.

Scenario 2: There is at least one misbehaved node distributed randomly in 50% of cells of the networks.

Scenario 3: There is at least one misbehaved node distributed randomly in 25% of cells rounded up to the nearest number of cells.

5.4.2 PARAMETER SETTINGS

We have implemented our algorithms using Matlab and OPNET 14.5.A and evaluated their execution time. Those algorithms consist of data randomisation, commitment generation, key generation, aggregated data encryption and data recovery. Furthermore, we have also implemented data verification and misbehaved node detection algorithms. The computations have been performed on a machine with a 3.60GHz Intel (R) Core (TM) i7 – 4790 CPU. Furthermore, the data transmission has been simulated on mobile devices with the data rate of 11Mbps and the buffer size of 256,000 bits. Table 3 provides the parameter settings of the implementations for both the two versions of APDA and VPA^{+}. Furthermore, Table 4 provides other parameter settings to investigate the performance of $APDA^{W}$, $APDA^{S}$ and $APDA^{H}$ for the whole data aggregation and misbehaved node tracking process. Such parameters are required to produce a signature for each aggregated data encryption so that each data can be verified and the nodes' behaviour can be determined based on the aggregated information received by the AS.

Table 3: Parameter Settings for $APDA^{W}$, $APDA^{S}$ and VPA^{+} for the process up to integrity checking

No.	Parameter	Description	Bit length	Who should know
1.	p	Public parameter	1024	All
2.	g	Public parameter	160	All
3.	q	Public parameter	186	All
4.	d_i	Plaintext data item	10	u_i only
5.	e_i	Randomised plaintext	160	u_i only
6.	b_i	Random integer	140	u_i only
7.	c_i	Commitment/encryption key	1024	u_i and AS only
8.	n	Number of nodes	9	All

Table 4: Parameter Settings for APDA^{W}, APDA^{S} and APDA^{H} for the whole process including misbehaved node detection

No.	Parameter	Description	Bit length	Who should know
1.	p	Public parameter	1024	All
2.	g	Public parameter	160	All
3.	q	Public parameter	587	All
4.	d_i	Plaintext data item	10	u_i only
5.	e_i	Randomised plaintext	160	u_i only
6.	b_i	Random integer	140	u_i only
7.	c_i	Commitment/encryption key	1024	u_i and AS only
8.	n	Number of nodes	9	All
9.	η	A secret key	120	AS only
10.	r	A secret key	289	AS only
11.	$h(\cdot)$	Hash function	160	All
12.	w	A random number	120	AS only
13.	q	Public parameter	578	All
14.	k_i	A private key	578	u_i and AS only
15.	μ_i	Public key	1024	Parent nodes

5.4.3 RESULTS AND DISCUSSIONS

In this section, we provide our experimental results to evaluate the performance of the different versions of our APDA together with the performance of the VPA^{+} scheme. Several experiments have been conducted to investigate the performance of each scheme. We divide our experiment into several parts. In the first part, we investigate the efficiency of implementing APDA^{W}, APDA^{S} and VPA^{+} up to the integrity checking.

In the second part of our experiment, we compare the total execution time for the whole data aggregation and misbehaved node detection process using only both APDA^{W} and APDA^{S} as VPA^{+} does not offer the capability of misbehaved node detection. In this experiment, we implement and evaluate the BU only approach for tracking misbahaved nodes within various sizes of networks.

To accelerate the misbehaved node detection within such networks, we apply the mixed TD and BU approach discussed in Sections 4.4 and 4.5. The result of such an improvement is illustrated in the third part of our experiment.

In the next part of our experiment, we investigate the probability of getting a correct maximal value based on the number of strings.

Finally, we compare the performance of our extended version of APDA^{M} to determine the maximum value among those contributed by the n nodes.

The results obtained from the above experiments using Matlab 15a and OPNET 14.5.A are shown in the figures below. Detailed discussions on the results are also provided for clarification and analysis purposes.

In addition to the aforementioned experiments, we have also tested some of our results using a statistical test, which is called a *t*-Test. We have executed this test by using Microsoft Excel 2013. The main purpose of this test is to determine the significant difference of two samples tested. This significant difference is important to prove that both samples are likely or unlikely based on the significant level, which we have set at 0.05. This value will be compared with the result of the *P*-value of the test. We say if the *P*-value is greater than the significant level value, then both samples are likely. In contrast, if the *P*-value is less or equal to the significant level value, then both samples are unlikely. Furthermore, if both samples are unlikely, the mean of both samples can be used to determine the best one.

5.4.3.1 An efficiency comparison for the process only up to the integrity checking.

In this experiment, the two versions of APDA, i.e. $APDA^{W}$ and $APDA^{S}$, have been compared with the chosen scheme VPA^{+} by executing the process only up to the integrity checking. This process covers the commitment generation, aggregated data encryption and integrity checking. Prior to comparing the efficiency performance of these schemes, we have tested $APDA^{W}$ with VPA^{+} to determine the significant difference and the best method between both schemes by using *t*-Test. For experimental purposes, samples of the *t*-Test are listed in Table 5 and the results of the test are illustrated in Table 6.

Table 5: Total execution time for the whole process up to integrity checking in one round data aggregation using $APDA^{W}$ and VPA^{+}

Total Nodes	Delay in seconds	
	$APDA^{W}$	VPA^{+}
103	2.84	5.50
154	3.00	5.66
205	3.16	5.83
256	3.33	5.99
307	3.49	6.16
358	3.65	6.32
409	3.82	6.49

Table 6: Results of *t*-Test for 2 samples of $APDA^W$ and VPA^+ in Table 5

Statistical Functions	*$APDA^W$*	*VPA^+*
Mean	3.327143	5.992857143
Variance	0.124324	0.127057143
Observations	7	7
Pooled Variance	0.12569	
Hypothesized Mean Difference	0	
df	12	
t Stat	-14.0668	
P(T<=t) one-tail	4.04E-09	
t Critical one-tail	1.782288	
P(T<=t) two-tail	8.08E-09	
t Critical two-tail	2.178813	

Table 6 shows the result of *t*-Test for the two samples listed in Table 5. Based on this test, there are two important results that provide a significant conclusion on both samples of the data, which are the *P*-value (our significant level at 0.05) and the mean of the two samples. From Table 6, the *P*-value of this test is $4 \times 10^{-9} < 0.05$. This means that both samples are unlikely or have significant difference. In order to determine the best method, we compare the means of both samples. Based on the result, we can conclude that $APDA^W$ is better than VPA^+ as its mean is lower than the mean of VPA^+. In Table 7, we compare the efficiency performance of both versions of our APDA and the VPA^+, and Figure 17 illustrates the graph of their performance.

Table 7: Total execution time for the whole process up to the integrity checking in one round data aggregation using $APDA^W$, $APDA^S$ and VPA^+

Total Nodes	Delay in seconds		
	$APDA^W$	$APDA^S$	VPA^+
103	2.84	5.56	5.50
154	3.00	5.74	5.66
205	3.16	5.92	5.83
256	3.33	6.10	5.99
307	3.49	6.28	6.16
358	3.65	6.45	6.32
409	3.82	6.63	6.49

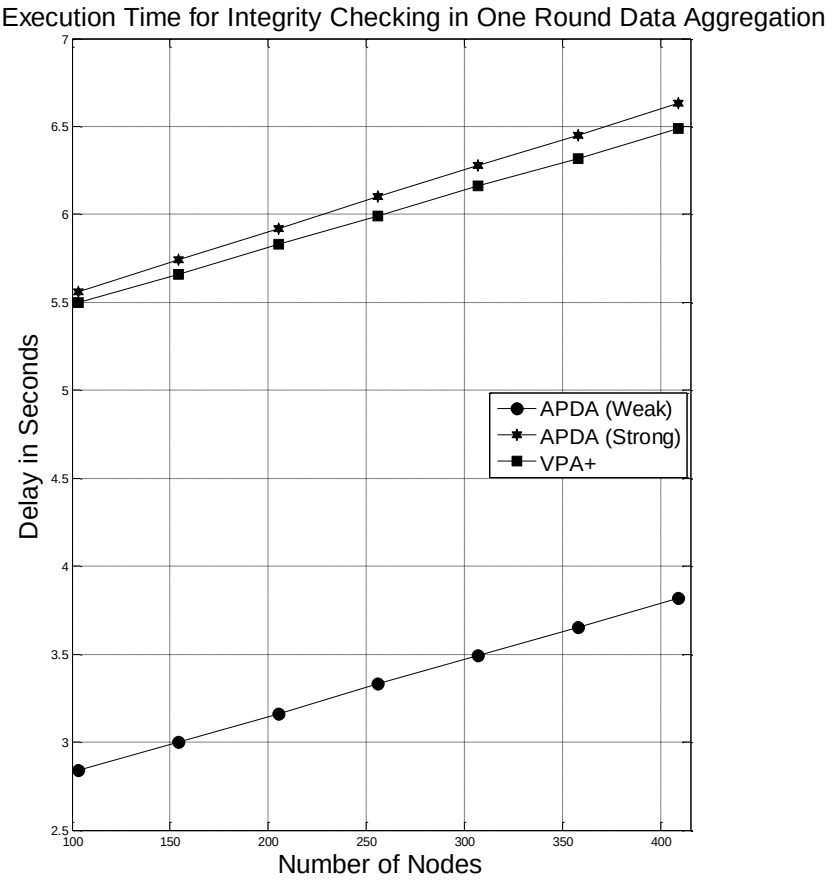


Figure 17: Total execution times for the process only up to integrity checking using $APDA^W$, $APDA^S$ and VPA^+.

Figure 17 illustrates the total execution times for the process only up to integrity checking in one round of data aggregation shown in Table 7. We can see that the execution time of $APDA^W$ is much faster than those of $APDA^S$ and VPA^+. The reason is that $APDA^W$ does not involve hash calculations in its privacy-preserving computation while $APDA^S$ and VPA^+ do. In addition, as we can see from the graph, the delays of implementing $APDA^S$ are higher than VPA^+. The reason is that $APDA^S$ requires more parameters to be computed in generating ciphertext data by each node.

5.4.3.2 An efficiency comparison of the whole misbehaviour nodes detection process via the BU only approach.

Table 8 compares the required total execution times of APDA for the whole process up to the misbehaviour nodes detection in an aggregation process. That is, the process covers the commitment generation, aggregated data encryption, data recovery, integrity checking and misbehaviour nodes detection. Both $APDA^W$ and $APDA^S$ have been implemented for

scenario 1 described in Sub-section 5.4.1 to compare their efficiencies. Furthermore, we have implemented the BU only approach to detect the misbehaved node in this experiment. As we can see in Table 8, the delays introduced by $APDA^S$ are higher than those of $APDA^W$. Furthermore, when the number of nodes increases, the delay difference also increases. The reason for such differences will be explained in the discussion of Figure 18.

Table 8: Total execution times for the whole process including misbehaved node detection using two versions of APDA in various sizes of networks

Total Nodes	Delay in Minutes		Delay Difference
	One Round Data Aggregation		
	$APDA^W$	$APDA^S$	
103	0.97	3.93	1.09
154	1.32	4.49	1.49
205	1.66	5.04	1.88
256	2.01	5.60	2.28
307	2.36	6.16	2.66
358	2.71	6.71	3.06
409	3.06	7.27	3.44

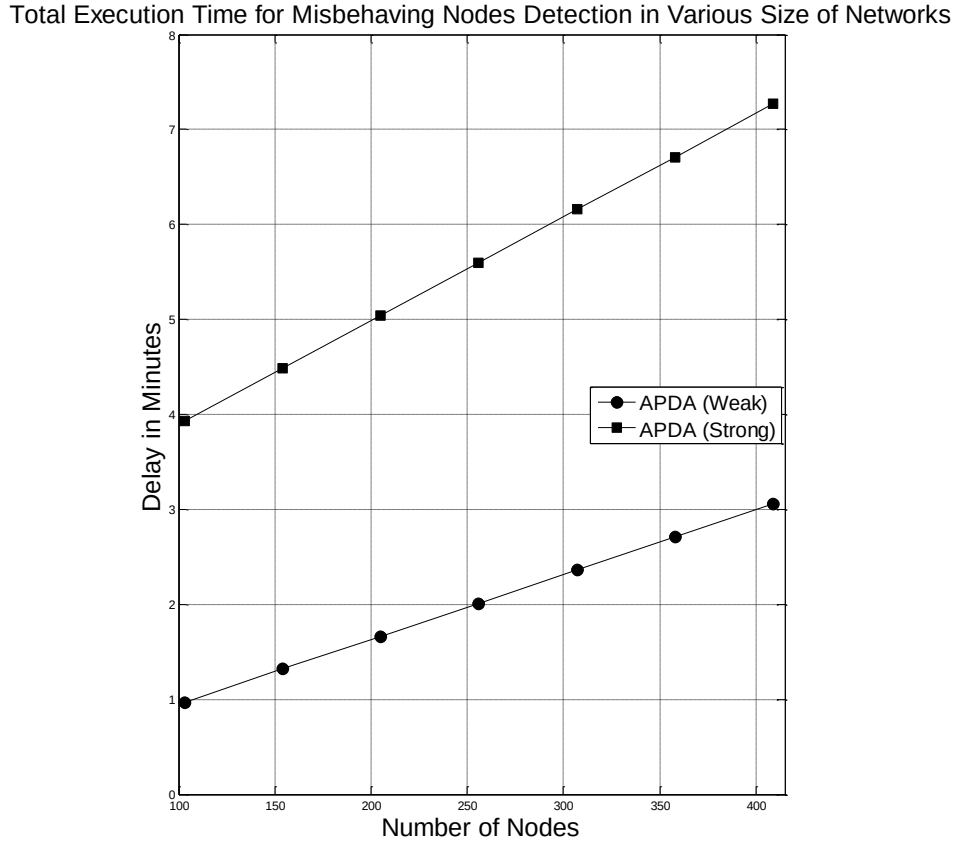


Figure 18: Total execution times for the whole process up to misbehaved node detection using the two versions of APDA in various sizes of networks.

Figure 18 displays the total execution times of the two APDA versions for the process up to misbehaved node detection. Such misbehaved nodes are randomly distributed in the networks as described in scenario 1. The graph shows that the execution time of $APDA^W$ is faster than that of $APDA^S$ within all the sizes of networks. The graph also shows that the percentage differences between the two lines of all the sizes of networks are greater than 50%. Furthermore, the delay difference between both lines increases as the network size increases. The main reason is that $APDA^W$ requires less computation on time-consuming exponentiations while executing the whole process up to misbehaved node detection. In contrast, $APDA^S$ has consumed more time as it requires to compute more exponentiations for data generation in addition to data transmission. Furthermore, we embed hash computation in each ciphertext generation using $APDA^S$ to allow the verification of the received aggregated data by each parent node, which leads to further delays.

5.4.3.3 An efficiency comparison of APDA via the combined approach for misbehaved node detection.

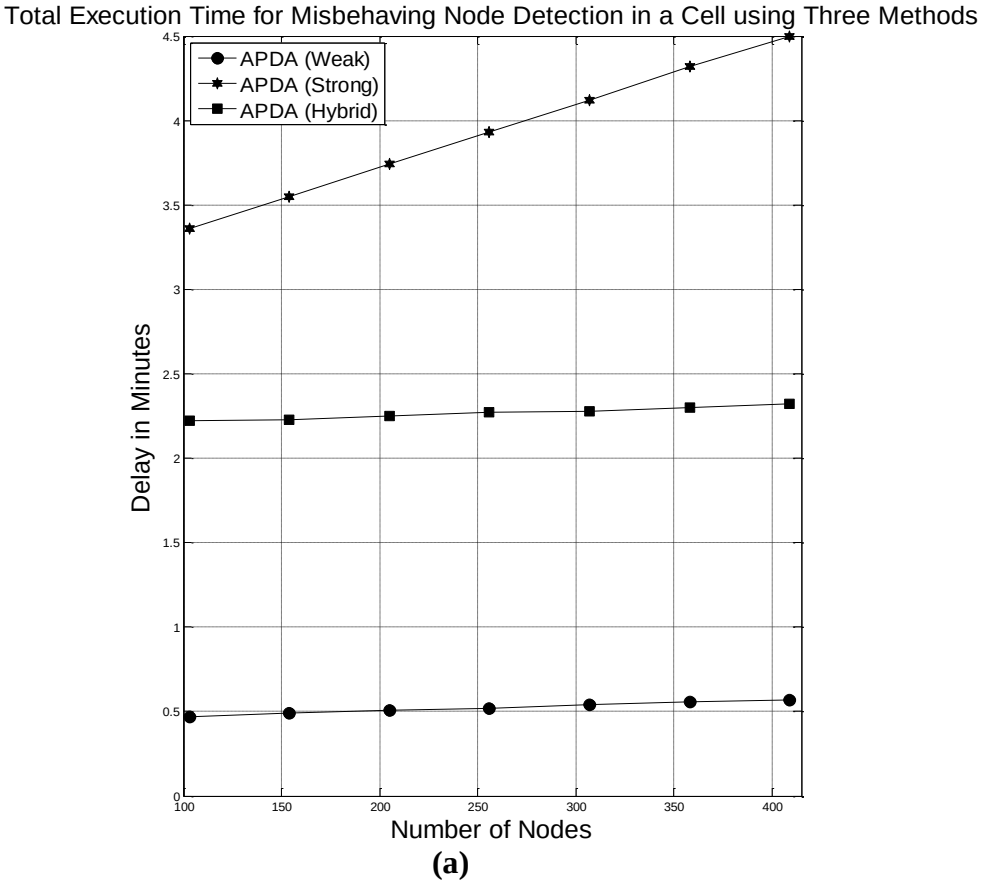
In this case study, the combination of the TD and BU approaches discussed in Sections 4.4 and 4.5 has been implemented to execute the whole process up to misbehaved node detection during an aggregation process. The main reason of this combination is to accelerate the detection of misbehaved nodes once the integrity check has failed. We have compared the delays of the whole process among the three versions of APDA in the three scenarios involving misbehaved nodes randomly distributed in one cell as well as 50% and 25% of the cells. The experiment results are given in Figure 19.

In the 25% (or 50%) scenario, the total numbers of suspicious cells are different from one network to another. For example, 25% of 6 and 8 cells are just 2 cells. Thus, in such cases, the time delays for determining misbehaved nodes are almost similar due to the same number of suspicious cells. In contrast, for a network of 10 or 12 cells, 25% of the cells provide more delays compared to a network of 8 cells due to the number of suspicious cells increased to 3. In addition, as each cell consists of 25 nodes, checking a cell with at least one misbehaved node requires all the nodes in the cell to be examined, which increases the delay.

Table 9 shows the total execution times for the whole process including the detection of various numbers of misbehaved nodes by the three versions of APDA.

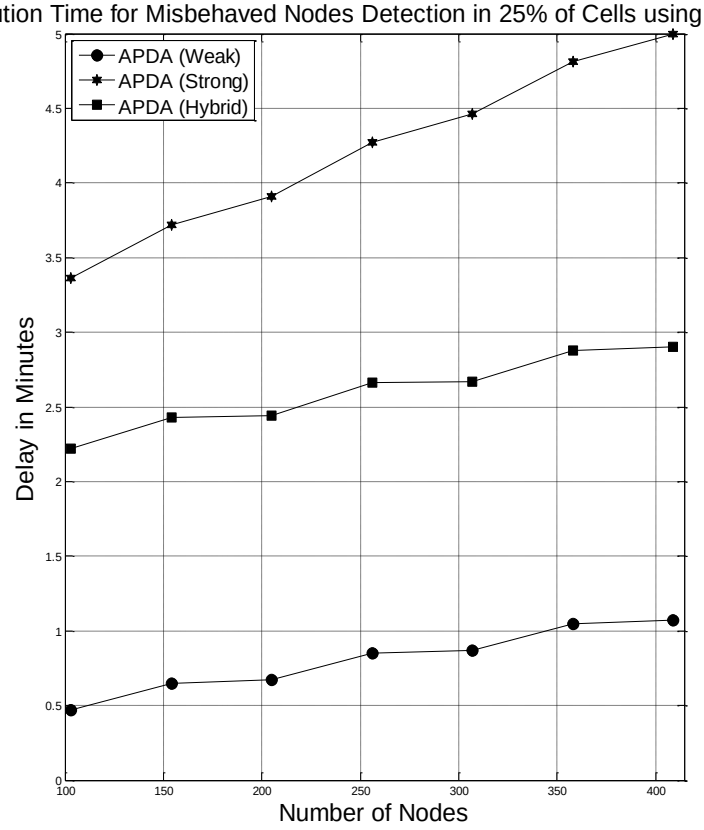
Table 9: Total execution times for the whole process using the combined TD and BU approach.

Node Total	Delay in Minutes								
	$APDA^W$			$APDA^S$			$APDA^H$		
	One Round of Data Aggregation						Two Rounds of Data Aggregation		
	One	50%	25%	One	50%	25%	One	50%	25%
103	0.47	0.64	0.47	3.36	3.53	3.36	2.22	2.41	2.22
154	0.49	0.82	0.65	3.55	3.89	3.72	2.23	2.62	2.43
205	0.51	1.00	0.67	3.74	4.24	3.91	2.25	2.84	2.44
256	0.52	1.18	0.85	3.93	4.60	4.27	2.27	3.05	2.66
307	0.54	1.36	0.87	4.12	4.95	4.46	2.28	3.26	2.67
358	0.56	1.55	1.05	4.32	5.31	4.81	2.30	3.48	2.88
409	0.57	1.73	1.07	4.50	5.66	5.00	2.32	3.69	2.90



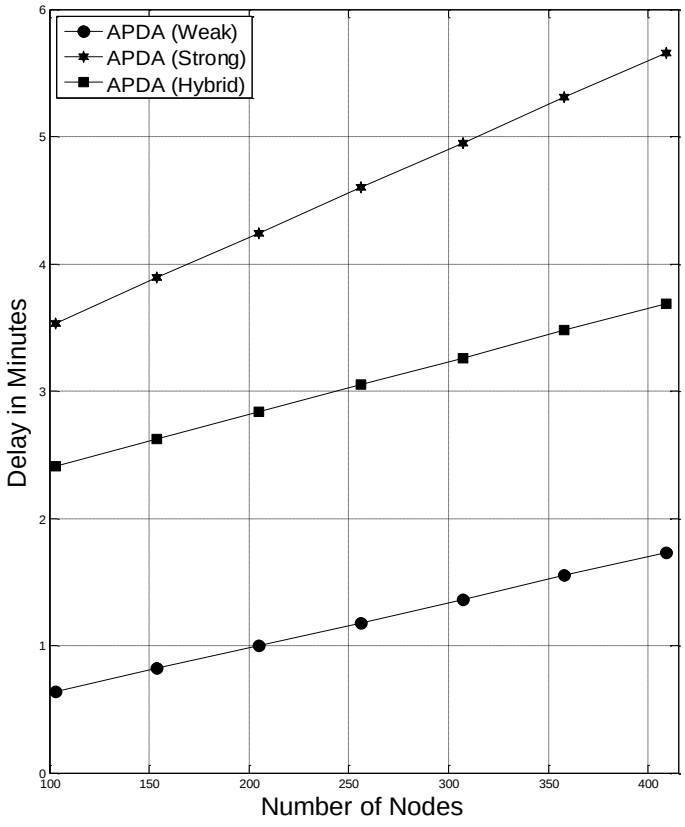
(a)

Total Execution Time for Misbehaved Nodes Detection in 25% of Cells using Three Methods



(b)

Total Execution Time for Misbehaved Node Detection in 50% of Cells using Three Methods



(c)

Figure 19: Total execution time for the whole process including misbehaved node detection in one cell, 50% and 25% of cells using $APDA^W$, $APDA^S$ and $APDA^H$.

Based on Figure 19, $APDA^W$ is the most efficient method compared to $APDA^S$ and $APDA^H$. This efficient performance is due to $APDA^W$ requiring the lowest number of time-consuming exponentiations during the misbehaved node detection. In contrast, $APDA^S$ is the least efficient method as it involves the highest number of exponentiations for data aggregation and verification as well as misbehaved node detection. By having more exponentiations in $APDA^S$ for data aggregation and data verification, the tracking of misbehaved nodes by the AS can be executed with certainty. Nevertheless, by using $APDA^W$, the AS can only detect with certainty those misbehaved nodes which are leaf nodes, as discussed in Sub-section 4.4.3. On the other hand, $APDA^H$ provides a more balanced performance in terms of the efficiency and assurance of identifying misbehaved nodes. This result has been achieved because only the high-level parent nodes initially run $APDA^W$ to detect any suspicious cell, and once detected, only the nodes in the suspicious cell(s) need to run $APDA^S$ while the rest still run $APDA^W$ if needed. As a result, $APDA^H$ remains relatively stable in all cases.

Furthermore, the results in Figure 19 show that the gap between the middle line for $APDA^H$ and the top line for $APDA^S$ is widening as the number of network nodes increases. The reason is that $APDA^S$ requires every parent node to verify the received data from its children and this verification process involves heavy computation on exponentiation. As a result, this verification increases the delay rapidly. In contrast, $APDA^H$ requires only the parent node within the suspicious cell, which is determined by $APDA^W$, to verify the received data. Conversely, the gap between the middle line for $APDA^H$ and the bottom line for $APDA^W$ remains constant as the number of network nodes rises. The reason is that, by running $APDA^H$ for every scenario, the number of nodes in the suspicious cells, which are determined by $APDA^W$ to run $APDA^S$, increases constantly as the network size increases.

5.4.3.4 A probability comparison for getting a correct maximum using different numbers of strings.

In this study, we have investigated the number of strings required for Max^B to determine the maximum value with a high probability. We have implemented two methods, a Matlab

based simulation of Max^B and the calculation of Equation 4.26, for a comparison. We have executed 1,000 rounds of the simulation for each number of strings up to 15. The experimental results together with the calculated ones are shown in Table 10 and Figure 20.

In addition, we have tested the results of both methods by using *t*-Test to determine whether both of them are significantly different, and the test results are provided in Table 11. From Table 11, the *P*-value of both samples is 0.5, which is greater than the significant level at 0.05. Thus, we conclude that both samples are likely and there is no significant difference between the means.

Table 10: Probabilities from our experimental result and Equation 4.26 for Max^B to get a correct maximum using different numbers of strings

Number of strings	The probability	
	Experimental result	Formula
0	0.000	0.000
1	0.508	0.500
2	0.749	0.750
3	0.876	0.875
4	0.930	0.938
5	0.965	0.969
6	0.987	0.984
7	0.990	0.992
8	0.998	0.996
9	0.998	0.998
10	0.999	0.999
11	0.999	0.999
12	0.999	0.999
13	0.999	0.999
14	0.999	0.999
15	0.999	0.999

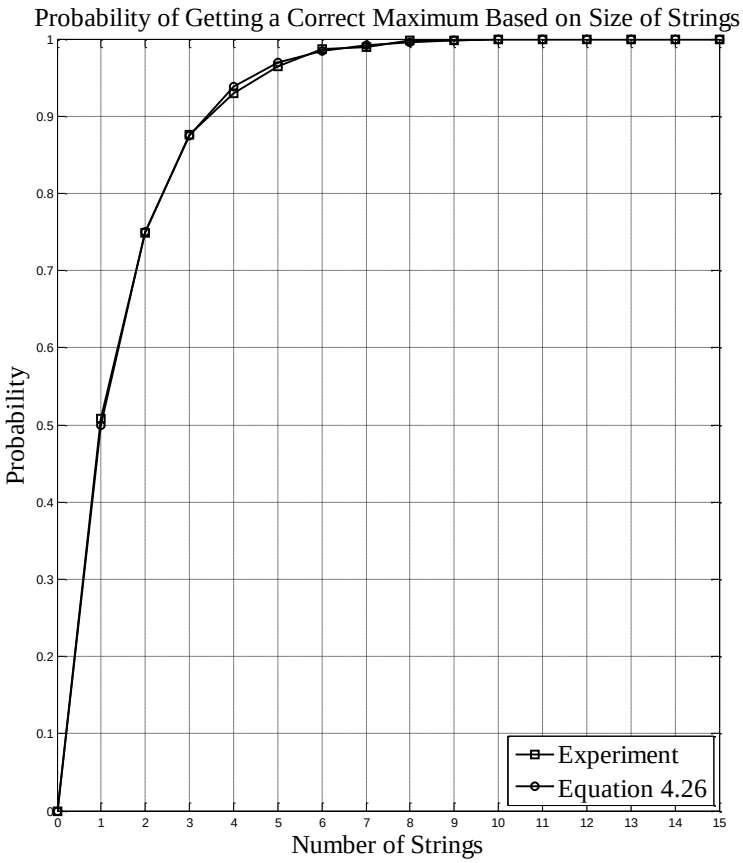


Figure 20: Probabilities for Max^B to get a correct maximum using different numbers of strings.

From Figure 20, we can see that the results from both methods are very close and the probability increases as the number of strings increases. Furthermore, in order to have a correct result of finding the maximum value, the probability must be close to one. From the graph, we can conclude that the total number of strings required for getting the correct result with a high probability of 0.999 is 10. If we further increase the number of strings, the probability will not significantly change as evidenced in Table 10.

Table 11: *t*-Test result for 2 samples in Table 10

Statistical Functions	*Experimental result*	*Formula*
Mean	0.8746875	0.87475
Variance	0.071706496	0.072151667
Observations	16	16
Pooled Variance	0.071929081	
Hypothesized Mean Difference	0	
df	30	
t Stat	-0.000659133	
P(T<=t) one-tail	0.499739226	
t Critical one-tail	1.697260887	
P(T<=t) two-tail	0.999478452	
t Critical two-tail	2.042272456	

5.4.3.5 An efficiency comparison between our solutions and VPA⊕ for maximal value finding.

In this case study, we compare the efficiency between the extended versions of our APDA scheme (i.e. Max^C, Max^B and Max^H) and the chosen scheme VPA⊕ in [23] for finding a maximum value among those contributed by the n nodes. The aim of comparing those methods is to determine the most efficient method for maximum value finding. We have set up the network with various sizes for the performance comparison. Furthermore, we have set up the parameter settings as illustrated in Table 12 to execute the different versions of our Max and VPA⊕. The efficiency on each scheme has been measured based on the total execution time taken in seconds for the completion of finding a correct maximum. The comparison results are shown in Table 13 and Figure 21.

Table 12: The Parameter Setting for implementing Max^C, Max^B and VPA⊕

No	Parameter	Description	Max^C	Max^B	VPA⊕
1.	τ	Bit length of d_i	10	10	10
2.	θ	Index of base 2 in γ	4	7	1
3.	ϵ	Number of decimal digits	2	2	2
4.	γ	Number of sub-ranges	16	128	2
5.	$\acute{n}$	Number of rounds	3	2	10
6.	τ'	Bit length of $\acute{d}_i$	112	128	14
7.	ϑ	Number of strings	1	10	1
8.	$\log_2 n$	Bit length of n	9	9	9

Table 13: Total execution times for the maximal value finding using Max^C and $VPA^{\oplus}$ for *t*-Test evaluation

Node Total	Delay in seconds	
	Max^C	$VPA^{\oplus}$
103	9.26	35.92
154	9.37	36.31
205	9.54	36.89
256	9.82	37.80
307	10.12	38.83
358	10.46	39.96
409	10.84	41.15

Table 14: *t*-Test result for 2 samples in Table 13

Statistical Functions	*Max^C*	*$VPA^{\oplus}$*
Mean	9.82	37.80125
Variance	0.369057143	4.084869643
Observations	8	8
Pooled Variance	2.226963393	
Hypothesized Mean Difference	0	
df	14	
t Stat	-37.50080309	
P(T<=t) one-tail	9.51026E-16	
t Critical one-tail	1.761310136	
P(T<=t) two-tail	1.90205E-15	
t Critical two-tail	2.144786688	

Table 14 illustrates the *t*-Test result of the two samples in Table 13. This test compares the significant difference between the two samples, which are generated via Max^C and $VPA^{\oplus}$. From the figures, the *P*-value of both samples is too small (i.e. 9.5×10^{-16}), which is lower than the significant level value at 0.05. This indicates that both samples are very unlikely and there is a significant difference between both samples. Furthermore, by comparing the means of both methods, we conclude that Max^C is better as its mean is lower than the mean of $VPA^{\oplus}$.

In Table 15, the three versions of our Max and $VPA^{\oplus}$ have been compared to investigate their efficiency performance in order to determine the maximal value based on a given data set.

Table 15: Total execution times for the maximal value finding using Max^C, Max^B, Max^H and VPA⊕

Node Total	Delay in seconds			
	Max^C	Max^B	Max^H	VPA⊕
103	9.26	1.48	4.56	35.92
154	9.37	1.56	4.69	36.31
205	9.54	1.66	4.84	36.89
256	9.82	1.71	4.98	37.80
307	10.12	1.84	5.22	38.83
358	10.46	1.94	5.42	39.96
409	10.84	2.03	5.51	41.15

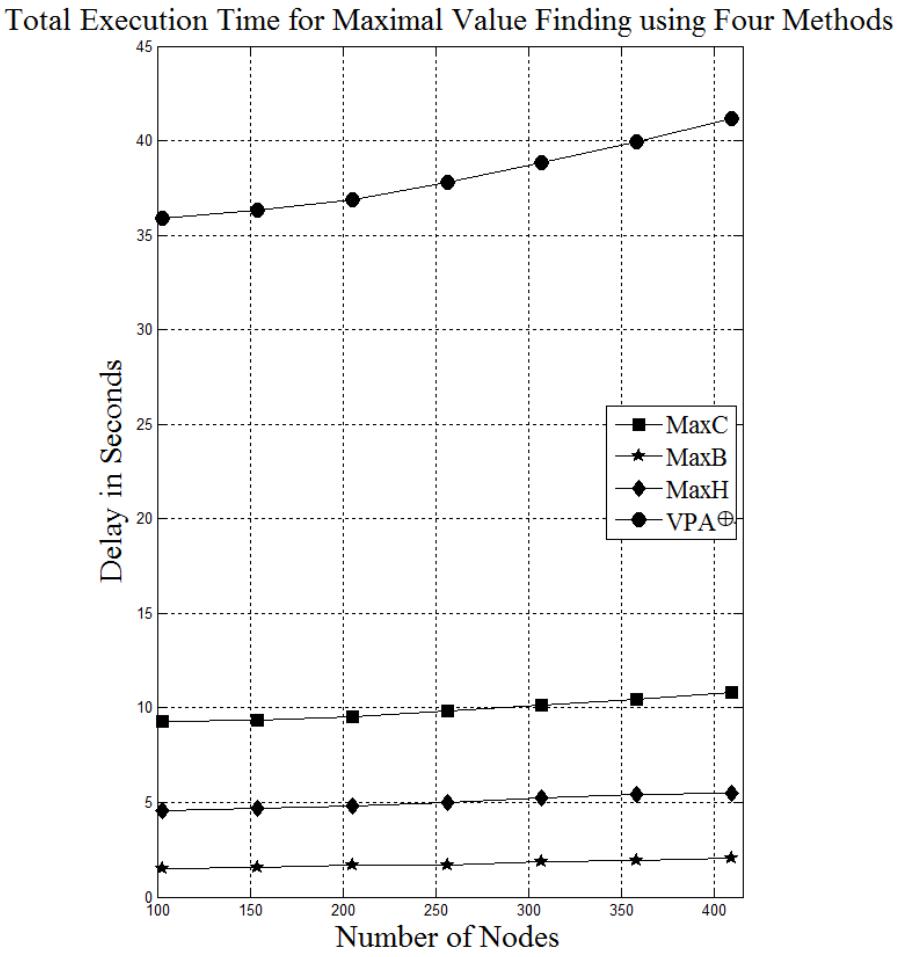


Figure 21: Total execution times for maximal value finding.

Figure 21 shows that Max^C is more efficient than VPA⊕. This is because Max^C utilises more sub-ranges in every round of maximal value finding so as to reduce the total number of rounds required to find the maximum. In contrast, VPA⊕ uses only 2 sub-ranges in each round, resulting in much more rounds of aggregation for finding the maximum and hence much lower efficiency. This highlights that reducing the number of aggregation

rounds for maximal value finding can lead to significantly lower execution times due to the reduced amounts of data computation and transmission. This is further evidenced from the results of Max^B, showing its best efficiency in the figure, as it increases even more sub-ranges in every round of aggregation and hence reduces the total number of rounds needed.

In addition, Figure 21 also shows the efficiency result of Max^H, which is between those of Max^C and Max^B. The reason is that Max^H implements Max^B in an efficient fashion to narrow down the range that contains the maximum, and then runs Max^C with a better accuracy in determining the maximum within the narrowed range.

5.5 SUMMARY

In this chapter, we have detailed the security analysis and proved that our proposed schemes are secured against the threat model described in Chapter 4. We have also shown that our schemes satisfied all the requirements stated in Chapter 4. Furthermore, the experimental results of our proposed schemes have demonstrated that our work can provide better efficiency compared to the best existing solutions VPA^+ and $VPA^{\oplus}$. Our APDA has shown that it is able to detect with certainty the misbehaved nodes in the aggregation process. Finally, our schemes' performance and security analysis have shown that they are suitable to be implemented in MSSs. In the next chapter, our LHE scheme will be introduced to allow resource-limited devices like smartphones to leverage rich-mobile applications provided by cloud providers in an efficient and secure manner.

CHAPTER 6

A NEW LIGHTWEIGHT HOMOMORPHIC ENCRYPTION SCHEME

6.1 INTRODUCTION

Even though MCC and MSS use a similar concept of leveraging cloud computing for alleviating all the burdens of heavy computations and complex tasks to a cloud server, both of them differ significantly as stated in Chapter 2. MCC allows mobile users to leverage rich-mobile applications provided by the clouds, mainly applications related to complex computation. In such a case, APDA proposed for MSS needs to be enhanced so that it can support fully homomorphic properties, which allows the scheme to compute arbitrary functions on ciphertext data. Thus, in this chapter, we propose a new scheme to suit the implementation of MCC. Prior to proposing the scheme, we will briefly describe the fundamental concept together with the implementation issues of MCC.

With the emerging technology of cloud computing, rich mobile applications have been offered and delivered through the Internet. Cloud computing offers tremendous advantages by allowing users to use its infrastructures, platforms and software. Such services are provided by Cloud Service Providers (CSPs) like Google, Amazon, and Salesforce at low costs. With the increasing number of mobile applications and the

support of cloud computing for a variety of services for mobile users, Mobile Cloud Computing (MCC) is introduced as an integration of cloud computing into the mobile environment. As a result, MCC brings new types of services and facilities for mobile users to take full advantages of cloud computing [4], [8], [9], [14], [27], [53], [129].

To leverage such a technology and service, mobile users need to outsource their data to the CSPs for storing and processing purposes. However, outsourcing such data, which is often private or sensitive, into the clouds with no physical and limited digital control by the users raises serious concerns on its security [60]. Furthermore, inappropriately handling such data could result in a disaster to the data owners due to data misuse, data leakage, or data theft by other parties that abuse the same services. Moreover, the CSPs do not offer proper security guarantees to the data owners [185]. Due to the scale, dynamicity, openness and resource-sharing nature of cloud computing, addressing security issues in such environments is a very challenging problem [186].

To ensure that the security and integrity of the data are preserved in clouds, encryption techniques should be implemented. Primitive encryption schemes such as RSA algorithm and Elliptic Curve Cryptosystem (ECC) are good for storing purposes [33] but they prevent the encrypted data from being processed by cloud-based applications [103]. Thus, a scheme that allows data to be processed in its ciphertext form, like a Fully Homomorphic Encryption (FHE) scheme, is extremely desirable for securing cloud-based data processing. Although a number of existing FHE schemes have been proposed and improved upon, none of them is practically efficient enough to be deployed in an open cloud environment, as efficiency is still a big challenge for their implementation. For instance, existing FHE schemes based on Lattices are suffering from efficiency issues due to the amount of noise introduced during the data processing stage [159]. Additionally, a scheme based on a bilinear map allows arbitrary additions but only one multiplication on encrypted data [146]. Furthermore, existing FHE schemes are computationally expensive, which draw a lot of computing resources to implement. This computational process inhibits mobile devices from computing in an efficient manner.

To address the above problem, in this chapter, we propose a new Lightweight Homomorphic Encryption (LHE) scheme that is constructed based on Gentry's scheme. We follow the same application settings as in [74]. The main difference between the two

schemes is that our choice of plaintext for encryption is an integer form, whereas Gentry's scheme is in the form of bits. This novel choice leads to our scheme being more efficient as the encryption on an integer is faster than encryption on every single bit of the integer [46]. Moreover, encryption over the integer increases input and output message spaces so as to consume less storage space and require less bandwidth for data transmission.

Based on the above choice, we have designed a symmetric encryption scheme as in [35] and [76] with better efficiency. This is because our scheme's input and output are in a single integer form, whereas both schemes in [35] and [76] represent their input and output in a matrix form. This argument is supported by our experimental results that show the total time to execute data summation and multiplication operations by using the proposed scheme is much less than the schemes in [35] and [76]. In fact, our scheme supports arbitrary functions on ciphertext data as long as the desired result satisfies some conditions to be detailed in Sub-section 6.4.1. To implement the new scheme, we have provided a process to securely communicate and handle the data in its ciphertext form. This process ensures that the privacy of the outsourced and processed data in the third party environment is preserved. Also the performance of our new LHE is thoroughly evaluated with detailed simulations, which demonstrates much better efficiency than related work.

Prior to detailing the threat model in the next section, Figure 22 shows the MCC setting used to implement our proposed scheme. This setting consists of mobile devices or nodes as data contributors, a data client as a data user, and a server managed by a CSP for data storage and processing, where each node has a dedicated destination address to avoid collusion and reduce any delay in data transmission. Here, all the mobile devices are connected wirelessly to the CSP, which exploit public communication links through either a high-speed mobile network technology 3G/4G or Wi-Fi access. Furthermore, we assume that every two connected nodes have already authenticated each other and established a secure communication channel between them if necessary.

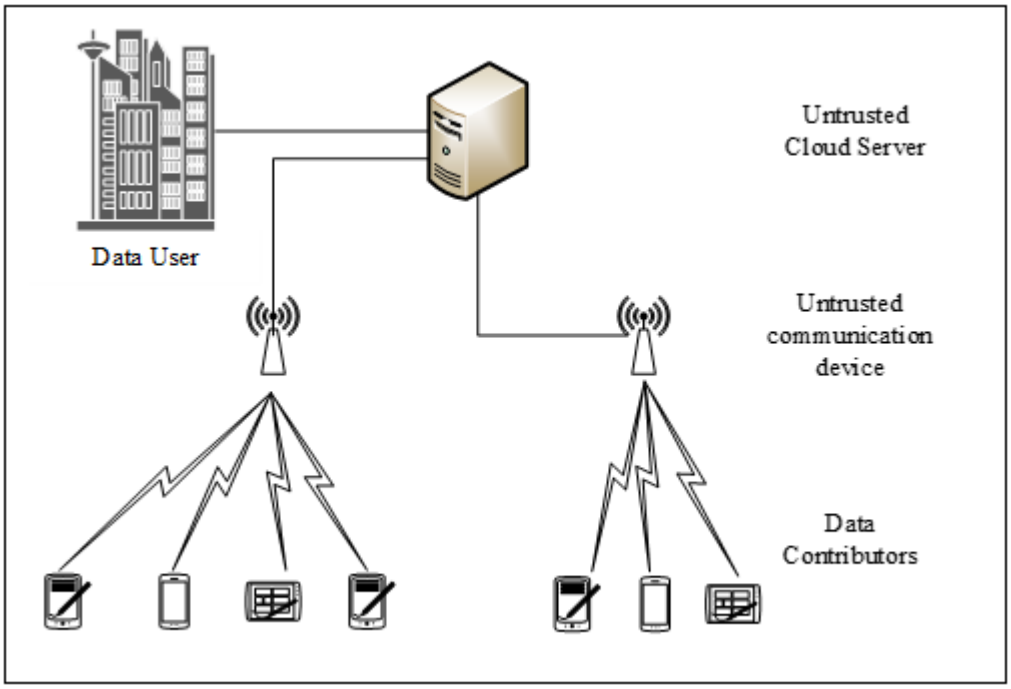


Figure 22: A MCC setting

6.2 THREAT MODEL

In our proposed MCC process, all parties in Figure 22 (i.e. the Data User, the Cloud Server and Data Contributors/mobile devices, which are denoted as DU, CS and DCs, respectively) communicate with one another to collect and process required information via a data offloading technique. This technique allows the collected data to be processed arbitrarily. By using this technique, DU sends a request to DCs via open wireless links to collect information from them, including personal data (e.g. health, monthly salary or home address information) or sensed data about the surrounding area (e.g. crowd in an event, road congestion or local weather information). Having received the request and got the requested information, DCs send the data to CS for necessary processing. However, as such information is closely related to DCs (e.g. their locations and health conditions), submitting the data in the aforementioned process could lead to the exposure of the data to a curious CS, malicious DCs or any intermediary party like an adversary that observes the process. In order to give a better description about the threats, we divide them into internal and external threats, which are elaborated separately below.

6.2.1 INTERNAL THREATS

In MCC, CS is normally assumed to be a trusted party. Nevertheless, it may have an intention to reveal the content of the transmitted or collected data for its curiosity or other purposes. Such an intention unavoidably invades the privacy of individual mobile users. In addition, as CS is an external party with diverse interests, it is difficult to ensure that CS is trustworthy in the MCC environment. Thus, it is essential that CS has no means to access any private data without authorisation.

6.2.2 EXTERNAL THREATS

In our proposed system, all the communications among DU, CS and DCs are executed via open wireless devices like Wi-Fi routers. Such devices are not secure and reliable as they are made public and thus vulnerable to external attacks like data interception and Denial of Service (DoS) [187], [188]. Those attacks can be orchestrated by an adversary in the following scenarios considered in this chapter:

(i) An adversary node that observes the transmission process and may want to eavesdrop on some communication devices between DU, CS and DCs.

(ii) An adversary or malicious node, which intends to obtain another nodes' secret key or leverage some cloud-based applications to retrieve relevant information about the key for decrypting the node's encrypted data.

6.3 DESIGN GOAL

Based on the above threats, we design our scheme to fulfil the following requirements:

(i) In the proposed process, DU uses a pair of master keys (p, r) to generate a secret key k_i for every DC node u_i. Then, it sends k_i to u_i securely (i.e. by encryption). The key generation may lead to a brute force attack on the master keys. Thus, we need to design the key generation algorithm in such a way that even if the attacker manages to hold a set of secret keys, it will not be able to retrieve any useful information about the master keys from the possessed keys.

(ii) Any cryptosystem is targeted by attacks on secret keys. Thus, we need to design our scheme with the ability to prevent any information about a secret key from being disclosed to any unauthorised users.

(iii) Every user should be assigned with a single and unique key for data protection and ease key management. Encrypting data using the same key by the same user may reveal some information about the plaintext. Thus, we need to design a scheme that prevents data leakage by adding extra information during the data encryption.

In the following section, we provide our proposed LHE solution in accordance with the above requirements. We describe all the processes executed by DU, CS and each DC, including key generation, data encryption, and data processing and recovery.

6.4 THE PROPOSED LHE SCHEME

Our LHE scheme consists of three algorithms, which are for key generation, data encryption, and data processing and recovery. To devise these algorithms for MCC, several parameters need to be set beforehand. Suppose that there are *n* mobile devices participating as DCs, each of which is denoted as u_i ($1 \leq i \leq n$). For each data item to be computed, we set the maximum length of l_d bits. In this setting, CS is employed to process the data received from the *n* mobile devices. We define two functions for data processing, which will be detailed in Sub-section 6.4.3.

The LHE scheme employs its key generation algorithm for DU to produce a unique symmetric secret key for each DC u_i. The data encryption algorithm of the scheme allows u_i to encrypt its contributed data with the key and outsources the encrypted data to CS for further processing. This enables all the DCs to contribute their values to CS in an encrypted form, which CS is unable to decrypt. The data processing and recovery algorithm of the scheme allows CS to process the received data without decryption based on the desired function requested by DU and transmit the result to DU. DU then decrypts the received result to recover its plaintext data without knowing any individual values contributed by different DCs. The details of these three algorithms are given in the sub-section below.

6.4.1 KEY GENERATION

The proposed LHE scheme employs a secret key for data encryption by each DC u_i. The secret key is shared only between its associated DU and u_i, and used for symmetric data encryption.

To produce this key, we adopt the parameter delineations for the verifiable encryption of RSA signatures [176]. That is, DU defines φ as the product of two safe primes p and w, i.e. $\varphi = pw$ where $p = 2p' + 1$ and $w = 2w' + 1$ with p' and w' being primes. φ will be used as a public number, w needs to be discarded without disclosing it to anyone and p should be kept securely. Additionally, DU selects a prime r ($< p$) and stores both p and r as its secret master keys.

To generate a key for each u_i, DU picks up random numbers $s_i < p$ and $q_i > p$ to produce the following symmetric secret key:

$$k_i = (rs_i + pq_i) \bmod \varphi \tag{6.1}$$

k_i is only given to u_i as its secret key. The n secret keys k_i need to meet the following conditions:

(a) For summation,

$$2^{l_d} n < r \text{ and } r(1 + 2^{l_{\tilde{c}} + l_s} n) < p \tag{6.2a}$$

(b) For multiplication,

$$2^{n l_d} < r \text{ and } (2^{l_d} + 2^{l_{\tilde{c}} + l_s} r)^n < p \tag{6.2b}$$

Here, l_d, $l_{\tilde{c}}$ and l_s are the maximal bit lengths of the data d_i, a random $\tilde{c}_i$ chosen by DC u_i for its data encryption, and random number s_i in key k_i for any i, respectively. The detailed reasons for the above conditions will be discussed later when the proposed data encryption and decryption are presented. In brief, the first part of both conditions says that the sum or the product of encrypted data items is less than r for the purposes of ensuring the recovery of the sum or product result. The second part of both conditions means that the calculation on the first part of each u_i's secret key together with the other items results in a number less than p. This condition also allows the summation or product result to be recovered.

For easy reference, the above key generation algorithm executed by DU is summarised below:

(i) Choose two safe primes p and w to calculate $\varphi = pw$, where $p = 2p' + 1$ and $w = 2w' + 1$ with p' and w' being primes.

(ii) Discard w.

(iii) Pick a prime number r ($< p$).

(iv) For i = 1 to n do

(v) Pick random numbers $s_i < p$ and $q_i > p$.

(vi) Compute $k_i = (rs_i + pq_i) \bmod \varphi$.

(vii) Send $\langle \varphi, k_i \rangle$ to DC u_i securely.

(viii) End for.

6.4.2 DATA ENCRYPTION

We now present how DC u_i generates its encrypted data to be sent to CS. To do so, u_i first performs the following calculation:

$$\tilde{\alpha}_i = (d_i + \tilde{c}_i k_i) \bmod \varphi \tag{6.3}$$

Here, $\tilde{\alpha}_i$ is the encrypted form of u_i data item d_i, and $\tilde{c}_i$ is a random number picked up by u_i and will be further explained in Section 6.5. Briefly, the reason to include $\tilde{c}_i$ in the encryption is to enhance the security of d_i. This is because the size of d_i may be too small when compared with secret key k_i, so some information about the key could be retrieved from the encrypted data if only the key was used in the encryption. Thus, by including such a random number $\tilde{c}_i$ during the encryption, it will hide the information about the secret key, as the encrypted data will be different from the key used for the encryption.

After the completion of the above calculation, u_i sends $\tilde{\alpha}_i$ to CS for storing and computing purposes. Upon the receipt of $\tilde{\alpha}_i$ from every u_i, CS starts its computation on the ciphertexts received to generate a result based on a function requested by DU.

The above data encryption algorithm performed by u_i is summarised as the following steps:

(i) Choose a random number $\tilde{c}_i$ ($< \varphi$).

(ii) Compute $\tilde{\alpha}_i = (d_i + \tilde{c}_i k_i) \bmod \varphi$.

(iii) Send $\tilde{\alpha}_i$ to CS securely.

6.4.3 DATA PROCESSING AND RECOVERY

In this sub-section, we first describe how CS computes the received data using addition and multiplication without the need for decryption. Once the computation on ciphertext is completed, the result will be sent to DU for decryption to recover the plaintext result. The following are the steps of addition and multiplication together with the reason for getting a correct result of ciphertext data computation. We will also show how the scheme supports homomorphism under both addition and multiplication. Homomorphism under those operations has been defined in the definition 3.9 in Sub-section 3.2.8.

(i) Summation

Let $f(d_1, d_2, \dots d_n) = \sum_{i=1}^{n} d_i$, i.e. the summation of all the data items d_i for $1 \leq i \leq n$. For summing n ciphertext in MCC, CS computes the received ciphertext data from each u_i as follows:

$$f(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) = (\sum_{i=1}^{n} \tilde{\alpha}_i) \bmod \varphi.$$

Then, this result will be sent to DU for recovering the sum. To obtain the sum, DU applies its master keys p and r to calculate:

$$\sum_{i=1}^{n} d_i = (f(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) \bmod p) \bmod r \tag{6.4}$$

This is valid due to the following relationships:

$(f(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) \bmod p) \bmod r$

$= \left(\left((\sum_{i=1}^{n} \tilde{\alpha}_i) \, mod \, \varphi \right) \bmod p \right) \bmod r$

$= \left(((\sum_{i=1}^{n} (d_i + \tilde{c}_i k_i)) \bmod \varphi) \bmod p \right) \bmod r$

$= \left(\left((\sum_{i=1}^{n} d_i + \sum_{i=1}^{n} \tilde{c}_i k_i) \, mod \, \varphi \right) \bmod p \right) \bmod r$

$= \left(\left((\sum_{i=1}^{n} d_i + \sum_{i=1}^{n} \tilde{c}_i (r s_i + p q_i)) \bmod pw \right) \bmod p \right) \bmod r$

$= \left((\sum_{i=1}^{n} d_i + \sum_{i=1}^{n} \tilde{c}_i r s_i + p\omega') \bmod p \right) \bmod r$

$= (\sum_{i=1}^{n} d_i + r \sum_{i=1}^{n} \tilde{c}_i s_i) \bmod r$

$= \sum_{i=1}^{n} d_i.$

Here, we have $(\sum_{i=1}^{n} \tilde{c}_i pq_i) \bmod pw = p\omega'$ with $\omega' < w$. Also the above fact is based on the following connections derived from condition (6.2a) defined in Sub-section 6.4.1, i.e. $2^{l_d} n < r$ and $r(1 + 2^{l_{\tilde{c}}+l_s} n) < p$:

$$\sum_{i=1}^{n} d_i < 2^{l_d} n < r, \sum_{i=1}^{n} d_i + r \sum_{i=1}^{n} \tilde{c}_i s_i < r + 2^{l_{\tilde{c}}+l_s} rn < p, \text{ and}$$

$$\sum_{i=1}^{n} d_i + r \sum_{i=1}^{n} \tilde{c}_i s_i + p\omega' < pw.$$

(ii) Product

Let $\tilde{\alpha}_i$ and $\tilde{\alpha}_j$ be the ciphertext of plaintext d_i and d_j respectively. The product of d_i and d_j can be recovered from the product of α_i and α_j as follows:

$$d_i d_j = \Big(\big((\tilde{\alpha}_i \tilde{\alpha}_j) \bmod \varphi\big) \bmod p\Big) \bmod r \tag{6.5}$$

This is valid due to the following relationship:

$$\Big(\big((\tilde{\alpha}_i \tilde{\alpha}_j) \bmod \varphi\big) \bmod p\Big) \bmod r$$

$$= \Big(\big((d_i + \tilde{c}_i k_i)\,(d_j + \tilde{c}_j k_j) \bmod \varphi\big) \bmod p\Big) \bmod r$$

Since $k_i = (rs_i + pq_i) \bmod \varphi$, then

$$= \Big(\big(\big(d_i + \tilde{c}_i (rs_i + pq_i)\big)\big(d_j + \tilde{c}_j (rs_j + pq_j)\big)\ mod\ \varphi\big)\ mod\ p\Big) mod\ r$$

$$= \Big(\big(((d_i + \tilde{c}_i rs_i) + \tilde{c}_i pq_i)\big((d_j + \tilde{c}_j rs_j) + \tilde{c}_j pq_j\big)\ mod\ \varphi\big)\ mod\ p\Big) mod\ r$$

$$= \Big(\big(((d_i + \tilde{c}_i rs_i)\big(d_j + \tilde{c}_j rs_j\big) + p\,\big((d_i + \tilde{c}_i rs_i)\tilde{c}_j q_j + \big(d_j + \tilde{c}_j rs_j\big)\tilde{c}_i q_i +$$

$$\tilde{c}_i q_i \tilde{c}_j pq_j\big))\ mod\ pw\big)\ mod\ p\Big) mod\ r$$

$$= \Big(\big((d_i + \tilde{c}_i rs_i)\big(d_j + \tilde{c}_j rs_j\big) + p\omega"\big)\ mod\ p\Big) mod\ r$$

$$= (d_i + \tilde{c}_i rs_i)\big(d_j + \tilde{c}_j rs_j\big)\ mod\ r$$

$$= \big(d_i d_j + r(d_i \tilde{c}_j s_j + \tilde{c}_i s_i d_j + \tilde{c}_i s_i \tilde{c}_j rs_j)\big)\ mod\ r$$

$$= d_i d_j$$

.

Here, we have $p\left((d_i + \tilde{c}_i r s_i)\tilde{c}_j q_j + \left(d_j + \tilde{c}_j r s_j\right)\tilde{c}_i q_i + \tilde{c}_i q_i \tilde{c}_j p q_j\right) \bmod pw = p\omega"$ with $\omega" < w$. Also the above fact is based on the following connections derived from condition (6.2b) in Sub-section 6.4.1, i.e. $2^{2l_d} < r$ and $\left(2^{l_d} + 2^{(l_{\tilde{c}}+l_s)}r\right)^2 < p$ for $n = 2$:

(i) $d_i d_j < 2^{2l_d} < r,$

(ii) $(d_i + \tilde{c}_i r s_i)\left(d_j + \tilde{c}_j r s_j\right) < (2^{l_d} + 2^{l_{\tilde{c}}+l_s}r)^2 < p,$ and

(iii) $(d_i + \tilde{c}_i r s_i)\left(d_j + \tilde{c}_j r s_j\right) + p\omega" < pw.$

We can generalise Equation (6.5) to retrieve a general relationship on a product of n plaintexts from n ciphertexts as follows. Let $d_1 \cdot d_2 \cdot \ldots \cdot d_n = \prod_{i=1}^{n} d_i$, i.e. the product of all the plaintexts d_i for $1 \leq i \leq n$. For multiplying their corresponding ciphertexts in MCC, CS computes the received ciphertext data from each u_i:

$$(\tilde{\alpha}_1 \cdot \tilde{\alpha}_2 \cdot \ldots \cdot \tilde{\alpha}_n) \bmod \varphi \; = \prod_{i=1}^{n} \tilde{\alpha}_i \bmod \varphi.$$

Then, this ciphertext product is sent to DU that recovers the plaintext result by the following computation:

$$\prod_{i=1}^{n} d_i = \left(\left(\left(\prod_{i=1}^{n} \tilde{\alpha}_i\right) \bmod \varphi\right) \bmod p\right) \bmod r \tag{6.6}$$

This is a valid operation due to the following relationships:

$$\left(\left(\left(\prod_{i=1}^{n} \tilde{\alpha}_i\right) \bmod \varphi\right) \bmod p\right) \bmod r$$

$$= \left(\left(\left(\prod_{i=1}^{n} (d_i + \tilde{c}_i k_i)\right) \bmod \varphi\right) \bmod p\right) \bmod r$$

$$= \left(\left(\left(\prod_{i=1}^{n} \left(d_i + \tilde{c}_i (r s_i + p q_i)\right)\right) \bmod \varphi\right) \bmod p\right) \bmod r$$

$$= \prod_{i=1}^{n} d_i.$$

The above relationships are similar to those detailed for Equation (6.5)

Based on the FHE concept given in Sub-section 3.2.8. it is clear that our LHE scheme is homomorphic under both addition and multiplication operations. For easy reference, the algorithms presented in this sub-section are summarised as below:

Ciphertext Addition by CS:

(i) Compute $\tilde{\alpha} = (\sum_{i=1}^{n} \tilde{\alpha}_i) \bmod \varphi$.

(ii) Send $\tilde{\alpha}$ to DU securely.

Ciphertext Multiplication by CS:

(i) Compute $\tilde{\alpha}' = \prod_{i=1}^{n} \tilde{\alpha}_i \bmod \varphi$.

(ii) Send $\tilde{\alpha}'$ to DU securely.

Sum Recovery by DU:

(i) Get master keys r and p.

(ii) Compute $d = (\tilde{\alpha} \bmod p) \bmod r$.

Product Recovery by DU:

(i) Get master keys r and p.

(ii) Compute $d' = (\tilde{\alpha}' \bmod p) \bmod r$.

In the following section, we describe how the LHE scheme can prevent several attacks on the scheme even though it has less complexity.

6.5 Security Analysis

In this section, we analyse the security of the scheme proposed in the previous section by considering both external and internal attacks described in Section 6.2. Those threats can be categorised as a brute force attack on DU's master keys, a brute force attack on DCs' secret keys and a many time pad attack.

6.5.1 Brute Force Attack on the Master Keys

By using our scheme to encrypt the data, this attack on the ciphertext can be formulated as follows. DU sends a request to the DC group. Suppose that several attackers are members of the group. The attackers pretend to have the data related to the request and will let DU know about it. As DU is not able to differentiate the attackers from other genuine contributors, DU will securely send a symmetric key to each DC, including those attackers, for data encryption. Having received the keys from DU, the attackers start the computation of deducing DU's master keys p and r. They subtract one key from another, hoping to remove r (or p). If r were removed, the attackers could compute the Greatest Common Divisor (GCD) of the remainder with the public value φ. By doing this, master key p would be discovered by the attackers, which could then be used to work out r in a similar way.

We now present the above attack in detail. Having received the keys from DU, the attackers start colluding for the computation of recovering the master keys. Suppose that A_1 and A_2 are attackers with received keys k'_1 and k'_2 in the following form:

$k'_1 = (r + pq_1) \bmod \varphi$, and

$k'_2 = (r + pq_2) \bmod \varphi$.

A_1 and A_2 then subtract both keys:

$$\tilde{k} = k'_1 - k'_2 = \big((r + pq_1) - (r + pq_2)\big) \bmod \varphi$$

$$= \big(p(q_1 - q_2)\big) \bmod \varphi.$$

Assume that $p(q_1 - q_2) < \varphi$. To determine master key p, they compute the GCD of the following values:

$$\text{GCD}\left(\tilde{k}, \varphi\right) = p.$$

This is valid due to the following relationships:

$$\text{GCD}\left(\tilde{k}, \varphi\right) = \text{GCD}\ (p(q_1 - q_2), pw) = p$$

Furthermore, the attacker can retrieve the value of r by computing $k'_1 \bmod p = r$.

After the completion of the above steps, the attackers obtain master keys (p, r) generated by DU. If they can intercept the communication between another group member u_i and CS, then they can gain the information from the encrypted data as they have the decryption keys.

To prevent such an attack, we attach a random parameter s_i to r in the definition of our keys k_i to avoid the above elimination of r when the attackers subtract two distinct keys they received from DU. This can be seen by repeating the elimination process as follows:

$k_1 = (rs_1 + pq_1) \bmod \varphi$, and

$k_2 = (rs_2 + pq_2) \bmod \varphi$.

If A_1 and A_2 subtract both keys:

$$\tilde{k}' = k_1 - k_2 = \big((rs_1 + pq_1) - (rs_2 + pq_2)\big) \bmod \varphi$$

$$= \big(r(s_1 - s_2) + p(q_1 - q_2)\big) \bmod \varphi$$

Now master key p cannot be retrieved by computing $\text{GCD}\left(\tilde{k}', \varphi\right)$ due to $\tilde{k}'$ being no longer a multiple of p, i.e.

$$\text{GCD}\left(\tilde{k}', \varphi\right) = \text{GCD}\ (r(s_1 - s_1) + p(q_1 - q_2), pw) \neq p$$

However, according to [147], such improvement allows known attacks like brute-forcing the remainders on the secret keys. Thus, we review such an attack for two keys k_1 and k_2 and prove the resilience of our scheme against the attack by using a contradiction approach described below.

A simple brute-force attack [147] tries to guess $\check{r}_1 = rs_1$ and $\check{r}_2 = rs_2$ and verify the guess with a GCD computation. Specifically, for two correctly guessed values $\check{r}_1'$ and $\check{r}_2'$ of $\check{r}_1$ and $\check{r}_2$ (i.e. $\check{r}_1' = \check{r}_1$ and $\check{r}_2' = \check{r}_2$), the attacker compute:

$k_1^{''} = (k_1 - \check{r}_1') \bmod \varphi$, and

$k_2^{''} = (k_2 - \check{r}_2') \bmod \varphi$.

By contradiction, if we assume $pq_1 < p$ and $pq_2 < p$, then p can be computed by:

$$\text{GCD}\left(k_1^{''}, k_2^{''}\right) = p.$$

The above equation is true due to the following relationships:

$$\text{GCD}\left(k_1^{''}, k_2^{''}\right)$$

$$= \text{GCD}\left((k_1 - \check{r}_1') \bmod \varphi, (k_2 - \check{r}_2') \bmod \varphi\right)$$

$$= \text{GCD}\left(\left((\check{r}_1 + pq_1) - \check{r}_1'\right) \bmod \varphi, \left((\check{r}_2 + pq_2) - \check{r}_2'\right) \bmod \varphi\right)$$

$$= \text{GCD}\ (pq_1 \bmod \varphi, pq_2 \bmod \varphi)$$

However, both pq_1 and pq_2 are in fact greater than φ as defined in Sub-section 6.4.1. Then, GCD $(pq_1 \bmod \varphi, pq_2 \bmod \varphi) \neq p$. Therefore, our selection of q_i prevents the brute-forcing on the master keys. In addition, to avoid the brute-force attack on the remainder, the length of $\check{r}_1$ and $\check{r}_2$ should be large enough to make it much harder for the attacker to guess them correctly, but it must be smaller than p to allow the recovery of the plaintext data during the decryption.

6.5.2 Brute Force Attack on the Secret Keys

To protect the proposed scheme from a brute force attack on a DC's secret key, a random parameter $\tilde{c}_i$ is added in ciphertext $\tilde{\alpha}_i$ as defined in Equation 6.3. Such a parameter can

improve the security of the encrypted data by avoiding any information about the encryption key being disclosed to unauthorised users. Having another random parameter $\tilde{c}_i$ means that, given items $\tilde{\alpha}_i$ and φ, an attacker with some knowledge about the plaintext related to $\tilde{\alpha}_i$, is unable to retrieve any useful information for successfully inferring the encryption key.

We now argue the above claim in detail. Suppose that the encryption algorithm for plaintext d_i with secret key k_i is:

$$\tilde{\alpha'}_i = (d_i + k_i) \bmod \varphi. \tag{6.7}$$

Such an encryption algorithm is vulnerable against a brute-force attack on key k_i. The reason is that, when the size of d_i is small compared to key k_i used for the encryption, then the high-end part of the ciphertext generated is likely to be identical to that of the key. This means that for different encryptions with the same key, the differences among them are just the bits at the lower end of the ciphertext, while the rest (the higher end of the key) remains the same. In case the attacker is able to obtain several ciphertexts, he/she can compare them to spot their identical part so as to gain that part of the key. If the remaining part of the key is short, then the attacker can guess it by a brute force attack.

Thus, to prevent such an attack, we add a unique random parameter $\tilde{c}_i$ in Equation 6.3 for our encryption. By adding this parameter, attempting to spot any identical part of the key will be intractable as it will be hidden by $\tilde{c}_i$, which is known only by u_i.

6.5.3 A MANY TIME PAD ATTACK

Our scheme could also face a so called many time pad attack [189]. This attack allows an adversary to obtain some information about plaintexts. Let $\dot{\tilde{\alpha}}_\iota = (d_i + k_i) \bmod \varphi$ and $\dot{\tilde{\alpha}}"_i = (d"_i + k_i) \bmod \varphi$ be two distinct ciphertexts of plaintext d_i and $d"_i$ respectively. DC u_i using the same key to produce both ciphertexts. If the attacker gets hold of $\dot{\tilde{\alpha}}_\iota$ and $\dot{\tilde{\alpha}}"_i$, it can compute the difference between them to gain some knowledge about d_i and $d"_i$. By using a crib dragging technique [190] on this result, the attacker would eventually discover all the information about the data.

We now elaborate such an attack in detail. Suppose that the attacker has gained $\dot{\tilde{\alpha}}_\iota$ and $\dot{\tilde{\alpha}}"_i$, it can compute:

$$d' = \left(\dot{\tilde{\alpha}}_\iota - \dot{\tilde{\alpha}}"_i\right) \bmod \varphi = \left((d_i + k_i) \bmod \varphi - (d"_i + k_i) \bmod \varphi\right) \bmod \varphi$$

$$= (d_i - d"_i) \bmod \varphi$$

This is caused by u_i using the same key k_i for the different encryptions. Hence the difference between the two ciphertexts produces the above result, revealing some information about the plaintexts (i.e. the difference between d_i and $d"_i$). This security flaw can be remedied by adding a random parameter $\tilde{c}_i$ to the encryption in Equation 6.3:

$$\tilde{\alpha}_i = (d_i + \tilde{c}_i k_i) \bmod \varphi.$$

In this case, the possession of $\tilde{\alpha}_i = (d_i + \tilde{c}_i k_i) \bmod \varphi$ and $\tilde{\alpha}''_i = (d''_i + \tilde{c}''_i k_i) \bmod \varphi$ does not allow the attacker to retrieve the difference between the two plaintexts as $\tilde{c}_i$ and $\tilde{c}''_i$ are randomly picked up by u_i and highly unlikely to be identical. This prevents any partial leakage to the attacker.

6.6 A Comparison with Existing Solutions

Although the efficiency of FHE schemes has received extensive attention and efforts have been made to improve it, there are still limitations on these methods as discussed in Section 3.3. In this section, we provide a comparison among selected existing solutions and highlight the improvements achieved by our scheme. Table 16 shows a brief summary of these selected schemes.

Table 16: Selected Existing Schemes

	Bits-based	Mixed-based	Integer-based
Description	Both input and output are in the form of bits.	Input is in the form of integers and the output is in the form of bits.	Both input and output are in the form of integers.
Existing Schemes	[128], [149], [153]	[44], [48], [147]	[35], [46], [76]
Advantage/s	Easier to achieve fully homomorphic properties.	Support arbitrary addition.	Support both addition and multiplication.
Limitation/s	More storage space required. Higher bandwidth for data transmission.	Only support one multiplication. High computational complexity and communication costs to implement by mobile devices [146].	[46] requires a large public key. [35], [76] designed for a higher performance device.

	More computing resources for encryption and decryption [46].	Require a large public key size for encryption [35], [76].

6.6.1 BITS-BASED ENCRYPTION SCHEMES

Most of the existing FHE schemes are suffering from an efficiency issue as their choice of plaintext for encryption and the generated ciphertext is in the form of bits [128], [149], [153]. The advantage of such individual bit based encryption is easier to achieve fully homomorphic properties. Nevertheless, these encryption schemes significantly reduce the storage and communication efficiency that leads to an increase in the computational time. Furthermore, the schemes require applications to convert computation tasks into binary addition and multiplication operations, which makes the computation more complex [46]. In contrast, our proposed scheme shows simplicity that improves its efficiency. Our scheme is also designed based on integers to reduce storage consumption and increase communication efficiency so as to allow resource-constraint devices like smartphones and tablets to prolong battery lifetime.

6.6.2 MIXED-BASED ENCRYPTION SCHEMES

Several schemes have been proposed so that the plaintext for encryption is in the form of integers while the output remains in the form of bits [44], [48], [147]. Those schemes support arbitrary functions on an encrypted form with better efficiency as they are designed based on integers. Nevertheless, such schemes are also hardly to be implemented by resources-constraint devices due to still high computational complexity and communication costs [46]. In addition, those schemes require a large public key size for encryption [35], [76], which rapidly reduces the battery lifetime of mobile devices during data encryption. To remedy these weaknesses, our scheme is designed with its key size depending on the computation function required. For instance, our addition only function requires a shorter key for encryption so as to improve the scheme's efficiency and reduce the battery consumption for the increase lifetime of mobile devices. This flexibility of the key size selection enables the scheme to incorporate any computation functions with better efficiency. As a result, our scheme allows mobile devices, which

have limited computing resources and storage spaces, to leverage rich-mobile applications provided by CSPs in an efficient and secure manner.

6.6.3 INTEGER-BASED ENCRYPTION SCHEMES

In the recent work by H. Zhou and G. Wornell [46], a new homomorphic encryption scheme has been developed. The scheme operates directly on integer vectors that support three operations, which are more specifically implemented in signal processing applications. The operations supported by this scheme are addition, linear transformation and weighted inner products. However, such a scheme has a limitation on the degree of a polynomial to be computed efficiently. Furthermore, this scheme suffers from an efficiency issue due to the large public key size adopted [46]. In contrast, our LHE scheme is able to compute both addition and multiplication on an encrypted form as long as the noise size (which is depending on the number of additions and multiplications) is less than the key for encryption. It can be implemented in various MCC related applications due to its lightweight property and strong security.

In addition, the works in [35] and [76] have proposed schemes with both plaintext and ciphertext in the form of integers. This approach improves the schemes' efficiency as discussed earlier. In our view, those are the only schemes that consider the plaintext and the generated ciphertext in the integer form. Both schemes allow arbitrary functions to be executed on an encrypted data. Nevertheless, such schemes are designed for devices with higher performance due to the plaintext and ciphertext data being represented as matrices. Processing and transmitting data in a matrix form requires more computing resources and bandwidths as well as storages. In contrast, our LHE scheme has its output in a single integer form, which requires less time for computation, less bandwidth for data transmission and less space for data storage. These merits enable our scheme to be implemented more efficiently by resource-constraint devices like smartphones. To support this claim, we have tested the efficiency of both schemes under a MCC setting with the details provided in the following section.

6.7 APPLICATION SETTINGS

Our new LHE scheme enable mobile data to be outsourced and processed in cloud. However, data outsourcing itself is not sufficient to overcome the limitation of the mobile

devices as the security and privacy of the data are highly important and should be consider carefully [4], [9], [14], [129]. Also, excessive computation for securing the data on mobile devices before the outsourcing degrades their battery lifetime [168]. Thus, the security and computation complexity need to be balanced in order to provide a better scheme for outsourcing the data. In this section, we describe the application settings for implementing our scheme.

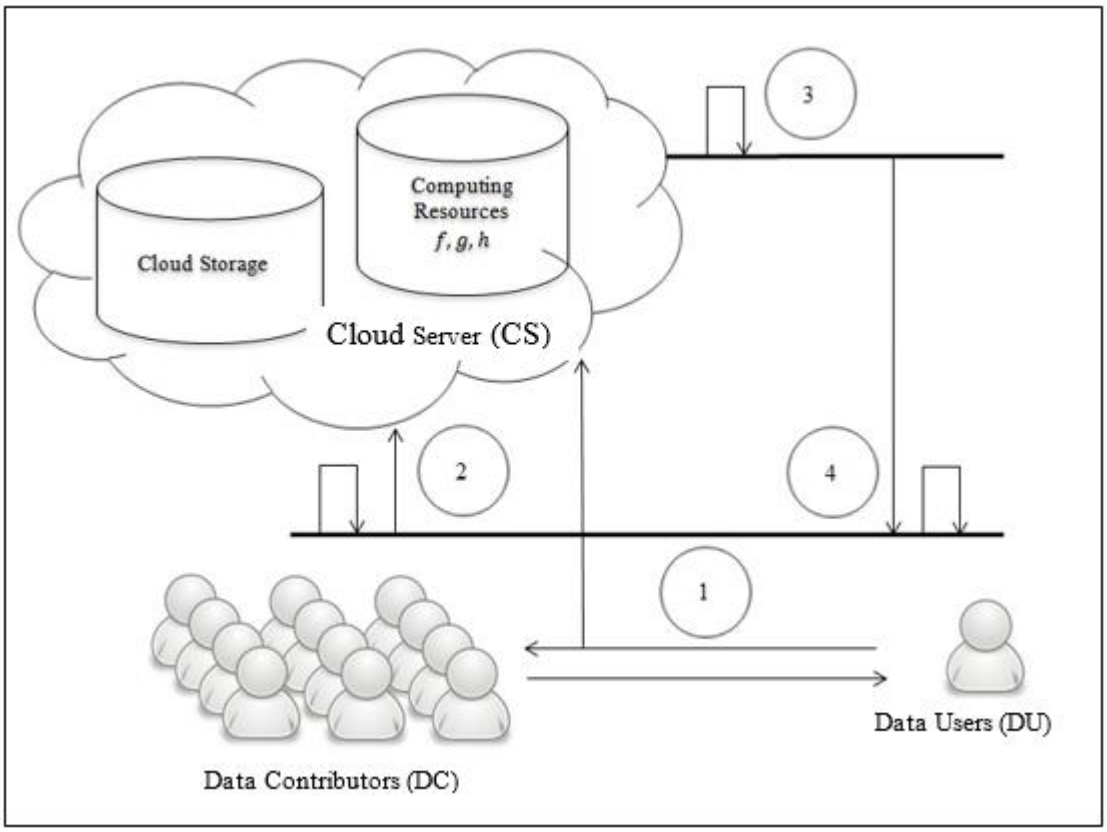


Figure 23: A process of implementing LHE in MCC

To process data in its ciphertext form, we propose a process of implementing the proposed LHE scheme in MCC as illustrated in Figure 23. We assume that there is a group of DCs, a DU and a CS managed by its CSP with their responsibilities described below:

(i) DCs:

They are a group of mobile users like smartphones and tablets. All the devices are connected to the Internet through wireless connections. The group members contribute their data to DU indirectly through CS after necessary processing.

(ii) DU:

A third party organisation requires information from DCs in relation to specific tasks and purposes. It has low computing resources and storage spaces. It leverages the technology provided by the cloud to compute and store data purposely.

(iii) CS:

A cloud server managed by a CSP, which possesses a huge amount of computing power and storage space for computing and storing purposes. It is an untrusted party to provide Internet based applications and deliver services through Internet connections. DU takes advantage of such services to ask CS to collect and process the data from DCs for certain purposes.

As an example, we consider a public health scenario where a group of individuals as DCs in the park share their personal health information like heart rates, blood pressures and weights with a local hospital as DU to get some statistical results based on the provided data. Here, the DCs are unwilling to disclose their personal data to CS and DU as there is no guarantee that their data privacy will not be violated. Thus, to allow such a scenario to be implemented securely, we propose a secure process as illustrated in Figure 23. The process involves four stages, which are described below:

In stage 1, DU broadcasts a request to the group to ask for specific data to be processed by the CS. Upon receipt of the request, each group member u_i in the possession of the requested data responds to DU to confirm its willingness for the data provision. After receiving the response from each of DC, DU utilies key generation algorithm in Sub-section 6.4.1 to generate a set of secret keys for the responded group members and send a different key to each member.

In stage 2, each participating DC u_i applies the data encryption algorithm in Sub-section 6.4.2 to encrypt its data with secret key k_i received from DU in stage 1. u_i then sends the encrypted data to CS for processing.

In stage 3, after receiving the ciphertext data from every participating DC u_i, CS uses the algorithm in Sub-section 6.4.3 to process the received data based on the requirements set by DU and then transfers the result to DU.

Finally, in stage 4, DU employs the algorithm in Sub-section 6.4.3 to decrypt the received result with its master keys to reveal the plaintext result.

For simulation purposes, we have set up various sizes of mobile networks using OPNET 14.5.A. The simulations have been designed and implemented based on the mobile cloud computing setting in Figure 23 and attribute configurations in Table 17. The table includes the packet size for data transmissions in the network, the traffic type of service, and necessary wireless connection parameters. Based on these settings, we have run each simulation 50 times for accuracy and consistency [23]. Some computation times needed for the simulation are taken from the results measured by MATLAB.

Table 17: Simulation attribute settings

Node Type	Packet Size (Bytes)	Traffic Type of Service	Wireless LAN Parameters				
			BSS ID	AP	Physical Characteristic	Data Rate (Mbps)	Buffer Size (Bits)
DC	128	Best Effort	0	Off	Direct Sequence	11	256000
DU	128	Best Effort	0	Off	Direct Sequence	11	256000
Server	128	Best Effort	0	On	Direct Sequence	11	256000

6.8 PERFORMANCE EVALUATION

In this section, we evaluate the total execution time of our LHE scheme and the chosen scheme for a comparison. The details of the chosen scheme can be found in [35]. The comparison results demonstrate the merit of our lightweight scheme in terms of its efficiency.

6.8.1 EXPERIMENTAL SETUP

In our total execution time tests, we have evaluated the total execution times of the two schemes for one round data processing in an encrypted form. We have simulated 50 runs for each round of data processing to get consistent results. This processing includes summation and multiplication on ciphertext data. We have implemented the two schemes with various numbers of DCs. The parameter settings of the schemes are given in the sub-section, while the results and discussions of the conducted experiments are given in 6.8.3.

6.8.2 PARAMETERS SETTINGS

In our experiments, we use the parameter settings shown in Table 18. These parameter settings satisfy the conditions stated in Sub-section 6.4.1 in order to execute the two functions defined in Sub-section 6.4.3.

Table 18 : The parameter settings

No.	Parameters	Bit length	
		LHE Scheme	The Chosen Scheme
1.	p	664	1024
2.	w	360	N/A
3.	r	80	16
4.	l_d	10	10
5.	$l_{\tilde{c}}$	80	N/A
6.	l_s	80	N/A
7.	q_i	1024	N/A
8.	φ	1024	N/A

6.8.3 RESULTS AND DISCUSSION

The main purpose for evaluating the total execution times is to compare the efficiency of the two schemes. In our experiments, we have measured the execution times for completing one round of data processing using the following functions:

(i) Addition-only operation:

$$f(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) = \sum_{i=1}^{n} \tilde{\alpha}_i \bmod \varphi.$$

(ii) Mixed addition and multiplication operation:

$$g(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) = \left(\sum_{i=1}^{n} {\tilde{\alpha}_i}^2 \right) \bmod \varphi.$$

The condition (i) defined in Sub-section 6.4.1 needs to be satisfied by the secret keys used in $f(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n)$. However, a new condition should be defined for $g(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n)$ as it involves a mixture of both additions and multiplications. By applying similar relationships discussed in Sub-section 6.4.3, we can set the condition as $2^{2l_d} n < r$ and $r\left(1 + \left(2^{l_d + l_{\tilde{c}} + l_s + 1} + 2^{2l_{\tilde{c}} + 2l_s + \log_2 r}\right) n\right) < p$ in order to enable DU to compute $d'' = (g(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) \bmod p) \bmod r$ (i.e. $d'' = \sum_{i=1}^{n} {d_i}^2$) for the plaintext result. This claim is supported by the following fact:

$$(g(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n) \bmod p) \bmod r$$

$$= \left(\left(\left(\sum_{i=1}^{n} {\tilde{\alpha}_i}^2 \right) \bmod \varphi \right) \bmod p \right) \bmod r$$

$$= \left(\left(\left(\sum_{i=1}^{n} \left((d_i + \tilde{c}_i r s_i) + \tilde{c}_i p q_i \right)^2 \right) \bmod \varphi \right) mod\ p \right) mod\ r$$

$$= \Big(\Big(\sum_{i=1}^{n} ((d_i + \tilde{c}_i r s_i)^2 + p\big(2(d_i + \tilde{c}_i r s_i)\tilde{c}_i q_i + \tilde{c}_i{}^2 q_i{}^2 p\big)) \bmod pw \Big) \bmod p \Big) \bmod r$$

$$= \Big(\Big(\sum_{i=1}^{n} d_i{}^2 + r \sum_{i=1}^{n} \big(2 d_i \tilde{c}_i s_i + \tilde{c}_i{}^2 s_i{}^2 r\big) + pw" \Big) \bmod p \Big) \bmod r$$

$$= \Big(\sum_{i=1}^{n} d_i{}^2 + r \sum_{i=1}^{n} \big(2 d_i \tilde{c}_i s_i + \tilde{c}_i{}^2 s_i{}^2 r\big) \Big) \bmod r$$

$$= \sum_{i=1}^{n} d_i{}^2$$

The above result is based on the condition $2^{2l_d} n < r$ and $r\big(1 + \big(2^{l_d + l_{\tilde{c}} + l_s + 1} + 2^{2l_{\tilde{c}} + 2l_s + \log_2 r}\big) n\big) < p$, which leads to $\sum_{i=1}^{n} d_i{}^2 < 2^{2l_d} n < r$ and $\sum_{i=1}^{n} d_i{}^2 + r \sum_{i=1}^{n} \big(2 d_i \tilde{c}_i s_i + \tilde{c}_i{}^2 s_i{}^2 r\big) < r + \big(2^{l_d + l_{\tilde{c}} + l_s + 1} + 2^{2l_{\tilde{c}} + 2l_s + \log_2 r}\big) rn < p$. Also, we use $pw"$ to represent the result of $p\big(2(d_i + \tilde{c}_i r s_i)\tilde{c}_i q_i + \tilde{c}_i{}^2 q_i{}^2 p\big) \bmod pw$ with $w" < w$.

Based on the parameter settings in Table 18, we now show that the n secret keys can meet the necessary conditions.

Claim 1: The addition only operation can satisfy the condition $2^{l_d} n < r$ and $r(1 + 2^{l_{\tilde{c}} + l_s} n) < p$.

Proof: Let $\log_2 n = 9$. Based on the settings in Table 18, we have:

$$2^{l_d} n \le 2^{l_d + \log_2 n} = 2^{10+9} = 2^{19} < 2^{80} = 2^{\log_2 r} = r.$$

Thus, $2^{l_d} n < r$ is satisfied.

Similarly, we can do the following calculation:

$$r(1 + 2^{l_{\tilde{c}} + l_s} n) \le 2^{\log_2 r} + 2^{l_{\tilde{c}} + l_s + \log_2 n + \log_2 r} = 2^{80} + 2^{80+80+9+80} = 2^{80} + 2^{249} < 2^{664} = 2^{\log_2 p} = p.$$

Hence, $r(1 + 2^{l_{\tilde{c}} + l_s} n) < p$ is also satisfied.

Claim 2: The mixed addition and multiplication operation meets the condition $2^{2l_d} n < r$ and $r\big(1 + \big(2^{l_d + l_{\tilde{c}} + l_s + 1} + 2^{2l_{\tilde{c}} + 2l_s + \log_2 r}\big) n\big) < p$.

Proof: In this case, we set the same value for n, i.e. $\log_2 n = 9$. It is clear that $2^{2l_d} n < r$ is satisfied due to $2^{2l_d} n \le 2^{2l_d + \log_2 n} = 2^{29} < 2^{80} = r$.

Similar to the proof of Claim 1, $r\left(1+\left(2^{l_d+l_{\tilde{c}}+l_s+1}+2^{2l_{\tilde{c}}+2l_s+\log_2 r}\right)n\right)<p$ is also met because of:

$$r\left(1+\left(2^{l_d+l_{\tilde{c}}+l_s+1}+2^{2l_{\tilde{c}}+2l_s+\log_2 r}\right)n\right) \leq 2^{\log_2 r}+2^{\log_2 r+\log_2 n+l_d+l_{\tilde{c}}+l_s+1}+$$
$$2^{2\log_2 r+\log_2 n+2l_{\tilde{c}}+2l_s}=2^{80}+2^{80+9+10+80+80+1}+2^{160+9+160+160}=2^{80}+2^{260}+$$
$$2^{489}<2^{664}=p.$$

Therefore, we conclude that those parameter settings have met all the conditions as stated in Section 6.4.1.

Through several experiments conducted based on those parameter settings in Table 18, our LHE scheme is faster than the chosen scheme as the computation complexity of the chosen scheme is higher than that of our scheme. Furthermore, by using our proposed scheme, the overall size of ciphertext data transmitted from each DC to CS is shorter than that of the chosen scheme. The reason is that the chosen scheme represents data as a 4 by 4 matrix. Thus, submitting a single ciphertext requires 16 elements of the matrix to be transmitted [35]. Such a data transmission introduces more delays and requires more bandwidth and storage space. The experimental results based on the delay of both schemes are compared to evaluate their efficiency for data summation (i.e $f(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n)$) and mixed operation (i.e. $g(\tilde{\alpha}_1, \tilde{\alpha}_2, \dots, \tilde{\alpha}_n)$) within one round of data processing. All the results are illustrated in Figure 24 for data summation and Figure 25 for the mixed operation. We have measured the delays based on the designed process depicted in Figure 23, where three parties communicate with one another to process data collection, through ciphertext processing, to plaintext recovery.

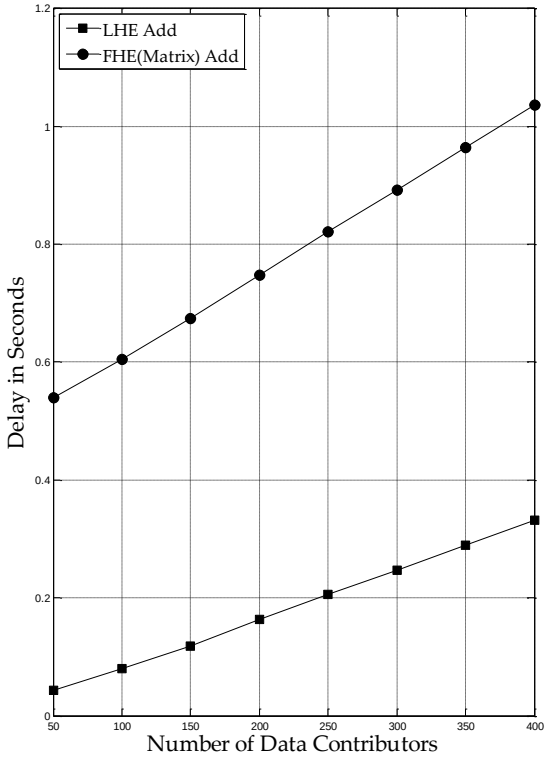


Figure 24: Total execution times for one round of data processing (summing) in the ciphertext form

Figure 24 illustrates the delay of one round of data summation on ciphertexts by the two schemes. The difference from the two lines shows that the delay introduced by our LHE scheme gradually increases as the number of contributors increases. However, for the chosen scheme, its delay is over 2 times higher than our scheme and goes higher as the number of data contributors gets larger. For example, when the number of contributors is 50, the delay caused by LHE is below 0.1 second, whereas the delay caused by the chosen scheme is nearly 0.6 second. Furthermore, when the number of contributors increases to 400, the delay caused by LHE is still lower, which is below 0.4 second, whereas the chosen scheme takes longer than 1 second. The main reason for the above differences is that the chosen scheme involves the matrix multiplication of keys and data [35]. Such computation incurs cubic complexity on each mobile node prior to data processing and hence extra delays. Furthermore, transmitting a ciphertext in a matrix form by a resource-constraint device with lower data rate and buffer size requires capacities require more bandwidth, leading to a further delay to the data processing. In contrast, transmitting a ciphertext via a single integer by the same device using our scheme requires less bandwidth and thus consumes shorter time for completing the transmission process.

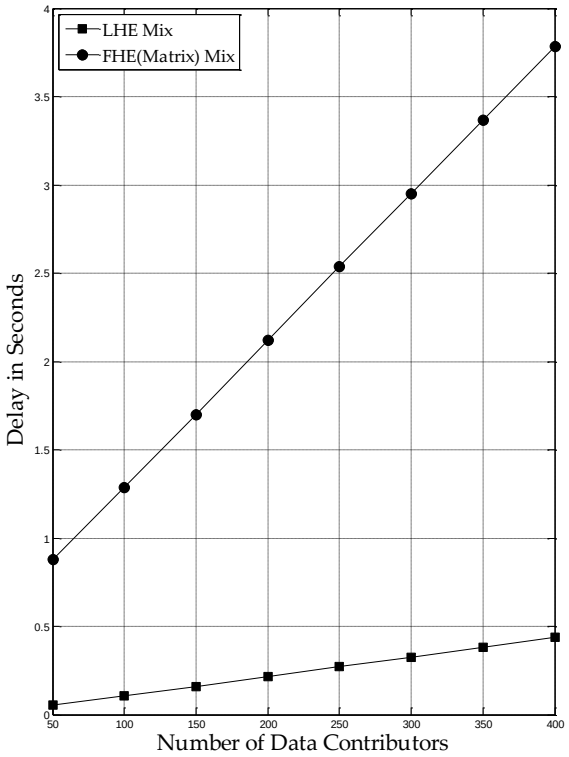


Figure 25: Total execution times for one round of data processing (mixed operation)

Figure 25 shows the delays of the two schemes on one round of data processing involving the mixed operation over ciphertext data. The two lines obviously demonstrate a significant difference in the delays between the two schemes. For our LHE scheme, the line gradually rises according to the rise in the number of contributors, but it is still acceptable and significantly smaller compared to the delays produced by the chosen scheme. In contrast, for the chosen scheme, the line increases rapidly in relation to the rise of the number of contributors. For instance, when the number of data contributors is 50, the delay caused by our scheme is nearly 0.1 second, but the delay by the chosen scheme is more than 0.7 seconds. Furthermore, when the number of data contributors is 400, the delay introduced by the chosen scheme drastically increases to more than 3.5 seconds to complete the process, while our scheme only takes less than 0.5 second. Similar to the previous case, the main reason for the above differences is due to multiplications on 4 by 4 matrices used by the chosen scheme, which contain a big integer at each entry. Moreover, as the number of contributors gets higher, more time is required to complete the operation. Thus, the result in Figure 25 shows that our scheme is much more efficient.

6.9 SUMMARY

In this chapter, we have proposed our new LHE scheme that enables arbitrary data to be processed by homomorphic addition and multiplication in an encrypted form. The scheme has achieved simplicity and improved on its efficiency as the scheme is designed based on single integer rather than a matrix of integers. The related experiments have been conducted to compare the total execution times of our scheme with the most relevant existing scheme. The results have shown that our scheme can operate faster than the chosen scheme as it has less complexity in terms of its computation. Such a result is essential to enable the scheme to be executed by resource-constraint devices in an efficient manner. Furthermore, we have also provided the security analyses of our scheme to show that although it has less complexity, it can achieve stronger security on the outsourced data. The next chapter will provide a summary of the whole thesis together with further developments that can be done to improve our schemes in the future.

CHAPTER 7

CONCLUSIONS AND FURTHER DEVELOPMENT

7.1 INTRODUCTION

This thesis has presented two novel schemes APDA and LHE to allow mobile data to be collected, processed and stored in MSS and MCC environments, respectively. With the enhancement of cloud-based applications like Google-apps, the need for processing data in its ciphertext form is very much demanded in order to protect the data from disclosure to CSPs. Such a level of security promises tremendous advantages to the mobile users to enjoy a variety of applications without having to be concerned about their privacy and mobile devices' limitations.

In this thesis, APDA has been proposed to allow data to be aggregated in a privacy-preserving manner. Data aggregation techniques that have widely been implemented in WSNs could be leveraged to extend the battery lifetime of mobile devices in a MSS. This is due to the emerging technology on mobile devices allowing more sensors to be embedded in such devices to sense and aggregate surrounding data to be further processed by the CSPs. For instance, an organisation like a hospital is interested in collecting and processing urban related information like personal health data instantly. Thus, users of such an application are requested to constantly upload their data so as to allow the

organisation to process the data and provide useful feedback to the users' immediate need. Such personal medical data is directly related to the privacy of the users and there is no guarantee that their data is protected properly. Consequently, the users may refuse to share the information and cooperate with the organisation.

In another scenario, the participating users may consist of both well-behaved and misbehaved users. For an example, misbehaved users such as a selfish driver may modify the information about his car speed to prevent other road users from choosing his road. Identifying such users in a privacy-preserving data aggregation is a very challenging task. Thus, the novel APDA scheme is proposed to hold the users' accountable for their misbehaviour while minimising device performance degradation. Also our APDA has been extended to support maximal value finding in a privacy-preserving manner. Our experimental results on both APDA and its extended versions have demonstrated that they can achieve better efficiency compared to the existing schemes.

Furthermore, being able to do addition and multiplication operations on encrypted data, the new LHE scheme is proposed. Our scheme works well as long as the noise size is less than the encryption key size. In our scheme, noise is embedded in each ciphertext for security reasons. The LHE scheme is really needed to allow resource-constrained devices like smartphones to efficiently leverage rich mobile applications provided by powerful computing resources and massive servers offered by CSPs. Our experimental results on the scheme have shown that its operability incurs a reasonable cost in terms of efficiency. Furthermore, the results have also shown that the scheme can achieve stronger security under the assumption that there is no collusion between mobile users and a CSP involved.

To summarise the above results, this chapter provides a summary and conclusion on our research findings together with future work in the subject area. It is organised as follows. We first present a summary of the thesis in Section 7.2. The future work is then investigated and proposed in Section 7.3. Finally, our concluding remarks are provided in Section 7.4.

7.2 THESIS SUMMARY

This sub-section provides a summary of the whole thesis and describes how our objectives have been achieved through the contributions of this project.

7.2.1 Overviews on the Previous Chapters

Here, we give an overview on previous five chapters. In Chapter 1 of this thesis, we have introduced the research area and presented the motivation behind the work carried out. The aim and objectives of the research have been discussed along with an overview of the solutions and the novelties of the work.

In Chapter 2, we have provided an overview of MSS and MCC in terms of its definition, benefits, limitations and existing solutions to the challenges. We have further discussed about the security challenges mainly on data security and integrity as well as users' privacy. We have described the potential solutions to overcome such challenges using the existing works. However, such solutions still have limitations and need to be enhanced. Therefore, we have provided our solutions to remedy such limitations and described in chapters 4 and 6.

In Chapter 3, the background on mathematical foundations has been presented. This chapter has defined some algebraic and number theory elements used throughout this thesis. Furthermore, the concepts of homomorphism and some related existing schemes have been introduced and discussed in this chapter. At the end of this chapter, we have detailed the research methodology taken for this research work.

In Chapter 4, our first solution to the problem by allowing privacy-preserving data aggregation to be executed securely and efficiently has been presented. Furthermore, we have shown that our proposed solution provides a salient feature, which is able to detect with certainty any misbehaving nodes in a privacy-preserving manner. Moreover, we have shown that our extended APDA features maximal value finding to be executed on ciphertext data without compromising a user's privacy. In addition, our extended APDA could be developed to compute other comparative operations in a privacy-preserving manner.

In Chapter 5, we have provided security analyses, comparison with existing solutions, results and discussions to demonstrate the merits of our proposed solutions. Based on those analyses have led us to claim that our proposed schemes are suitable for implementing in MSS.

In Chapter 6, our second solution to overcome the efficiency problems with MCC has been provided with some justifications on why our scheme is highly required in such an environment. Furthermore, we have described the design and implementation settings of our proposed scheme. At the end of this chapter, we have shown some results obtained from the conducted experiments and provided some discussions on the results.

7.2.2 A Comparison with Existing Researches

A verifiable privacy-preserving data aggregation has received a great attention as mobile data is shared and processed by an AS like a CSP for various purposes. Several existing schemes have allowed mobile data to be protected and verifiable in a privacy-preserving manner [23], [57], [70], [72], [191]. Nevertheless, all of them do not provide further steps for node detection. Thus, our main objective of proposing APDA is to provide a verifiable privacy-preserving data aggregation scheme that can support an accountability of mobile users for their behaviour in the aggregation process. Our APDA offers a salient feature, which is capable of detecting misbehaved nodes with certainty and efficiency. Although our scheme has an extra functionality, our scheme still can provide better efficiency compared to the existing schemes.

Furthermore, the main objective of having an aggregated result is to allow additive and non-additive functions like Count, Average, Min/Max and Percentile to be retrieved from such a result. Existing schemes that exploit homomorphism have allowed those functions to be computed in an encrypted form without any need for decryption [23], [42], [57], [67], [72]. However, such schemes still have limitations mainly in terms of their functionality and efficiency. Therefore, to overcome these limitations, we have extended our APDA to support maximal value finding in a privacy-preserving manner. Our simulation results show that the extended APDA provides better efficiency compared to the existing solutions.

Finally, homomorphic encryption schemes based on integers are getting more attention due to their simplicity and efficiency. These schemes reduce computing resources while encrypting and decrypting, and require less bandwidth for data transmission, so as to reduce battery and storage consumptions [35], [46], [74], [76]. However, allowing ciphertext data to be executed by arbitrary functions in an encrypted form is very

challenging tasks in order to achieve a balanced performance between efficiency and functionality. Existing schemes that support fully homomorphic properties suffers from efficiency issues. Thus, going for this challenge as one of the objectives has brought to the introduction of a new LHE scheme that supports homomorphic addition and multiplication on ciphertext data. From the experimental result, we have shown that our LHE provides better efficiency compared to the chosen counter scheme.

In summary, Table 19 provides an indication on whether each listed scheme delivers the objectives stated in Section 1.6 and listed in Table 20 for easy referencing.

Table 19: Research Objectives achieved by the Proposed and Existing Schemes

The Compared Schemes	Objectives of our Proposed Schemes				
	Objective 1	Objective 2	Objective 3	Objective 4	Objective 5
LHE				√	√
APDA	√	√	√	√	√
MAI [57]		√			
PPDM [25]	√				
VPA [23]	√	√		√	
PPSense [66]	√				
PLAM [70]	√	√			
PDA [67]	√			√	

Table 20: List of Research Objectives

Objective	Description
1	To develop a novel scheme, which can verify the integrity of aggregated data in MSS while preserving mobile users' privacy. This objective has been achieved by using an extensive literature survey on privacy-preserving data aggregation techniques to provide the proposed novel APDA, which is able to verify the integrity of aggregated data without disclosing any information related to mobile users to the CSPs as described in Chapters 2 and 4.
2	To propose an innovative scheme that can detect any misbehaved nodes in a data aggregation process in MSS without compromising individual nodes' data privacy. Such an objective has been achieved

	by using the proposed APDA with various versions to detect any misbehaved nodes in the aggregation process certainly and efficiently. We have evaluated those versions to confirm the fulfilment of this objective as described in Chapter 4.
3	To extend the functionality of the above proposed scheme to support additive and non-additive statistical functions over encrypted data certainly and efficiently. We have developed an extended APDA by introducing three versions of a maximal value function in Chapter 4 and evaluated those versions in Chapter 5 to fulfil this objective.
4	To design a new FHE scheme, which has a lightweight property applicable to resource-constrained devices in MCC. This objective has been achieved by the proposed new LHE scheme that supports both addition and multiplication in an encrypted form as described in Chapter 6. Furthermore, we have evaluated this scheme for its fulfilment of this objective and described it at the end of Chapter 6.
5	To design secure schemes without scarifying the privacy of mobile users during data outsourcing. This objective has been fulfilled by the extensive security analyses provided on the proposed schemes against several attacks launched by a curious AS or malicious devices as well as attacks on the schemes themselves such as key recovery attacks and a many time pad attack. We have shown in Chapter 5 and 6 that our schemes are secure against those attacks.

7.2.3 NOVEL CONTRIBUTIONS TO KNOWLEDGE

The research findings presented in this thesis offer significant contributions to the advance of HE, particularly in terms of its applicability to the privacy-preserving data aggregation in MSSs and its efficient operation in MCC. More specifically, the main novel contributions of our research work can be described as follows:

- In MSSs, data is sensed and aggregated by networked mobile devices, which could consist of both well-behaved and misbehaved nodes [58], [59]. Holding a

device accountable for its behaviour in such an environment is a very challenging task as data is encrypted to preserve the privacy of the users. Thus, the novel privacy-preserving data aggregation scheme APDA is proposed. This scheme allows the data to be aggregated in a privacy-preserving manner with an improved functionality that enables misbehaved nodes to be detected by an AS with certainty. APDA includes three sub-schemes with different benefits. $APDA^{W}$ is proposed to efficiently determine misbehaved nodes with less certainty. It offers lower computation and less communication as there is no digital signature included in ciphertext data. Conversely, $APDA^{S}$ is devised to pinpoint with certainty which nodes have misbehaved, but it has higher complexity and thus lower efficiency as the use of digital signatures is essential for holding misbehaved nodes accountable for their misbehaviour. To take advantage of the strengths of $APDA^{W}$ and $APDA^{S}$ as well as rectify their weaknesses, $APDA^{H}$ is introduced by mixing both versions to certainly spot misbehaved nodes with more balanced efficiency.

- In data aggregation, the purpose of having aggregated data is to generate additive and non-additive statistical results like Sum, Average, Count, Min/Max and Percentile. Thus, a number of researchers have proposed schemes to allow such functions to be executed on ciphertext data so as to preserve the users' privacy. Nevertheless, their limitations on efficiency and functionality are still the major challenges for their implementation [66], [67]. Therefore, the innovative extension of APDA is proposed to allow the AS to find the maximal value among those contributed by the n nodes certainly and efficiently. Such an extension could be developed further for other comparative operations, which is beyond the scope of this thesis. The extension includes three sub-schemes, Max^{C}, Max^{B} and Max^{H}, for finding a maximum value. Max^{C} is designed to accurately determine a maximum value without compromising any privacy and security of the data involved, but its efficiency is lower. On the other hand, Max^{B} is produced to efficiently determine a maximum value but have less certainty in terms of whether the maximum determined is the right one. To take advantage of both methods, Max^{H} is introduced to accurately identify a maximum value with more balanced efficiency.

- The existing FHE schemes cannot be directly implemented by resource-constrained devices as complexity and efficiency are the big obstacles for their implementation [37], [44]. Implementing such schemes requires extra computing resources to encrypt and decrypt the data, more bandwidth for transmitting the data and more storage spaces for storing the data [1], [11], [38], [46], [59]. Thus, the new LHE scheme with less complexity but strong security is proposed for MCC, so as to prolong the battery lifetime of mobile devices. This scheme allows mobile devices with limited computing resources and storage spaces to leverage rich mobile applications provided by CSPs in an efficient and secure manner.
- During the design of our APDA methods, we have proposed a BU only approach for misbehaviour node detection. However, by using such an approach, the AS needs to check all the nodes when the integrity checking has failed even though some of them behaved well. Therefore, such a procedure still demands high communication and computation costs. To remedy such a problem, we have proposed a new method, which mixes the BU and TD approaches to accelerate the process of misbehaved node detection with better efficiency. The reason is that fewer exponentiations are computed in the verification process as the well-behaved nodes are skipped from this check. Thus, the whole process requires less communication and computation.

7.3 THE PROPOSED SCHEMES APPLICABILITY

As described in the previous sub-section, we strongly believe that both MSS and MCC can be highly benefited by implementing our proposed schemes. This is due to our schemes offering effective and efficient solutions to remedy security issues inherent in both computing paradigms. Our schemes preserve user privacy as well as support user accountability without much resources degradation of mobile devices. In addition, our schemes guarantee confidentiality and integrity of the data generated by mobile devices against the external parties, which are responsible for processing the data.

In MSS, the widespread implementation of smart city demands that data generated by mobile devices like smartphones is shared and processed by external parties to produce some useful information that could benefit other mobile users. Air quality [10], [192], [193] and transportation [194] are two relevant examples of main applications in smart

cities that require instant accurate information from the crowd to generate some useful real-time results. Our scheme suits both applications embedded with security solutions to the data generated by mobile users.

In MCC, mobile users can access massive elastic resources and leverage rich-mobile applications provided by the cloud server in a very convenient way. For instance, file transfer via electronic mail using MCC proposed in [195] is a good implementation of MCC applications. Nevertheless, such files need to be filtered as they may contain malicious activity. Thus, our LHE is good to be implemented as it can filter all the files without disclosing the content of the files. Furthermore, our scheme can offer better efficiency compared to other methods such as a scheme proposed in [35] as transferring files online is resource consuming.

7.4 Future Directions

Although a number of novel contributions have been presented in this thesis, it is possible to incorporate other features, which will enhance our schemes' capabilities. In the following sub-sections, several possible future research directions stemming from this thesis are discussed.

7.4.1 Searchable Encryption

Our APDA scheme allows the maximal value to be retrieved by the AS in a privacy-preserving manner. Thus, similar methods could be developed for other comparative operations like finding a percentile, histogram and average on a given dataset. Furthermore, our LHE scheme allows homomorphic addition and multiplication to be executed in an encrypted form. In searchable encryption, more functions and operations are needed on data processing and retrieving such as data filtering based on keyword search, data sorting (e.g. ascending and descending order) and data grouping (e.g. sex, age and location). Thus, our schemes could be enhanced by adding more functionalities to allow those processes to be executed in a privacy-preserving manner.

7.4.2 Parent Nodes Selection

In MSSs, mobile nodes are embedded with sensors to sense surrounding data and aggregate such data to be processed by an AS. In the aggregation process, available parent

nodes need to be assigned prior to aggregate the data. However, as mobile nodes are mobile dynamically in and out of the coverage area of its children, the selection of parent nodes is crucial to ensure the suitable parent nodes to be selected at the first place of the aggregation process. As a result, the node mobility makes the selection of a parent node from the available ones hard. Thus, a suitable technique for determining a parent node from the available nodes in a MSS, e.g. the work in [196], can be used to allow the best available parent nodes to be chosen prior to aggregate the data.

7.4.3 PUBLIC KEY DISTRIBUTION

In APDA^S, the public keys need to be shared among the relevant nodes for signature verifications. Existing approaches have provided solutions to share the key securely [197]. For example, in a certificate-based public key Infrastructure (PKI), where the certificate authority, i.e. the AS in our study, can issue a certificate to each node for its public key. This certificate is then passed to other nodes for signature verification. Alternatively, an approach like Identity Based Encryption (IBE) can also be used to distribute the key by allowing the AS to create a pair of private and public keys for each node. In such a scheme, a user can encrypt using users' identities such as email addresses. Such an approach avoids the use of public key certification. Based on these two examples, it is essential to select a suitable technique to distribute the public key securely without degrading the performance of mobile nodes.

7.4.4 PHYSICAL THREATS

In MSS and MCC, a cloud server and mobile devices communicate via open communication devices such as Wi-Fi routers. Those devices may vulnerable to physical threats such as Denial of Service (DoS) and passive capturing [198] as they belong to the public and are beyond the control of mobile users. Nonetheless, as described in the threat model of MSS and MCC, we are not considering physical attacks as such attacks need different approaches for their solution and some of them can be found in [199]–[203].

7.4.5 FHE BASED ON INTEGERS

In our proposed scheme, we have improved the scheme efficiency by reducing the scheme complexity while achieving the strong security of the scheme. The proposed schemes

allow data to be processed in its ciphertext form. However, our schemes can only retrieve a correct result if and only if the noise generated from the process is less than the secret key. Thus, to allow such schemes to be implemented, the expected noise size needs to be estimated to ensure that a suitable secret key size is selected. Therefore, our schemes can be further improved so that it can handle arbitrary functions without the need to know the function to be executed in order to set the noise size. Our scheme can achieve fully homomorphic properties if it can handle the added noise in the decryption without any limitation and also improve its efficiency.

7.4.6 FUNCTIONAL FHE

A scheme, which enables some specific functions to be executed on ciphertext data without disclosing other information about the data, is really interesting. Such a scheme is known as Functional Encryption (FE) [204]. A FE scheme can be described as a scheme where the decryption key allows a user to learn a specific function of the encrypted data and nothing else [204]. Furthermore, [204] has also described that in a functional encryption system, a trusted authority holds a master key (m) known only to the authority. When the authority is given the description of some function f as input, it uses its master key to generate a derived secret key $sk[f]$ associated with f. Now anyone holding $sk[f]$ can compute $f(x)$ from an encryption of any x.

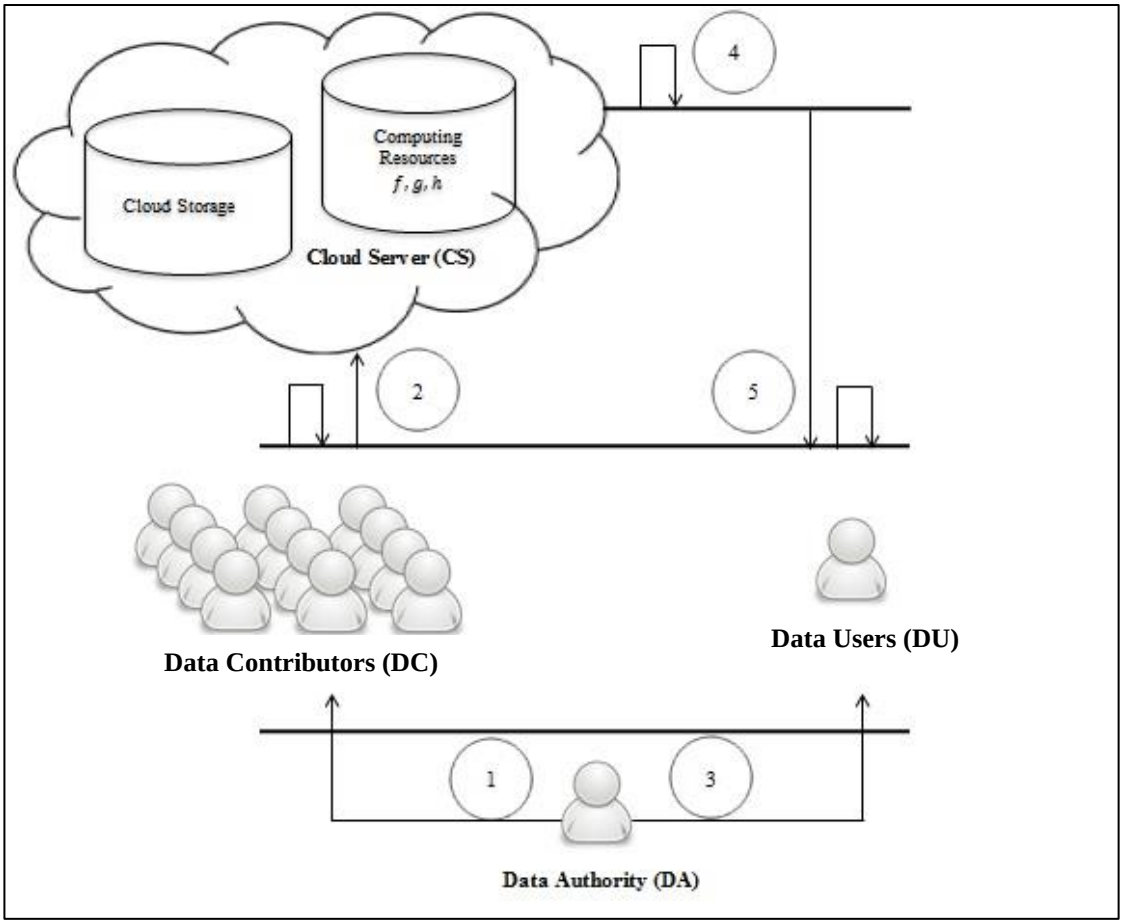


Figure 26: The proposed protocol with added Data Authority (DA)

An FHE scheme holds a promise to allow data to be processed in its ciphertext while the FE scheme allows only a specific ciphertext to be revealed to the users and keeps the rest as a secret. In cloud settings, the combination of FHE and FE schemes will provide more flexibility to the output of the processed data by CSPs. Furthermore, it also provides an improved security feature to the user's data. In such a combination, an additional party, i.e. a trusted authority (Data Authority (DA)), needs to be introduced as illustrated in Figure 26. In the figure, the DA generates a master key (m) and a secret key $sk[f]$ based on the requested function f by a Data User (DU). Then, the DA sends the $sk[f]$ to the DU so that the DU could decrypt the specific result based on the requested function f.

The Fully Functional Homomorphic Encryption (FFHE) scheme provides an improved version of FHE and FE to our setting as in such a scheme, the inherent security flaw in our proposed scheme can be dissolved. That is, the problem of possible collusion between a DU and a CSP could be solved by introducing functional encryption mixed with the

FHE scheme. Such a combination allows the DU only to receive specific information about the users' data based on the function requested.

7.5 Concluding Remarks

Recent technologies in mobile devices and cloud computing have brought a new and efficient way for data to be collected, processed and stored. Mobile devices like smartphones have evolved on their capabilities to leverage rich mobile applications provided by CSPs. Furthermore, such devices are not only being used as communication tools, but also their role has been extended to act as sensors to collect surrounding data mainly in urban areas. Local weather, crowd monitoring and road traffic are part of the surrounding data that can be collected by the devices to share with other users. On the one hand, such sharing can benefit local people so that they are able to better plan and manage their daily life efficiently and productively. On the other hand, the privacy of the data collected has to be protected and preserved. By introducing a technique that can protect and preserve mobile users' data, they can enjoy tremendous benefits provided by cloud computing and take full advantage of data aggregation techniques while improving data security and maintaining the battery lifetime of mobile devices.

Privacy-preserving data aggregation is an approach that allows users' privacy to be preserved against any curious CSPs in MSSs. Nevertheless, holding the users accountable for their behaviour using such a technique is a very challenging task as their data is encrypted. Thus, in this thesis, we have proposed a novel APDA to support the accountability of mobile users without compromising the users' privacy. To efficiently detect any misbehaved nodes in the aggregation process, APDA^{W} is composed of a weak version of accountability but high efficiency. Conversely, to certainly pinpoint the misbehaved nodes, APDA^{S} is devised with lower efficiency due to its higher algorithmic complexity. To take advantage of both versions, APDA^{H} is proposed to provide strong accountability with balanced efficiency, which is supported by our security analysis and experimental results.

The successfully aggregated data also needs to be accompanied by some summary information based on necessary additive and non-additive functions. Thus, our APDA has been extended to allow maximal value finding to be computed on the ciphertext data so

as to preserve user privacy with good efficiency. The extension includes three versions for maximal value finding with different levels of accuracy and efficiency. To confidently determine a maximal value, Max^C is proposed but its efficiency is lower. In contrast, to efficiently determine the maximum value, Max^B is designed but its accuracy is lower. Thus, Max^H is produced to combine both versions to balance their strengths and weaknesses for high accuracy and balanced efficiency. The efficiency measurements on each version have been conducted through our experiments. The results have shown that Max^H is the most balanced method to implement in a MSS.

To further leverage the tremendous benefits provided by CSPs, MCC is introduced. MCC is receiving great popularity due to the emerging technology of mobile devices together with the widespread 3G/4G networks and Wi-Fi access. MCC has brought rich mobile application experiences from CSPs to mobile users. In addition, FHE schemes have allowed mobile users to enjoy such experience from their mobile devices securely. Nevertheless, efficiency is still a big obstacle in their practical implementation mainly in mobile nodes, which have limitations on computing resources and storage spaces. These limitations are due to several reasons, which are mainly related to a large key size for encryption and the size of ciphertext data, which consumes a large amount of bandwidth for data transmission and requires huge storage spaces for data storing.

The aforementioned reasons have led us to propose a new LHE scheme for data to be outsourced and processed in its ciphertext form without decryption, so as to preserve the privacy of mobile users. We have simulated the related processes to measure the efficiency of our proposed scheme. Based on our comparison with the related work, we claim that our scheme provides better efficiency and can achieve stronger security at a reasonable cost.

As described in the thesis, our APDA method has its own novelties, mainly the scheme's efficiency while guaranteeing data integrity, user privacy and accountability as well as data security. On the one hand, our APDA is plausible to incorporate other comparative functions on encrypted data like finding Average, Percentile and Histogram. Furthermore, by using this method, it is good to have a suitable technique like the one in [196] to determine the best node to be a parent node for aggregating the data via a hierarchical structure in a network. This is due to the nature of a mobile node, which is dynamically

in and out of the coverage area of its children. Moreover, as the existing key management schemes are computational expensive, it is good to have an efficient algorithm for public key distribution to run on resource-constraint devices without degrading their performance.

On the other hand, our LHE scheme requires that the expected noise size be estimated to ensure the selection of a suitable secret key size. Therefore, it is good to have a scheme that can handle arbitrary functions without the need to know the function to be executed in order to set the noise size. In addition, the implementation of Functional Encryption (FE), which supports fully homomorphic properties, is desirable as such an encryption scheme could improve data privacy and processing efficiency. This is because implementing existing schemes leads to the decryption of all data or nothing. In contrast, the Fully Functional Homomorphic Encryption (FFHE) scheme allows arbitrary operations on a targeted result to be obtained by using a restricted secret key on the whole encrypted data. This reveals nothing about the rest of the encrypted data as the data remains in its ciphertext form.

REFERENCES

[1] W. Z. Khan, Y. Xiang, M. Y. Aalsalem, and Q. Arshad, "Mobile Phone Sensing Systems: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 1, pp. 402–427, 2013.

[2] K. Wibowo, "Mobile Security: Suggested Security Practices for Malware Threats," *Competition Forum*, vol. 14, no. 1, pp. 119–125, 2016.

[3] M. Kummer, P. Schulte, M. Kummer, and P. Schulte, "When Private Information Settles the Bill : Money and Privacy in Google ' s Market for Smartphone Applications," Germany, 16, 2016.

[4] H. Ba, W. Heinzelman, C. A. Janssen, and J. Shi, "Mobile computing -A green computing resource," *IEEE Wireless Communications and Networking Conference, WCNC*, pp. 4451–4456, 2013.

[5] M. B. Terefe, H. Lee, N. Heo, G. C. Fox, and S. Oh, "Energy-efficient multisite offloading policy using Markov decision process for mobile cloud computing," *Pervasive and Mobile Computing*, vol. 27, pp. 75–89, 2016.

[6] C.-M. Chen and B.-Y. Ann, "Efficiencies vs. importance-performance analysis for the leading smartphone brands of Apple, Samsung and HTC," *Total Quality Management & Business Excellence*, vol. 27, no. 3, pp. 227–249, 2016.

[7] X. Fan, J. Cao, and H. Mao, "A survey of mobile cloud computing," *ZTE Corporation*, vol. 16, no. 1, pp. 393–413, 2011.

[8] H. Qi and A. Gani, "Research on Mobile Cloud Computing : Review , Trend and Perspectives," in *2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP),* 2012, pp. 195–202.

[9] N. Fernando, S. W. Loke, and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 84–106, 2013.

[10] J. Dutta, S. Roy, J. Dutta, C. Chowdhury, S. Roy, A. I. Middya, and F. Gazi, "Towards Smart City : Sensing Air Quality in City based on Opportunistic Crowd-sensing," in *The 18th International Conference on Distributed Computing and Networking, ACM*, 2017, pp. 1–6.

[11] J. Fan, Q. Li, and G. Cao, "Privacy-Aware and Trustworthy Data Aggregation in Mobile Sensing," in *IEEE Conference on Communication and Network Security (CNS)*, 2015, pp. 31–39.

[12] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, R. a. Peterson, H. Lu, X. Zheng, M. Musolesi, K. Fodor, and G.-S. Ahn, "The Rise of People-Centric Sensing," *IEEE Internet Computing*, vol. 12, no. 4, pp. 12–21, Jul. 2008.

[13] A. Bourdena, C. Mavromoustakis, G. Mastorakis, J. Rodrigues, and C. Dobre, "A Context Sensitive Offloading Scheme for Mobile Cloud Computing Service," in *8th IEEE International Conference on Cloud Computing (CLOUD)*, 2015, pp. 869–876.

[14] L. Fangming, P. Shu, H. Jin, L. Ding, and J. Yu, "Mobile Cloud Computing Gearing Resource-poor Mobile devices with Powerful Clouds: Architectures, Science and Technology," *IEEE Wireless Communications*, vol. 20, no. 3, pp. 14–22, 2013.

[15] S. K. Madria, "Secure Data Aggregation and Collaboration in Wireless Sensor

Networks," in *2011 International Conference on Collaboration Technologies and Systems (CTS)*, 2011, pp. 273–273.
[16] J. Jose, J. Jose, and M. Princy, "A Survey on Privacy Preserving Data Aggregation Protocols for Wireless Sensor Networks," *Journal of Computing and Information Technology - CIT 22*, vol. 1, no. 1, pp. 1–20, 2014.
[17] Y. Liu and a B. T. Ab, "A Novel Trust-Based Secure Data Aggregation for Internet of Things," in *The 9th International Conference on Computer Science & Education*, 2014, pp. 435–439.
[18] S. Ben Othman, A. A. Bahattab, A. Trad, and H. Youssef, "Confidentiality and Integrity for Data Aggregation in WSN Using Homomorphic Encryption," *Wireless Personal Communications*, vol. 80, no. 2, pp. 867–889, 2014.
[19] F. Diao, F. Zhang, and X. Cheng, "A Privacy-Preserving Smart Metering Scheme Using Linkable Anonymous Credential," *IEEE Transactions on Smart Grid*, vol. 6, no. 1, pp. 461–467, 2015.
[20] M. Rezvani, A. Ignatovic, E. Bertino, and S. Jha, "Secure Data Aggregation Technique for Wireless Sensor Networks in the Presence of Collusion Attacks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 1–18, 2015.
[21] X. Jian, Y. Geng, C. Zhengyu, and W. Qianqian, "A Survey on the Privacy-Preserving Data Aggregation in Wireless Sensor Networks," *Communications, China*, pp. 162–180, 2015.
[22] X. Li, D. Chen, C. Li, and L. Wang, "Secure Data Aggregation with Fully Homomorphic Encryption in Large-Scale Wireless Sensor Networks," *Sensors*, vol. 15, no. 7, pp. 15952–15973, 2015.
[23] R. Zhang, J. Shi, Y. Zhang, and C. Zhang, "Verifiable Privacy-Preserving Aggregation in People-Centric Urban Sensing Systems," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 9, pp. 268–278, 2013.
[24] M. K. Sandhya, K. Murugan, and P. Devaraj, "Selection of aggregator nodes and elimination of false data in wireless sensor networks," *Wireless Networks*, vol. 21, no. 4, pp. 1327–1341, 2015.
[25] J. Zhou, Z.-F. Cao, X. Dong, and X. Lin, "PPDM: Privacy-preserving Protocol for Dynamic Medical Text Mining and Image Feature Extraction from Secure Data Aggregation in Cloud-assisted e-Healthcare Systems," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 7, pp. 1332–1344, 2015.
[26] V. Cardellini, V. De Nitto, V. Di, V. Francisco, V. Grassi, F. Lo, and V. Piccialli, "A game-theoretic approach to computation offloading in mobile cloud computing," *Mathematical Programming*, vol. 157, no. 2, pp. 421–449, 2016.
[27] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A Survey of Mobile Cloud Computing : Architecture , Applications , and Approaches," *Wireless Communications and Mobile Computing*, no. 13, pp. 1587–1611, 2013.
[28] R. Niu, W. Song, and Y. Liu, "An Energy-Efficient Multisite Offloading Algorithm for Mobile Devices," *International Journal of Distributed Sensor Networks*, vol. 2013, pp. 1–6, 2013.
[29] S. Kumar, S. Ramalingam, and R. Buyya, "An efficient and secure privacy-preserving approach for outsourced data of resource constrained mobile devices in cloud computing," *Journal of Network and Computer Applications*, vol. 64, pp. 12–22, 2016.
[30] Z. Xia, X. Wang, X. Sun, S. Member, and Q. Wang, "A Secure and Dynamic

Multi-keyword Ranked Search Scheme over Encrypted Cloud Data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 2, pp. 340–352, 2015.

[31] R. K. L. Ko, P. Jagadpramana, M. Mowbray, S. Pearson, M. Kirchberg, Q. Liang, and B. S. Lee, "TrustCloud: A Framework for Accountability and Trust in Cloud Computing," in *2011 IEEE World Congress on Services*, 2011, pp. 584–588.

[32] H. Hu, J. Xu, C. Ren, and B. Choi, "Processing Private Queries over Untrusted Data Cloud through Privacy Homomorphism," in *Proceedings of the 2011 IEEE 27th International Conference on Data Engineering*, 2011, pp. 601–612.

[33] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 1–11, Jan. 2011.

[34] M. R. Baharon, Q. Shi, D. Llewellyn-Jones, and M. Merabti, "Secure rendering process in cloud computing," *2013 11th Annual Conference on Privacy, Security and Trust, PST 2013*, pp. 82–87, 2013.

[35] C. P. Gupta and I. Sharma, "A fully homomorphic encryption scheme with symmetric keys with application to private data processing in clouds," in *The 4th International Conference on the Network of the Future, NoF 2013*, 2013, pp. 1–4.

[36] X. Zhang, H. Chen, K. Wang, H. Peng, Y. Fan, and D. Li, "Rotation-based Privacy-preserving Data Aggregation in Wireless Sensor Networks," in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 4184–4189.

[37] D. J. Wu, "Fully Homomorphic Encryption: Cryptography's holy grail," *XRDS: Crossroads, The ACM Magazine for Students*, vol. 21, no. 3, pp. 24–29, 2015.

[38] J. H. Cheon and J. Kim, "A Hybrid Scheme of Public-Key Encryption and Somewhat Homomorphic Encryption," *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 5, pp. 1052–1063, 2015.

[39] P. V.Parmar, S. B. Padhar, S. N. Patel, N. I. Bhatt, and R. H. Jhaveri, "Survey of Various Homomorphic Encryption algorithms and Schemes," *International Journal of Computer Applications*, vol. 91, no. 8, pp. 26–32, 2014.

[40] C. Zhi-gang, W. Jian, C. Liqun, and S. Xin-xia, "Review of how to construct a fully homomorphic encryption scheme," *International Journal of Security and its Applications*, vol. 8, no. 2, pp. 221–230, 2014.

[41] O. Zibouh, A. Dalli, and H. Drissi, "Cloud Computing Security Through Parallelizing Fully Homomorphic Encryption Applied to Multi-cloud Approach," *Journal of Theoretical and Applied Information Technology*, vol. 87, no. 2, pp. 300–307, 2016.

[42] Y. Yao, N. Xiong, J. H. Park, L. Ma, and J. Liu, "Privacy-preserving max/min query in two-tiered wireless sensor networks," *Computers & Mathematics with Applications*, vol. 65, no. 9, pp. 1318–1325, May 2013.

[43] W. Jia, H. Zhu, Z. Cao, L. Wei, and X. Lin, "SDSM : A Secure Data Service Mechanism in Mobile Cloud Computing," *The First International Workshop on Security in Computers, Networking and Communications*, pp. 1060–1065, 2011.

[44] J. Kim, M. S. Lee, A. Yun, and J. H. Cheon, "CRT-based Fully Homomorphic Encryption over the Integers," in *IACR Cryptology ePrint Archive 2013*, 2013, pp. 1–18.

[45] M. Albanese, S. Jajodia, R. Jhawar, and V. Piuri, "Privacy-Preserving Keyword Search Over Encrypted Data in Cloud Computing," in *Secure Cloud Computing*, 2014, pp. 189–212.

[46] H. Zhou and G. Wornell, “Efficient homomorphic encryption on integer vectors and its applications,” in *Information Theory and Applications Workshop (ITA)*, 2014, pp. 1–9.
[47] Y. Doröz, E. Öztürk, and B. Sunar, “A million-bit multiplier architecture for fully homomorphic encryption,” *Microprocessors and Microsystems*, vol. 38, no. 8, pp. 766–775, 2014.
[48] M. Tibouchi, “Batch Fully Homomorphic Encryption over the Integers,” in *Lecture Notes in Computer Science*, vol. 7881, 2013, pp. 315–355.
[49] C. Zhigang, W. Jian, Z. Zengnianb, and S. Xinxia, “A Fully Homomorphic Encryption Scheme with Better Key Size,” *China Communications*, vol. 11, no. 9, pp. 82–92, 2014.
[50] K. Kumar and Y.-H. Lu, “Cloud Computing for Mobile Users: Can Offloading Save Energy?,” *IEEE Computer*, vol. 43, no. 4, pp. 51–56, 2010.
[51] R. Gao, Y. Wen, H. Zhao, and Y. Meng, “Secure Data Aggregation in Wireless Multimedia Sensor Networks Based on Similarity Matching,” *International Journal of Distributed Sensor Networks*, vol. 2014, pp. 1–6, 2014.
[52] A. N. Khan, M. L. Mat Kiah, S. U. Khan, and S. a. Madani, “Towards secure mobile cloud computing: A survey,” *Future Generation Computer Systems*, vol. 29, no. 5, pp. 1278–1299, 2012.
[53] S. A. Z. Sanaei A. Gani, R. Buyya, “Heterogeneity in Mobile Cloud Computing:Taxonomy and Open Challenges,” *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2013.
[54] S. Subramaniyan, W. Johnson, and K. Subramaniyan, “A distributed framework for detecting selfish nodes in MANET using Record- and Trust-Based Detection (RTBD) technique,” *EURASIP Journal on Wireless Communications and Networking 2014*, vol. 2014, no. 1, pp. 205–215, 2014.
[55] W. Kozma Jr. and L. Lazos, “REAct: Resource-Efficient Accountability for Node Misbehavior in Ad Hoc Networks based on Random Audits,” in *Proceedings of the second ACM conference on Wireless network security*, 2009, pp. 103–110.
[56] A. Kapadia, D. Kotz, and N. Triandopoulos, “Opportunistic Sensing: Security Challenges for the New Paradigm,” in *First International Communication Systems and Networks and Workshops*, 2009, pp. 1–10.
[57] H. Li, K. Li, W. Qu, and I. Stojmenovic, “MAI: Secure and energy-efficient data aggregation with malicious aggregator identification in wireless sensor networks,” *Future Generation Computer Systems*, vol. 37, pp. 108–116, 2014.
[58] V. S. A. K. Umar and V. T. R. K. Umar, “Securing Data using Fully Homomorphic Encryption Schemes Over Cloud Computing,” *Compusoft*, vol. 3, no. 42, pp. 8526–8532, 2014.
[59] C. Cornelius, A. Kapadia, D. Kotz, D. Peebles, M. Shin, and N. Triandopoulos, “AnonySense : Privacy-Aware People-Centric Sensing Categories and Subject Descriptors,” in *Proceeding ACM 6th International Conference on Mobile Systems, Applications and Services (Mobisys)*, 2008, pp. 211–224.
[60] S. Singh and I. Chana, “Cloud Based Development Issues : A Methodical Analysis,” *International Journal of Cloud Computing and Services Science*, vol. 2, no. 1, pp. 73–84, 2013.
[61] M. Renuka and P. Thangaraj, “Multi-path Encrypted Data Security Architecture for Mobile Ad hoc Networks,” in *2011 National Conference on Innovations in Emerging Technology*, 2011, pp. 153–156.

[62] D. Dachman-soled, T. Malkin, and M. Raykova, “Secure Efficient Multiparty Computing of Multivariate Polynomials and Applications,” in *Applied Cryptography and Network Security: 9th International Conference, ACNS 2011*, 2011, pp. 130–146.
[63] N. J. Patel, “Detecting Packet Dropping Nodes using Machine Learning Techniques in Mobile Ad-hoc Network: A survey,” in *International Conference on Signal Processing And Communication Engineering Systems (SPACES)*, 2015, pp. 468–472.
[64] T. Dimitriou, “Secure and Scalable Aggregation in the Smart Grid,” in *6th International Conference on New Technologies, Mobility and Security (NTMS)*, 2014, pp. 1–5.
[65] A. Houmansadr, S. a. Zonouz, and R. Berthier, “A Cloud-based Intrusion Detection and Response System for Mobile Phones,” in *IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2011, pp. 31–32.
[66] Z. Wei, B. Zhao, Y. Liu, and J. Su, “PPSense: A novel Privacy-Preserving system in people-centric sensing networks,” in *8th International ICST Conference on Communications and Networking in China, CHINACOM 2013*, 2013, pp. 461–467.
[67] Z. Wei, B. Zhao, and J. Su, “PDA: A novel privacy-preserving robust data aggregation scheme in people-centric sensing system,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 11, pp. 1477–1550, 2013.
[68] D. Mashima and A. Roy, “Privacy Preserving Disclosure of Authenticated Energy Usage Data,” in *IEEE International Conference on Smart Grid Communications*, 2014, pp. 866–871.
[69] K. Ohara, “Privacy-Preserving Smart Metering with Verifiability for Both Billing and Energy Management,” in *Proceedings of the 2nd ACM workshop on ASIA public-key cryptography*, 2014, pp. 23–32.
[70] R. Lu, X. Lin, Z. Shi, and J. Shao, “PLAM: A privacy-preserving framework for local-area mobile social networks,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, 2014, pp. 763–771.
[71] S. M. Erfani, S. Karunasekera, C. Leckie, and U. Parampalli, “Privacy-preserving data aggregation in Participatory Sensing Networks,” in *IEEE Eighth International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 2013, pp. 165–170.
[72] S. Han, S. Zhao, Q. Li, C. Ju, and W. Zhou, “PPM-HDA : Privacy-preserving and Multifunctional Health Data Aggregation with Fault Tolerance for Cloud Assisted WBANs,” in *IEEE Transactions on Information Forensics and Security*, 2015, vol. 11, no. 9, pp. 1940–1955.
[73] S. Rana and P. S. Thilagam, “Hierarchical Homomorphic Encryption Based Privacy Preserving Distributed Association Rule Mining,” in *International Conference on Information Technology*, 2014, pp. 379–385.
[74] M. Tibouchi, “Scale-Invariant Fully Homomorphic Encryption over the Integers,” *Lecture Notes in Computer Science*, vol. 8383, pp. 311–328, 2014.
[75] J. H. Cheon, H. Hong, M. S. Lee, and H. Ryu, “The Polynomial Approximate Common Divisor Problem and its Application to the Fully Homomorphic Encryption,” *Information Sciences*, vol. 326, pp. 41–58, 2016.
[76] L. Xiao, O. Bastani, and I.-L. Yen, “An Efficient Homomorphic Encryption

Protocol for Multi-User Systems," *IACR Cryptology ePrint Archive 2012*, pp. 193–212, 2012.
[77] J. Shi, R. Zhang, Y. Liu, and Y. Zhang, "PriSense: Privacy-Preserving Data Aggregation in People-Centric Urban Sensing Systems," in *Proceedings IEEE INFOCOM*, 2010, pp. 1–9.
[78] A. T. Campbell, S. B. Eisenman, N. D. Lane, E. Miluzzo, and R. a. Peterson, "People-centric urban sensing," in *Proceedings of the 2nd annual international workshop on Wireless internet - WICON '06*, 2006, vol. 18, pp. 2--5.
[79] W. Guo, W. Hong, B. Zhang, Y. Chen, and N. Xiong, "Reliable Adaptive Data Aggregation Route Strategy for a Trade-off between Energy and Lifetime in WSNs," *Sensors*, vol. 14, no. 9, pp. 16972–16993, 2014.
[80] A. Bharti, "Comparative Study of Clustering based Routing Protocols for Wireless Sensor Network," *International Journal of Computer Applications*, vol. 66, no. 21, pp. 9–12, 2013.
[81] K. Du, J. Wu, and D. Zhou, "Chain-Based Protocols for Data Broadcasting and Gathering in the Sensor Networks," in *Proceedings International Parallel and Distributed Processing Symposium*, 2003, pp. 8–16.
[82] G. Ahn, M. Musolesi, H. Lu, R. Olfati-saber, and A. T. Campbell, "MetroTrack : Predictive Tracking of Mobile Events using Mobile Phones," in *Distributed Computing in Sensor Systems*, vol. 6131, Springer Berlin Heidelberg, 2010, pp. 230–243.
[83] E. Novak, "Challenges and Software Architecture for Fog Computing," *IEEE Internet Computing*, vol. 21, no. 2, pp. 44–53, 2017.
[84] I. Stojmenovic, S. Wen, X. Huang, and H. Luan, "An Overview of Fog Computing and its Security Issues," *Concurrency and Computation: Practice and Experience*, vol. 28, pp. 2991–3005, 2016.
[85] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing , Fog et al .: A survey and analysis of security threats and challenges," *Future Generation Computer Systems*, 2016.
[86] A. Botta, W. De Donato, V. Persico, and A. Pescap, "Integration of Cloud Computing and Internet of Things : a Survey," *Journal of Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.
[87] S. Yi, Z. Qin, and Q. Li, "Security and Privacy Issues of Fog Computing : A Survey," in *International Conference on Wireless Algorithms, Systems, and Applications*, 2015, pp. 685–695.
[88] S. Yi, C. Li, and Q. Li, "A Survey of Fog Computing : Concepts , Applications , and Issues," in *Proceedings of the Workshop on Mobile Big Data*, 2015, pp. 37–42.
[89] A. Ahmed and E. Ahmed, "A Survey on Mobile Edge Computing," in *10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, pp. 1–8.
[90] W. Shi and E. Computing, "The Promise of Edge Computing," *Computer*, vol. 49, no. 5, pp. 78–81, 2016.
[91] T. Taleb, S. Dutta, A. Ksentini, M. Iqbal, and H. Flinck, "Mobile Edge Computing Potential in Making Cities Smarter," *IEEE Communications Magazine*, vol. 55, no. 3, pp. 38–43, 2017.
[92] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative Mobile Edge Computing in 5G Networks : New Paradigms , Scenarios , and Challenges," *IEEE*

Communications Magazine, vol. 55, no. 4, pp. 54–61, 2017.
[93] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile Edge Computing : Progress and Challenges," in *4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, 2016, pp. 83–84.
[94] L. M. Vaquero, L. Rodero-merino, L. M. Vaquero, and L. Rodero-merino, "Finding your Way in the Fog : Towards a Comprehensive Definition of Fog Computing," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 5, pp. 27–32, 2014.
[95] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog Computing : Platform and Applications," in *Third IEEE Workshop on Hot Topics in Web Systems and Technologies*, 2015, pp. 73–78.
[96] R. P. Rudenko A., "Saving Portable Computer Battery Power through Remote Process Execution," *Mobile Computing and Communication Review*, vol. 2, no. I, pp. 19–26, 1998.
[97] A. Boldyreva, G. Tech, P. Grubbs, and S. Networks, "Making encryption work in the cloud," *Network Security*, vol. 2014, no. 10, pp. 8–10, 2014.
[98] M. R. Baharon, Q. Shi, and D. Llewellyn-jones, "A New Lightweight Homomorphic Encryption Scheme for Mobile Cloud Computing," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, pp. 618–625.
[99] W. Itani, A. Kayssi, and A. Chehab, "Energy-efficient incremental integrity for securing storage in mobile cloud computing," in *2010 International Conference on Energy Aware Computing, ICEAC 2010*, 2010, pp. 26–27.
[100] S. C. Hsueh, J. Y. Lin, and M. Y. Lin, "Secure cloud storage for convenient data archive of smart phones," *Discovery*, vol. 18, no. 51, pp. 35–40, 2011.
[101] Y. J. Song, K. Y. Park, and J. M. Kang, "The Method of Protecting Privacy Capable of Distributing and Storing of Data Efficiently for Cloud Computing Environment," in *The 1st International Conference on Computers, Networks, Systems and Industrial Engineering*, 2011, pp. 258–262.
[102] F. Rocha and M. Correia, "Lucy in the Sky without Diamonds : Stealing Confidential Data in the Cloud," in *2011 IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2011, pp. 129–134.
[103] Z. Mahmood, "Data Location and Security Issues in Cloud Computing," in *2011 International Conference on Emerging Intelligent Data and Web Technologies*, 2011, pp. 49–54.
[104] Y. Shen, W. Cui, Q. Li, and Y. Shi, "Hybrid Fragmentation to Preserve Data Privacy for SaaS," in *2011 Eighth Web Information Systems and Applications Conference*, 2011, pp. 3–6.
[105] C. Delettre, K. Boudaoud, M. Riveill, and L. I. Cnrs, "Cloud Computing , Security and Data Concealment," in *2013 IEEE Symposium on Computers and Communications (ISCC)*, 2013, pp. 424–431.
[106] S. Bajaj, "TrustedDB : A Trusted Hardware based," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 3, pp. 752–765, 2011.
[107] C. Gentry, "Computing arbitrary functions of encrypted data," *Communications of the ACM*, vol. 53, no. 3, pp. 97–105, 2010.
[108] R. Curtmola, "Searchable Symmetric Encryption : Improved Definitions and Efficient Constructions," in *Proceedings of the 13th ACM conference on Computer*

and communications security, 2006, pp. 79–88.
[109] H. S. Rhee, W. Susilo, and H.-J. Kim, "Secure searchable public key encryption scheme against keyword guessing attacks," *IEICE Electronics Express*, vol. 6, no. 5, pp. 237–243, 2009.
[110] M. Yoshino, K. Naganuma, and H. Satoh, "Symmetric Searchable Encryption for Database Applications," in *2011 14th International Conference on Network-Based Information Systems*, 2011, pp. 657–662.
[111] J. Bringer, H. Chabanne, and B. Kindarji, "Error-Tolerant Searchable Encryption," in *2009 IEEE International Conference on Communications*, 2009, pp. 1–6.
[112] N. Cao, C. Wang, M. Li, K. Ren, and W. Lou, "Privacy-preserving multi-keyword ranked search over encrypted cloud data," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 829–837.
[113] Y. Yang, "Towards Multi-User Private Keyword Search for Cloud Computing," in *2011 IEEE 4th International Conference on Cloud Computing Towards*, 2011, pp. 758–759.
[114] C. Wang, S. Member, N. Cao, K. Ren, S. Member, and W. Lou, "Enabling Secure and Efficient Ranked Keyword Search over Outsourced Cloud Data," in *IEEE Transaction on Parallel and Distributed Systems*, 2011, pp. 1–14.
[115] C. Wang and Q. Wang, "Toward Secure and Effective Data Utilization in Public Cloud," *IEEE Network*, vol. 26, no. 6, pp. 69–74, 2012.
[116] C. Liu, L. Zhu, L. Li, and Y. Tan, "Fuzzy keyword search on encrypted cloud storage data with small index," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 2011, pp. 269–273.
[117] H.-A. Park, J. H. Park, and D. H. Lee, "PKIS: practical keyword index search on cloud datacenter," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 64–80, 2011.
[118] M. Chuah and W. Hu, "Privacy-Aware BedTree Based Solution for Fuzzy Multi-keyword Search over Encrypted Data," in *2011 31st International Conference on Distributed Computing Systems Workshops*, 2011, pp. 273–281.
[119] C. Wang, N. Cao, J. Li, K. Ren, and W. Lou, "Secure Ranked Keyword Search over Encrypted Cloud Data," in *2010 IEEE 30th International Conference on Distributed Computing Systems*, 2010, pp. 253–262.
[120] S. Yu, C. Wang, K. Ren, and W. Lou, "Achieving Secure, Scalable, and Fine-grained Data Access Control in Cloud Computing," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.
[121] M. Li, S. Yu, N. Cao, and W. Lou, "Authorized Private Keyword Search over Encrypted Personal Health Records in Cloud Computing," in *31st International Conference on Distributed Computing Systems (ICDCS), 2011*, 2011, pp. 1–12.
[122] A. A. Almutairi, M. I. Sarfraz, S. M. Basalamah, W. G. Aref, and A. Ghafoor, "A Distributed Access Control Architecture for Cloud Computing," *IEEE Software*, vol. 29, no. 2, pp. 36–44, 2012.
[123] J. Naruchitparames and M. H. Gunes, "Enhancing data privacy and integrity in the cloud," in *2011 International Conference on High Performance Computing & Simulation*, 2011, pp. 427–434.
[124] A. Sahai, "Computing on Encrypted Data," in *Information Systems Security*, vol. 5352, Springer Berlin Heidelberg, 2008, pp. 148–153.
[125] M. Brenner, J. Wiebelitz, G. Von Voigt, and M. Smith, "Secret Program Execution in the Cloud Applying Homomorphic Encryption," in *5th IEEE International*

Conference on Digital Ecosystems and Technologies (IEEE DEST 2011), 2011, vol. 5, pp. 114–119.
[126] Y. Tian, J. Xie, S. Yin, J. Zhang, X. Qin, M. I. Alghamdi, M. Qiu, and Y. Yang, "A Secure File Allocation Algorithm for Heterogeneous Distributed Systems," in *2011 40th International Conference on Parallel Processing Workshops*, 2011, pp. 168–175.
[127] J. Zhang, X. Qin, and M. I. Alghamdi, "Secure Fragment Allocation in a Distributed Storage System with Heterogeneous Vulnerabilities," in *2011 Sixth IEEE International Conference on Networking, Architecture, and Storage*, 2011, pp. 170–179.
[128] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(Leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference on - ITCS '12*, 2012, pp. 309–325.
[129] M. Louk, "Homomorphic Encryption in Mobile Multi Cloud Computing," in *2015 International Conference on Information Networking (ICOIN)*, 2015, pp. 493–497.
[130] W. Stallings, *Cryptography and Network Security: Principle and Practice*, 5th ed. New York, USA: Prentice Hall, 2011.
[131] N. Smart, *Cryptography: An Introduction*, 3rd ed. London: McGraw-Hill, 2003.
[132] K. Rosen, *Discrete Mathematics and Its Applications*, 7th ed. NewYork: McGraw-Hill, 2011.
[133] M. R. Baharon, "Elliptic Curve Group Over Finit Field and Its Application," University Technology Malaysia, 2006.
[134] A. M. Richard, *An Introduction To Cryptography*, 2nd ed. USA: Chapman & Hall, 2008.
[135] S. Goluch, "The development of homomorphic cryptography," Vienna University of Technology, 2011.
[136] M. Fu and C. Wei, "Elliptic Curve Cryptosystem EIGamal Encryption and Transmission Scheme," in *2010 International Conference on Computer Application and System Modeling (ICCASM 2010)*, 2010, pp. 51–53.
[137] J. Katz, *Digital Signatures*. Springer US, 2010.
[138] O. Goldreich, L. A. Levin, and N. Nisan, *On Constructing 1-1 One-Way Functions*, vol. 6650. Springer Berlin Heidelberg, 2011.
[139] A. Menezes, P. van Oorschot, and S. Vanstone, "Chapter 01: Overview of Cryptography," in *Handbook of Applied Cryptography*, CRC Press, 1996, pp. 1–48.
[140] K. S. McCurley, "The Discrete Logarithm Problem," in *Proceeding of Symposia in Applied Mathematics*, 1990, vol. 42, pp. 49–74.
[141] M. Chenal and Q. Tang, "On Key Recovery Attacks against Existing Somewhat Homomorphic Encryption Schemes," in *Proceedings of LATINCRYPT 2014*, 2014, pp. 1–28.
[142] A. Kiayias, N. Leonardos, H. Lipmaa, K. Pavlyk, and Q. Tang, "Optimal Rate Private Information Retrieval from Homomorphic Encryption," in *Proceedings on Privacy Enhancing Technologies 2015*, 2015, pp. 222–243.
[143] P. S. Pisa, M. Abdalla, O. Carlos, and M. Bandeira, "Somewhat Homomorphic Encryption Scheme for Arithmetic Operations on Large Integers," in *Proceeding Global Information Infrastructure and Networking Symposium (GIIS)*, 2012, pp. 1–8.
[144] X. Yi, P. Russell, and B. Elisa, "Homomorphic Encryption," in *Homomorphic*

Encryption and Applications, 2014, pp. 27–47.
[145] C. Gentry, "A Fully Homomorphic Encryption Scheme," Ph.D. Dissertation, Stanford University, 2009.
[146] D. Boneh, "Evaluating 2-DNF Formulas on Ciphertexts," in *Second Theory of Cryptography Conference, TCC 2005, Cambridge Proceedings*, 2005, pp. 325–341.
[147] M. Van Dijk, C. Gentry, S. Halevi, and V. Vaikuntanathan, "Fully Homomorphic Encryption over the Integers," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2010, pp. 24–43.
[148] C. Gentry and S. Halevi, "Implementing Gentry ' s Fully-Homomorphic Encryption Scheme," in *30th Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2011, pp. 1–29.
[149] Z. Brakerski, "Fully Homomorphic Encryption without Modulus Switching from Classical GapSVP," in *Proceedings of the 32nd Annual Cryptology Conference on Advances in Cryptology*, 2012, vol. 7417, pp. 868–886.
[150] A. C.-F. Chan, "Symmetric-Key Homomorphic Encryption for Encrypted Data Processing," in *IEEE International Conference on Communications*, 2009, pp. 1–5.
[151] N. P. Smart and F. Vercauteren, "Fully Homomorphic Encryption with Relatively Small Key and Ciphertext Sizes," in *Proceedings of the 13th international conference on Practice and Theory in Public Key Cryptography*, 2010, pp. 420–443.
[152] C. Gentry and N. P. Smart, "Better Bootstrapping in Fully Homomorphic Encryption," in *Proceedings of the 15th international conference on Practice and Theory in Public Key Cryptography*, 2012, pp. 1–16.
[153] C. Gentry and S. Halevi, "Fully Homomorphic Encryption without Squashing Using Depth-3 Arithmetic Circuits," in *2011 IEEE 52nd Annual Symposium on Foundations of Computer Science*, 2011, pp. 107–109.
[154] D. J. Bernstein, N. Heninger, P. Lou, and L. Valenta, "Post-quantum RSA," in *Cryptology ePrint Archive, Report 2017/351*, 2017, pp. 1–20.
[155] D. Micciancio, "Lattice-based Cryptography," in *Post-Quantum Cryptography*, Springer Berlin Heidelberg, 2009, pp. 147–191.
[156] M. Naehrig, K. Lauter, and V. Vaikuntanathan, "Can homomorphic encryption be practical?," in *Proceedings of the 3rd ACM workshop on Cloud computing security workshop - CCSW '11*, 2011, pp. 113–124.
[157] C. Gentry and N. P. Smart, "Homomorphic Evaluation of the AES Circuit," in *Proceedings of the 32nd Annual Cryptology Conference on Advances in Cryptology*, 2012, vol. 7417, pp. 850–867.
[158] M. S. Lee, "On the sparse subset sum problem from Gentry-Halevi ' s implementation of fully homomorphic encryption," in *Cryptology ePrint Archive, Report 2011/567*, 2011, pp. 1–7.
[159] J. Fan and F. Vercauteren, "Somewhat Practical Fully Homomorphic Encryption," in *Proceedings of the 15th international conference on Practice and Theory in Public Key Cryptography*, 2012, pp. 1–16.
[160] Z. Brakerski and V. Vaikuntanathan, "Fully homomorphic encryption from ring-LWE and security for key dependent messages," in *Proceedings of the 31st annual conference on Advances in cryptology*, 2011, pp. 505–524.
[161] O. Regev, "The learning with errors problem," in *Proceedings of the Annual IEEE*

Conference on Computational Complexity, 2010, pp. 191–204.
[162] J. Sen, "Homomorphic Encryption: Theory & Applications," in *Theory and Practice of Cryptography and Network Security Protocols and Technologies*, Croatia: INTECH, 2013, pp. 1–32.
[163] Z. Zhang, "Revisiting Fully Homomorphic Encryption Schemes and Their Cryptographic Primitives," PhD Thesis, University of Wollongong, 2014.
[164] M. Li, S. Yu, N. Cao, and W. Lou, "Privacy-Preserving Distributed Profile Matching in Proximity-Based Mobile Social Networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2024–2033, 2013.
[165] J. H. Cheon, H. Hong, M. S. Lee, and H. Ryu, "The polynomial approximate common divisor problem and its application to the fully homomorphic encryption," *Information Sciences*, vol. 326, pp. 41–58, 2016.
[166] A. Mandal, D. Naccache, and M. Tibouchi, "Fully Homomorphic Encryption over the Integers with Shorter Public Keys," in *Proceedings of the 31st annual conference on Advances in Cryptology*, 2011, pp. 487–504.
[167] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," *Lecture Notes in Computer Science*, pp. 377–394, 2010.
[168] M. Sookhak, F. R. Yu, M. Khurram, and Y. Xiang, "Attribute-based data access control in mobile cloud computing : Taxonomy and open issues," *Future Generation Computer Systems*, 2016.
[169] Y. Zhang, J. Yan, and X. Fu, "Reservation-based Resource Scheduling and Code Partition in Mobile Cloud Computing," in *2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, 2016, pp. 1–6.
[170] L. Chen, R. Lu, and Z. Cao, "PDAFT: A privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer Networking and Applications*, pp. 1–11, 2014.
[171] R. L. Rivest, M. L. Dertouzos, and L. Adleman, "On data Banks and Privacy Homomorphisms," in *Foundation of Secure Computation*, 1978, pp. 169–177.
[172] Q. Zhou, G. Yang, and L. He, "An Efficient Secure Data Aggregation Based on Homomorphic Primitives in Wireless Sensor Networks," *International Journal of Distributed Sensor Networks*, vol. 2014, pp. 1–11, 2014.
[173] M. M. Groat, W. He, and S. Forrest, "KIPDA : k -Indistinguishable Privacy-preserving Data Aggregation in Wireless Sensor Networks," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 2024–2032.
[174] F. Borges, D. Demirel, L. Bock, J. Buchmann, and M. Muhlhauser, "A privacy-enhancing protocol that provides in-network data aggregation and verifiable smart meter billing," in *2014 IEEE Symposium on Computers and Communications (ISCC)*, 2014, pp. 1–6.
[175] G. Sagers, B. Hosack, R. J. Rowley, D. Twitchell, and R. Nagaraj, "Where's the security in WiFi? An argument for industry awareness," in *Proceedings of the Annual Hawaii International Conference on System Sciences*, 2015, pp. 5453–5461.
[176] G. Ateniese, "Verifiable encryption of digital signatures and applications," *ACM Transactions on Information and System Security*, vol. 7, no. 1, pp. 1–20, Feb. 2004.
[177] V. Lyubashevsky and T. Prest, "Efficient Identity-Based Encryption over NTRU Lattices," in *Advances in Cryptology – ASIACRYPT 2014*, Springer Berlin Heidelberg, 2014, pp. 22–41.

[178] M. Zhandry, “Secure Identity-Based Encryption in the Quantum Random Oracle Model,” in *Proceedings of the 32nd Annual Cryptology Conference on Advances in Cryptology*, 2012, vol. 7417, pp. 758–775.
[179] C. Jeong and Y. Kim, “Implementation of Efficient SHA-256 Hash Algorithm for Secure Vehicle Communication using FPGA,” in *2014 International SoC Design Conference (ISOCC)*, 2014, pp. 224–225.
[180] A. Abidi, B. Bouallegue, and F. Kahri, “Implementation of Elliptic Curve Digital Signature Algorithm (ECDSA),” in *2014 IEEE Global Summit on Computer & Information Technology (GSCIT)*, 2014, pp. 1–6.
[181] A. A. Ciss and A. Youssef, “A Factoring and Discrete Logarithm based Cryptosystem,” *International Journal Contemporary Mathematical Sciences*, vol. 8, no. 11, pp. 511–517, 2013.
[182] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J. A. Halderman, N. Heninger, D. Springall, E. Thomé, L. Valenta, B. Vandersloot, E. Wustrow, and S. Z. Paul, “Imperfect Forward Secrecy : How Diffie-Hellman Fails in Practice,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 2015, pp. 5–17.
[183] D. Johnson, A. Menezes, and S. Vanstone, “The Elliptic Curve Digital Signature Algorithm (ECDSA),” *International Journal of Information Security*, vol. 1, no. 1, pp. 36–63, 2001.
[184] A. Narayan, A. Feldman, and A. Haeberlen, “Verifiable Differential Privacy,” in *Proceedings of the Tenth European Conference on Computer Systems*, 2015, pp. 1–14.
[185] K. Nahrstedt, R. Campbell, E. Burger, J. Giffin, X. H. Gu, A. D. Joseph, E. Keller, D. Ma, and H. Weatherspoon, “Security for Cloud Computing,” in *A Report: Directorate for Computer and Information Science and Engineering (CISE)*, 2012, pp. 1–19.
[186] D. Zissis and D. Lekkas, “Addressing cloud computing security issues,” *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.
[187] F. Zhao, C. Li, Feng Zhao, Chao Li, and Chun Feng Liu, “A cloud computing security solution based on fully homomorphic encryption,” in *16th International Conference on Advanced Communication Technology*, 2014, pp. 485–488.
[188] J. Jiang, G. Han, F. Wang, L. Shu, and M. Guizani, “An Efficient Distributed Trust Model for Wireless Sensor Networks,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 5, pp. 1228–1237, 2015.
[189] D. E. Denning, “Digital Signatures with RSA and Other Public-Key Cryptosystems,” *Communications of the ACM*, vol. 27, no. 4, pp. 388–392, 1984.
[190] J. Huang, H. Kannan, Y. K. Xiao, and E. Zhang, “6 . 857 Final Project - One Time Pad,” 2015.
[191] I. Leontiadis and M. Li, “Collusion Resistant Aggregation from Convertible Tags,” in *IACR Cryptology ePrint Archive 2015*, 2015, pp. 1147–1168.
[192] J. Dutta, F. Gazi, S. Roy, and C. Chowdhury, “AirSense : Opportunistic Crowd-Sensing based Air Quality monitoring System for Smart City,” in *IEEE Sensor*, 2016, pp. 5–7.
[193] Y. Mehta, M. P. M. M, S. Mallissery, and S. Singh, “Cloud enabled Air Quality Detection , Analysis and Prediction - A Smart City Application for Smart Health,” in *3rd MEC International Conference on Big Data and Smart City*, 2016, pp. 1–7.
[194] Z. Xiao, H. Lim, and L. Ponnambalam, “Participatory Sensing for Smart Cities : A

Case Study on Transport Trip Quality Measurement," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 759–770, 2017.
[195] L. Chen, Y. Ho, W. Kuo, and M. Tsai, "Intelligent file transfer for smart handheld devices based on mobile cloud computing," *International Journal of Communication Systems*, vol. 30, no. 1, pp. 1–12, 2017.
[196] K. Kifayat, "Group-based Secure Communication for Wireless Sensor Networks," Ph.D. Thesis, Liverpool John Moores University, 2008.
[197] X. Li and Q. Wen, "A revocation scheme for the cloud computing environment," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 2011, pp. 254–258.
[198] B. Y. Zou, J. Zhu, X. Wang, and L. Hanzo, "A Survey on Wireless Security : Technical Challenges , Recent Advances , and Future Trends," *Proceeding of the IEEE*, vol. 104, no. 9, pp. 1727–1765, 2016.
[199] G. Ho, D. Leung, P. Mishra, and D. Song, "Smart Locks : Lessons for Securing Commodity Internet of Things Devices," 2016.
[200] J. Chen, "Threats to Networking Cloud and Edge Datacenters in the Internet of Things," *IEEE Cloud Computing*, vol. 3, no. 3, pp. 64–71, 2016.
[201] S. O. Damodaran and A. K. Haridas, "A Study on Jamming and Wormhole Attacks in Wireless Mesh Network," *International Journal of*, vol. 5, no. March, pp. 105–109, 2016.
[202] N. S. V Rao, S. W. Poole, C. Y. T. Ma, F. He, J. Zhuang, and D. K. Y. Yau, "Defense of Cyber Infrastructures Against Cyber-Physical Attacks Using Game-Theoretic Models," *Risk Analysis*, vol. 36, no. 4, pp. 694–710, 2016.
[203] K.Sumanth, G. Sridevi, S. Umar, K. K. Kumar, and M. A. Hussain, "A Proposal for Mitigation of Gray Hole Attack in Wireles Mesh Ad-Hoc Networking using S-DSDV," *Journal of Theoretical and Applied Information Technology*, vol. 84, no. 1, pp. 79–87, 2016.
[204] D. Boneh, "Functional Encryption : Definitions and Challenges," in *Proceeding TCC'11 Proceedings of the 8th conference on Theory of cryptography*, 2011, pp. 253–273.