2018

Privacy-Preserving Protocols for IEEE 802.11s-based Smart Grid Advanced Metering Infrastructure Networks

Samet Tonyali
stony002@fiu.edu

DOI: 10.25148/etd.FIDC004067

FLORIDA INTERNATIONAL UNIVERSITY

Miami, Florida

PRIVACY-PRESERVING PROTOCOLS FOR IEEE 802.11S-BASED SMART GRID ADVANCED METERING INFRASTRUCTURE NETWORKS

A dissertation submitted in partial fulfillment of the

requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ELECTRICAL & COMPUTER ENGINEERING

by

Samet Tonyali

2018

To: Dean John L. Volakis
College of Engineering and Computing

This dissertation, written by Samet Tonyali, and entitled Privacy-Preserving Protocols for IEEE 802.11s-based Smart Grid Advanced Metering Infrastructure Networks, having been approved in respect to style and intellectual content, is referred to you for judgment.

We have read this dissertation and recommend that it be approved.

A. Selcuk Uluagac

Osama A. Mohammed

Ahmed S. Ibrahim

Leonardo Bobadilla

Kemal Akkaya, Major Professor

Date of Defense: January 19, 2018

The dissertation of Samet Tonyali is approved.

Dean John L. Volakis
College of Engineering and Computing

Andres G. Gil
Vice President for Research and Economic Development
and Dean of the University Graduate School

Florida International University, 2018

DEDICATION

To my beloved father Yilmaz and my softhearted mother Fatma...

ACKNOWLEDGMENTS

I would like to thank and pay my gratitude to the members of my committee for their insightful comments, encouragement, and generous support. In addition, I would like to express my deepest gratitude to my major professor, Prof. Kemal Akkaya, for his valuable guidance and immense support for completing this research. He steered me in the right direction during my research and has always been there for help whenever I ran into troubles. His guidance helped me in all the time of research. I would like to thank my fellow lab mates for all the fun we have had in the last two and a half years. Thanks to them, I learned that the time is a precious asset that should not be wasted. My special sincere thanks from the bottom of my heart have to go to my parents for being patient of longing for me and their moral and any kind of material support. Finally, my wholeheartedly thanks go to my one-of-a-kind friend, Ugurtan Demirtas. His support, encouragement, and being there whenever I wanted to share my thoughts on anything made the past four years endurable.

I would also like to acknowledge the support by Department of Energy, National Science Foundation, and University Graduate School at Florida International University. This dissertation is mostly based upon the work supported by National Science Foundation under the grant number CNS-1550313. In addition, it partially depends on the work supported by Department of Energy under Award Number DE-OE0000779 and the Dissertation Year Fellowship granted by University Graduate School at Florida International University.

ABSTRACT OF THE DISSERTATION

PRIVACY-PRESERVING PROTOCOLS FOR IEEE 802.11S-BASED SMART GRID ADVANCED METERING INFRASTRUCTURE NETWORKS

by

Samet Tonyali

Florida International University, 2018

Miami, Florida

Professor Kemal Akkaya, Major Professor

The ongoing Smart Grid (SG) initiative proposes several modifications to the existing power grid in order to better manage power demands, reduce CO_2 emissions and ensure reliability through several new applications. One part of the SG initiative that is currently being implemented is the Advanced Metering Infrastructure (AMI) which provides two-way communication between the utility company and the consumers' smart meters (SMs).

The AMI can be built by using a wireless mesh network which enables multi-hop communication of SMs. The AMI network enables collection of fine-grained power consumption data at frequent intervals. Such a fine-grained level poses several privacy concerns for the consumers. Eavesdroppers can capture data packets and analyze them by means of load monitoring techniques to make inferences about household activities. To prevent this, in this dissertation, we proposed several privacy-preserving protocols for the IEEE 802.11s-based AMI network, which are based on data obfuscation, fully homomorphic encryption and secure multiparty computation. Simulation results have shown that the performance of the protocols degrades as the network grows. To overcome this problem, we presented a scalable simulation framework for the evaluation of IEEE 802.11s-based AMI applications. We proposed several modifications and parameter adjustments for the network protocols being used. In addition,

we integrated the Constrained Application Protocol (CoAP) into the protocol stack and proposed five novel retransmission timeout calculation functions for the CoAP in order to increase its reliability.

Upon work showing that there are inconsistencies between the simulator and a testbed, we built an IEEE 802.11s- and ZigBee-based AMI testbed and measured the performance of the proposed protocols under various conditions. The testbed is accessible to the educator and researchers for the experimentation.

Finally, we addressed the problem of updating SMs remotely to keep the AMI network up-to-date. To this end, we developed two secure and reliable multicast-over-broadcast protocols by making use of ciphertext-policy attribute based signcryption and random linear network coding.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the development of new communication technologies, the current Power Grid is going through a transformation to the Smart Grid (SG), which will enable two-way communications of data between the utility company (UC) and the consumers' smart meters (SMs) [1] to collect fine-grained power consumption data through the deployment of SMs and smart data collection techniques [2]. The ongoing SG initiative proposes several modifications to the existing power grid in order to better manage power demands, reduce CO_2 emissions and ensure reliability. This is particularly important at the distribution level of the grid where there is a need for more smart applications. Consequently, several new applications were implemented at this level [3, 4]. The advanced metering infrastructure (AMI) is one of these applications that enables utilities to collect, measure, and analyze energy consumption data at more frequent rates, thereby adjusting power demands and the prices accordingly (demand response and dynamic pricing).

The SG communication infrastructure involves a home area network of the electric consuming devices in the consumers residence, a neighborhood area network (NAN) of various devices including SMs which measure the power consumption, and a wide area network that connects the consumers to the utilities [5]. In this dissertation, we mostly focus on the NAN part of the AMI network and refer to this part when we use the term *the AMI network*.

The AMI network can be built by using power-line communication (PLC) and radio frequency (RF) technologies individually or together [6]. When compared to the PLC, RF solutions require far less cabling work, thereby lowering the infrastructure, deployment and maintenance costs. Considering that a typical AMI network consists of thousands of SMs, a wireless mesh network (WMN) is the ideal RF solution to

cover such a large-scale area, which can be supported by a variety of wireless radio technologies such as 802.16 (WiMAX), 802.11 (WiFi), and 802.15 (Bluetooth and ZigBee). A wireless mesh network (WMN) is a wireless communications network which enables any two nodes in the network to communicate with each other even if they are not in each others communication range [7, 8]. WMNs have numerous features such as self-organization, self-healing and multi-hop communication, which make them flexible and adaptable, e.g., the coverage area of the network can be extended by adding new nodes.

1.1 Data Privacy in the AMI Network

The AMI network enables collection of data at different periods. The frequency of the data collection depends on the application and the premise. The period can be in the order of seconds or minutes rather than daily [4]. However, such a fine-grained level poses several privacy concerns for the consumers [9, 10].

The IEEE 802.11s-based AMI network is a WMN in which each node relays data packets in order to enable multi-hop data communication [7,8]. Since the communication medium is air, eavesdroppers can capture data packets. The collected consumption data can be analyzed by means of load monitoring techniques to infer household activities and behavior patterns of the consumers [11, 12]. Obviously, this is against their privacy and can have social impacts. For example, a curious person can run simple signal capturing devices to know what his/her neighbors are doing. Similarly, at the commercial level, some companies may want to spy on their competitors. For example, smart grid traffic analysis can reveal how long a factory works, the number of workers present in the factory, etc. Revealing this information can cause financial losses, e.g., if a company knows that its competitors are producing too many products, it can work on reducing the price of this product by offering sales on their

products. Thus, any effective method of collecting and using fine-grained consumption information from SMs must provide sufficient protection of consumer privacy while preserving the suitability of the data for legitimate uses.

Due to such privacy concerns, preserving consumer privacy in SG attracted a lot of attention from the research community and several solutions were proposed to address different needs by making various assumptions on the available resources [10]. Despite such efforts, the privacy issue has been creating several problems in the deployment of SMs throughout the US and making the consumers reluctant to participate in SG programs [13]. This largely stems from the fact that all of these approaches at some point assume a trust relationship between the utility companies (UCs) and the consumers. The bottom line is that consumers may not even be comfortable with the UCs that have the right to access their data whenever needed.

While a number of studies addressed these problems through some theoretical approaches, their adoption in a realistic wireless mesh-based AMI has not been addressed. This dissertation aims to fill this gap by proposing various privacy-preserving protocols that can be readily employed in the IEEE 802.11s-based AMI network with comprehensive security and reliability features.

Under these circumstances, our major problem we addressed in this dissertation can be defined as follows: *Given the IEEE 802.11s-based AMI network, collect the consumers' power consumption data in a secure way such that the consumers' actual power consumption data cannot be obtained, and it is still possible to perform arithmetic operations on the concealed power consumption data for the AMI applications.* To this end, we proposed several privacy-preserving protocols based on data obfuscation [14] and data aggregation [15–17]. For privacy-preserving data aggregation, we employed the fully homomorphic encryption (FHE) [18–21] and secure multiparty computation (secure MPC) [22–24]. We assessed and compared the performance of

the proposed protocols under the widely used ns-3 simulator. During the performance evaluation, we observed that the performance of the protocols degrades as the network scales. To overcome the scalability issues, we came up with a scalable simulation framework for the AMI network.

1.2 A Scalable Simulation Framework for the AMI Network

The AMI network is built as a WMN, and WMNs suffer from scalability issues as the network grows. Therefore, in this dissertation, we present a scalable simulation framework for the evaluation of the IEEE 802.11s-based AMI applications. We proposed several modifications and parameter adjustments for the network protocols being used. Specifically, several parameters at the MAC layer were adjusted. Furthermore, we integrated a modified Address Resolution Protocol to take advantage of Hybrid Wireless Mesh Protocol's proactive route requests/replies, which is IEEE 802.11s standard's default routing protocol. Moreover, we introduced the Constrained Application Protocol (CoAP) to the application layer of the stack and proposed five novel retransmission timeout calculation functions for the CoAP in order to increase the reliability [25]. However, upon work showing that there are some inconsistencies between the simulator and a testbed [26, 27], we built an IEEE 802.11s- and ZigBee-based AMI testbed and measured their performance under security and privacy requirements of the AMI network.

1.3 Development of a ZigBee- and IEEE 802.11s-based AMI Network Testbed

There has been some efforts to assess the performance of the IEEE 802.11s-based AMI network that runs secure and privacy-preserving protocols [14,15,17,26–28]. However, to the best of our knowledge, there is not any comprehensive work in the literature that studies the performance of a ZigBee-based AMI network for privacy-preserving protocols. ZigBee has different features than IEEE 802.11s mesh and considered more lightweight that would consume less resources in terms of computation and communication. Therefore, in this dissertation, we compare the performance behavior of ZigBee- and IEEE 802.11s-based AMI network under different privacy-preserving protocols. In addition to standard encryption, we utilize the FHE- and secure MPC-based protocols. The performance comparison is specifically done under an actual testbed that was built at Florida International University Engineering Center. Note that we opt not to utilize simulation as it has been shown that the ns-3 simulation results for the AMI network do not match well with the testbed results [26, 27].

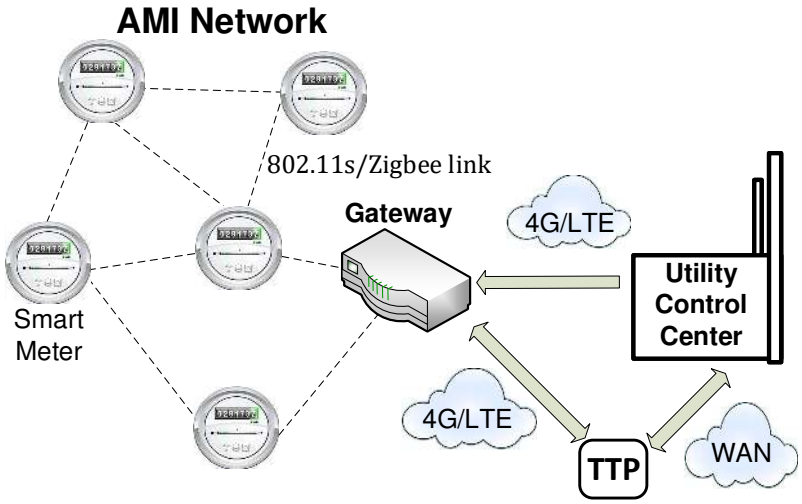


Figure 1.1: A sample AMI communication network, gateway and long-distance communication to a utility company.

We did not include the other technologies mentioned above in our work because WiMAX is not a commonly used technology, and having WiMAX-supported devices and maintaining their services are much more expensive compared to WiFi and ZigBee devices [29]. Also, Bluetooth Low Energy mesh has just been introduced in July, 2017 [30].

A typical infrastructure for the considered AMI is shown in Fig. 1.1. SMs can communicate their readings to the gateway of the network. The gateway reports the readings to the utility company directly or through a trusted third party (TTP) over any 4G/LTE network or the Internet.

After the implementation and deployment of the protocols, the state is changed to maintenance in the software development life cycle [31]. Based on the bugs reported and technological advancements, the protocols running on the installed SMs need to be updated. However, this is a challenging operation [32,33]. Therefore, we proposed two secure and reliable firmware update protocols for the AMI network.

1.4 Secure and Reliable Firmware Updates in the AMI Network

SMs can rule their physical components thanks to some programs called firmware. The firmware occasionally needs to be updated to fix bugs and improve the services. The SM firmware is proprietary, so the update file should be communicated to the SMs in a secure way. In addition, the firmware update can target a specific subgroup of the SMs rather than all of them in which case access control is required.

In this dissertation, we addressed the problem of multicasting the firmware update remotely in the IEEE 802.11s-based AMI network. Since it is not feasible to turn them off and update one-by-one manually, in order to enable remote updates, we

develop two secure and reliable multicast-over-broadcast protocols by making use of ciphertext-policy attribute-based signcryption (CP-ABSC) [34] to provide not only confidentiality and access control but also message authentication. The CP-ABSC signcrypts based on an access tree such that the signcrypted data can be designcrypted by those possessing the attributes that can satisfy the access tree.

The preliminary tests showed that increased size of the request due to signcryption reduces reliability of the protocol. Therefore, we employed random linear network coding [35] along with CP-ABSC in order to increase the reliability and use the bandwidth and processing resources efficiently.

1.5 Major Contributions

In this dissertation, we focused on developing secure and privacy-preserving protocols for the IEEE 802.11s-based AMI network. Specifically, we filled a gap between the theory and the practice by introducing privacy-preserving techniques into the AMI network context. In order to simulate a scalable and more realistic AMI network environment for the researchers and utility companies, we presented a scalable simulation framework. Although there are some inconsistencies between the simulator and testbed behaviors, the proposed changes and methods can be applied to the AMI network to solve the scalability issues. In addition to simulations, we analyzed the feasibility of the proposed privacy-preserving protocols by conducting experiments on an IEEE 802.11s- and ZigBee-based AMI testbed which is built at the Florida International University Engineering Center and made accessible to the educator and researchers at `https://amitestbed.fiu.edu/`. Finally, we developed secure and reliable firmware update protocols to keep the protocols up-to-date without any human intervention as new technologies are emerged, and new bugs are detected in the firmware.

1.6 Organization of the Dissertation

The rest of the dissertation is organized as follows. In the following chapter, we give a concise background information about the fundamental building blocks of this dissertation. It is followed by a comprehensive literature review on which every individual work in this dissertation depends. In Chapter 4, we investigate how to enable privacy-preserving state estimation in the AMI network via data obfuscation. In Chapter 5, we present the FHE- and secure MPC-based privacy-preserving protocols. In Chapter 6, we present a scalable simulation framework for the AMI network. We develop a ZigBee- and IEEE 802.11s-based AMI network testbed and compare their performance under security and privacy requirements of the AMI network in Chapter 7. It is followed by the secure and reliable firmware update protocols in Chapter 8. Finally, we conclude the dissertation and present some future work for further studies in Chapter 9.

CHAPTER 2

PRELIMINARIES

In this chapter, we present an introductory background information about the fundamental building blocks of this dissertation. More detailed information can be found in related chapters.

In the following section, we introduce the privacy-preserving techniques we used to solve the research problem defined in the previous chapter.

2.1 Privacy-preserving Techniques

In this dissertation, we develop several protocols to be used for preserving privacy on the IEEE 802.11s-based AMI network. Specifically, we use data obfuscation, fully homomorphic encryption (FHE) and secure multiparty computation (MPC) in these protocols.

2.1.1 Data Obfuscation

Data obfuscation techniques provide unique opportunities that can be exploited at the distribution level. They have the ability of protecting consumer privacy while also allowing the UC to perform state estimation, billing and dynamic pricing [36, 37].

State estimation is used to monitor the state of a power system (i.e., voltage magnitude and phase angle of every bus) in order to maintain reliable power supply. Recently, there is some interest to do state estimation in low-voltage distribution networks using meters and their instantaneous measurements (real power, reactive power and voltage magnitude) [37] in addition to the measurements collected from the distribution system substation [38].

Due to data collection from smart meters, privacy came to picture in state estimation in distribution networks. One possible solution to this issue is data obfuscation.

The idea of data obfuscation is to hide the actual energy usage by randomizing the fine-grained meter data. By perturbing the collected reading values in a linear space, the UC can still monitor the distribution network and calculate billing for given intervals.

2.1.2 Data Aggregation

Another method for concealing the consumers' private data is data aggregation. It relies on aggregation of the private data at intermediate meters on the path towards the gateway meter.

A minimum spanning tree of the network is found by the root node as illustrated in Fig. 2.1 and each node is informed about its parent node. Each node sends its data to its own parent node. The parent nodes aggregate the data they receive from their child nodes with their own data and send the aggregated data to their parent node. This procedure goes on until the root node.

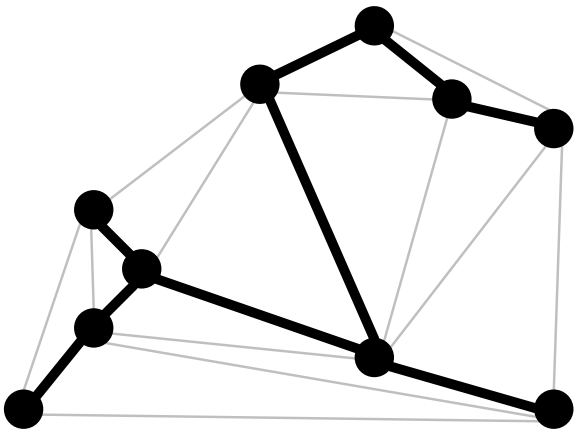


Figure 2.1: A minimum spanning tree of a mesh network shown as black links and nodes.

Data aggregation prevents the UC from accessing the consumers' individual power consumption data. However, since the medium is air, an eavesdropper can still capture and analyze the individual data. Encrypting the data before transmission can be a

solution. Moreover, if the traditional encryption methods (symmetric or asymmetric) are used one node's data is revealed to another node (the parent node). This problem can be overcome by employing FHE [18, 20, 21, 39–47] and secure MPC [23, 24, 48, 49] methods.

Fully Homomorphic Encryption

Homomorphic encryption enables to perform a set of operations on encrypted data without revealing the plaintext such that when the resulting ciphertext is decrypted, the decrypted value is equal to the resulting plaintext obtained when the same set of operations are performed on the plaintext.

Two typical operations in homomorphic encryption are addition and multiplication. We can define homomorphic encryption on addition and multiplication operations in a more formal way as:

Let m_1 and m_2 be two plaintexts.

$$D_{S_K}(E_{P_K}(m_1) \square E_{P_K}(m_2)) = m_1 \triangle m_2 \tag{2.1}$$

where $\triangle, \square \in \{+, \text{x}\}$ and D, E, P_K and S_K stand for decryption, encryption, public key and secret key, respectively.

The most commonly used homomorphic encryption method for data aggregation in the AMI network is Paillier cryptosystem. The Paillier cryptosystem is a partially homomorphic encryption (PHE) method, which means that it can perform only addition operation on encrypted data [50]. In addition, there is a newly emerged homomorphic encryption mechanism: fully homomorphic encryption (FHE).

The FHE enables to perform both addition and multiplication operations on encrypted data. The most important disadvantages of the FHE are size of keys and encrypted data, and very high computational delays [15, 28]. In this dissertation, we

adapt the FHE for preserving privacy on the IEEE 802.11s-based AMI network since the PHE is already very-well studied in the literature.

Secure Multiparty Computation

Secure multiparty computation (MPC) [22] is another method for privacy-preserving data aggregation. Secure MPC makes use of secret sharing to implement data aggregation. One of the most commonly used secret sharing scheme is Shamir Secret Sharing (SSS). The protocol we develop in this dissertation mostly relies on the SSS.

In the SSS, we assume that there are n nodes in the network and all computations are done in a finite field $\mathbb{Z}_p$, where p is a prime number. Let r_i be the private secret of node i. Node i chooses a unique point $x_i \in \mathbb{Z}_p$ other than zero and selects an $(n-1)$ degree random secret sharing polynomial $f_i(x)$ with $f_i(0) = r_i$. It sends its unique point x_i to all other nodes and receives share values $f_j(x_i)$ computed by the other $(n-1)$ nodes. Then, it computes $F(x_i) = \sum_{k=1}^{n} f_k(x_i)$. These steps are done by all n nodes and $F(x_i)$ values are sent to the gateway. The gateway can construct an $(n-1)$ degree polynomial $g(x)$ by using the $F(x_m)$ values along with Lagrange interpolation, where $m \in \{1, ..., n\}$. The constant term of $g(x)$ is the aggregation of all individual n private secrets.

2.2 Wireless Mesh Networking Standards

In this section, we introduce the technologies we used to build the AMI testbed.

2.2.1 WiFi Mesh Networking

Wireless Fidelity (WiFi) is the name of a technology for wireless local area networks, which is based on IEEE 802.11 standards. WiFi mesh network was developed on top

of IEEE 802.11s standard. IEEE 802.11s is a wireless local area network (WLAN) standard and an amendment to the IEEE 802.11 standard for mesh networking. It defines how wireless devices can communicate with each other to build a mesh network. IEEE 802.11s integrates mesh networking services and protocols with existing IEEE 802.11 protocols at the medium access control (MAC) layer.

2.2.2 ZigBee Mesh Networking

ZigBee is a WMN stack specification for personal area networks (PANs) which require less power and cost less when compared to LANs. ZigBee has different features than IEEE 802.11s, and it is considered more lightweight that would consume less resources in terms of computation and communication.

ZigBee is based on IEEE 802.15.4 MAC/PHY layer standard [51]. Although the standard does not define the mesh networking, ZigBee stack employs some routing protocols such as Ad hoc On-Demand Distance Vector (AODV) to implement mesh networking.

CHAPTER 3

LITERATURE REVIEW

In this chapter, we categorize and present the related work which we referred for each individual work in this dissertation.

3.1 Privacy Preservation in Smart Grid AMI Network

In general, the utility companies need fine-grained meter data for each customer to monitor demand and the state of the distribution network as well as utilizing dynamic pricing to reduce peak demand. Also, the utility companies would prefer to have the data to generate the bill on-site rather than relying on each individual smart meter (SM). Thus, the Smart Grid (SG) should allow the utility companies to collect and use this fine-grained data while protecting it from being used to monitor or profile an individual consumer's behavior.

Various approaches of providing consumer privacy are surveyed in [52]. This work categorizes these approaches into three groups: approaches that anonymize the fine-grained meter data, approaches that mask or obfuscate the individual consumption, and approaches that focus only on protecting communication of meter data from threats in the communication network. In practice, the third category does not provide any additional mechanism for ensuring privacy other than relying on the trustworthiness of the utility companies.

[53] is another study in which recently proposed SG privacy protection mechanisms are surveyed. Pros and cons of the methods are investigated in terms of their implementation complexity, efficiency, robustness and simplicity. In [54] a privacy-preserving, usage-based dynamic pricing scheme is proposed for SG, which employs Brakerski and Vaikuntanathan scheme which is a somewhat homomorphic encryption scheme. [55] proposes a privacy-preserving demand response scheme employing

Paillier cryptosystem to achieve privacy-preserving demand aggregation. [56] introduces human-factor-aware differential aggregation (HDA) attack which cannot be prevented by existing privacy-preserving aggregation protocols. Two novel protocols are proposed to resist HDA attack while achieving privacy-preserving metering data aggregation. [57] proposes an efficient privacy-preserving scheme, which employs identity-committable signatures and partially blind signatures, for incentive-based demand response programs in order to enable the demand response provider to compute individual demand curtailments in the SG.

Different from the above works, our privacy approach in this chapter is based on data obfuscation or hiding. One of these obfuscation approaches in [58] explains the necessary theoretical background on the ability of protecting consumer privacy while also allowing the utility company to perform state estimation, billing and dynamic pricing. However, this approach does not touch to the aroused problems during the deployment of this approach in a real environment.

One of the major issues when obfuscation is used is to be able to securely distribute the obfuscation values to each SM. In fact, we identified several new attacks that are possible when obfuscation values are to be distributed in a wireless AMI network. This problem by itself is an important networking problem since it requires new communication protocols to securely and reliably transmit these values. As a result, we introduce a secure distribution protocol from the gateway to the SMs as one of the major contributions of this work. Second, securing the communication will not be sufficient if the protocol suffers from intense interference and data traffic in a wireless environment, causing some of the packets to be dropped. Today's AMI network are typically wireless and rely on a multi-hop architecture whether it is based on WiFi or Zigbee standards. None of the works including [58] considers the impact of such a wireless multi-hop network on the performance of the distribution of both data

obfuscation values and data. Our work is in a sense a bridge between SG privacy research and wireless networking. Final contribution of this work is to introduce multiple gateways for increased security. When the single gateway is compromised, it is possible for the entire obfuscation vector to be captured. In multiple gateway version, the process of creating obfuscation values is split among the gateways.

3.2 State Estimation in Smart Grid Distribution System

SG should monitor the states of the grid to be able to take the most suitable action for the network, generations, and consumers [59]. Therefore, state estimation is crucial to utility centers for the reliability of SG functionality.

In power transmission networks, state estimation at the transmission level has been studied since 1970s [60]. At the the distribution level, there has been several studies in recent years. For instance, Dzafic et al. [61] presented a new approach for a three-phase distribution system state estimation (DSSE). The approach makes use of real time measurements such as current magnitude, real and reactive power measurements. The weighted least squares error method is used for estimation of the state variables. In another recent study for distribution state estimation, [62] investigate the use of SMs' power injection measurements for low-voltage system observability and controllability. Failure resistance tests demonstrated that the state estimation accuracy can be kept at a good level unless approximately 40% of all the SMs in the network are lost. In addition to this study, [63] develop a state estimator for low-voltage networks with and without distributed generations.

3.3 Data Aggregation in Smart Grid AMI Network

In addition to preserving the consumers' privacy, we utilize data aggregation to reduce packet traffic and consequently minimize the number of dropped packets in the network. Power consumption data from different meters are collected and aggregated at prespecified aggregator meters hierarchically. All collected data is aggregated at the gateway and the aggregated data is sent to the utility server. The aggregator meters perform the aggregation by using some arithmetic operations on the collected data before they are transmitted to the next aggregator meter.

In order to preserve consumer privacy, several works made use of homomorphic encryption and homomorphic arithmetic operations. For instance, Li et al. [64] used Paillier cryptosystem to provide in-network data aggregation while protecting user consumer privacy. The aggregation is performed at each level of a tree topology whereas the other applications perform the aggregation only at the gateway. Li and Luo [65] used homomorphic signatures for homomorphically encrypted data in order to make in-network data aggregation more robust to errors and internal/external attacks. Ruj and Nayak [66] proposed a decentralized security framework for data aggregation and access control in SGs. Consumers' private data are encrypted by using homomorphic encryption. In [67], the authors focused on finding the optimal placement for the data aggregation service, which minimizes the cost of in-network processing.

Contrary to these studies, Ambrosin et al. [68] discourage to perform aggregation on meter readings since it decreases the accuracy of the measured data. Instead, they proposed a secure protocol that achieves anonymous metering data delivery to a metering data management system (MDMS). Since the metering data report visits at least one other SM in the network, the MDMS cannot associate the report with a certain SM.

While these useful approaches considered different aspects of data aggregation, none of them studied the networking aspects such as reliability and delay. In particular, none of them considered the use of TCP in a multi-hop wireless environment such as the one in the AMI network when privacy is considered. Our approach would be complementary to these approaches as it will allow others to work under TCP especially if the data sizes are larger.

3.4 TCP Modifications for Smart Grid AMI Network

There are a number of works which investigated the TCP performance for SGs. For instance, the work in [69] proposes a scalable protocol that can handle both security and reliability using a TCP-friendly congestion control scheme. Due to similar motivations of the work in [69], Khalifa et al. [70] proposed a TCP-based scheme, which is called Split and Aggregated-TCP (SA-TCP). The scheme aggregates separate TCP connections to the utility server at SA-TCP aggregators and those incoming packets are forwarded over a single TCP connection between the SA-TCP aggregator and the utility server. This scheme has a different goal from ours. There is no in-network aggregation while in our work we utilize in-network data aggregation at intermediate nodes.

3.5 Homomorphic Systems

3.5.1 Partially Homomorphic Encryption

PHE has attracted most of the researchers' attention studying SG privacy preserving [44]. Among many PHE cryptosystems, Paillier [71] is widely proposed for data aggregation in SG thanks to its addition property, smaller message expansion factor

compared to others, and security features [10]. There are many SG privacy preserving aggregation applications based on Paillier [55, 64, 72].

Ozgur et al. [26,27] carried out an experimental study. They built an AMI network testbed comprised of Beaglebone Black boards and tested it with various parameters. End-to-End and Hop-by-Hop data aggregation applications were implemented on plaintext, Paillier and AES (Advanced Encryption System) encryption algorithms. ECDSA (Elliptic Curve Digital Signature Algorithm) and OpenSSL (Secure Sockets Layer) certificates were used for two-factor authentication. These aggregation mechanisms were run on top of TCP and UDP transport layer protocols. By varying these parameters, the aggregation mechanisms were tested both on the testbed and in the ns-3 network simulator, and their performance was compared.

Our work in this chapter is different than other relevant work since we consider encryption systems with the capability of supporting all arithmetic operations. Our goal is to investigate how the overhead in such systems compare to PHE in a realistic testbed using IEEE 802.11s-based mesh networks.

3.5.2 Fully Homomorphic Encryption

Gentry proposed the first FHE system using ideal lattices in 2009 [47]. While this was a great breakthrough for achieving FHE systems, the implementation of the proposed approach was still far from being a reality. This is because FHE generates large-size keys and ciphertexts when compared to other encryption schemes and the ciphertext at some point become too noisy due to bootstrapping-needed that it may not be decryptable at all. Therefore, since 2009 there have been a lot of efforts to build practical FHEs based on Gentry's work.

To this end, Smart and Vercauteren [40] presented an FHE scheme which had both relatively small key and ciphertext size. However, it lacked the implementation of

bootstrapping functionality. After a while, a faster FHE scheme was proposed in [46]. Besides these efforts, Gentry and Halevi [20] developed a working implementation of a variant of Gentry's FHE scheme. Despite such efforts, there was still not publicly available implementation of any FHE scheme until recently when Perl et al. [39] presented a working implementation of the Smart-Vercauteren scheme [40]. Brakerski et al. [45] later presented a new FHE scheme that dramatically improved performance, but based its security on weaker assumptions. This scheme did not need Gentry's bootstrapping procedure to evaluate arbitrary polynomial-size circuits.

While such implementations of FHE started to emerge, the adoption of such systems to be used in SG applications has yet to be investigated. So far, the only study that utilizes a somewhat FHE is about wide-area supervisory control and data acquisition (SCADA) security [73]. To the best of our knowledge, our work is the first to consider the feasibility and practicality of an FHE scheme for IoT-enabled Smart Metering systems.

3.6 Secure Multiparty Computation-based Protocols

There have been a few studies using secure MPC-based protocols to perform data aggregation in the AMI network. These protocols can be implemented with different cryptographic schemes in order to make data aggregation private and secure. For instance, Rottondi et al. [74–76] proposed a security architecture and a secure communication protocol for distributed aggregation of energy consumption metering data. A light variant of Cramer-Shoup cryptosystem and Shamir's secret sharing are used to provide security and privacy in data aggregation. Thoma et al. [77, 78] proposed a privacy preserving, secure MPC-based protocol along with Paillier cryptosystem for smart meter based load management and billing framework. The proposed system is able to conceal consumers' data and preserve its integrity without needing a

trusted third party. Yang et al. [79] analyzed the privacy risks of currently used smart metering techniques which collect fine-grained data in plaintext. They proposed a secure MPC-based solution as well as a data sanitization method which removes any identifying data that enables to associate the data with a certain consumer.

Our work differs from these studies in two aspects. The network topology used for the proposed systems is a kind of ring topology whereas we use mesh network in our work geared for the AMI applications. A ring does not apply to the AMI network. Also, they collect data once a day whereas in our application the meter data is collected in a more realistic fashion with much higher frequency that introduces additional overhead.

3.7 Reliability in Wireless Mesh Networks

Although the TCP-like protocols provide reliability, they introduce computational delay and message overhead. These factors degrade the performance of a protocol as the network scales. Instead, connectionless (without any handshake procedure) and lightweight protocols can be preferred. The CoAP is a very good example to this kind of protocols since it comes with a low header overhead and does not require any handshake procedure before sending a message.

There have been a few studies investigating the CoAP in SG systems. For example, Jung et al. [80] compare the performance of different web technologies including the CoAP for SM data exchange over the Internet. The protocols were tested on a testbed of two virtual machines (the service provider and consumer) with the ad-hoc and periodic (15 mins) meter data reporting scenarios. The test results show that the performance of the CoAP is comparable to the other technologies in both scenarios in terms of computational demand, throughput and financial cost.

Tanganelli et al. [81] set up a demonstration environment comprised of different 6LoWPANs (IPv6 over Low power Wireless Personal Area Networks) for smart metering and home automation systems in order to test the feasibility of the CoAP. In addition to smart metering, the CoAP can be used for other purposes in the AMI network. Kim et al. [82] proposed a lightweight SDN (Software Defined Networking) controller based on the CoAP for resource-constrained AMI devices. The virtual testbed and simulation results show that the proposed controller outperforms the traditional controller in terms of reliability, communication overhead and latency.

The major complain about the CoAP in the aforementioned studies is its default congestion control mechanism. The initial timeout value is randomly chosen between 2s and 3s, and it is simply doubled at each time the transmission timeouts. Since the maximum number of retransmissions is 4, they can be exhausted easily as the network scales [83]. Moreover, the CoAP considers all timeouts as congestion even if it is due to the lossy paths. In such a case, the timeout value is doubled and this causes the device to wait unnecessarily more than required at the next retransmission. Therefore, the researchers have proposed several alternatives to the CoAP's default congestion control mechanism.

Balandina et al. [84] proposed a new method for calculating the retransmission timeout (RTO) value in which a ratio between the estimated round-trip time (RTT) and the RTO value is used to update the RTO value. Betzler et al. [85] compared the CoAP's default congestion control mechanism with the Congestion Control/Advanced (CoCoA) [86] which provides an improved congestion control although it is more complex. The mechanisms were tested on the Cooja simulator [87]. The test results show that the CoCoA performs as good as the default mechanism at least in terms of throughput, and that it is able to reduce MAC layer buffer overflows in case the network is congested. In [88], they evaluated the performance of the CoCoA by using the

Californium framework [89] on the FlockLab testbed [90]. Thereafter, they improved the CoCoA (CoCoA+) [91] and conducted a comparative performance analysis of CoCoA and a variety of alternatives [92] including state-of-the-art TCP mechanisms. The test results show that CoCoA consistently outperforms its alternatives.

Bhalerao et al. [93] proposed a variant of CoCoA (CoCoA 4-State-Strong) which uses a 4-state estimator for variable backoff timings to improve the throughput performance. Järvinen et al. [94] evaluated the scalability of alternative congestion control mechanisms for the CoAP. They claim that all those alternative mechanisms are scalable although they tested them on a system with 80 clients at most.

In our work, we modify the CoAP to provide reliability for large-scale IEEE 802.11s-based AMI network. Instead of its default congestion control mechanism, we propose fixed, logarithmically increasing and exponentially increasing functions for RTO calculation.

3.8 Scalability in Wireless Mesh Networks

The network scalability can be defined as the ability of keeping the throughput at an acceptable level while increasing the coverage area by adding new nodes. There have been a few studies investigating the scalability in WMNs. Akyildiz et al. [8] attract the researchers' attention to the different layer protocols such as routing, transport, and MAC protocols for the reasons of the scalability issues in WMNs. Huang et al. [95] investigate scalable WMN for dense urban areas and scalable ring-based WMN for wide area coverage. They present the architecture of both WMN model and analyze them analytically. Then, they define the scalability issues as optimization problems and propose computation-based solutions. Also, they present some open issues about quality of service, cross-layer design and cooperative communications. Srivathsan et al. [96] investigate the scalability issues from both the hardware and software's

point of view. Different architectures for the WMNs are also discussed in their work. They investigate the use of directional antennas and the effect of the radios used. In software side, they highlight the importance of routing and MAC protocol design. Finally, they make the observation that none of the proposed approaches have been tested in a large-scale real network which is one of the several motivations of our work in this chapter.

Nassereddine et al. [97] evaluated the scalability of the HWMP based on the ns-2 [98] simulations. They used the Constant Bit Rate traffic type on top of UDP with the packet size of 512 bytes. The simulation results obtained in a 225-node topology show that the performance of the HWMP is highly sensitive to the number packets traveling throughout the network and the network size. Hence, the HWMP does not assure the scalability. This is the sole work in the literature that investigates the scalability issues in IEEE 802.11s standard.

Our work differs from the aforementioned studies in that we use a modified version of the ARP and HWMP [99–101]. In addition to the modifications, we adjust some of the parameters in the protocols running at different layers. The last but not the least is that we use 1024-node (32 x 32 grid) topology for our tests, and that all of the nodes send their message periodically and simultaneously. To the best of our knowledge, this is the first study that investigates a network simulator's stack to develop a scalable simulation framework for the AMI applications.

3.9 Wireless Mesh Network Testbeds

There have been several studies investigating the performance of IEEE 802.11- and 802.15.4-based testbeds in the literature. For example, Garroppo et al. [102] built an IEEE 802.11s-based WMN testbed consisting of four nodes to analyze the routing protocol HWMP, and they proposed some enhancements to improve its performance.

They tested their changes in a simulation environment with a larger network size, and showed that fine tuning the HWMP parameters is at least as significant as the metric computation. Singh et al. [103] built an IEEE 802.11s-based WMN testbed by using the personal and laptop computers in their lab and evaluated its throughput performance through UDP transmission protocols with varying data packet sizes. Our work differs from this work in that we build the testbed so as to cover a larger area with more nodes, and we test the privacy-preserving protocols running on top of both UDP and TCP. Also, we test the performance of ZigBee mesh network technology.

Imboden et al. [104] built a linear WMN testbed using IEEE 802.11s and IEEE 802.11n. They implemented the IEEE 802.11n-based WMN by creating virtual interfaces at the IP layer. They evaluated the throughput performance of TCP and UDP at both 2.4GHz and 5GHz frequency bands with different number of hops. The test results demonstrated that the increased number of hops significantly deteriorates the performance of IEEE 802.11n, and that network layer routing outperforms the link layer path selection. In our work, we build a larger network by using IEEE 802.11s and ZigBee standards and compare the performance of privacy-preserving protocols.

Takahashi et al. [105] built a WMN testbed of 22 nodes located in a rural area for disaster management and presented the details about its design, operation and management. They tested the link quality between the nodes in the mesh network and showed that the packet delivery ratio is around 80%. When compared to our testbed, they implemented a more sophisticated mesh access point consisting of an OpenBlockS, two Atheros wireless LAN cards and a wireless LAN access point along with a fan, timer and a temperature sensor whereas our nodes are comprised of more simplistic devices. Also, we conducted our experiments in a building in which there is almost all possible obstacles (e.g., walls, metal cabinets, workstations, personal

computers, etc.) that can exist and can affect the performance of the communication adversely in an urban area.

Ozgur et al. [26, 27] built an AMI network testbed and simulated the same testing environment in ns-3. They tested several privacy-preserving protocols along with two-factor authentication via Elliptic Curve Digital Signature Algorithm (ECDSA) signatures and SSL certificates. They developed a data collection application with different data collection modes and tested them on both TCP and UDP to show that if the testbed experiment results match the ns-3 experiment results. Test results demonstrated that there are major discrepancies among the two environments particularly in data collection completion time metric. Our work differs from these studies in the tested protocols, and that we tested the ZigBee-based AMI network testbed as well.

Zimmermann et al. [106] introduced a hybrid WMN testbed called *UMIC-Mesh* which is a compromise of real mesh network and a virtualization environment. The real hardware used in the hybrid testbed enables to transfer the experiment results to the real world deployments while the virtualization part makes it flexible to develop various network protocols. However, they neither performed any experiments on the proposed hybrid testbed nor compared its performance with that of a real world WMN.

IEEE 802.11s standard uses the HWMP as its default routing protocol. However, it is not the only protocol for mesh routing. Optimized Link State Routing (OLSR), Better Approach to Mobile Ad hoc Network (BATMAN) and BABEL are some other protocols that can be used for mesh routing. Song et al. [107] designed and implemented mesh routers and mesh clients. Then, they used these mesh routers and clients to develop an IEEE 802.11-based WMN testbed. The Optimized Link State Routing (OLSR) was selected as the multi-hop routing protocol. Abolhasan et

al. [108] investigated the performance of OLSR, BATMAN and BABEL by focusing on the multi-hopping performance and recovering from link failures. The test results showed that BATMAN and BABEL outperformed OLSR in all performance metrics examined. In another performance evaluation study based on a testbed, Hamidian et al. [109] built a WMN based on Microsoft Mesh Connectivity Layer in an office-wide area and analyzed its performance under multi-hop heterogeneous traffic. They concluded that even if the performance degrades as the network scales it is still promising because the network can meet the users' requirements if a good design that can limit the number of consecutive hops between the source and the destination can be created.

Eriksson et al. [110] deployed a WMN having 21 mesh nodes in an office. They investigated how different network designs affect the performance of user traffic. They observed that the delays incurred by different designs are negligible and concluded that all-wireless office meshes are feasible. Wu et al. [111] built a wide-area WMN testbed called QuRiNet on Quail Ridge Natural Reserve in Napa, CA. The network uses IEEE 802.11b/g as its underlying MAC protocol while using OLSR as its routing protocol. They tested several research projects such as channel assignment algorithms, rate control and routing relationship, evaluating the performance of queuing theories and bandwidth estimation on the testbed. They have provided a web site including a visualization and graphing tool, and an interactive map to get detailed information about the individual mesh nodes. In our work, we build IEEE 802.11s and ZigBee-based WMN testbeds with more simple and cheaper devices. Moreover, we test periodic data collection protocols under security and privacy requirements of the AMI network and assess their computational and communication overhead.

Furrer et al. [112] evaluated the performance of wireless communication technologies such as IEEE 802.15.4/ZigBee networks, Bluetooth WPANs, and IEEE 802.11b

WLANs in the IBM wireless sensor networking testbed. Several new and lightweight messaging applications such as remote metering, location sensing and mesh networking protocols were tested in terms of efficiency and scalability. However, the tested network consists of four devices, and the devices are located very close to each other. Also, our work differs from this one in the tested applications because we tested privacy-preserving protocols, which are comprised of computationally expensive operations that can push the resources to the limits.

Bansal and Sofat [113] deployed a WMN testbed at PEC University of Technology campus and evaluated its performance under heterogeneous traffic. They measured the SNR values to justify the location of the mesh points. Also, they tested the testbed performance by downloading a 4.9MB file by a client with varying number of hops. They did not provide any information about how they implemented multi-hop routing although they used IEEE 802.11g as the underlying MAC protocol as we did in this work. Casey et al. [114] built an IEEE 802.15.4-based testbed and tested it in four different environments. Test results indicated that IEEE 802.15.4 standard provides a reliable communication, and that the performance of the network heavily depends on the position of the transmitter and the receiver. In contrast, we built a ZigBee-based mesh network that runs on top of IEEE 802.15.4 standard.

Franceschinis et al. [115] compared the performances of ZigBee Pro and ZigBee IP stacks. They tested the ZigBee IP stack with both HTTP over TCP and CoAP over UDP. For the tests, they built two testbeds comprised of five nodes. ZigBee Pro nodes used AODV as the routing protocol while ZigBee IP nodes used RPL. The test results showed that ZigBee IP can outperform ZigBee Pro if CoAP is used. Abrignani et al. [116] built a ZigBee-based testbed and an equivalent simulation environment to show under which conditions the simulation can be considered reliable. The test results showed that the simulation can give a similar results that can be obtained in a

real environment in case of a specific range of data generation rates. The experiments were conducted in a restricted area, so the nodes are very close to each other. Our testbed deployments cover larger area and the tested privacy-preserving protocols generate more overhead in the network. Also, we compare the performance of ZigBee which is different than ZigBee IP and IEEE 802.11s on our AMI network testbed.

Unlike the aforementioned studies, we used two open-source wireless mesh networking technologies, IEEE 802.11s and ZigBee to build the AMI network testbed. We evaluated and compared their performance on FHE- and secure MPC-based privacy-preserving data aggregation protocols. Also, we implemented different data collection mechanisms. We investigated how the message overheads and the crypto-computational delays affect the performance of the network by testing them on a real testbed.

3.10 Firmware Update in the AMI Network

Updating the firmware of the SMs in an AMI network is one of the crucial operations triggered by out-of-network agents such as the UC and the SM vendor. It is crucial because a firmware update can remove security vulnerabilities as well as improve/enhance the SMs' functionalities. However, there have been a few studies investigating the firmware update process. The primary work about updating a SM firmware is the technical requirements document [117] in which the functional and security requirements of the firmware update process are defined. In addition, [118] presented a functional requirements specification for updating the firmware of a SM, remotely. Kim et al. [119] introduced firmware update management and network service management systems (FAN architecture) and describe the process of remote firmware update for the devices in the AMI network. Similarly, [120] described the remote firmware update process via the AMI network in a use case.

Although the process is a non-real-time and non-time-constrained operation [121], it still needs to be carried out in a secure manner in order to avoid downloading and installing a malicious firmware update. For this purpose, Katzir and Schwartzman [122] proposed a secure firmware update method for the devices connected to an alternating current network in order to avoid malicious firmware updates. In the method, a pre-defined pattern of changes in base frequency opens an update window in which the devices accept firmware update requests. Fadlullah et al. [123] investigated the applicability of KP-ABE in the AMI network and proposed a KP-ABE based broadcast mechanism which eliminates the need for issuing multiple unicast messages.

Contrary to the aforementioned studies, in our work, we propose complete firmware update protocols which are not only secure but also reliable. We utilize ciphertext-policy instead of key-policy in order not to renew the private key of the users for each firmware update with different access tree. Our protocols are based on CP-ABSC [34] which provides data integrity, message authentication and access control as well as confidentiality. Also, we introduce network coding to one of the protocols in order to use network bandwidth more efficiently. To the best of our knowledge, this is the first study that investigates the performance of a secure and reliable firmware update protocol employing network coding.

CHAPTER 4

PRIVACY-PRESERVING STATE ESTIMATION IN THE AMI NETWORK

The future SG is envisioned as a viable solution for finding efficient and economic methods of addressing a combination of several challenges: 1) using electricity more efficiently [124–126]; 2) reducing the impact of energy production on the environment; 3) integrating renewable energy generated by individuals; and 4) building the framework necessary for the use of electrical vehicles [4, 127, 128]. One part of the SG initiative that is currently being implemented is the AMI, which provides two-way communication between the utility company and consumers' "smart" meters (SMs) [1]. The utility companies can use this infrastructure to monitor power demands over short periods, provide more accurate billing as well as utilize dynamic pricing to facilitate the reduction of peak demand. Typically, two-way AMI communication is ensured via a wireless mesh network (WMN) which can be based on either IEEE 802.15.4 or IEEE 802.11s standards [4, 129].

Despite its potential, the implementation of the AMI has arisen concern of consumer privacy, since the fine-grained meter data being collected could be used to infer activities and behavior patterns of consumers [12]. The frequency of the data depends on the application and the premise and can be from 6 secs (for businesses) to 15 mins (for residential) as opposed to once a month in the existing grid [4]. This fine-grained data can reveal the types of activities going on in the house. Obviously, this is against their privacy and can have social impacts. For example, a curious person can run simple signal capturing devices to know what his/her neighbors are doing. Similarly, at the commercial level, some companies may need to spy on their competitors. For example, smart grid traffic analysis can reveal how long a factory works, the number of workers present in the factory, etc. Revealing this information

can cause financial losses, e.g., if a company knows that its competitors are producing too many products, it can work on reducing the price of this product by offering sales on their products. Thus, any effective method of collecting and using fine-grained consumption information from SMs must provide sufficient protection of consumer privacy while preserving the suitability of the data for legitimate uses.

Recently, there has been much research for addressing this privacy issue under different assumptions [52]. While some of the approaches focus solely on the confidentiality of the meter data during its transit, others additionally strive to hide it from utility companies by leaving the data handling to trusted third parties (TTP). In this way, privacy can be provided by only giving the utility company the chance to do monthly billing and thus no access to individual readings; however, the utility companies then cannot perform distribution state estimation of the power grid which includes collecting real-time state information from the grid including SMs. State estimation is indispensable for SG reliability, particularly to be able to take the most suitable action for the network, generations, and consumers [130]. Therefore, there is a need to be able to perform both privacy-preservation and state estimation at the same time.

To achieve both functions, data obfuscation techniques provide unique opportunities that can be exploited at the distribution level [58]. This work aims to realize this unique opportunity by bringing together state estimation, privacy and wireless data collection tasks in a realistic setup. The idea of data obfuscation is to hide the actual energy usage by randomizing the fine-grained meter data. By perturbing the collected reading values in a linear space, the utility company can still monitor the distribution network and calculate billing for given intervals. Nonetheless, the obfuscation operation necessitates the distribution of obfuscation values to each of the SMs in a secure and efficient way. Such distribution of values within the AMI network is

crucial in order to ensure the plausibility of the privacy in addition to providing classical security services such as authentication and integrity. Another major missing component in the existing privacy approaches is the feasibility and scalability of the implementation in a realistic network by considering different parameters regarding the network architecture. This chapter addresses these two issues in a comprehensive manner.

However, the proposed obfuscation value distribution approach with a single gateway poses some issues regarding security and efficiency. First of all, a single gateway will become a bottleneck when IEEE 802.11s is used with increased node count. This is because of the increased hop counts, congestion and interference in the network. Previous studies suggest using clustering within the AMI network and adjust the size of each cluster based on network conditions [131]. Therefore, it is preferable to use multiple gateways for scalability concerns.

Another motivation to use multiple gateways is due to the security of the distribution of obfuscation values. If a single gateway is being used, it is possible for all obfuscation vector to be captured when the gateway is compromised. Splitting the process of generating the obfuscation vector among multiple gateways could be used to reduce this vulnerability.

The solution proposed in this work is to separate the vectors among multiple gateways. Each gateway selects the weights for its given vectors and creates one part of the obfuscation vector. Since each gateway is responsible for the SMs in a portion of the grid, each gateway sends to the other gateways the elements of its partial obfuscation vector that are associated to the meters for which these gateways are responsible. Since each gateway only has a portion of the basis from which the full obfuscation vector is derived, an adversary compromising one gateway would be limited in the number of actual readings it could acquire. However, a compromised

gateway can affect the distribution of obfuscation values to other gateways and hence able to modify the final obfuscated meter readings. Since our main motivation in this work is preserving privacy of meter readings and providing efficient communication with the gateway, we have not addressed data falsification/injection issues. Such an attempt by a compromised meter, for instance, can be detected and eliminated by integrating an anomaly detection based mechanism developed for the AMI network [65].

Specifically, by assuming an 802.11s-based WMN for the AMI, we propose secure obfuscation value distribution approaches in order to implement data obfuscation in an efficient manner based on a number of security goals identified. We then implement this distribution approach and simulate obfuscated data traffic on ns-3 [132] by using a draft version of the 802.11s implementation. Our goal is also to assess the overhead it brings to the network and compare the overhead to that of a regular 802.11s network which does not provide privacy preservation. The simulation results revealed that such use of obfuscation does not bring any overhead in terms of packet losses or packet delay compared to other approaches. With the ability to perform state estimation and providing consumer privacy, the approach is feasible to be used in the 802.11s-based AMI network.

Following the ability of SMs to contribute to the state estimation computations, in this work, we also target distribution system state estimation where SM readings are used. However, since the use of SMs would introduce privacy concerns, we focus on the problem of privacy preserving state estimation which has not been studied before in power grid state estimation research.

The rest of the chapter is organized as follows. In the next section, we provide some background on AMI network, state estimation in power grids, and describe the assumptions, security goals and problem. In Section 4.2, we present our approach for

obfuscation value distribution and obfuscated reading collection in a secure manner. Section 4.3 introduces multiple gateways approach to the same problem. Section 4.4 is dedicated to security analysis and experiment results. Finally, we conclude the chapter in Section 4.5.

4.1 Preliminaries

4.1.1 Underlying AMI Network

We assume that the AMI communication network consists of SMs that are connected via a WMN with a gateway serving as a relay between the SMs. SM measures mainly the real-time electrical energy consumption of the customers in addition to power quality and instantaneous values such as voltage and current at their connection points. In the current AMI systems, this data is either collected by the utility company control center or by a trusted third party (TTP). In this work, we assume the existence of a TTP as well since handling every type of processing in a central control center is not a scalable option [58]. A typical infrastructure for the considered AMI in this chapter is shown in Fig. 1.1.

The mesh network is created using the new IEEE 802.11s standard which allows mesh networking among the SMs through 802.11 MAC/PHY layer standard [133] [134]. All nodes in 802.11s WMN are considered as Mesh Points (MP) and are able to provide connectivity at the data link layer between other MPs. If an MP also provides connectivity to the Internet, it is termed a Mesh Portal Point (MPP). In our mesh network, the gateway is the MPP which collects meter readings that are obfuscated at the MPs (i.e., SMs) via multi-hop routes. IEEE 802.11s standard provides a routing protocol called Hybrid Wireless Routing Protocol (HWMP) as its default routing protocol to find a multi-hop path towards the destination.

TTP collects data from the gateway and stores this data for future access but also forwards collected data in form of a data vector (e.g., an array) to the utility. In addition, TTP is responsible for monthly billing computations. The utility is responsible for creating and transmitting obfuscation base information for privacy preservation purposes to the gateway which in turn creates the individual obfuscation values and distributes to the SMs. We assume that the gateway communicates with the TTP and utility via a long distance communication (e.g., LTE). However, the utility and TTP may have other means for communication.

For security and privacy, we assume the availability of public key schemes since symmetric key systems require a lot of overhead in terms of key management. Each SM is initialized with a public/private key based on elliptic curve cryptography (ECC). This was chosen since its overhead is the minimal in comparison to other public cryptography schemes. ECC also uses a key size comparable to current symmetric cryptographic schemes, avoiding the higher computation of other public key schemes due to the larger key size. The gateway knows the public key of every SM in its mesh network. Every SM knows the public key of the gateway.

4.1.2 Weighted Least Squares State Estimation

A power system consists of a collection of buses, transmission lines and power meters. State estimation is used to monitor the state of a power system (i.e., voltage magnitude and phase angle of every bus) in order to maintain reliable power supply. Recently, there is some interest to do state estimation in low-voltage distribution networks using meters and their instantaneous measurements (real power, reactive power and voltage magnitude) in addition to the measurements collected from the distribution system substation [38]. One of the techniques for this state estimation process is called common weighted least squares (WLS) state estimation.

In this technique, the state of the network is estimated as a vector of variables $x = (x_1, \ldots, x_n)^T$ using $z = (z_1, \ldots, z_m)^T$ consisting of measurements from the power meters, where n, m are positive integers such that $m > n$ and $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^m$. Then, the state of the system is represented by:

$$z = h(x) + e \tag{4.1}$$

where $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ represents nonlinear dynamics such as the configuration of transformers and buses in the grid and $e \in \mathbb{R}^m$ is measurement errors and unmodeled dynamics. The state x is estimated to be $\hat{x}$ by the following unbiased linear estimation:

$$\hat{x} = (H^T W H)^{-1} H^T W z \tag{4.2}$$

where W^{-1} is the covariance matrix of e.

4.1.3 Privacy-Preserving State Estimation

Due to data collection from SMs, privacy came to picture in state estimation in distribution networks. One possible solution to this issue is to perturb the collected SM data. To this end, the authors in [58] create a *distortion free obfuscation space* from the span of a basis set $\mathbb{O} = \{o_1, \ldots, o_{m-n}\}$ of *kernel* denoted as $ker((H^T W H)^{-1} H^T W)$. Each $o_i \in \mathbb{O}$ is a vector with m elements that will perturb the SM values. Note that there are $m - n$ such vectors that can be used for perturbation. The authors also create an *obfuscated measurement vector* named z_{obf}, where $z_{obf} = z + o$, $o \in \mathbb{O}$. They show that z_{obf} can be used in place of z to calculate the same estimated state $\hat{x}$. In this way, without having access to actual power readings, the state of the power grid can be estimated.

$\mathbb{O}$ is derived from the state of distribution of the grid such as the configuration of the transformers stepping up or down voltages or buses branching off to multiple

distribution lines. It can be sent by the utility company only when any one of these dynamics changes. It can be reused for multiple readings until new ones are provided.

In order to create an obfuscation vector o, some random η weight values need to be generated. At measurement time t_j, the goal is to choose a random weight $\eta_i^j \in \mathbb{R}$ for a vector $o_i \in span(\mathbb{O}) = \{\eta_1^j o_1 + \eta_2^j o_2 + \eta_3^j o_3 + \ldots + \eta_{m-n}^j o_{m-n}\}$. The values of the weights η_i^j at each data collection time t_j in a billing period T are chosen so that the sum of the values of η_i^j in T equals to 0 (i.e., $\sum_{j \in T} \eta_i^j = 0$).

After the above computation is done, an element from the vector o is sent to the corresponding SM. Each SM adds this element to its actual power measurement in order to conceal it and to preserve the privacy by this way.

4.1.4 Problem Definition

Our problem can be defined as follows: *Given an 802.11s-based WMN, our primary goal is to distribute the obfuscation values to the SMs and collect the obfuscated values from them in a secure and efficient way via TTP. Our secondary goal is to assess the overhead of this process in a large scale AMI network and thus analyze the feasibility of the approach for future SG applications.*

4.1.5 Threat Model and Security Goals

We identified the following attacks to the privacy and security of the collection of fine-grained meter data in the AMI and established the associated security goals. They are organized into two sets: those targeting the consumer and those targeting the utility company. The first set relates to the privacy of a consumer's fine-grained meter data:

Attack 1: The utility company misuses fine-grained meter data it obtains to analyze

consumer behavior or shares the data with a third party for this purpose.

Security Goal 1: Obfuscate the collected fine-grained meter data to protect it from misuse by the utility company or related third party.

Attack 2: An eavesdropper monitors the communication channel to capture meter data in messages between a targeted SM and the gateway to determine the behavior of its consumer.

Security Goal 2: Protect communications containing SM readings.

Attack 3: An eavesdropper compromises a gateway to gather the obfuscation basis $\mathbb{O}$ that is stored to re-generate actual meter readings.

Security Goal 3: Limit the amount of obfuscation data that could be obtained if a gateway is compromised.

The second set of attacks relates to accurate state estimation and billing:

Attack 4: An attacker impersonates the gateway and sends fabricated obfuscation values to the SMs to change the state of the power grid.

Security Goal 4: Provide sender authentication to verify the sender and contents of messages.

Attack 5: An attacker captures the obfuscation values and replay them to change the state or billing.

Security Goal 5: Identify and discard replayed messages.

4.2 Data Obfuscation on a WMN

4.2.1 Overview

In this section, we describe in detail the design of a realistic architecture and procedures for obfuscating and collecting SM data. The approach has two phases: First, obfuscation values are created by the gateway and distributed to the SMs. Second,

each SM creates its obfuscated power reading and transmits it to the gateway. Note that our approach avoids the assumption that each SM has a communication link with the TTP. The gateway transmits all the data to the TTP which is responsible to create the data vector and transmit to the utility control center (UCC) for state estimation. TTP also performs billing computations at the end of each billing period and stores all obfuscated customer data for archival purposes.

4.2.2 Creating the Obfuscation Vector

The gateway is responsible for creating the obfuscation vector. To do this, the utility company first sends the basis of the obfuscation space, $\mathbb{O}$, to the gateway directly. The gateway randomly selects weights (η) for each of the vectors in $\mathbb{O}$ and constructs an obfuscation vector. An example for a simple mesh topology is provided in Fig. 4.1. In this example, upon receiving the obfuscation basis $\mathbb{O}$ from the utility, the gateway randomly chooses a weight η for each vector v_i in $\mathbb{O}$ and constructs the actual obfuscation vector o by multiplying each vector v_i by associated weight and adding them up. If there is single gateway as assumed in our case, each SM will be assigned one element from this vector o. Hence, each obfuscation element (i.e., $o[j]$ where $j : 1$ to 7) is communicated to its corresponding SM.

However, given the large size of the AMI network, this may not be feasible and the network may need to be divided into multiple clusters of SMs each led by a different gateway. For those cases, our approach will still apply with one difference: Each gateway will get the same $\mathbb{O}$ and will create an obfuscation vector for all the SMs. However, each gateway only serves a subset of all the SMs. Therefore, when distributing the obfuscation values, only the SMs that are within the cluster of that gateway will be contacted. Nonetheless, this approach may not be efficient in terms

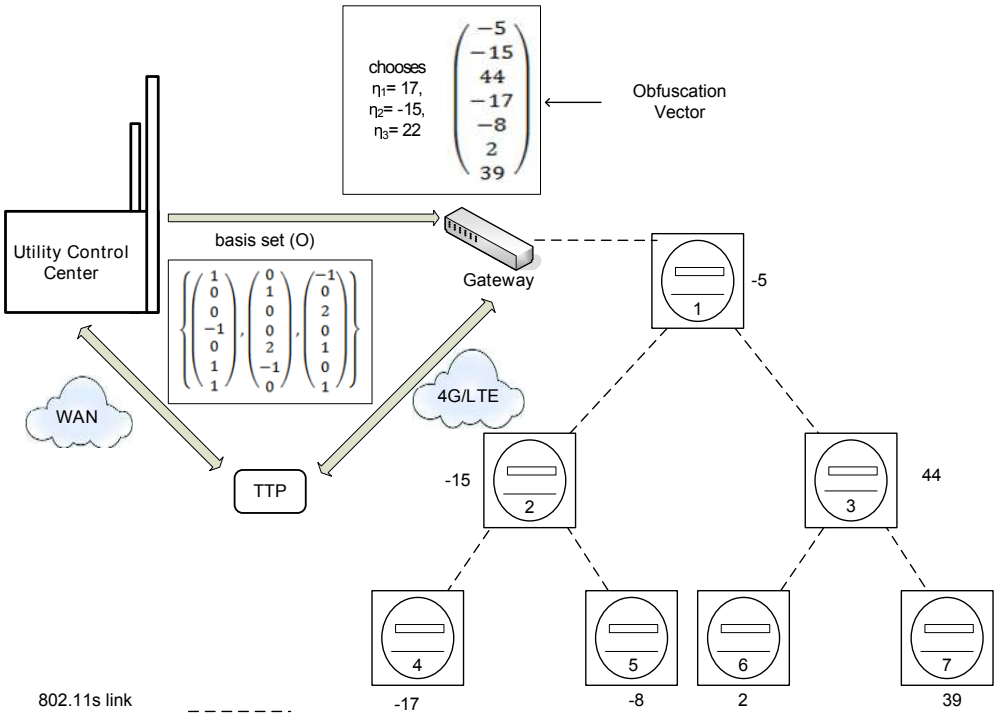


Figure 4.1: A simple illustration of obfuscation vector creation.

of $\mathbb{O}$ transmissions and computations. We will propose a more efficient approach later in this chapter.

Let v_i denote the i^{th} vector in every $\mathbb{O}$, the gateway stores a sum of (η) values for all v_i during a particular billing period T where i can get values from 1 to the number of vectors in $\mathbb{O}$. Let T be divided into the following epochs: $\{t_1, t_2, ...t_n\}$, and $\eta_i^{t_j}$ denotes the η value for v_i at time t_j, then the sum of all η values for all v_i is: $sum_i = \sum_{j=1}^{j=n} \eta_i^{t_j}$. When the final meter reading for a billing period is collected, the gateway chooses the weight $(\eta_i^{t_n})$ for a particular vector v_i so that the sum_i becomes zero. Thus, $\eta_i^{t_n}$ is chosen as $-\sum_{j=1}^{j=n-1} \eta_i^{t_j}$ so that $\sum_{j=1}^{j=n} \eta_i^{t_j} = 0$.

4.2.3 Secure Distribution of Obfuscation Values

Once the obfuscation vector o is created at the gateway, the next task is to send these values to SMs in a secure and efficient way. To reduce the traffic, one possibility is to broadcast the whole vector within the network and let each SM pick its corresponding obfuscation value. However, there are issues regarding this method. First of all, we use TCP which does not support broadcast. Even if we use UDP without acknowledgments, this creates unnecessary flooding in the network where some SMs receive the same vector multiple times from their neighbors. In addition, the size of the whole vector will grow with the increased SM count and thus may necessitate additional broadcasts due to exceeding maximum transfer unit (MTU) for IEEE 802.11 standard. Given that SMs send readings at regular intervals, we opt to use inter-interval times to distribute the obfuscation values using unicasting capability of IEEE 802.11s standard through its routing protocol HWMP. The gateway prepares a packet for each SM and transmits to each SM separately.

Specifically, the gateway employs 128-bit AES block cipher to encrypt the elements in the vector. First, it creates a unique key for each SM and exchanges them with the corresponding SM by encrypting it with the public key (PU_i) of its corresponding SM_i. The gateway then sends each SM its corresponding element of the obfuscation vector (which is represented as $o[i]$) by encrypting it with the shared key (SK), signing it with its own private key PR_G and adding a timestamp (TS) as follows. This is also illustrated in Fig. 4.1.

$$Gateway \rightarrow SM_i : \{< o[i], TS >\}_{SK}, Sig_{PR_G}(\{< o[i], TS >\}_{SK})$$

4.2.4 Calculating & Transmitting Obfuscated Measurements

When an SM_i receives its element $o[i]$, it calculates its obfuscated power measurement (op_i) by adding its current power reading (p_i) and $o[i]$: $op_i = p_i + o[i]$. SM_i then timestamps (TS) the sum and digitally signs the message for the gateway using its private key, PR_i. SM_i then transmits this to the gateway again by using HWMP:

$$SM_i \rightarrow Gateway :< TS, op_i >, Sig_{PR_i}(< TS, op_i >)$$

Upon receiving the obfuscated measurements from each SM, the gateway verifies the digital signatures and timestamps. It then sends them to the TTP. For simplicity, we assume that the gateway can wait for all the SM readings and send them as a single packet.

TTP prepares the obfuscated measurement vector for the utility. In addition, when the billing period ends, it can sum all the measurements to obtain the total usage for each SM for the billing period to charge the customer. The utility receives the obfuscated measurement vector from the TTP and uses it for performing state estimation. Based on the $\mathbb{O}$ in Fig. 4.1, the calculation and collection of the measurements are depicted in Fig. 4.2. Each meter adds its current reading (circled) to the received obfuscation values to calculate its obfuscated reading (underlined). SM_4, for example, has a current reading of 7. It sums it with the obfuscation value it received, -17, obtaining an obfuscated reading of -10. The obfuscated readings are securely communicated back to the gateway which constructs the obfuscated measurement vector, z_{obf}. This is sent to the TTP.

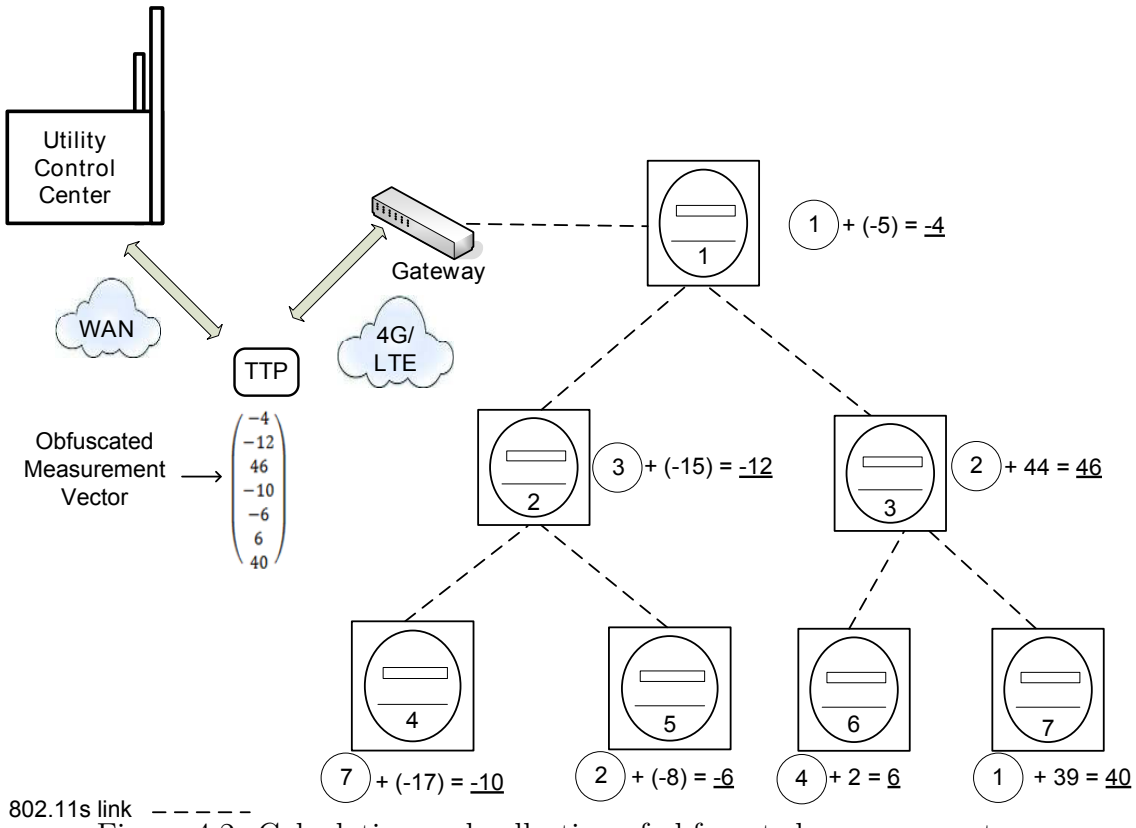


Figure 4.2: Calculation and collection of obfuscated measurements.

4.3 Multiple Gateways for Increased Efficiency and Security

4.3.1 Multi-Gateway Communication Protocol via LTE D2D

Multi-gateway communication protocol is geared for information exchange among the gateways. We assume that the AMI network is divided into multiple clusters, where each is led by a different gateway. Each gateway node has two radios one for 802.11s and one for LTE. Each gateway knows the IDs of SMs within its cluster and the public keys of other gateways in advance. The gateways need to communicate with each other to exchange the obfuscation elements using device to device (D2D) communication architecture that is recently being standardized for LTE-Advanced under proximity services (ProSe) studies [135] (also known as LTE-Direct [136]). While direct

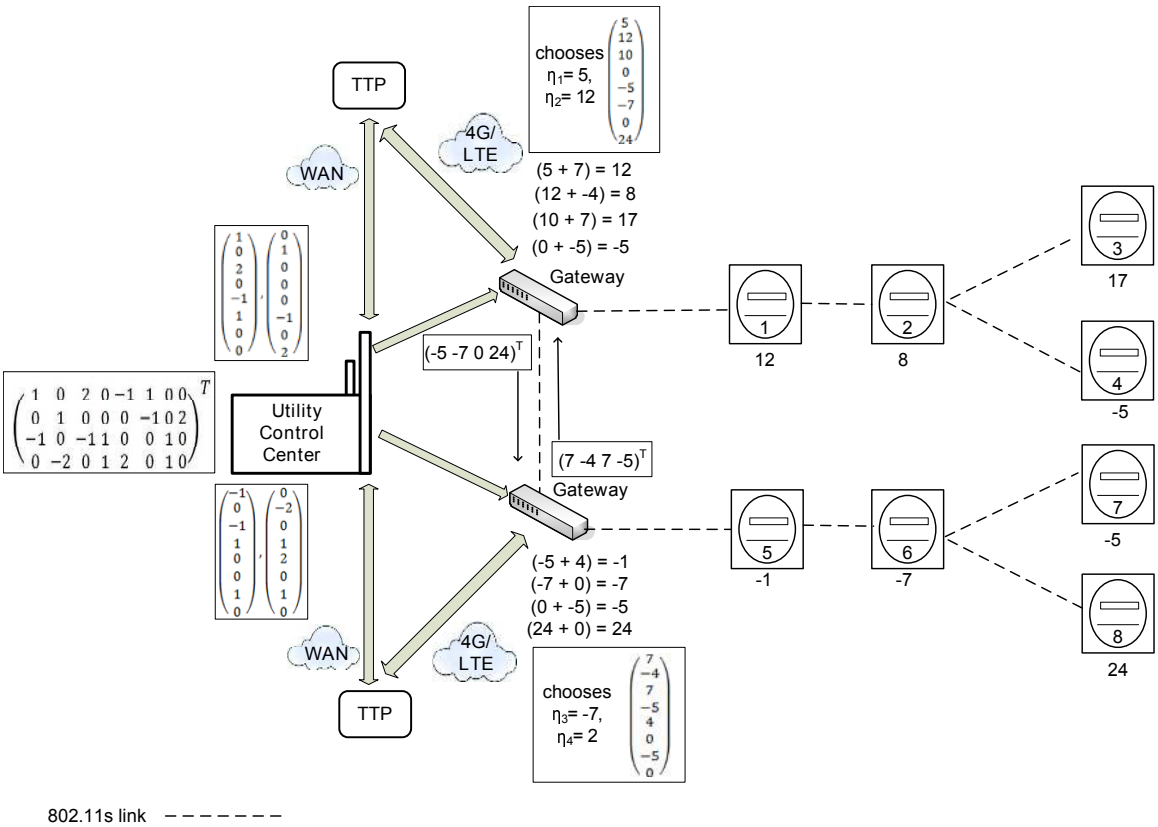


Figure 4.3: A simple illustration of generation, exchange and distribution of the obfuscation vector with two gateways.

communications have been possible in unlicensed bands, e.g., using WiFi-direct, there are various challenges, such as the high interference in the unlicensed bands, as well as the difficulty of pairing and synchronizing D2D transmissions. In LTE-Advanced Release-12 standardization, preliminary studies on D2D and ProSe communications were initiated for commercial and public safety applications [137]. A major benefit of ProSe D2D communications in LTE-Advanced is that device pairing, resource allocation, and power control can be coordinated through a base station (which e.g. can be embedded in the UCC in Fig. 4.3). This allows improved spectral efficiency and lower scheduling delays compared to completely uncoordinated scheduling of D2D transmissions. Use of licensed bands also allows improved quality of service for D2D transmissions.

Each gateway would run the same protocol explained in the previous section within its cluster once the obfuscation vector is formed. However, the formation of this obfuscation vector is different in this case. The protocol begins when the utility provider derives the obfuscation basis $\mathbb{O}$. It splits $\mathbb{O}$ into g components where g represents the number of gateways involved. Then each component is encrypted with each gateway's public key and sent to that gateway through LTE network.

When a gateway involved in the obfuscation generation receives its *partial* $\mathbb{O}$, it chooses the η weight(s) and calculates its *partial* obfuscation vector. This vector would have obfuscation values for all the SMs in the network but the gateway distributes only the ones that belong to its cluster. The other portions are transmitted to the other gateways (in encrypted form) so that they can distribute corresponding values within their own clusters.

In this way, a reading of an SM in a particular cluster will be obfuscated by adding g obfuscation values as opposed to one in the previous approach. Again, for a particular billing period T, the η weights for the same vector are chosen in such a way that the sum of these weights would be zero. Note that the same security operations of signing and encryption still apply. In Fig. 4.3, you can see an illustration of how two gateways generate obfuscation values, exchange them and distribute them to SMs. The UCC generates the obfuscation basis and splits it for the gateways. It encrypts them and sends to the gateways. The gateways extract the bases, exchange the values and generate the obfuscation vector by picking random η values and multiplying them with each vector of the $\mathbb{O}$. Each gateway sends to the other gateway the obfuscation values belonging to the meters that are in the other gateway's area of responsibility. Each gateway adds the obfuscation value it received from the other to its obfuscation value and obtains ultimate obfuscation values. Then, each obfuscation element (i.e., $o[j]$ where $j : 1$ to 8) is transmitted to its corresponding SM by each gateway.

4.3.2 Algorithm and Analysis

When an SM receives its obfuscation element, it decrypts it and adds it to its current reading to calculate its obfuscated reading. The obfuscated readings are signed, timestamped and securely transmitted back to the gateway, which collects all of readings in this cluster. The gateways construct the obfuscated measurement vector, z_{obf} and send the vectors to the TTP via LTE. This is the second phase of what is given in Fig. 4.3. Each SM follows the same procedure given in Fig. 4.2. Then, the gateways send the obfuscated measurement vectors to the TTPs over a 4G/LTE network.

If there are g gateways, the utility provider needs to send g messages for transmitting out the partial basis. Each gateway involved in the obfuscation vector generation would need to contact $g-1$ gateways to send its own obfuscation vector. The remaining steps are similar to those of the single gateway approach and at the end there will be g total messages sent from the gateways to the TTP. Since the inter-gateway communication will be using LTE-Direct, we argue that this will not put any burden on the gateways. Typically, there will be 648 bits allocated for each entry of a vector. Since LTE-Direct can send up to 3Gbps, this would allow sending up to 5000 SMs' readings at the same time under perfect conditions.

4.4 Evaluations

4.4.1 Security Analysis

In this section, we evaluate our proposed approach based on the security goals listed in Section 4.1.5.

Security Goal 1: Since the fine-grained meter data is obfuscated, the actual reading cannot be determined at any time. Because of this, it cannot be analyzed to determine any activity or behavior of the consumer.

Security Goal 2: The obfuscated reading that the SM sends to the gateway does not reflect the actual reading. Therefore, even if an eavesdropper captures this reading, its inference about the activity in the house will be wrong. Also, since the gateways disseminate different obfuscation values at each reading collection period, the eavesdropper cannot extract a pattern of the consumer's power consumption.

Security Goal 3: If a gateway is compromised, the obfuscation information regarding that cluster could be obtained. However, since there are other obfuscation values coming from the other gateways, the attacker needs to have access to all other gateways as well. Therefore, obtaining actual meter readings is not possible with a single gateway being compromised.

Security Goal 4: Since all the SMs use digital signatures for messages containing obfuscation information and measurements, the digital signature can be verified to confirm the identity of the message sender. In addition, since the messages are digitally signed, they cannot be modified without invalidating the signature, providing message integrity.

Security Goal 5: Since all messages are timestamped and digitally signed, the timestamp can be checked to verify that the received message is for the current reading.

4.4.2 Experimental Setup

We implemented the proposed approach under the widely used network simulator ns-3 [132], which has an implementation of IEEE 802.11s. ns-3 can realistically simulate the physical layer and any random interference among SMs. Randomly connected

AMI network topologies were created containing 25, 36, 49, 64, 81 and 100 nodes in an area of size 1200m x 1200m. This area mimics the size of a neighborhood which uses a single gateway to communicate with the utility company.

The transmission range of each SM is set to 100m [131]. The underlying MAC protocol is IEEE 802.11g. TCP protocol is used to ensure reliability. The data frequency of the SMs is set to 10sec [131]. The simulation is run for 1000 secs. We tested each run for 20 different topologies and reported the average of these topologies. All the results are shown with the error bars in the graphs for statistical significance.

For encryption, we used crypto++ library [138]. The Elliptic Curve Digital Signature Algorithm (ECDSA) is an approved signature algorithm for the US government use [139] and the Elliptic Curve Integrated Encryption Scheme (ECIES) is a well-known scheme having several standards [140]. ECDSA is used when only signature is required and ECIES is used when encryption and signature are required. In both cases, we used the ASN.1 secp128r1 standard curve with SHA1, having a key length of 256 bits.

Although PKC is computationally more expensive than symmetric key encryption we used Elliptic Curve Cryptography (ECC) which is an emerging and promising PKC technique because use of PKC eliminates the overhead of key management which is a major issue in symmetric key systems.

128-bit AES generates 16-byte message authentication code (MAC) whereas ECC generates 32-byte signature, which means ECC provides signature sizes that are comparable to that of AES with a slightly increased computational time.

4.4.3 Baselines and Performance Metrics

We considered three baselines for comparison with our approach. The first baseline (represented as "baseline" in the graphs) sends meter readings in clear, providing no

privacy. The second baseline (represented as "baseline sign") provides authentication but does not provide any confidentiality in transmission and the utility provider has access to the fine-grained meter data. The third baseline (represented as "baseline sec" in the graphs) provides authentication as well as confidentiality, but the utility provider still has access to the fine-grained meter data.

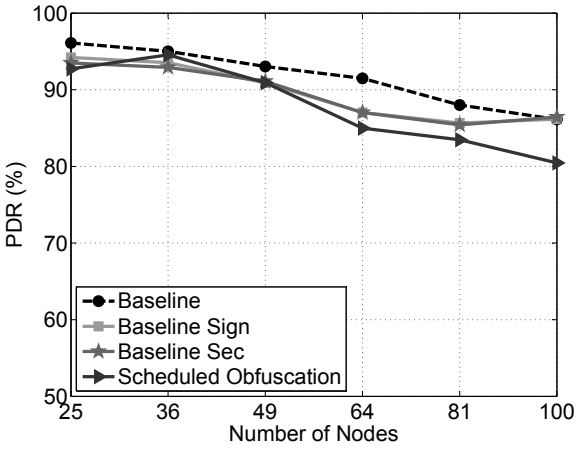
In addition, we analyzed our approach in terms of different setting. For instance, we set the meter reading frequency to both 10sec. and 20sec. in order to observe its effects especially on end-to-end (ETE) delay. We implemented three obfuscation mechanisms based on different data transmission strategies:

1. **Scheduled Obfuscation:** The scheduled transmission in which a SM sends its obfuscated reading value to the gateway at every 10sec. even if the meter receives the obfuscation value earlier. The gateway sends obfuscation values to the SMs for the next reporting time, simultaneously. This is the default mechanism used in all experiments.
2. **Reactive Obfuscation:** The immediate sending in which the meter sends its obfuscated reading value to the gateway as soon as it receives the obfuscation value.
3. **Relaxed Obfuscation:** It is similar to the Scheduled Obfuscation, but the gateway does not send the obfuscation values simultaneously. Instead, there is either 10 sec. or 20 sec. between obfuscation value distribution and meter reading reporting processes.

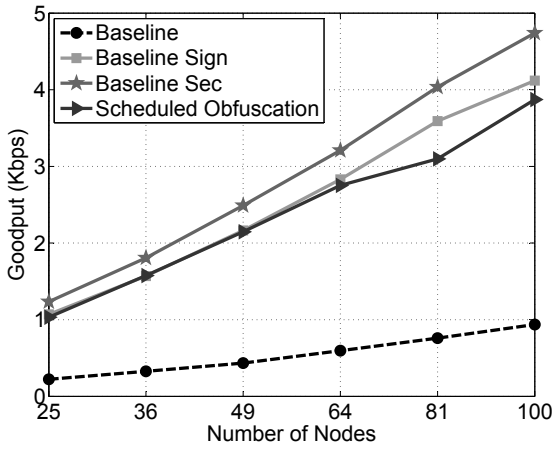
In analyzing the results, we considered three metrics: goodput (i.e., the amount of data received at the application layer by the gateway per second), data delay (i.e., the total time it takes for a reading to reach the gateway) and packet delivery ratio (PDR) (i.e., the ratio of packets that are delivered to the gateway compared to the number of packets sent by the SMs).

4.4.4 Simulation Results

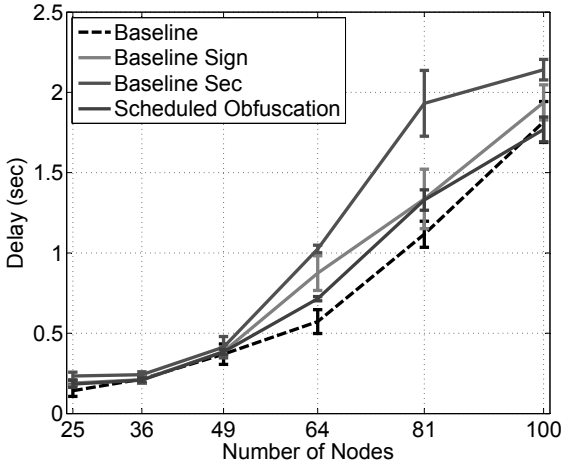
The results of the experiments conducted for comparing the performance of the scheduled obfuscation approach with those of the other three baseline approaches are shown in Fig. 4.4a, 4.4b and 4.4c. We discuss each of the metrics separately below.



(a) PDR at different number of nodes.



(b) Goodput at different number of nodes.



(c) Average ETE delay at different number of nodes.

Figure 4.4: Scheduled Obfuscation simulation results.

PDR

The PDR as shown in Fig. 4.4a decreases slightly for all approaches as the number of SMs increases. This is due to increasing number of packets transmitted throughout the network from SMs, which increases the collisions. We observe that PDR is the highest for the *baseline* because it has the smallest packet size compared to the other three approaches. The *baseline sign* and *baseline sec* approaches achieve almost the same PDR, but mostly the *baseline sign* is slightly better because its packet size is smaller than that of the *baseline sec*. Note that the *baseline sign* and the proposed approach generate 44-byte packets whereas the *baseline* and the *baseline sec* generate 12-byte and 65-byte packets, respectively.

Although the proposed obfuscation approach and *baseline sign* generate the same size packets, the former mostly achieves lower PDR. This is due to the two-way packet traffic between the SMs and the gateways. Contrary to the other baselines, the proposed approach needs to have obfuscation values coming from the gateway. Our approach is scheduled and waits for the end of 10secs to send all the SM readings. At the same time, the gateway starts sending obfuscation values for the next cycle. Note that the obfuscation values coming from the gateway are signed and encrypted and thus the packet size becomes 65 bytes which is even greater than the readings' size (i.e., 44 bytes after signing). Therefore, the obfuscation values sent for the next data cycle may collide with some of the SM readings. This slightly increases the number of dropped packets and results in a decrease in the PDR. We will examine the effect of using a reactive approach where the readings are sent immediately in the next subsection.

Goodput

As seen in Fig. 4.4b, the goodput increases as the network size grows due to the contribution of more nodes as expected. The goodputs of the *baseline sign* and the proposed approach are similar as expected. However, there seems to be a very slight decrease in the rate of increase of the proposed approach. This is again due to the crossing of traffic when the obfuscation values are being sent from the gateways to the SMs and shortly after the readings are being sent from SMs to the gateway through the same paths. While these transmissions are happening one after another, there may still be some traffic in the network during the transmission of obfuscation values to the leaves (i.e., the nodes at the far end of the network) of the network topology. This can cause some interference and keep the channel busy at certain parts of the network which eventually causes the rate of increase in goodput to reduce slightly. However, overall these results indicate that there is no major adverse effect of the proposed distribution and obfuscation approach in terms of goodput.

We observe that the *baseline sign*, *the baseline sec* and the proposed approach have significantly higher goodput than the *baseline* has. This can be attributed to the sizes of the packets they send. Even if the *baseline* has the highest PDR, the sizes of the packets the other three approaches generate compensate this difference and cause higher goodput.

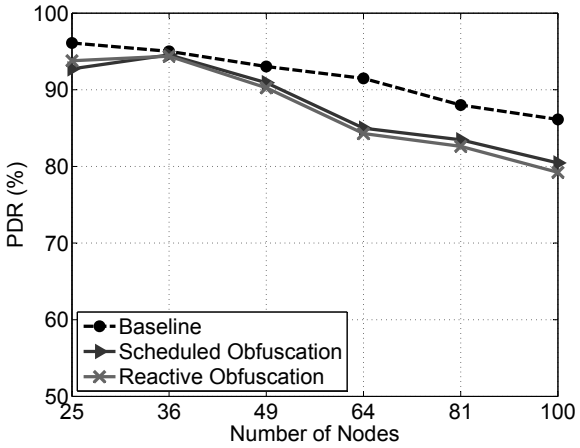
ETE Delay

We also looked at the impact of the approach on ETE delay, an important metric for some of the AMI applications such as demand and response. The delay of all of the baselines is similar when the network size is smaller (i.e., up to 49 nodes) as shown in Fig. 4.4c, and it increases because of the increase in congestion cases as the network size grows.

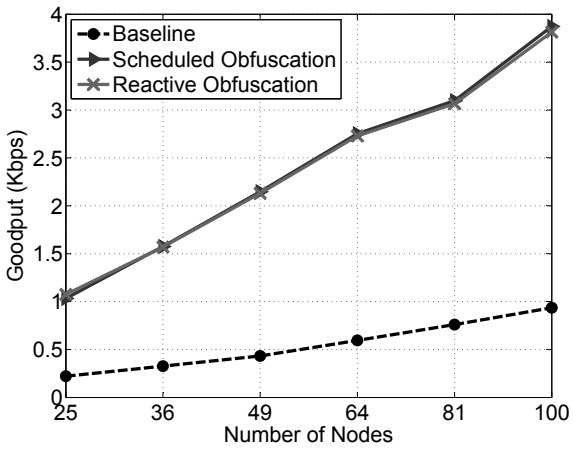
The proposed approach is expected to experience almost the same delay with the *baseline sign* for all of the number of nodes, but it demonstrates lower delays after 49 nodes. This can be explained as follows: Since the SMs are waiting for obfuscation values from the gateway, they cannot send their readings around the same time. The obfuscation values reach the destinations at different times due to the topological structure of the network and SMs are scheduled to send their readings at the next sending time. Since ns-3 does not schedule the sending operation to an exact time value but schedules it so as to be performed after a given time interval, the SMs cannot be scheduled to exactly the same time for transmission. They are scheduled between the same seconds, but there are time lags (some milliseconds) between each scheduling. This apparently reduces the contention among the nodes for accessing the channel in the network and thus MAC layer delay is reduced. Note that in the other approaches, more nodes become involved in message sending at the same time and thus channel access delay increases significantly due to heavy contention and interference. In this way, our proposed approach amortizes the impact of obfuscation distribution, making it feasible for practical cases.

Scheduled Obfuscation vs Reactive Obfuscation

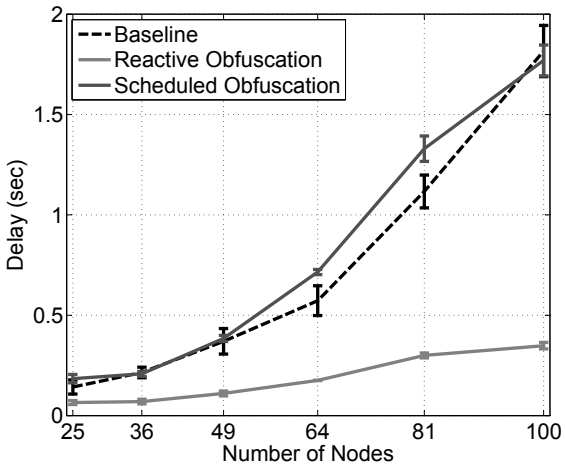
In a different version of our proposed approach, the gateways send obfuscation values at every 10 sec as in the scheduled version, but a SM sends its obfuscated reading to the gateway as soon as it receives its obfuscation value instead of waiting for the next reporting time. We refer to this approach as *Reactive Obfuscation*. We implemented this version in ns-3, ran the same experiments as in the case of the *Scheduled Obfuscation* approach and compared their performance in terms of PDR, goodput and ETE delay.



(a) PDR at different number of nodes.



(b) Goodput at different number of nodes.



(c) Average ETE delay at different number of nodes.

Figure 4.5: Scheduled Obfuscation vs. Reactive Obfuscation simulation results.

In Fig. 4.5a, we observe that PDR of the *Reactive Obfuscation* is almost same as the *Scheduled Obfuscation*. The baseline performs better as there is no traffic coming from the gateway for obfuscation value distribution that would cause congestion and interference. The use of TCP ensures that the packets are delivered even if there would be contention among the SMs to access the channel.

We also looked at the goodput of both approaches as seen in Fig. 4.5b. The goodputs for the *Reactive Obfuscation* and *Scheduled Obfuscation* approaches show

the same trend as in the case of PDR and they are almost the same due to receiving similar number of packets at the gateway.

We tested ETE delay performance of *Reactive Obfuscation* and compared it with that of *Scheduled Obfuscation*. Our results stayed within 1%-13% of the sample mean with a 95% confidence interval. These are shown with the error bars in the graph.

While the *Reactive Obfuscation* performs almost as good as *Scheduled Obfuscation* in terms of PDR and goodput, we observed that it reduces ETE delay significantly, which can be seen in Fig. 4.5c. This can be attributed to the fact that the transmission times of the readings from the SMs would be different in *Reactive Obfuscation* as the obfuscation values arrive to SMs at different time. This is not the case in *Scheduled Obfuscation* where all the nodes wait for a specific time to transmit their readings even though they may have received their obfuscation values before. In this approach, a lot of SMs experience back-off repeatedly due to unavailability of the WiFi channel at MAC layer. This increases delay significantly. Such a situation is not the case in *Reactive Obfuscation*.

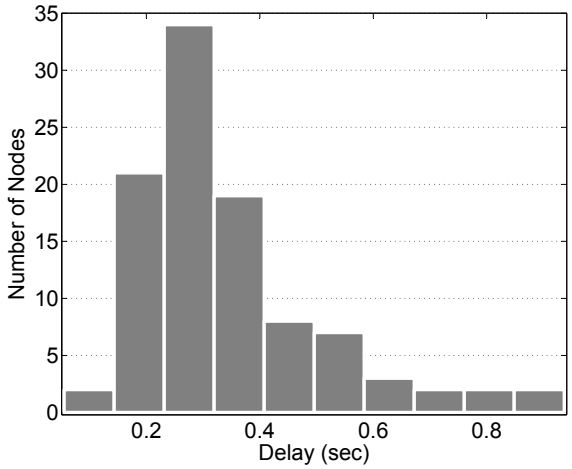


Figure 4.6: Reactive obfuscation delay value distribution for 100 node topologies.

The Impact of Scheduling Meter Readings and Obfuscation Values

Since congestion and contention in these two obfuscation approaches increase the number of dropped packets, we decided to investigate if this can be further alleviated by interleaving a time gap between the transmission of obfuscation values from the gateway and the meter readings from the SMs. This is referred to as *relaxed obfuscation*. Recall that in the regular *scheduled obfuscation* approach both the gateway and the SMs transmit their messages to the network around the same times. The main motivation behind this is to either reduce possible clashes among obfuscation values and data readings or wait enough time for the network to have less traffic.

We conducted two experiments by using 100-Node topologies. In the first experiment, the time interval between the obfuscation value distribution and meter reading reporting processes is 10 seconds while in the second one, it is 20 seconds. The results compared with the other approaches are shown in Table 4.1. Also, a histogram of ETE delay values is given for *reactive obfuscation* in Fig. 4.6, which indicates that the values show a normal distribution which is expected due to various hop distances of SMs from the gateway.

From Table 4.1, we observe an interesting result in terms of ETE Delay. Basically, *Reactive Obfuscation* beats all the other approaches including the relaxed ones although they wait for the network to become quiet for 10 or 20secs. The difference is significant (i.e., reduce by 5 times) while the PDR values keep around the same rates. This takes us to the conclusion that the problem was not the clash of values but the contention among the SMs to access the channel. In the case of *Reactive Obfuscation*, the obfuscation values come at different times and thus the contention among SMs is minimal. This is not the case of the scheduled and relaxed approaches as they tend to transmit readings at the same scheduled time. Therefore, it is wiser to work on data scheduling approaches in order to reduce ETE delay further.

Table 4.1: Delay and PDR comparison for different approaches

100-Node Topologies		
Approaches	***Delay (sec)***	***PDR*** (%)
Relaxed Obfuscation-10sec	1.61	82.89
Relaxed Obfuscation-20sec	1.31	79.92
Scheduled Obfuscation	1.77	80.46
Reactive Obfuscation	0.35	79.23

4.5 Conclusion

In this work, we tackled the problem of user privacy preservation in SG that will also enable distribution state estimation. We followed a data obfuscation mechanism and proposed secure and efficient algorithms to distribute obfuscation values within the AMI network. Specifically, we used AES for hiding and ECC for authenticating the obfuscation values that are distributed within the network. We also considered multiple-gateway implementations for increased security. We proposed a protocol that utilizes LTE-Direct for exchanging of data among multiple gateways.

We implemented all the proposed approaches in ns-3 using a draft version of 802.11s for a 802.11s-based mesh network to assess their overhead. We investigated the performance under scheduled, reactive and relaxed transmitting strategies. Simulation results showed that the obfuscation approaches are promising in terms of ETE delay without introducing additional overhead on PDR and goodput compared to other existing approaches. We analyzed the approaches in terms of the security goals they provide and showed that they can ensure consumer privacy while still allowing state estimation and billing.

CHAPTER 5

PRESERVING PRIVACY VIA FULLY HOMOMORPHIC ENCRYPTION AND SECURE MULTIPARTY COMPUTATION IN THE AMI NETWORK

Automatic Meter Reading (AMR) systems collect consumption, diagnostic and status data [141] from the consumers' utility meters by means of a drive-by vehicle or a hand-held device. The collected data suffice to bill the consumer and to monitor the status of the meters on a monthly basis in the existing grid. In order better to manage the power demand, reduce CO_2 emissions, and ensure reliability [4, 129], the ongoing Smart Grid (SG) initiative in the US proposes several modifications to the existing grid. This requires a communication infrastructure to enable two-way communication between the utility companies (UCs) and the meters, and the ability of making decisions autonomously, which makes the meters "smart" [142,143]. However, AMR systems are far from providing the required data to implement smart functions such as demand-load matching, demand response, dynamic pricing, etc. [144].

The necessity of such an infrastructure brings the Internet of Things [145] concept to the existing grid. The "Things" in SG are the sensors/intelligent electronic devices that are deployed along with the transmission/distribution lines and the smart meters (SMs) at the consumer side. SMs are the IoT devices that have the capabilities of processing and accessing the Internet. They are able both to send the fine-grained power consumption data they measure to the UC and to receive instructions from the UC. Also, they can adjust energy usage based on the cost or availability of energy, depending on the preferences set by the consumers. These functions can be enabled through several new applications such as Advanced Metering Infrastructure (AMI). AMI applications are run on a network infrastructure that connects SMs and the UC,

typically via a wireless mesh-based network, referred to as the AMI network in the rest of the chapter.

Collection and storage of such fine-grained data, however, raises the issue of privacy for the consumers who have to use the SMs daily [9,10]. Specifically, the collected consumption data can be analyzed using load monitoring techniques to infer activities of the consumers [146]. Hence, typical privacy threats include, but not limited to: 1) Determining personal behavior patterns (can be used by marketers, government); 2) Determining specific appliances used (can be used by insurance companies); 3) Performing real-time surveillance (can be used by law enforcement and press); 4) Target home invasions (can be used by criminals); and 5) Location tracking based on electric vehicle usage patterns (can be used by law enforcement). The problem is compounded with the involvement of third party service providers (TSPs) for the management of the collected data [147]. These service providers provide cloud services to maintain, store, and analyze the consumers' data on behalf of the utility companies.

Due to such privacy concerns, partially homomorphic encryption and secure data obfuscation schemes were employed to prevent eavesdroppers from making inferences about the consumer activity by making various assumptions on the available resources [10, 14]. Despite such efforts, the privacy issue has been creating several problems in the deployment of SMs throughout the US and making the consumers reluctant to participate in SG programs [13] because all of the proposed approaches at some point assume a trust relationship between the UC/TSPs and the consumers. The consumers may not be comfortable with UCs/TSPs that have the right to access their private data.

Data aggregation can be used to both hide individual meter readings and reduce packet traffic in the network due to the high frequency metering data [129]. The idea is to perform the aggregation within the network as meter readings are routed

towards the gateway from the SMs. Each intermediate SM performs an aggregation. However, this exposes private data of a particular meter to another meter in the network because the aggregation is performed on clear meter readings. To solve this problem, several studies [15,16,50,74–76] suggested using partially homomorphic encryption (PHE) [71], fully homomorphic encryption (FHE) [47] or secure multiparty computation (Secure MPC) [22] that are capable of performing certain arithmetic operations on concealed data in a privacy-preserving fashion. Of these homomorphic encryption systems, PHE is widely used for simple aggregation since it allows addition on the encrypted data. However, PHE is not able to perform other operations on the encrypted data. This may eventually affect many other SG Distribution side operations such as state estimation, demand response, direct load control, etc.

FHE and secure MPC systems are becoming more popular since they allow both addition and multiplication on the encrypted data, giving flexibility to the applications to perform different computations for their needs without endangering privacy of the consumers. However, FHE systems suffer from generating large size ciphertexts and longer computational times, particularly for multiplication. This makes it challenging to be used for in-network aggregation in the AMI network. Secure MPC approaches, on the other hand, are lightweight, but they require excessive messaging which may not be feasible to be used in an AMI network that does not allow direct communication among all members. This work aims to address these issues by introducing the necessary mechanisms and then assessing the overhead and performance of the use of the aforementioned mechanisms. To the best of our knowledge, this is the first work to implement and investigate a secure MPC-based protocol with highly reduced messaging complexity for the IEEE 802.11s-based [148] SG AMI network.

Our contributions are three-fold in this work.

1. For the adaptation of FHE systems in the AMI network, we propose mechanisms to reduce the large ciphertext size and deal with packet reassembly problem [16] when TCP is used as the underlying transport protocol. Specifically, we first tackle a new problem due to excessive fragmentation of FHE packets. Note that data aggregation cannot be performed in such cases since TCP does not know the packet sizes in advance and thus cannot determine where to cut the streams arrived at the receiver. To this end, in this work we propose a novel solution by adding a presentation layer above the transport layer to include packet size information at the sender side.

2. For the adaptation of secure MPC, we propose a mechanism to reduce the message complexity. In a classical secure MPC-based protocol using secret sharing techniques, the shares are exchanged between the meters at each data collection round. However, this protocol consumes the bandwidth significantly. Instead, in this work, we propose a privacy-aware communication protocol to lower the required bandwidth. Specifically, the meters use a pseudo-random number generator (PRNG) to compute the shares locally that are computed by the other meters. Hence, the meters do not need to exchange the shares before each data collection round; so, the bandwidth and the other network re/sources are used more efficiently. In addition, we further improve the bandwidth usage by employing in-network data aggregation.

3. Finally, we implemented the aforementioned privacy-preserving data aggregation protocols by using the ns-3 [132] network simulator. We compared the performance of both FHE and secure MPC-based protocols to that of PHE in terms of packet delivery ratio, throughput, and average data collection completion time in order to investigate if the use of FHE and secure MPC is feasible under realistic settings. The experimental results indicate that the secure

MPC-based protocol is a viable option for preserving privacy with a comparable performance to PHE while it can support multiple operations. In addition, the simulation results indicate that the proposed packet reassembly protocol enables the realization of FHE-based data aggregation using TCP in terms of the data collection completion time and used bandwidth.

The rest of this chapter is organized as follows. In the next section, we provide some background on PHE, FHE, secure MPC, the network and attack models, and define the problem. Section 5.2 investigates the adaptation of an FHE system to the AMI network, assesses the feasibility of FHE aggregation operations, and presents the details of the proposed packet reassembly protocol. We present the adaptation of a secure MPC-based data aggregation protocol in Section 5.3. In Section 5.4, we assess the performance of the proposed approaches. Finally, Section 5.5 concludes the chapter.

5.1 Preliminaries

In this section, we provide a background information about partially and fully homomorphic encryption systems, secure MPC, and network model we used for this work.

5.1.1 Partially and Fully Homomorphic Encryption Systems

Homomorphic encryption systems enable performing a set of operations on ciphertexts without disclosing their actual value. When the resultant ciphertext is decrypted, the decrypted value is equal to the value to be obtained when the same set of operations are performed on the actual value of the ciphertexts.

In this work, we use two types of homomorphic encryption systems: PHE and FHE. PHE is an encryption system that enables performing either addition or multiplication operation on encrypted data. Paillier cryptosystem [71] is the most commonly used PHE system. It is an additive homomorphic cryptosystem, which means that it is able to perform only homomorphic addition operation on a ciphertext. Below is a more formal representation of Paillier's homomorphic addition operation:

Let m_1 and m_2 be two plaintexts.

$$D_{S_K}((E_{P_K}(m_1)\, x\, E_{P_K}(m_2))\ mod\ n^2) = (m_1 + m_2)\ mod\ n\,, \tag{5.1}$$

where x and $+$ operators represent modular multiplication and addition operations, respectively. n is the first component of the public key ($P_K = (n,\ g)$ where g is a random integer and $g \in \mathbb{Z}^*_{n^2}$).

As opposed to PHE systems, an FHE system can perform both addition and multiplication operations on encrypted data. In this work, we use Smart-Vercauteren (SV) scheme to provide privacy which is an FHE system. SV scheme consists of key generation, encryption, decryption, addition/multiplication, and recryption functions [39].

We will explain two aspects here as others are already well-known: key generation and recryption. Key generation is different in SV since some portion of the public-key is used for recryption purposes. In addition, the key size in SV is in the order of kilobytes which is much higher than the keys in traditional schemes that are in the order of bits. SV is a member of public-key cryptography family, so it generates a key pair: public and secret (private) key.

The keys are generated considering three important parameters: The number of bits ($|B|$) which is used to create random coefficients for the variables of the polynomials that are used to generate a *hint*, the number of *shares* (S_1), and the number of

cells (S_2) in which the shares of the hint are stored. We call each tuple ($|B|/S_1/S_2$) a "key geometry".

As more operations are performed on a ciphertext noise is accumulated in the ciphertext. The recryption function removes this noise in the ciphertext without decrypting it and the cleartext is kept unchanged. The function utilizes the hint whose pieces are distributed into an array in public-key randomly. In the lack of such a function, we are limited to a fixed number of homomorphic operations. When we exceed this number of homomorphic operations the ciphertext becomes undecipherable.

5.1.2 Secure Multiparty Computation

Secure multiparty computation makes use of secret sharing to implement data aggregation. Secret sharing differs from PHE and FHE in the way of concealing the data. It is based on dividing a secret into shares and distributing them amongst a group of participants such that the secret cannot be reconstructed unless a certain number of the participants collude. However, in PHE or FHE, it is sufficient to obtain the private key in order to decrypt any message encrypted with the corresponding public key.

Shamir's Secret Sharing (SSS) [149] is the most commonly used secret sharing scheme. In SSS, we assume that there are n nodes in the network and all computations are done in a finite field $\mathbb{Z}_p$, where p is a prime number. Let r_i be the private secret of node i. Node i chooses a unique point $x_i \in \mathbb{Z}_p$ other than zero and selects an $(n-1)$ degree random secret sharing polynomial $f_i(x)$ with $f_i(0) = r_i$. It sends its unique point x_i to all other nodes and receives share values $f_j(x_i)$ computed by the other $(n-1)$ nodes. Then, it computes $F(x_i) = \sum_{k=1}^{n} f_k(x_i)$. These steps are done by all n nodes and $F(x_i)$ values are sent to the gateway. The gateway can construct an $(n-1)$ degree polynomial $g(x)$ by using the $F(x_m)$ values along with Lagrange

interpolation, where $m \in \{1, ..., n\}$. The constant term of $g(x)$ is the aggregation of all individual n private secrets.

5.1.3 Network Model

We assume an AMI network that consists of SMs (e.g., IoT devices) and a gateway that can communicate with a UC. The communication between SMs is based on IEEE 802.11s-based mesh standard which allows SMs to determine a route to the gateway for sending their readings [150–152]. The gateway collects all the SM readings and sends them to the UC using a wide area network connection such as WiMAX or LTE [153]. A sample AMI network based on IEEE 802.11s is given in Fig. 1.1.

5.1.4 Problem Definition

Traditional encryption methods can be used to provide security for data communication, but they require decryption before data aggregation. This reveals private meter readings to another meter and breaches the consumers' privacy. While this can be addressed using PHE systems, the aggregated encrypted data cannot be further used for other applications such as distribution state estimation or direct load control where more sophisticated computations are needed. Hence, our problem in this work can be defined as follows: *Devise network protocols that will help adapt FHE and secure MPC for deployment in the AMI network. In addition, assess their performance with respect to PHE solutions in a realistic network to understand the overhead of achieving comprehensive privacy.*

5.1.5 Threat Model and Security Goals

We have the following threats to the privacy and security of SM data collection in the AMI network and identify the relevant security goals.

Threat 1: The UC can misuse fine-grained meter data to analyze consumer behavior or worse, it can share the collected data with a third party for this purpose.

Security Goal 1: Aggregate the collected fine-grained meter data in-network before sending to the UC to protect them from misuse by the UC or any third party.

Threat 2: An eavesdropper can monitor the communication channel to capture meter data in messages between a targeted SM and the gateway to determine the behavior of the SM's user.

Security Goal 2: Protect communications containing SM readings via data concealment.

Threat 3: An attacker can compromise a SM and analyze behavior of its child meters.

Security Goal 3: Employ data aggregation techniques that can perform arithmetic operations on concealed data.

Threat 4: An attacker can impersonate the gateway and send fabricated data collection requests to the SMs more frequently to keep them busy and to waste the network bandwidth.

Security Goal 4: Provide sender authentication to verify the sender and to check the content integrity.

Threat 5: An eavesdropper can capture and replay the data packets to change the state estimation or billing.

Security Goal 5: Identify and discard replayed messages.

5.2 FHE Scheme for the AMI Network

In this section, we first examine the complexity of the used FHE system and then tackle the problem of packet reassembling when it is to be used in AMI systems.

5.2.1 The Complexity of Smart-Vercauteren Addition and Multiplication Operations

As mentioned, we use an implementation of Smart-Vercauteren scheme [39] which is an FHE system. In this work, we extended [39] so that the operations can be performed on multi-bit operands (rather than single bits) without losing the ability to perform decryption. We also incorporated recryption operation to provide noise cleaning whenever needed. These operations and the communication between the meters are highly secure because meter readings are transmitted in ciphertext and all operations are performed on encrypted data. Also, recryption does not require to have the original of the encrypted data. Hence, unless an attacker has the secret (private) key, no confidential data can be revealed.

Table 5.1: Delay comparison of addition and multiplication

# of operands	***Delay (sec)***	
	Addition	***Multiplication***
2	3.99	625.09
3	8.49	1593.62
4	13.49	3562.15
5	19.04	7624.68

Before we use SV scheme in an AMI network, we investigated the complexity of its operations. Specifically, we assessed the feasibility of addition and multiplication

Table 5.2: Data size comparison of addition and multiplication

# of operands	*Data Size (bits)*	
	Addition	***Multiplication***
2	52,237	101,556
3	55,348	153,626
4	58,353	206,014
5	61,486	258,403

operations of SV scheme for 16-bit operands. We performed sequential homomorphic operations on encrypted data and assessed the time and storage complexity. The tests were performed on a Raspberry Pi 3 Model B [154] having four 64-bit ARM Cortex-A53 processors at 1.2 GHz with 1 GB RAM using Raspbian OS. The results are given in Table 5.1 and 5.2. As shown in Table 5.1, multiplication suffers from excessive processing times. Even for two operands, its processing time is more than 10 minutes. For the generated data size, we observed that addition generates far less data than multiplication does. For instance, for five operands, addition generates less than four fold of that multiplication generates. As can be seen from these results, the multi-bit multiplication processing times in the order of minutes which may not be applicable to all SM data collection applications. However, these types of operations can be run on more powerful servers in the utility control centers.

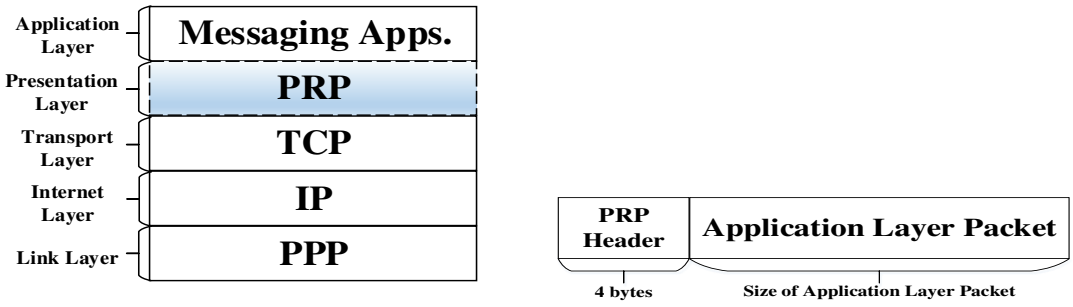
Thus, for the rest of the chapter, we focus on the multi-bit homomorphic addition that can be run on SMs. We analyze its feasibility and performance when used in an AMI network under TCP.

5.2.2 Packet Reassembling with Secure Aggregation

In this section, we first introduce the packet reassembly problem when secure aggregation is employed. We then propose a solution to address it.

The Packet Reassembly Problem under TCP

Given the critical nature of the SM data, we use TCP in order to ensure reliability. Nonetheless, when data packets are transmitted over a TCP connection using FHE, we identified that a *packet reassembly* problem occurs at the receiver side which needs to be solved. Specifically, data flow in a TCP connection is controlled by the *window size* (WS) field in a TCP header. The receiver of a segment states how many bytes of data it is willing to receive. Accordingly, the sender of the segment does not send more data than the stated value in the WS field. In this way, data flow in each direction of the connection is adjusted so that hosts are not overwhelmed by more data than they can handle (i.e., flow control). However, this adjustment may cause some portions of a packet to be transmitted in different segments due to changing WS value especially when the packet size is large. This case typically shows up in FHE systems since large size ciphertexts are fragmented into many segments. At the receiver side, the packet needs to be reassembled from the collected segments since it will be aggregated with other packets coming from other child meters. In this case, the receiver (meter) does not know the size of the sent packet from a particular sender and thus cannot know where to cut the byte stream (consisting of multiple segments). Note that each of the receiver's child meters may send different size packets in case the child meters have different number of child meters. We call this problem the *packet reassembly problem*.



(a) Placement of the *PRP* in protocol stack. (b) An illustration of a *PRP* packet.

Figure 5.1: The Packet Reassembly Protocol in the network stack.

In order to overcome this problem, we propose a new protocol which enables the receiver meter to know the total size of the packet it will receive. We develop this new protocol on top of the TCP layer, in the presentation layer as shown in Fig. 5.1a. The proposed *Packet Reassembly Protocol (PRP)* enables an aggregator meter to reassemble a packet from its segments. The protocol adds a minimal header that includes the packet size to the packet at the sending side while it removes the header, reads the packet size and gathers this size of bytes to reassemble the packet at the receiving side.

Algorithm 1 $Receive(segment, from)$

```
1: buffer ← bufferMap.RetrieveBuffer(from)
2: if buffer == null then
3:    header ← segment.GetPRPHeader()
4:    buffer ← CreateBuffer(header.GetPacketSize())
5: end if
6: residualBytes ← buffer.Add(segment)
7: if buffer.IsFull() then
8:    appPacket ← CreateAppPacket(buffer)
9:    ReportUpperLayer(appPacket)
10:   bufferMap.RemoveBuffer(from)
11:   if residualBytes.Size() ≠ 0 then
12:      resSegment ← CreateSegment(residualBytes)
13:      Receive(resSegment, from)
14:   end if
15: end if
```

As such, a *PRP* packet consists of the *PRP* header and the application layer packet. An illustration of a *PRP* packet is shown in Fig. 5.1b. The size of the header is kept minimum with 4 bytes and it includes the size of the application layer packet and the identifier of the meter. Even if a packet is exposed to TCP segmentation, the first segment is received first by the receiver meter since the TCP guarantees ordered delivery of a stream of bytes. Thus, a meter will be able to know the total size of the packet by using the header information in the first segment it receives.

Protocol Pseudocode

The *PRP* implements two crucial functions: *Send* and *Receive*. *Send* function is called by the application layer. It is utilized to send application layer packets of a meter to another meter. *Receive* function is called by the transport layer when there is a packet in the receive buffer. We provide a pseudocode for only *Receive* function in Algorithm 1 because *Send* function is straightforward.

The algorithm, first, checks if there is a $buffer$ dedicated to $from$. If there is no such a $buffer$, a $buffer$ is created in the size of the received *segment* and the *segment* is pushed into the $buffer$. If the size of the *segment* is more than the size of the $buffer$, excess bytes are put into a byte array $residualBytes$. If the $buffer$ is full, an application layer packet *appPacket* is created out of the *segment*s in the $buffer$. The *appPacket* is sent up to the application layer and the $buffer$ dedicated to the $from$ is deleted from the $bufferMap$. If there is any data in the $residualBytes$ array, a segment $resSegment$ is created out of $residualBytes$ and *Receive* function is called with $resSegment$ and $from$ to handle the excess bytes, recursively.

5.3 Adapting Secure MPC for the AMI Network

As mentioned in Section 5.1.2, secure MPC requires communication among all the nodes (e.g., $n(n-1)$ messages need to be exchanged), which not only increases the communication complexity, but may also render the implementation infeasible due to the topologies of the AMI network. The challenge is to adapt secure MPC in such a way that it can be used in an AMI mesh network topology without significant overhead. To address this issue, we adopt the idea used in [155]. Specifically, instead of exchanging the shares, each set of two meters agrees upon a shared key and uses this key as an initial feed to a pseudo-random number generator (PRNG) to locally

compute the shares that will be received from the other meters. The keys can be preloaded on the meters or the Diffie-Hellman [156], which is the most commonly used key-exchange protocol can be used to share the secret keys.

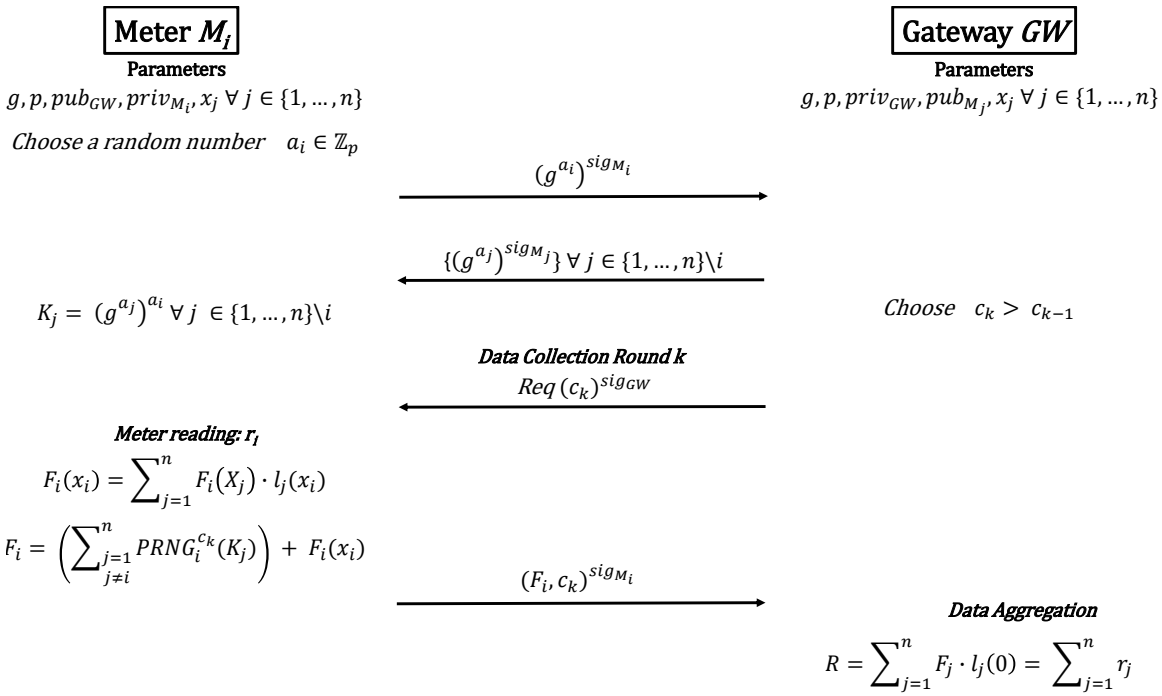


Figure 5.2: Overview of the secure MPC-based protocol we used in this work

We give an overview of the protocol we used in our work in Fig. 5.2. In the protocol, each data collection round is initiated by the gateway. The gateway chooses a round value c_k which is larger than the values used for the previous rounds and sends it to all meters in the network. Each meter i applies the $PRNG_i(\cdot)$ function c_k times with an initial seed K_j to compute $f_j(x_i) = PRNG_i^{c_k}(K_j)$ values locally, where $j \in \{1, ..., n\}/i$. These values are the shares that would be computed by the other meters. Now, we have n points: $\{(0, r_i), (x_1, f_1(x_i), \cdots, (x_n, f_n(x_i))\}/(x_i, f_i(x_i))$. For the sake of clarity, we represent these points with a new tuple $(X_i, F_i(X_i))$. We can construct an $(n-1)$ degree polynomial $F_i(X)$ over these points. However, the coefficients of this polynomial cannot be random, but they have to be computed. The

Lagrange polynomials l_i can be used to pre-compute the coefficients by each meter i as given in Formula (5.2):

$$l_i(x) = \prod_{\substack{j=1 \\ j \neq i}}^{n} \frac{x - x_j}{x_i - x_j} \cdot \tag{5.2}$$

Hence, the polynomial $F_i(X)$ can be derived as in Formula (5.3):

$$F_i(X) = \sum_{j=1}^{n} F_i(X_j) \cdot l_j(X) \cdot \tag{5.3}$$

From Formula (5.3), the meter i can compute its own share by substituting X in the formula with x_i. Now that we have computed all shares, we can sum them up and send the result to the gateway. The gateway constructs a polynomial over received F_i values by using the method given above. The constant term of this polynomial is the aggregated value of all r_i values.

5.3.1 Hierarchical Secure MPC in the AMI Network

Due to the nature of secure MPC, each meter computes the sum of its shares including the shares that would be computed by other meters; signs, and sends it to the gateway directly. The gateway verifies the signature of the packets received and derives a new polynomial over these summed shares. The constant term of this polynomial is the aggregated value of the meters' reading. Finally, the gateway signs and sends the aggregated value to the UC.

However, in our case the AMI network is a multi-hop network where a hierarchical relationship can be defined between the nodes in the network. Therefore, we would like to take the advantage of in-network processing and revise the protocol to work in a multi-hop manner. Specifically, we propose the following modifications: The Lagrange polynomials to be computed by the gateway can be computed by each

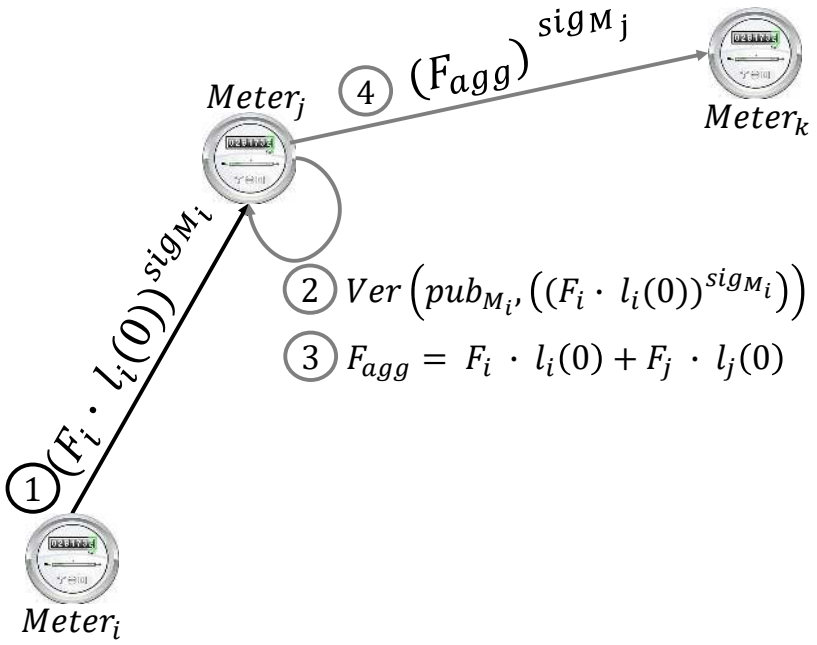


Figure 5.3: A simple example for hierarchical secure MPC of a parent meter with one child meter

meter. The meters compute their total share (F_i in Fig. 5.2) and multiply it by the associated Lagrange polynomial $l_i(0)$. Then, they sign and send it to their parent meter. The parent meters verify the signature of the multiplied total shares and aggregate them with their own multiplied total share. They sign the result and send it to their parent meter (illustrated in Fig. 5.3). This process goes on up until to the gateway. The gateway verifies the multiplied total shares and aggregates them with its own multiplied total share. Finally, it signs the result and sends it to the UC. By following this protocol, both the total bandwidth usage and the computational overhead at the gateway can be reduced further.

The protocols given above are used to perform addition operation. The secure multiparty multiplication [157] can be implemented by applying $PRNG(\cdot)$ function twice consecutively followed by a degree reduction [158]. For the sake of a fair comparison with FHE and PHE, we have not implemented and discussed the multiplication operation in this work.

5.4 Performance Evaluation

In this section, we, first, analyze the security of the proposed approaches, then, present the simulation results.

5.4.1 Security Analysis

In this section, we evaluate our proposed protocols based on the security goals listed in Section 5.1.5.

Security Goal 1: Let $m_i \forall i \in \{1, 2, \cdots, n\}$ be the reading value of meter i. It is encrypted with the public key of the UC (PK_{UC}) before transmitting.

$$Enc_{PK_{UC}}(m_i)$$

The fine-grained meter data is aggregated in-network and the resultant value (c_{GW}) is communicated to the UC by the gateway.

$$\sum_{i=1}^{n} Enc_{PK_{UC}}(m_i) = c_{GW}$$

After decrypting the resultant value, the problem turns into obtaining individual meter readings from their summation, which is obviously impossible.

$$Dec_{SK_{UC}}(c_{GW}) = \sum_{i=1}^{n}(m_i)$$

The same approach applies to the secure MPC-based protocol because all operations are performed on concealed data (distributed shares of the meter readings). In the course of operations, what the UC can obtain is only the summation of all of the meter readings.

Security Goal 2: The concealed data packets that the SMs transmit do not reflect actual meter readings. Therefore, even if an eavesdropper capture a data packet, his/her inference about the activity of the consumer will be wrong. For PHE or FHE,

in order to capture the actual reading the eavesdropper needs to know the private key that only the UC possesses. For the secure MPC-based protocol, s/he needs to know the $(n-1)$ 256-bit random numbers generated by the targeted SM as the shares from the other SMs.

Security Goal 3: Since the employed protocols are able to perform data aggregation on concealed data, they do not disclose the actual readings even to the SMs that perform data aggregation.

Security Goal 4: This threat applies to the secure MPC-based protocol because the data collection in this protocol depends on data collection requests sent by the gateway. Since all the SMs use an authentication mechanism called Elliptic Curve Digital Signature Algorithm (ECDSA) for data packets they transmit, the digital signature can be verified to confirm the identity of the packet sender and a signature cannot be forged without the private key that created that signature. In addition, the content of the packets cannot be modified without invalidating the signature, providing data integrity.

$$\{Enc_{PK_{UC}}(m_i), Sig_{SK_i}(Enc_{PK_{UC}}(m_i))\}$$

Security Goal 5: Since all data packets are timestamped, the timestamp (TS) of a packet can be checked if the packet is for the current data collection round.

$$\{< Enc_{PK_{UC}}(m_i), TS >, Sig_{SK_i}(< Enc_{PK_{UC}}(m_i), TS >)\}$$

5.4.2 Experimental Setup

We assessed the performance of our protocols using network simulator ns-3 [132], which has an implementation of the IEEE 802.11s mesh networking protocol. We created random multi-hop network topologies of size **N**, where **N** $\in$ (36, 49, 64, 81, 100). For each topology, a mesh node acts as the gateway/data collector and (**N**-1)

mesh nodes act as SMs that send their reports to the gateway periodically at every 60s [159] reflecting the worst cases scenarios. The data size generated at the SMs is assumed to be 16 bits, large enough to hold the power readings. Also, we assume that the network is synchronized with a global clock in order to have a reliable timestamp mechanism. For each $\mathbf{N}$, we created 30 random network topologies and reported the average from these random network topologies. For TCP, we set the Maximum Segment Size (MSS) to 1500 bytes [160].

There are two types of data aggregation mechanisms defined for the AMI network [50]. Both mechanisms are implemented: End-to-End (EtoE) aggregation and Hop-by-hop (HbyH) aggregation. In the HbyH aggregation, a minimum spanning tree of the network is found by the gateway meter [161] as illustrated in Fig. 2.1. The gateway meter designates parent-child relationships to each meter based on this aggregation network tree. Leaf meters in the network send their meter reading to their parent meter periodically. The parent meter aggregates its own reading with the readings received from its child meter(s). Then, it sends the resultant value to its own parent. This process goes on up until to the gateway meter. Finally, the gateway aggregates its reading with the aggregated readings received from its child meter(s) and sends the result to the UC. In the EtoE aggregation, all the meters send their reading directly to the gateway. The gateway aggregates its own reading with the readings received from the other meters and sends the result to the UC.

The secure MPC-based protocol we used in this work makes use of SSS for data aggregation. For the SV scheme, we used the implementation of [15]. The SV scheme runs on top of the *PRP* and uses the key geometry of (384/8/5). Paillier cryptosystem uses 1024 bit keys and the PRNGs generate 256 bit random numbers. ECDSA was employed to provide authentication since it is an approved signature algorithm by the US NIST [139]. We used the ASN.1 secp128r1 standard curve with SHA1, having a

key length of 256 bits. The SMs are assumed to possess all required public/private keys required for a secure communication with other SMs.

5.4.3 Baselines and Performance Metrics

In our simulations, we employed the SV scheme and the secure MPC-based protocol in both EtoE and HbyH aggregation and used Paillier cryptosystem as a baseline for comparison. The SV scheme and the secure MPC-based protocols were represented as *SV-EtoE*, *SV-HbyH*, *SMPC-EtoE*, and *SMPC-HbyH* for EtoE, and HbyH aggregation, respectively in the figures. We compare the performance of the SV scheme and the secure MPC-based protocol to the following baselines that utilize Pallier PHE. Our goal is to see how close the performance of FHE approaches to PHE.

- *Paillier & EtoE Aggregation (Pai-EtoE)*: In this test, the meter readings were encrypted with Paillier cryptosystem and sent directly to the gateway.
- *Paillier & HbyH Aggregation (Pai-HbyH)*: In this test, the meter readings were encrypted with Paillier cryptosystem and subject to data aggregation at intermediate meters.

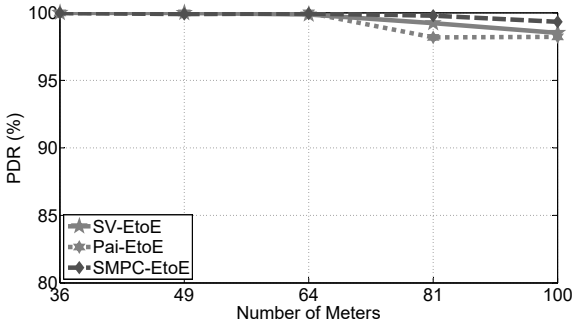
For performance evaluation, we used the following metrics:

- *Packet Delivery Ratio (PDR)*: The ratio of packets that are delivered to the gateway compared to the number of packets sent by the SMs.
- *Throughput (TP)*: The total amount of data received by the gateway per second.
- *Average Data Collection Completion Time (CT)*: The average elapsed time for receiving all the power readings from all the SMs at the gateway in one round. It is measured at the application layer and thus it takes into account the cryptosystem/Lagrange interpolation operations.

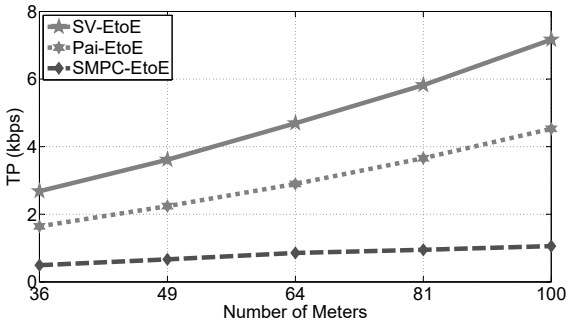
Note that we assessed the PDR only for EtoE aggregation mechanism because in HbyH mechanism, the throughput is reduced as there is in-network computation and, thus, gateway throughput is not comparable to that of EtoE.

5.4.4 Simulation Results

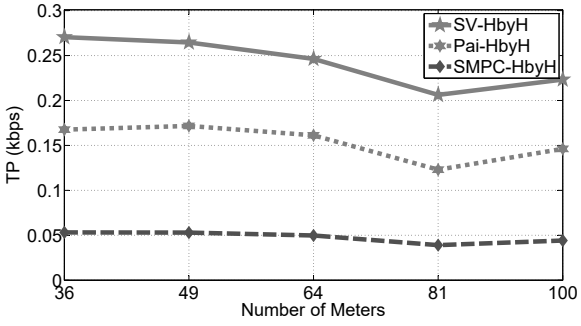
In this subsection, we present results of the simulations we conducted to compare the performance of the protocols with that of the baseline. We discuss each of the metrics separately below.



(a) The EtoE PDR values at different number of nodes.



(b) The EtoE TP values at different number of nodes.



(c) The HbyH TP values at different number of nodes.

Figure 5.4: EtoE and HbyH simulation results.

Packet Delivery Ratio

As mentioned before, we give the PDR only for EtoE mechanism. As shown in Fig. 5.4a, the PDR is almost 100% until 81-node topology for all approaches. After 64-node topology, the PDR decreases very slightly for **Pai-EtoE** and **SV-EtoE** approaches. This is due to the fact that the size of packets these approaches generate is larger compared to **SMPC-EtoE**. The larger the data size, the higher probability the more congestion occurs. Overall, increased number of meters do not deteriorate the PDR performance of the approaches significantly.

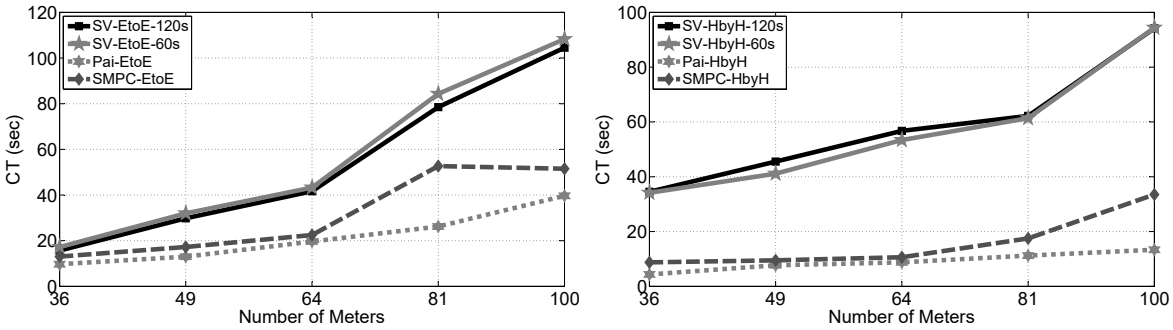
Throughput

We investigate the throughput (TP) performance to analyze bandwidth usage of the proposed approaches. The goal is to use as less bandwidth as possible to accommodate other types of traffic. We give throughput figures for both EtoE and HbyH mechanisms in Fig. 5.4b and Fig. 5.4c, respectively. Overall, it can be seen that the HbyH TP values are smaller than the EtoE TP values. This is because the gateway receives meter readings from its child meter(s) in HbyH mechanism whereas it receives meter readings from all other meters in the network in EtoE mechanism.

As shown in 5.4b, the EtoE TP values increase as the number of meters in the network increases. The approaches produce TP based on the size of data packets they generate. In this manner, **SMPC-EtoE** produces the least TP as expected because it generates smaller data packets compared to the other approaches.

We observe an interesting tendency of the TP values for HbyH mechanism given in Fig. 5.4c. For all the approaches, the values for 36 and 49-node topologies are almost fixed. Then, it decreases until 81-node topology. Finally, it increases at 100-node topology. This is related to the number of meters that send their reading directly to the gateway (e.g., 1-hop meter neighbors of it), and the packet delivery delay within

the network. As the number of meters in the network increases, the time required for the gateway to receive aggregated meter readings increases. However, the number of child meters of the gateway does not increase with the same ratio, which causes a decrease in TP. The increment at 100-node topology can be attributed to a significant increment in the number of the child meters of the gateway. When we compare the approaches, we can see that the order of the TP values are the same as in Fig. 5.4b. This order stems from the same reasons mentioned above for the EtoE TP values.



(a) The EtoE CT values at different number of nodes.

(b) The HbyH CT values at different number of nodes.

Figure 5.5: EtoE and HbyH simulation results with different periods.

Average Data Collection Completion Time

Another metric we investigated is the average data collection completion time because it is an important metric for some of the AMI applications such as demand/response. We give the simulation results for EtoE and HbyH mechanisms in Fig. 5.5a and Fig. 5.5b, respectively. From both figures, we can see that the CT values increase for all the approaches as the network grows. Also, from the figures, it can be seen that it is not feasible to collect meter readings at every 60s for **SV** approach. Therefore, we ran another simulation in which meter readings are collected at every 120s to investigate if giving more time to SV will make an impact on the CT. We used **-60s** and **-120s** suffixes to distinguish the approaches.

Pai-EtoE/HbyH and **SMPC-EtoE/HbyH** require less time to complete a data collection round than **SV-EtoE/HbyH-60s** and **SV-EtoE/HbyH-120s** approaches since size of the data packets generated by Paillier cryptosystem and PRNG is much more smaller than that of the packets generated by the SV scheme. The increased data size causes to segment the data into smaller packets based on the window size by the TCP. This increases the probability of the collision while having access to the channel to transmit the data. Each collision increases the backoff waiting times, so the collection completion time.

SMPC-EtoE/HbyH require more time than **Pai-EtoE/HbyH** because the meters need to receive c_k from the gateway to compute the shares that would be received from the other meters in the network. This procedure increases the data collection completion time of **SMPC-EtoE/HbyH**. When we compare the data collection mechanisms, we can see that EtoE mechanism takes more time to complete a round than HbyH mechanism. We attribute this to the large number of meters that want to send their readings to the same meter, i.e., to the gateway. This causes more back-off waitings compared to those in HbyH mechanism because all of the meters attempt to send their readings to the gateway at the same time. However, in HbyH mechanism, meter readings are aggregated at intermediate aggregator meters rather than only one meter and these meters receive meter readings from relatively smaller number of meters compared to the gateway collecting meter readings by using EtoE mechanism. This reduces the contention on accessing the medium, so the collisions.

As shown in the figures, **SV-EtoE/HbyH-120s** approaches complete data collection within 120s, which makes **SV** approach feasible. Both in EtoE and HbyH, 60s and 120s approaches show a very similar tendency because the meters experience the same delay since they try to send their readings at the same time. This results in the same contention on accessing the medium, consequently, the same back-off timings.

We expected to observe that **SV-HbyH-60s/120s** show better performance than **SV-EtoE-60s/120s** due to the same reasons given above for **Pai-HbyH** and **SMPC-HbyH**. However, **SV-EtoE-60s/120s** outperform **SV-HbyH-60s/120s** from 36-node topology to 64-node topology. This is due to the packet reassembly process at the intermediate meters when HbyH mechanism is employed. The *PRP* is not used for EtoE mechanism because size of the encrypted meter reading is fixed and the same for each meter. The computational overhead at the gateway is due to the data aggregation process in EtoE mechanism. This overhead exceeds the overhead of the packet reassembly process after 64-node topology. Thus, **SV-HbyH-60s/120s** outperform **SV-EtoE-60s/120s** for 81-node and 100-node topologies.

5.5 Conclusion

In this chapter, we tackled the problem of reliable and privacy-preserving in-network data aggregation in the IEEE 802.11s-based AMI network. We utilized both FHE and secure MPC for AMI applications.

We identified a new problem called the packet reassembly problem, which stems from varying aggregated data sizes of SV scheme when HbyH mechanism is employed and proposed a new protocol at the presentation layer in order to overcome this problem. Also, we proposed a new secure MPC-based protocol that can perform data aggregation with HbyH mechanism as well.

The proposed approaches fulfill several crucial goals to provide a secure and privacy-preserving communication environment. First of all, the messages are timestamped to prevent replay attacks and signed for message authentication (Security Goals 4 and 5). The approaches conceal the actual meter readings by either encrypting or dividing them into shares computed over a polynomial. This prevents the eavesdroppers from capturing the consumption information and analyzing the con-

sumers' consumption pattern (Security Goal 2). Since FHE and secure MPC are able to perform arithmetic operations on concealed data, the proposed approaches implement in-network data aggregation in order not to reveal the actual meter readings to the UC or a compromised SM (Security Goals 1 and 3).

We implemented all the approaches in ns-3 using a draft version of 802.11s for a 802.11s-based mesh network to assess their overhead. We investigated the performance under EtoE and HbyH data aggregation mechanisms. Simulation results showed that HbyH mechanism performs better than EtoE mechanism for all approaches except SV scheme for Completion Time metric. From the results, we inferred that there is a threshold network size for SV scheme to employ EtoE mechanism in periodic data collection, and that HbyH mechanism may not be a good choice for medium-scale networks due to the computational overhead brought by the *Packet Reassembly Protocol*.

For both data collection mechanisms, the secure MPC-based protocol consumes far less channel bandwidth than SV scheme consumes. In addition, an increased data collection period makes SV scheme more acceptable in terms of bandwidth usage. Also, in average data collection completion time, the secure MPC-based protocol outperforms SV scheme for both data collection mechanisms. Particularly in HbyH mechanism, the time gap between the approaches is considerable. Overall, we conclude that the secure MPC-based protocols are much more scalable than SV scheme in terms of bandwidth usage and average data collection completion time. They can also match the performance of PHE and thus can be an attractive option for preserving privacy in AMI applications.

CHAPTER 6

A SCALABLE SIMULATION FRAMEWORK FOR THE AMI NETWORK

Wireless Mesh Networks (WMNs) have numerous features such as self-organization, self-healing and multi-hop communication, which make them flexible and adaptable, e.g., the coverage area of the network can be extended by adding new nodes. However, this arises reliability and scalability issues because maintaining the throughput while expanding the network becomes a difficult challenge due to more collisions from a large number of nodes [162]. This is also a concern for our privacy-preserving protocols in ns-3 since we experienced reliability and scalability problems during the simulations in the previous chapter. Therefore, we decided to analyze the performance of the network further and detect the factors that thwart reliable communication and scalability in the Advanced Metering Infrastructure (AMI) network.

Large-scale WMNs already suffer from the reliability and scalability issues. Moreover, periodic and simultaneous packet generating applications complicate mitigating the issues. One of those applications in the AMI network collects the consumers' power consumption data from their smart meters (SMs) and communicates the collected data to the utility companies (UCs). In addition to the existing applications, the UCs and the researchers have been developing new applications. Although the UCs have a ready-to-use infrastructure, they want to test their applications' performance on a scalable simulation environment before installation because the installation of an application to each node in a large-scale network requires too much time and introduces an extra traffic to the network. Regarding the researchers, they completely depend on the simulators since there is a lack of large-scale AMI testbeds and the UCs do not allow them to use their AMI network. While there is a number of simulators such as ns-2 [98] or ns-3 [132] that can be used to simulate AMI applications,

they do not scale well beyond a hundred nodes [14, 15, 17, 28, 163–166]. Therefore, in this work, we investigate the reliability and scalability issues of a network simulator's network stack and propose a scalable simulation framework for IEEE 802.11s-based AMI applications using ns-3 network simulator. IEEE 802.11s is the mesh standard that can be used to form a WMN among SMs. We analyze the protocols running in the network stack in order to find the reasons of performance degradation as the network scales and propose several parameter tunings and modifications to the existing stack as minimum as possible as will be detailed below.

The AMI applications need reliable data communication because it is required for the reliability of the entire Smart Grid (SG). For example, the demand-response application is implemented to improve the system reliability through emergency-based direct-load control programs [167, 168]. Therefore, each power consumption reading by the SMs is critical to communicate to the UC reliably. The TCP is one of the commonly used transport layer protocols to provide reliability. However, its three-way handshake procedure and header overhead introduce a burden to the network and make it infeasible to use in large-scale networks. Also, its default congestion control mechanism significantly reduces the throughput in case of a packet loss [169]. Instead, a more lightweight protocol UDP can be used by incorporating a congestion control and an acknowledgment mechanism at the application layer. The Constrained Application Protocol (CoAP) [83] perfectly fits to this description. It is an application layer protocol developed for resource-constrained devices to have access to the Internet. Since it is reliable and connectionless, it can be used for the AMI applications. However, it employs a primitive congestion control mechanism that can keep the device idle longer than needed after several retransmissions. Therefore, we propose several simple but useful congestion control functions in this work. Also, we proposed adding a random jitter less than or equal to 1s such that there is at least 1ms between

any two meters' reading reporting times for the same reporting round to alleviate any potential congestion.

The existing studies investigating the scalability issue in WMNs [8, 96, 162] attract the attention to the importance of routing protocols in finding a reliable routing path to the destination. IEEE 802.11s standard runs the Hybrid Wireless Mesh Protocol (HWMP) at the data link layer. Contrary to the classical routing protocols, the HWMP uses Medium Access Control (MAC) addresses instead of IP addresses. Hence, an address resolution from IP to MAC is required through the Address Resolution Protocol (ARP) However, this process introduces significant packet overhead to the network [100], consequently deteriorates the reliability and increases the latency. Therefore, we propose to implement a new interface between the HWMP and the ARP which modifies the proactive path discovery process by adding a new field to the route request (RREQ) and route reply (RREP) messages.

Besides modifying the congestion control mechanism and the HWMP, there are some parameters to be adjusted. For example, the maximum hop count parameter used in the HWMP ($maxTTL$) needs to be increased since the default value (32) is not sufficient to forward a packet to its destination. Similarly, the limits for packet retransmission ($dot11MeshMaxRetries$) and packet failure ($maxPacketFail$) on a peer link need to be increased since the increased number of collisions while accessing the medium cause to quickly reach these limits.

We have implemented and tested the proposed solution in ns-3 network simulator. Instead of coding the CoAP from scratch in ns-3, we used an implementation of the CoAP in ns-3 Direct Code Execution (DCE) [170]. The DCE is a framework that enables executing user- and kernel-space protocols in either ns-3 or the Linux network stack. The simulation results have shown that the proposed simulation framework

provides a scalable simulation environment for the UCs and the researchers to test the applications they develop for the AMI network.

The rest of the chapter is organized as follows. In the next section, we provide background information about the protocols mentioned above and the ns-3 DCE. In Section 6.2, we introduce our scalable simulation framework. We present and discuss the simulation results in Section 6.3. Finally, we summarize our work and conclude the chapter in Section 6.4.

6.1 Background

6.1.1 The Constrained Application Protocol

The CoAP is an application layer protocol that has been developed for connecting the resource-constrained devices to the Internet [83]. It can easily communicate with the HTTP (Hypertext Transfer Protocol) for integration with the Web. Contrary to the HTTP, it runs on top of UDP, thereby introducing very low overhead. Also, it needs a very small header which is only 4 bytes when compared to the HTTP's header.

In the CoAP, the application end points interact based on the request/response model. It provides both reliable and unreliable message transmission. If an endpoint sends a "CONFIRMABLE" message the recipient(s) replies with an "ACK" message. The sender endpoint expects to receive the "ACK" within a specific time interval called retransmission timeout (RTO). Due to the simultaneous accesses to the medium by neighbor devices, the transmission collisions may happen. The transmission collisions can cause the message or the "ACK" to be lost on the path to their destination. Since the endpoint cannot receive the "ACK" within the expected time interval, the transmission is timeout. Hence, the endpoint retransmits the message.

The message retransmission is scheduled based on some functions called *congestion control mechanisms*. The CoAP's default congestion control mechanism simply

doubles the RTO value. Since the default RTO value is a random value between 2s and 3s, and the maximum number of retransmissions is 4 by default, the retransmissions may exhaust quickly. Therefore, more complicated but useful congestion control mechanisms were proposed [84–86, 88, 91–93]. The most accepted alternative is the CoAP Congestion Control/Advanced (CoCoA).

The CoCoA relies on round-trip time (RTT) information. Two types of RTT information are used to calculate two types of RTO estimators. RTO_{strong} and RTO_{weak} are calculated based on RTT_{strong} and RTT_{weak}, respectively. RTT_{strong} is the RTT value obtained without a retransmission whereas RTT_{weak} is the RTT value obtained after a retransmission occurs.

6.1.2 The Address Resolution Protocol

The ARP is a communication protocol that is used to fetch the MAC address corresponding to the IPv4 address of a device's targeted interface and to maintain a table of IPv4-to-MAC address pairs [171]. This table is called *ARP cache*. The ARP operations are vital for IEEE 802.11s-based networks because the routing protocol HWMP works with MAC addresses instead of IP addresses.

When a device in a network has a packet to send to another device in the same network it needs to know the MAC address of the targeted recipient because the receiver of a packet can understand if the packet is for itself by checking if the target MAC address in the packet header is equal to its own MAC address. Therefore, the sender device checks the ARP cache if there is an entry with the targeted IPv4 address. If so, the corresponding MAC address is used. Otherwise, the ARP starts a broadcast of an ARP request (ARPREQ) with the targeted IPv4 address (and a targeted MAC address of FF:FF:FF:FF:FF:FF). If the targeted interface receives the

ARPREQ it unicasts an ARP reply with its IPv4 and MAC addresses back to the requester [172].

6.1.3 The Hybrid Wireless Mesh Protocol

The HWMP is a hybrid routing protocol which is defined in the IEEE 802.11s standard [173]. It uses MAC addresses for routing operations instead of IP addresses in contrast to the most of the routing protocols.

The HWMP is a hybrid protocol because it supports two modes of operation: proactive mode (tree-based) and reactive mode (AODV (Ad hoc On-Demand Distance Vector Routing)). In this work, we tackle the proactive mode since we are seeking a way of contacting all of the Mesh Points (MPs) in the network with the minimal effort. This mode uses two methods to disseminate the routing information for reaching the root Mesh Point (MP). The first method creates routes between the root and all MPs in the network by broadcasting proactive Route Request (RREQ) messages. The second method distributes the routing information by using Root Announcement messages.

The proactive RREQ mechanism builds a tree of the network so as to enable the MPs in the network to reach the root easily. The root starts the tree building process by broadcasting a RREQ message. Any MP that receives the RREQ message creates or updates its forwarding information to the root and transmits the updated RREQ. If the "proactive Route Reply (RREP)" bit in the RREQ is set the receiver MP unicasts a RREP back to the root. Thus, not only routing information to the root is disseminated to all other MPs in the network but also a route from the root to each MP is established. If an MP receives multiple RREQs it updates its routing information to the root if and only if the sequence number of the RREQ is greater, or the RREQ contains the same sequence number but offers a better metric.

6.1.4 The ns-3 Direct Code Execution

The ns-3 DCE [170] is an ns-3 module that enables to execute existing userspace or kernelspace applications and network protocols on ns-3 or Linux network stack without needing to change the source code. It supports C/C++ applications as well as IPv4, IPv6, TCP, UDP, and DCCP protocols. Since it loads the executables similar to shared libraries, the simulations use the memory in an efficient way.

In this work, we use the ns-3 DCE to execute an implementation of the CoAP, *libcoap* [174], on ns-3 network stack. The libcoap is a C implementation of the CoAP.

6.2 The Proposed Scalable Simulation Framework

The Advanced Metering Infrastructure (AMI) network is the last mile communication network in the Smart Grid (SG). It enables the two-way communication between the consumers and the utility companies (UCs) as shown in Fig. 1.1. Thus, the UCs can provide some services that can reduce both the peak energy demand and the unit cost for the consumers such as dynamic pricing [175] and demand-response [168]. Moreover, the UCs and the researchers have been developing new applications and communication protocols for the SG since it is still evolving. However, there is a lack of an adequately large-scale environment to test the performance of their applications considering that a typical AMI network consists of thousands of smart meters (SMs). The UCs only have such a big infrastructure, but they, not surprisingly, do not allow any external access to their system. As a result, the researchers have built small-scale testbeds and made some inferences about the performance of their applications when they are applied in a large-scale network. However, these analyses may not reflect the real performance. An alternative to overcome this problem is to use simulators. The Cooja and ns-3 are the most commonly used open-source network simulators

that support mesh networks. The problem with the simulators is that they cannot guarantee a reliable and scalable testing environment for the developed applications. Therefore, in this chapter, we present a scalable simulation framework for the evaluation of AMI applications. In our work, we use the ns-3 network simulator since we build the mesh network based on IEEE 802.11s standard which is not supported by the Cooja.

We present our work in modules as in Fig. 6.1. The CoAP module needs to use the ns-3 DCE to communicate with the ns-3 network stack while the Jitter and ARP-HWMP modules are able to reside in the ns-3 stack without using any other means.

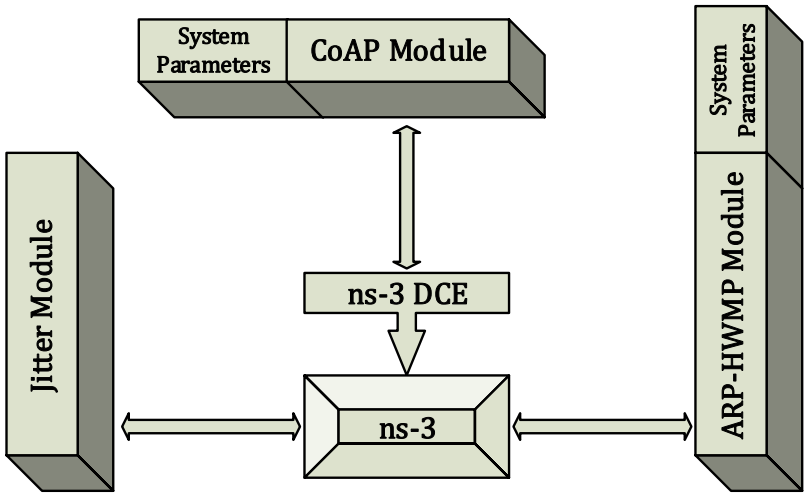


Figure 6.1: The architecture of the proposed simulation framework.

6.2.1 The CoAP Module

As we mentioned in Section 6.1.1, the CoAP's default congestion control mechanism is primitive and may cause the device to wait longer than needed for retransmission because of its doubling method. Therefore, the CoCoA (CoAP Congestion Control/Advanced) was proposed in order to better converge to the optimal retransmis-

sion timeout (RTO). The CoCoA relies on the round-trip time (RTT) information. It is expected that the RTO value converges to the optimum after some RTT measurements are obtained. However, it is vital to communicate the power consumption data in the period that it belongs. Therefore, we need an efficient and effective congestion control mechanism which is independent of the RTT information for the AMI applications.

In this work, we propose three RTO calculation methods which are defined below.

Fixed RTO: It is already a well-known problem that packet loss in wireless communication is not necessarily due to the network congestion. The packet may be dropped due to lossy paths or multi-path fading [176]. Hence, increasing the RTO at each packet loss may result in waiting longer than needed for the retransmission unnecessarily. Instead, we propose generating a random RTO value between the approximate minimum and maximum time (considering the size of the network) required for communicating the power consumption data to the gateway of the network. The randomly generated RTO value is fixed and not updated upon packet loss. In this work, we generated a random number between 4 and 6 for the fixed RTO method.

Logarithmically Increasing RTO: A function is called "logarithmically increasing" if it is increasing more slowly than any nonconstant polynomial. We propose using a logarithmically increasing function to calculate the RTO values in order to find the optimum RTO at minimum number of retransmissions possible. This function quickly approaches to the optimum RTO in the case of large RTTs. In this work, we propose two logarithmically increasing functions given below:

- **Function 1 (Func1):** $5.24 * ln(x) + 4$
- **Function 2 (Func2):** $log_{1.24}(x) + 4$

where x is the number of retransmissions.

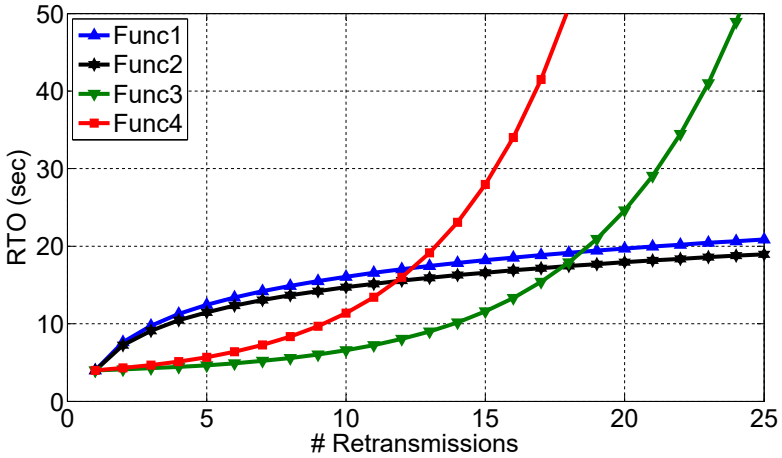


Figure 6.2: RTO calculation functions.

Exponentially Increasing RTO: In contrast to the logarithmically increasing functions, exponentially increasing functions increase more quickly than any polynomial. An exponentially increasing function is time-efficient in the case of small RTTs. The main drawback is that it requires more retransmissions than a logarithmically increasing function does in the case of large RTTs. We propose the exponentially increasing functions given below:

- **Function 3 (Func3):** $e^{(x-4)/5.24} + 3.436$
- **Function 4 (Func4):** $1.24^x + 2.76$

where x is the number of retransmissions.

The initial RTO value is randomly generated as in the default congestion control mechanism when the proposed functions except the **Fixed RTO** are employed. We used constant values in the functions to start the RTO values at 4s after the first timeout for a fair comparison as shown in Fig. 6.2.

6.2.2 The Jitter Module

In addition to the aforementioned changes and methods, a small and random jitter can be added to the data reporting time. Since the metering devices report their power consumption data simultaneously, medium access collisions are highly-likely to happen [14, 15, 17, 28]. Adding jitter can decrease the probability of simultaneous medium access attempts, consequently the packet losses.

6.2.3 The ARP-HWMP Module

The simultaneous data reporting is not the only factor that causes packet losses. Since the AMI applications make use of IP-based technologies, they use IP addresses to identify any device that is connected to the network, so the destination device. However, they need to find the corresponding physical address (MAC address) of the destination device before data transmission. ARP is used to fetch the destination's MAC address. However, as explained in Section 6.1.2, the ARP broadcasts the ARP requests, and this results in network congestion and an increase in medium access collisions. To overcome this broadcast storm problem, we replaced the default ARP with an efficient piggybacking-based ARP (PARP) [99–101] which takes advantage of proactive Route Request (RREQ) and proactive Route Reply (RREP) messages in the HWMP (Hybrid Wireless Mesh Protocol) which is the default routing protocol of the IEEE 802.11s standard.

The PARP modifies proactive RREQ and proactive RREP messages and creates another interface between the HWMP and the ARP. The modified RREQ message contains the IPv4 and corresponding MAC address of the root (the gateway of the network) whereas the modified RREP message contains the IPv4 and corresponding MAC address of the node that received the RREQ message. When a node receives an IPv4-MAC address pair, it adds this tuple to its ARP table. Thus, MAC address

resolution is handled during the proactive path discovery process (the routing tree creation), i.e., the ARP request messages to find out the MAC address of the gateway are completely eliminated while building the mesh network at the link layer. Moreover, a route from each node to the root and a route from the root to each node are established.

The PARP eliminates the ARP request broadcast and significantly reduces the number of packets traveling through the network. However, it cannot decrease the minimum number of hops required for a packet to arrive to its destination. There is a parameter called Time-to-Live (TTL) which is defined in the HWMP. It restricts the maximum number of hops that the packet is allowed to be forwarded. In case that the default value of the TTL parameter is less than the minimum number of hops from a node to its destination, the packet cannot be delivered to the destination. Therefore, we adjusted the TTL parameter based on the network size.

The HWMP uses the Link Management Protocol (LMP) [177] to discover the nodes that are in the communication range and to keep track of the links between the peers. It provides transmission reliability at the medium access layer in a similar way that is used for reliable transport layer protocols. When a device transmits a packet it waits for an ACK. If it does not receive an ACK within a specific time interval the packet is retransmitted. A parameter (*dot11MeshMaxRetries*) restricts the number of retransmissions. Moreover, another parameter (*maxPacketFail*) limits the maximum number of packet transmission failures before closing the link. In our analysis, we revealed that the value of these parameters are vital for the scalability of the simulation environment. Therefore, we fine-tuned the value of both parameters based on the network size.

6.3 Performance Evaluation

6.3.1 Experimental Setup

We developed the proposed simulation framework in the most commonly used network simulator ns-3 [132], which has a draft implementation of IEEE 802.11s standard which is a wireless mesh standard that connects wireless hosts in a multi-hop fashion. The underlying MAC protocol is IEEE 802.11g. The ns-3 does not have an implementation of the CoAP, so we used the libcoap [174] which is a C implementation of the CoAP. Since we needed to run an executable, we used the ns-3 DCE module.

The experiments were conducted on a grid topology containing 1024 nodes (32 nodes by 32 nodes) with a distance of 100m between each neighbor nodes. The gateway was placed at one of the corners of the grid.

We assumed that the smart meters report their 512-byte power consumption readings at every 5mins [178]. We added random jitter to the starting time of each node's application such that the time difference between the starting time of any two applications cannot be more than 1s and less than 1ms. Before starting the application, the nodes are given 1min to create links to their neighbors.

We removed the maximum retransmission constraint in the CoAP. We chose *maxTTL* to be 70, and *dot11MeshMaxRetries* and *maxPacketFail* to be 10. Also, we removed the maximum number of data retransmissions constraint in the TCP for a fair comparison with the proposed CoAP.

6.3.2 Baselines and Performance Metrics

In the simulations, we used the UDP, TCP and CoAP as baselines to demonstrate the importance of the transport protocol in reliability and scalability. We integrated these protocols into the proposed framework for a fair comparison. Note that the CoAP

modifications are not included in the framework for these simulations. We used the following metrics to compare their performance with that of the RTO calculation functions given in Section 6.2:

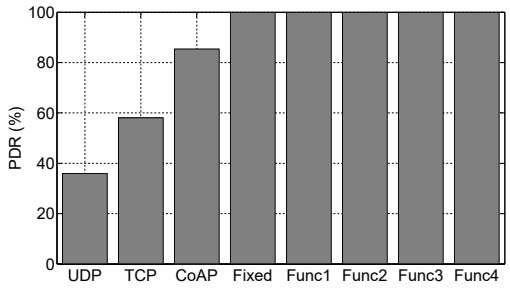
- **Packet Delivery Ratio:** This is the ratio of the number of smart meters that are able to communicate their reading to the gateway of the network to the total number of smart meters to report in the network.
- **Throughput:** This is the amount of data received by the gateway of the network per second.

We tested the proposed RTO calculation methods for the CoAP and compared their performance inter se. We used the following metrics to measure their performance:

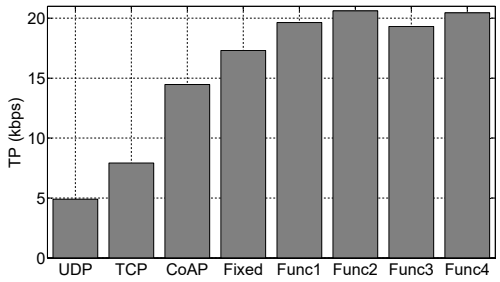
- **Redundant Receptions:** This is the total number of redundant receptions of the same packet at the application layer of the gateway. The redundant receptions mostly occur when the RTO value is less than the RTT.
- **Completion Time:** This is the time elapsed between the first transmitted meter reading and the last reading received by the gateway of the network. It includes the computational delay that is introduced by the piggybacking-based ARP.

6.3.3 Simulation Results and Discussion

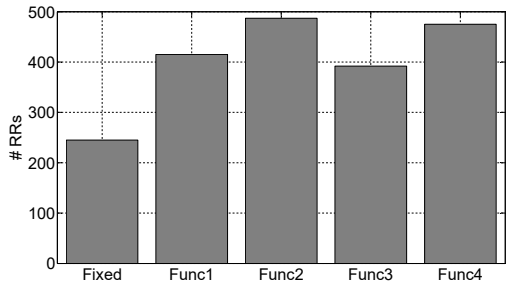
In this section, we evaluate and discuss the simulation results in two stages. First, we compared the performance of the protocols and methods mentioned in the previous subsection in terms of packet delivery ratio and throughput in order to investigate their scalability. Then, we eliminated those that are not scalable and compared the



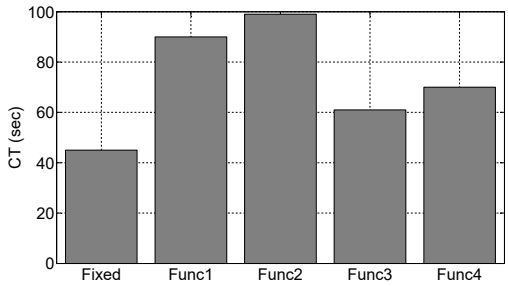
(a) The packet delivery ratio as an indication of the reliability.



(b) The throughput produced at the gateway.



(c) The total number of redundant receptions by the gateway.



(d) The completion time of all meter readings.

Figure 6.3: Simulation results.

performance of the proposed RTO calculation functions in terms of total number of retransmissions and completion time to show their efficiency.

Packet Delivery Ratio (PDR): As shown in Fig. 6.3a, the UDP achieves the lowest PDR as expected because it is an unreliable protocol which means that after it transmits a packet, it does not care about whether the packet was received by the destination or not. It is followed by the TCP which is a reliable protocol. However, its PDR is less than 60% although we removed the data retransmission constraint. This is due to the three-way handshake procedure because the TCP needs to establish a connection before data transmission. The connection is established after the communication of the three messages (SYN, SYN+ACK and ACK) between the peers. The increased network congestion interferes this procedure, and the connection cannot be established. However, the CoAP does not need a connection but provides

reliability with ACK messages and handles the network congestion by means of the congestion control mechanisms. Thus, it outperforms the TCP with a PDR of 85%. Since it is allowed to retransmit a data packet 4 times at most, and its default congestion control mechanism can exceed this limit easily, the CoAP cannot scale as the network grows. Therefore, we removed this limit and proposed five novel RTO calculation functions. Hence, they can provide reliability by achieving a PDR of 100%.

Throughput (TP): The data traffic that we use for this study is expected to produce approximately 14 kbps throughput at the gateway in the end of a meter reading collection period ((1023 SMs * 512 bytes * 8 bits/byte) / (1000 bits/Kbit) / (300 seconds per data collection period)) when assumed that there is no redundant receptions at the gateway. In Fig. 6.3b, we give the throughput produced by each method. As expected, the UDP and TCP do not generate this amount of throughput because they cannot achieve a 100% packet delivery to the gateway. However, the CoAP produces more throughput than the expected although it cannot achieve a PDR of 100%. This is due to the redundant receptions of the same packet at the gateway. Finally, we can make the inference that the RTO calculation functions in the modified CoAP provides reliability at the expense of high bandwidth consumption. When compared to each other, it can be seen that the fixed RTO calculation function prominently produces the least throughput. This can be attributed to the fact that the fixed RTO function generates random RTO values in a larger interval (4s and 6s) than the CoAP does (2s and 3s). This both decreases probability of the medium access collisions and prevents from premature RTO values which are less than the RTT.

Redundant Receptions (RRs): In the analysis of the simulations, we revealed that the functions made a SM to cause five redundant receptions at most. As can be

seen in Fig. 6.3c, the **Fixed RTO** function causes less redundant receptions. This is due to the same reason given in the throughput discussion. In our further analysis, we realized that the **Func3** causes approximately 20% more retransmissions than the **Func1** does although it generates less redundant receptions at the gateway. This indicates that the **Func3** incurs more packet losses when compared to the **Func1**.

Completion Time (CT): As shown in Fig. 6.3d, the **Fixed RTO** function takes the least time to complete a meter reading reporting period from all SMs to the gateway when compared to the others. This is due to the fact that the randomly generated RTO value is fixed and greater than the RTT. Since the RTO value is fixed, the waiting time for the next retransmission does not increase. Similarly, since the **Func3** and **Func4** are almost fixed and slowly increasing in the beginning, their waiting time for the next retransmission and the increase at the waiting time are far less than those of the **Func1** and **Func2**. Consequently, the exponentially increasing RTO functions complete the meter reading reporting period earlier than the logarithmically increasing RTO functions can.

6.4 Conclusion

In this chapter, we have investigated a scalable simulation framework for evaluation of the IEEE 802.11s-based Advanced Metering Infrastructure (AMI) applications. We analyzed the ns-3 DCE's network stack and revealed the drawbacks that hinder the reliability and scalability. We proposed to use a modified version of the Constrained Application Protocol (CoAP) for the scalability. Moreover, we replaced the Address Resolution Protocol (ARP) with an efficient and piggybacking-based ARP (PARP) to dissolve the address resolution traffic in the proactive path discovery process of the Hybrid Wireless Mesh Protocol (HWMP). Furthermore, two critical parameters of the Link Management Protocol were updated so as to increase one-hop data transmission

reliability. Simulation results indicate that the proposed framework can successfully scale provided that the protocol parameters highlighted in this chapter are adjusted based on the network size. In addition, the **Fixed RTO** function consistently outperforms the other proposed functions.

CHAPTER 7

A REALISTIC PERFORMANCE EVALUATION OF PRIVACY PRESERVING PROTOCOLS FOR THE AMI NETWORK

It has been reported that cities are responsible for consuming 60-80% of energy generated worldwide [179], and contributing CO_2 emissions significantly [180]. To reduce CO_2 emission and use energy resources efficiently, the cities should be designed and built in a *smart* way [181]. A smart city can monitor and integrate the aspects of its critical infrastructures such as transportation, communication and power infrastructure, thereby optimizing resource use and maximizing benefits for its citizens [182]. From the point of power management, smart cities require a *smart* grid that can monitor generation, transmission, distribution and even consumption of power so that it can take required actions for a more stable, reliable and scalable grid, consequently a more consistent grid for a smart city. The Smart Grid (SG) is the integration of sensors and advanced communication technologies with the conventional power grid, which enables monitoring real-time power consumption, thereby billing accurately, efficient power generation, and enabling demand response [183, 184].

The Advanced Metering Infrastructure (AMI) network has been one of the most important components of the SG that relates to ordinary consumers. It is the last mile communication network in the SG, which enables two-way communication between the consumers and the utility companies (UCs). Thus, it can collect fine-grained power consumption data from the consumers and communicate them to the UCs for not only providing better demand response but also help in distribution state estimation and increase the efficiency [14].

The AMI network is comprised of smart meters, communications networks, and data management systems [185,186]. The communication within the network of these components can be carried out via different technologies such as power-line commu-

nication (PLC), IEEE 802.16 (WiMAX), 802.11 (WiFi), and 802.15 (Bluetooth and ZigBee) [6]. The AMI network serves thousands of consumers, so it covers large areas. Therefore, using wireless technologies to build the AMI network is more feasible since they require far less cabling work and infrastructure, deployment and maintenance costs [187–190].

Building wireless mesh networks (WMNs) is the best option to communicate through such large area networks with the least effort because it makes end-to-end communication easy by distributing the routing task among the intermediate nodes located between the communicating hosts. IEEE 802.11s [148] and ZigBee [191] are the two most commonly used open standards that can support WMNs. IEEE 802.11s is a wireless local area network (LAN) standard and an amendment to the IEEE 802.11 standard for mesh networking whereas ZigBee is an IEEE 802.15.4-based WMN stack specification for personal area networks (PANs) which require less power and cost less when compared to LANs. In addition, IEEE 802.15.4g [192] Smart Utility Networks Task Group developed a physical layer amendment to IEEE 802.15.4 in order to provide control over very large scale networks with minimal infrastructure.

Although WMNs can be used to build the AMI infrastructure, the power consumption data transmitted through the air is highly vulnerable to be eavesdropped. To overcome this problem, the data can be encrypted (confidentiality) and signed (integrity and authentication) before transmission. However, the secure communication alone is not sufficient to provide consumer privacy in the AMI network because the utilities can still have access to the consumers' actual power consumption data. In particular, with the availability of smart meters, utilities collect data more frequently (e.g., from seconds to minutes [4]) as opposed to once a month for billing purposes. It has been shown that such fine-grained power consumption data can be used to

infer the consumers' instantaneous activities [146]. Therefore, recent years witnessed a variety of solutions to preserve the consumers' privacy [50].

One solution is to aggregate meter readings in the network before sending them to the UCs, either at each hop (Hop-by-Hop (HbyH) aggregation) or at the gateway of the network (End-to-End (EtoE) aggregation) [17,26,27,193]. In this way, individual readings which are encrypted will not be exposed to any other meter and only an aggregate reading will be available at the gateway. However, both methods have their own risks. In HbyH aggregation, actual meter readings of the consumers at the edge of the network are revealed to the one-hop-ahead consumers whereas in EtoE aggregation, each meter's reading can be disclosed in case of a collusion with the gateway. To eliminate this flaw, several privacy-preserving protocols were proposed for the AMI network [14,15,17]. The unique feature of these protocols is the ability of performing arithmetic operations on concealed data of which homomorphic encryption and secure multiparty computation [24] are the two most known examples.

While homomorphic encryption and secure multiparty systems are bringing privacy-preserving features, their computation and communication costs are typically high, which may not fit well in to the characteristics of the AMI network. As future AMI systems are expected to run various applications in addition to billing (e.g., demand response, outage management, privacy-preservation, etc.), their performance abilities need to be evaluated under different conditions. In particular, since privacy has been a hotly debated issue for the AMI, the overhead of privacy-preservation on the underlying wireless AMI infrastructure is crucial.

To this end, there has been some efforts to assess the performance of IEEE 802.11s-based AMI networks that run secure and privacy-preserving protocols [14, 15, 17, 26–28]. However, to the best of our knowledge, there is not any comprehensive work in the literature that evaluates the performance of these protocols in a realistic

testbed under various conditions such as transport/application layer protocols and data aggregation protocols. This is not surprising because both the actual AMI networks are not accessible to the researchers, and there is not any realistic AMI testbed that is available to them. Therefore, one of our motivations in this work is to build a realistic AMI testbed and make it accessible to the researcher and educators.

In addition to IEEE 802.11s, ZigBee is another well-known wireless mesh networking technology that can be used to build the AMI network. It has different features than IEEE 802.11s mesh and considered more lightweight that would consume less resources in terms of computation and communication. Similarly, the literature lacks performance evaluation work of these protocols in a ZigBee-based AMI network testbed as well. Therefore, in this work, we aim to compare the performance of ZigBee- and IEEE 802.11-based AMI networks under different privacy-preserving protocols. In addition to standard encryption, we utilize the fully homomorphic encryption (FHE) [18–21] and secure multiparty computation (secure MPC) [22–24]. We test them with both UDP and TCP. In addition, we integrate the Constrained Application Protocol (CoAP) into the FHE-based protocol to provide a reliable but lightweight communication. Also, we run them with both EtoE and HbyH aggregation mechanisms to investigate the effect of data collection method used. The performance comparison is specifically done under actual testbed that was built at Florida International University Engineering Center. Note that we opt not to utilize simulation as it has been shown that the ns-3 simulation results for AMI networks do not match well with the testbed results [26, 27]. We exclude IEEE 802.15.4g in this work because 802.15.4g-compliant devices on the market operate at sub-1GHz channels [194–197] in order to provide long-range connectivity which would prevent us testing multi-hop data communication which is one of the most essential features of a mesh network within a limited area (e.g., a building). In addition, in some of the

countries, the sub-1GHz bands are not licensed exempt, which will make it impossible to be tested by researchers.

The testbed consists of 20 nodes. We attached a Digi XBee S2C [198] to a Raspberry Pi 3 (Raspi3) [199] through one of its USB ports via Sparkfun's XBee explorer dongle [200] to imitate a smart meter [201]. For IEEE 802.11s-based testbed, we used the Protronix wireless USB adapter having a Ralink RT3070 chipset [202]. Our contributions can be listed as follows. To the best of our knowledge, this is the first study using ZigBee technology to build an AMI network testbed. We test the FHE and secure MPC-based privacy-preserving protocols running on top of UDP and TCP on this testbed. The CoAP is introduced to increase reliability of the FHE-based protocol. We compare and contrast the abilities of ZigBee and IEEE 802.11s standards in terms of AMI applications. Finally, we would like to note that the testbed will be open to researcher and educators for experimentation [203].

We conducted comprehensive experiments with the privacy-preserving protocols in our AMI testbed by varying several system (both hardware and software) parameters. Experiment results indicate that the secure MPC-based protocol is a better option in case it is equipped with EtoE data aggregation and TCP in an IEEE 802.11s-based mesh network. In addition, the FHE-based protocol with CoAP is also promising in terms of message delivery. Zigbee, on the other hand, is not suitable for heavy privacy-preserving protocols.

The rest of the chapter is organized as follows. In the next section, we present the components of a privacy-preserving and secure AMI network. In Section 7.2, we give the details about the ZigBee- and IEEE 802.11s-based AMI network testbed development. We present and discuss the experiment results in Section 7.3. Finally, we summarize our work and conclude the chapter in Section 7.4.

7.1 Components of a Privacy-Preserving and Secure AMI Network

In this section, we present a background information about the AMI network, the tested privacy-preserving protocols along with the implemented data aggregation mechanisms and authentication components.

7.1.1 Network Model

We considered an AMI communication network consisting of smart meters that are connected via a WMN (IEE 802.11s and ZigBee) with a gateway serving as a relay between smart meters and the UC. The smart meters measure mainly the real-time electrical energy consumption of the consumers in addition to power quality and instantaneous values such as voltage and current at their connection points. A typical infrastructure for the considered AMI in this chapter is shown in Fig. 1.1.

IEEE 802.11s standard allows mesh networking among the meters through 802.11 MAC/PHY layer standard [148]. It uses the HWMP as its default routing protocol to find a multi-hop path towards the destination. The nodes in 802.11s WMN are given names based on their roles. All nodes are considered as Mesh Points (MP) and are able to provide connectivity at the data link layer between other MPs. If an MP also provides connectivity to the Internet, it is termed a Mesh Portal Point (MPP). In our mesh network, the gateway is the MPP which collects meter readings that are sent by the MPs (i.e., meters) via multi-hop routes.

ZigBee is the other wireless mesh networking technology we used in this work. It is based on IEEE 802.15.4 MAC/PHY layer standard [51]. Although the standard does not define the mesh networking, ZigBee stack employs some network layer protocols such as Ad hoc On-Demand Distance Vector (AODV) and Routing Protocol for

Low-Power and Lossy Networks (RPL) to implement mesh networking. In a ZigBee network, one of the nodes must be the ZigBee coordinator which is the root of the network and refers to the gateway in our mesh network. The rest of the nodes can be either a ZigBee router or a ZigBee end device. In our mesh network, the rest of the nodes are ZigBee router because they should be capable of forwarding data packets towards the destination [204].

There are some differences between TCP/IP and ZigBee stacks. ZigBee does not have a transport layer between application and network layers. Hence, it implements application layer acknowledgments for end-to-end reliable communication. However, it retransmits the data, only up to 2 times until it receives an acknowledgment. Also, MAC layer acknowledgments are used to provide reliability between neighbor nodes [205].

7.1.2 Data Aggregation Mechanisms

In-network data aggregation is used to aggregate the meter readings and send an aggregated power consumption information instead of the individual readings to the UC. Typically, two data aggregation mechanisms are implemented in the AMI network: Hop-by-Hop and End-to-End aggregation.

Hop-by-Hop Aggregation (HbyH)

A minimum spanning tree of the network is found, and the root of this tree is designated as the gateway/data collector of the network. After that, the parent-child relationships are assigned to each meter. Leaf meters in the network send their consumption readings to their parent meter. The parent meter aggregates its own reading with the readings from its child meter(s). Then, it sends the resultant to its own parent. This process goes on up until to the gateway meter. Finally, the gateway

aggregates its reading with the readings from its child meter(s) and sends the result to the UC.

End-to-End Aggregation (EtoE)

In this mechanism, the aggregation is not performed at the intermediate meters. Instead, every packet is aggregated when it arrives at the gateway which acts as the data collector. One of the meters acts as the gateway meter and all other meters send their readings to the gateway. The gateway aggregates the readings from all other meters with its own reading and send the resultant to the UC.

7.1.3 Fully Homomorphic Encryption - The Smart-Vercauteren Scheme

Homomorphic encryption enables performing arithmetic operations such as addition and multiplication on encrypted data. Roughly speaking, homomorphic encryption can be divided into two: partially homomorphic encryption (PHE) and fully homomorphic encryption (FHE). The former allows to perform only one operation (either addition or multiplication) whereas the latter allows both addition and multiplication.

While PHE cryptosystems have been known for years, FHE cryptosystems are gaining more attention in recent years since the work of Gentry in 2009 [47]. In his FHE scheme, a "hint" about secret key is placed into public key. Even if the hint is not enough to decrypt a ciphertext, it can be used to process the ciphertext. To this end, Smart and Vercauteren [40] presented an FHE scheme which has both relatively small key and ciphertext size. Smart-Vercauteren (SV) scheme consists of key generation, encryption, decryption, homomorphic addition/multiplication and recryption functions. The recryption function removes noise in the ciphertext without

decrypting it and the cleartext is kept unchanged. The function utilizes the hint whose pieces are distributed into an array in public-key randomly. In the lack of such a function, we are limited to a fixed number of homomorphic operations. When we exceed this number of operations the ciphertext becomes undecipherable. In this way, the ciphertext can be used for more addition and multiplication operations.

SV scheme is a member of public-key cryptography family, so it generates a public-secret (private) key pair. Key generation is different in SV since some portion of the public-key is used for recryption purposes. In addition, the key size in SV is in the order of kilobytes which is much higher than the keys in traditional schemes which are in the order of bits. The keys are generated considering three important parameters: The number of bits ($|B|$) used to create random variables for the coefficients of the polynomials that are used to generate a *hint*, the number of *shares* (S_1) which the hint is divided into and the number of *cells* (S_2) in the array containing the shares of the hint. Each tuple ($|B|/S_1/S_2$) is called a "key geometry". We used (384/8/5) geometry in this work.

There had not been a publicly available implementation of any FHE scheme until when Perl et al. [39] presented a working implementation of the SV scheme [40]. The SV implementation in [39] was meant only for single bits. However, we need multi-bit operations that will be used on smart meters. Therefore, we used the algorithms in [15, 17, 28] to support multi-bit addition and multiplication. These algorithms include some functions from external libraries. We also note that data types used in these operations are large integers since cryptographic operations need more complicated and larger data types than primitive data types. In order to implement the functions, the GNU Multiple Precision Arithmetic Library (GMP) is used, which further relies on another library called Fast Library for Number Theory (FLINT).

7.1.4 Secure Multiparty Computation

Secure multiparty computation (secure MPC) is based on dividing a secret into shares and distributing them amongst a group of participants such that the secret cannot be reconstructed unless a certain number of the participants collude. One of the most commonly used secret sharing scheme is Shamir's Secret Sharing (SSS) [149].

In SSS, we assume that there are n nodes in the network and all computations are done in a finite field $\mathbb{Z}_p$, where p is a prime number. Let r_i be the private secret of node i. Node i chooses a unique point $x_i \in \mathbb{Z}_p$ other than zero and selects an $(n-1)$ degree random secret sharing polynomial $f_i(x)$ with $f_i(0) = r_i$. It sends its unique point x_i to all other nodes and receives share values $f_j(x_i)$ computed by the other $(n-1)$ nodes. Then, it computes $F(x_i) = \sum_{k=1}^{n} f_k(x_i)$. These steps are done by all n nodes and $F(x_i)$ values are sent to the gateway. The gateway can construct an $(n-1)$ degree polynomial $g(x)$ by using the $F(x_m)$ values along with Lagrange interpolation, where $m \in \{1, ..., n\}$. The constant term of $g(x)$ is the aggregation of all individual n private secrets.

Tonyali et al. [17] proposed secure MPC-based data aggregation protocols for EtoE and HbyH aggregation in the AMI network that does not require exchanging the shares, so reduces communication overhead. In this work, we test and compare the performance of these secure MPC-based protocols in the testbed we built.

The tested secure MPC-based protocol for EtoE data aggregation works as follows. In the very beginning of the protocol, each pair of two-meters agrees upon a shared key, and this key is used as a feed to locally compute the shares that would be received from the other SMs if we used a classical secret sharing scheme. This is done only once, so we assume that these keys are either preloaded on the SMs or shared via a key-exchange protocol such as the Diffie-Hellman [156] in our experiments. In each data collection round, the gateway chooses a round value which is greater than the

value used for the previous round and unicasts it to each SM in the network. Each SM computes its own shares locally that would be received from the other SMs. A polynomial is constructed by using these values and the Lagrange polynomials. Each SM computes its own share based on this polynomial, sums all shares up and send the result to the gateway. The gateway constructs a polynomial over the received values. The constant term of this polynomial is the aggregated value of all SM readings. Please refer Chapter 5 for more detail.

The HbyH aggregation version of this protocol slightly differs from its EtoE aggregation version in the way of computing the Lagrange polynomial for the gateway. This polynomial is computed by each meter instead of the gateway, and each meter multiplies its total share by the associated Lagrange polynomial and sends the result to the parent. This goes on up until to the gateway. Finally, the gateway aggregates all received values with its own multiplied total share. The aggregated value is the aggregation of the readings from all meters.

7.1.5 Digital Signatures and Public Key Certificates

Although the FHE and the secure MPC can provide privacy, this is not sufficient for a secure communication system. A secure communication system should be able to guarantee the integrity of the data in transit between hosts, which means that it enables to detect if the data has been changed on the way to the destination. In addition, the receiver of a message should be able to verify the source of the message (sender authentication) [206]. Such a system can be implemented by integrating message authentication codes or digital signatures into the privacy-preserving protocols. The message authentication codes require to have a separate pair of symmetric keys for each communicating peers. Therefore, we used digital signatures in this work.

Digital signatures are based on public-key cryptography (PKC). In PKC, two mathematically-linked keys are used: public and private keys. While keeping the private key secret, the public key is shared with all peers to communicate. To create a digital signature, a one-way hash of the message is created, and then encryption is applied with the *private key*. The encrypted hash is appended to the original message and sent to the destination. The receiver can check the integrity of the message and verify the source of the message by using the encrypted hash that is appended to the message and the public key of the sender [207].

To avoid fake public keys and rely upon the signatures, the key pairs are created by a trusted third party which is called certificate authority (CA). The CA creates the key pair, signs the public key with its private key in a certificate (public key certificate) and send them to the requester entity. In this way, communicating peers can verify that the received public key belongs to that particular entity. Due to some reasons such as private key compromise, malicious activity, the CA can revoke a certificate. In such a case, the relevant public key should not be used for any encryption or verification operation. Hence, the CA adds this certificate's serial number into a public list which is called certificate revocation list (CRL). Therefore, an entity should check the public key of another entity to communicate against the CRL [208].

7.1.6 Data Transport/Application Protocols

In this section, we present the transport/application layer protocols in the TCP/IP stack we used in this work.

User Datagram Protocol (UDP)

UDP is a connectionless transport layer protocol that does not guarantee message delivery. Once a UDP message is sent no state is retained by the protocol. Hence, it

is not exposed to excessive header overhead. Thus, it avoids processing delays in the protocol stack.

Transmission Control Protocol (TCP)

TCP is a stream-based and connection-oriented transport layer protocol that provides reliable, ordered, and error-checked message delivery. TCP needs to establish a connection before data transmission. To provide a reliable communication channel, it keeps track of state of the connection. Therefore, it has a bigger header when compared to UDP.

Constrained Application Protocol (CoAP)

CoAP is an application layer protocol that has been developed for connecting the resource-constrained devices to the Internet [83]. CoAP runs on top of UDP, thereby introducing very low overhead. It provides both reliable and unreliable message transmission. If an endpoint sends a *CONFIRMABLE* message the recipients reply with an *ACK* (acknowledgment) message. The sender endpoint expects to receive the *ACK* within a specific time interval called retransmission timeout (RTO). If the *ACK* is not received before RTO, the transmission is timeout. Hence, the endpoint retransmits the message. The message retransmission is scheduled based on some functions called congestion control mechanisms. The CoAPs default congestion control mechanism simply doubles the RTO value.

7.2 The AMI Network Testbed Development

In this section, we describe our IEEE 802.11s- and ZigBee-based AMI network testbed. We give the technical details about the electronic components and the software tools we used to build the testbed.

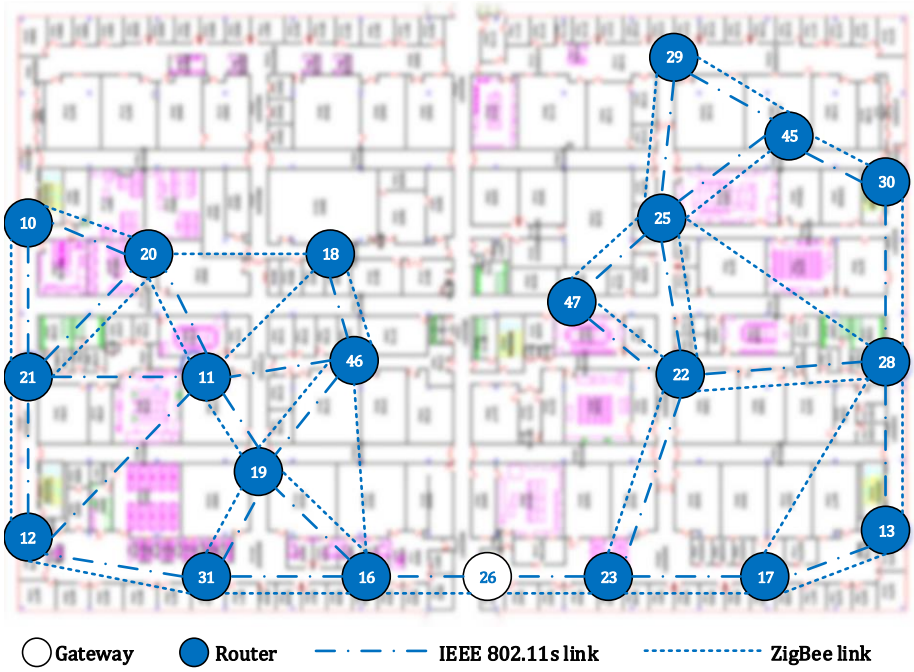


Figure 7.1: The layout of the floor which hosts the testbed.

We built the testbed in a 64m by 96m block on the third floor at Florida International University Engineering Center. Fig. 7.1 provides the representation of the network with IEEE 802.11s and ZigBee links. We tried to place the nodes as distant as possible from each other. In this figure, the white circle represents the gateway of the testbed while the blue ones represent the ordinary meters. The testbed is comprised of 20 nodes. Please, note that node #45 is the node that we used as a relay to access the other nodes. Since it is a part of the mesh network, it can forward network packets to its neighbors, but it is not capable of generating data packets.

7.2.1 Nodes

The nodes should be capable of processing as well as forwarding data packets, i.e., networking. To this end, we used a Raspberry Pi 3 (Raspi3) shown in Fig. 7.2a with each node as a processing unit. Raspi3 includes the Broadcom BCM2837 system-on-chip (SoC) having four high-performance ARM Cortex-A53 processing cores running

at 1.2GHz with 32KB Level 1 and 512KB Level 2 cache memory, which is built specifically for Raspi3 and linked to a 1GB LPDDR2 memory module. Also, it includes an 802.11n supporting WiFi module. Raspbian [209] is Raspberry Pi Foundation's officially supported operating system which is an optimized Debian distribution based on Linux kernel. Please note that the kernel version should be greater than 4.1 to support mesh point mode in the IEEE 802.11s-based AMI testbed.

(a) Raspberry Pi 3.

(b) Sparkfun XBee Explorer Dongle.

(c) Protronix USB WiFi Adapter.

(d) Digi XBee S2C.

Figure 7.2: Major components of the testbed nodes.

We integrated a complete authentication mechanism in the communication between meters based on certificates. Specifically, we created a root certificate authority (CA) and made it authorize an intermediate CA in order to avoid a single point of failure in case the root CA is compromised. We used the intermediate CA to create the public key certificates and the private keys on behalf of the root CA. In addition, we maintained a CRL for the certificate authentication. As shown in Fig. 7.3, the

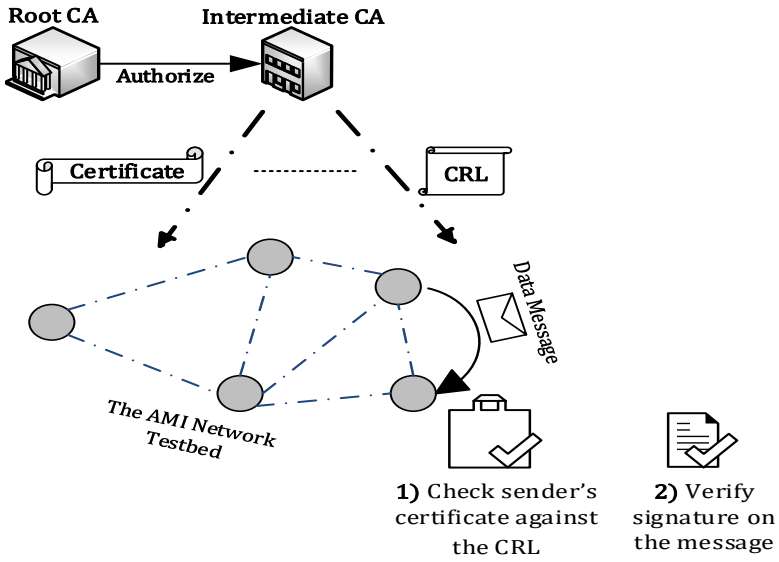


Figure 7.3: A complete authentication mechanism for secure data communication.

intermediate CA issues both the public key certificates and the CRL for all meters in the AMI network testbed. When a meter sends a data message to another meter, the recipient, first, checks the sender meter's certificate against the CRL to see if it is revoked for a reason. If the certificate is revoked the message is discarded. If it is not, then the signature on the message is verified with the sender's public key. If the signature is valid, then it is forwarded to the relevant application. Otherwise, it is discarded.

7.2.2 Gateway

The gateway is an interface of the AMI network opening to the Internet. Hence, it is responsible for the communication between the meters and out-of-network entities. For example, it collects power consumption data from the meters and communicate them to the UCs. The other nodes in the testbed send their readings to the gateway periodically. This is implemented by using the data aggregation mechanisms defined

in Section 7.1.2. In addition, the certificate authority contacts the gateway when there is a new certificate or CRL issued. Then, the gateway distributes the CRL to all of the meters or sends the certificate to its owner in the network. Since all of these extra tasks require much more computational power and storage capacity, it is wiser to equip the gateway with more powerful devices.

7.2.3 Communication Interfaces

The IEEE 802.11s-based Mesh Networking

All of the nodes in the IEEE 802.11s-based testbed are run as mesh points. Although Raspi3 has a wireless module, we need to use an external device to make it a mesh point because the built-in module driver (Broadcom brcmsmac) does not support mesh mode [210]. Therefore, we use a Protronix USB WiFi dongle shown in Fig. 7.2c along with RasPi3. The WiFi dongle is a high gain wireless USB adapter. It has a 4dBi detachable antenna and a Ralink RT3070 chipset which is run by $rt2800usb$ driver, so enables mesh networking.

In order to create a WMN, we need some tools to create and configure the network. The first tool we need is *iw* which is a configuration tool for wireless devices. Before creating and configuring the mesh by using the *iw* tool, we need to stop the $NetworkManager$ service because it interferes with the new interface for the mesh network.

After stopping the *NetworkManager* service and installing all required tools, we created a mesh interface and configured it to join a specific mesh network. We started with creating a mesh point interface. Although it is not obliged, we set the communication channel to 11. We do not run the DHCP on the nodes since we want to have a complete control on the network. Hence, we assign an IP address and the netmask

to determine the subnet for each node manually and set the interface up. Finally, the Raspi3 joined the mesh network.

In order to avoid Address Resolution Protocol (ARP) messages which creates additional traffic [101], we manipulate the ARP cache by adding the IP-MAC address pair of each node in the network.

All commands to configure and build the testbed are given in Appendix .1.

The ZigBee-based Mesh Networking

The other technology we used to build the AMI network testbed is ZigBee technology. In this testbed, we use the Digital International's XBee ZigBee S2C module shown in Fig. 7.2d for mesh networking.

XBee is a low-cost and low-power consuming wireless connectivity module as expected from a ZigBee module. It includes Silicon labs' EM357 transceiver that enables XBee to communicate in a range of up to 60m indoor/urban and up to 1200m outdoor (line-of-sight). The XBee has 15 General Purpose Input/Output ports available, but it does not have a USB port. Therefore, we employ the Sparkfun's XBee explorer dongle shown in Fig. 7.2b. This dongle enables data communication between Raspi3 and XBee thanks to FT231X USB-to-Serial converter. Also, it includes an on-board voltage regulator to supply XBee the required power to operate.

In a ZigBee-based mesh network, a node can be either a ZigBee coordinator, router or end device. In this testbed, we have one ZigBee coordinator and 19 ZigBee routers. In order to use an XBee as a coordinator or a router, we need to configure it before joining the mesh network. We use the XCTU [211] to configure the XBees. The XCTU is a graphical network configuration platform developed by Digi International Inc. The layout of the XCTU is given in Fig. 7.4.

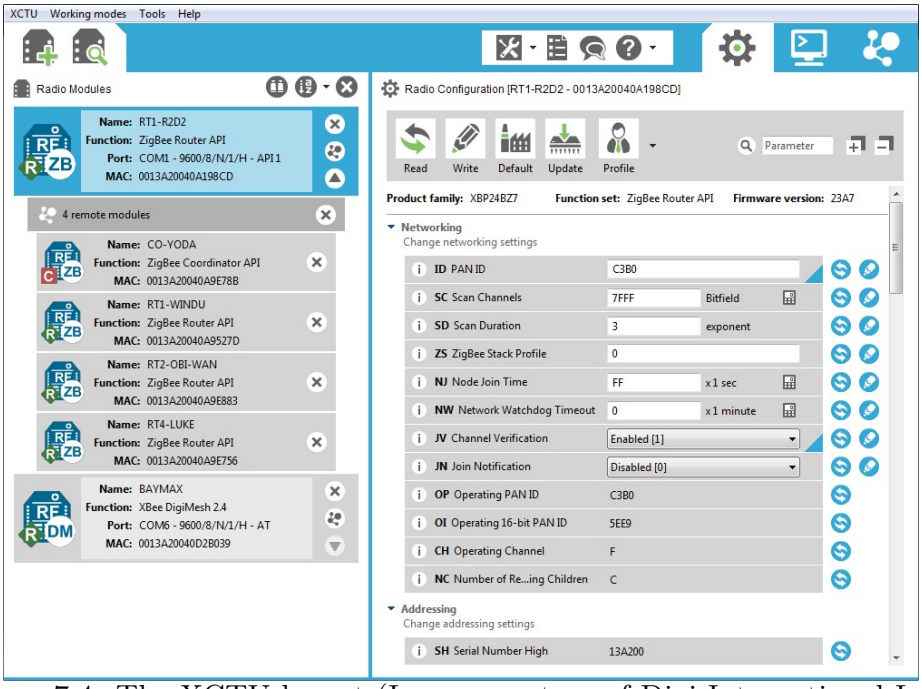


Figure 7.4: The XCTU layout (Image courtesy of Digi International Inc.).

There are three main differences between the coordinator and router in configuration. First of all, the coordinator should set the coordinator enable (**CE**) flag. Later, each device should have a unique node identifier (**NI**). Finally, the router should enable the power-on join verification (**JV**) check so that the XBee discovers 64-bit address of the coordinator when it first joins a mesh network. After the network is built, it is visualized in the XCTU as shown in Fig. 7.5.

In order to send our custom data packets that the privacy-preserving protocols generate, we need to run the XBee in API mode. Hence, we can structure the data frames to be transmitted. The frame begins with the start delimiter which is 0x7E in hexadecimal. It is followed by the most and least significant bytes of the packet length. The frame ends with the frame data and its checksum.

We encountered a communication issue between the RasPi3's Universal Serial Bus (USB) which is connected to the Universal Asynchronous Receiver-Transmitter (UART) and XBee dongle's serial interface. Although the XBees were able to com-

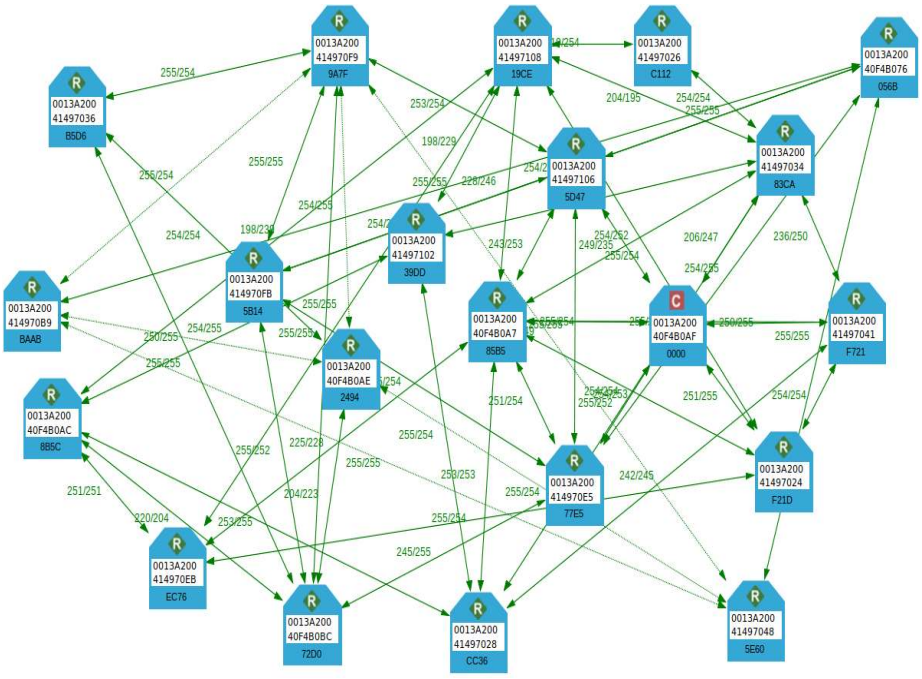


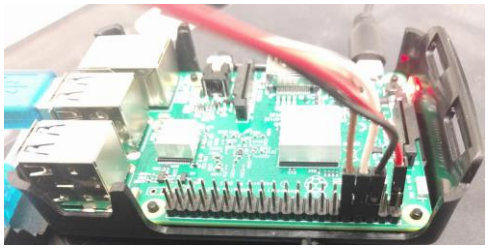
Figure 7.5: A visual representation of the testbed in XCTU.

municate to each other, which can be understood from the LED on the dongle, they were unable to transfer the received data to the RasPi3. We tried to reconfigure USB and UART parameters of the Linux kernel running on the RasPi3s. However, the problem was related to the freescale bypass application that is installed on the XBee module. To communicate through serial interface, serial communications peripheral of the freescale should be connected to the microprocessor's UART channel. To this end, the XBee module should be switched to bootloader mode first, then bypass mode. This can be implemented by issuing a 'U' character, and then a 'B' character from the RasPi3 to the XBee module.

The Raspi3 communicates with the XBee over the serial port. Therefore, we need to initialize a serial port between the devices. Fortunately, The RXTX.org team provides a Java API [212] for communicating with the XBee modules in API mode.

7.2.4 Integrating a Raspberry Pi 3 with a Smart Meter

Smart meters are the devices that make a metering infrastructure really *advanced*. Therefore, we used a hand-made smart meter which is capable of measuring voltage, current, power, sold power, and consumed power along with our RasPi3s in our testbed as can be seen in Fig. 7.6b. We integrated the RasPi3s with the meter so that power consumption values can be read from the meter and communicated to the gateway through the communication interfaces. We used an RS232-to-TTL converter to connect RS232 serial bus on the meter to the General-Purpose Input/Output (GPIO) pins on the RasPi3 for serial communication. However, we found out the mini-UART can limit the GPIO capabilities on the RasPi3. This is due to a software change in Raspbian that allows the main UART to use the GPIO pins. To this end, we needed to disable the UART for bluetooth.



(a) The GPIO pins on a Raspberry Pi 3.

(b) A smart meter integrated with a Raspberry Pi 3.

Figure 7.6: Components of an integrated AMI testbed node.

The GPIO pins have a 3V tolerance, so the RasPi3 uses TTL communication. In TTL communication, a 1 is represented with 3.3 to 5V while a 0 is represented with 0V. This is a problem since the smart meters use RS232 which is a more widely used standard in which 1 is represented with -3V through -25V and 0 is represented with 3V through 25V. Due to the differences, we needed to have a converter to convert

from TTL to RS232. The converter has to be connected to the GPIO pins such that the VCC goes to the pin 1, the RXD to pin 10, TXD to pin 8, and GND to pin 6 as shown in Fig. 7.6a. After connecting them, we executed a simple python script to read data from the smart meter and used Minicom [213] to send data. We tested the integrated system and found out that a continuous reading of data was not working effectively, so we programmed the RasPi3 so as to send a request character '?' and then the smart meter replies with the data.

7.3 Performance Evaluation

This section describes the details of the testbed setup, performance metrics and baselines and provides a thorough analysis of the performance results.

7.3.1 Experimental Setup

We first describe how we setup the testbed using IEEE 802.11s or Zigbee standards. The IEEE 802.11s module in Linux kernel requires to run some commands to build the mesh network. Since it requires to connect a monitor to each Raspi3 every time we want to build the network and run the commands given in .1, we wrote a shell script with these commands and defined it as a service to run on boot. Hence, the nodes can join the mesh network without needing physical human intervention. This is not required for the ZigBee-based mesh network because once the XBees are configured properly, they can join the mesh network automatically. However, we still need to run the data aggregation protocols manually so that all of the nodes start reporting their readings at the same time. For this purpose, we use ClusterSSH tool [214] which is able to control multiple SSH sessions from a single input window. We wrote a shell script including the command which is unique for each node, to run the protocol to be

tested and copied it into each node in the network with the same name. Once we run the script via the ClusterSSH, all of the nodes execute their own specific command.

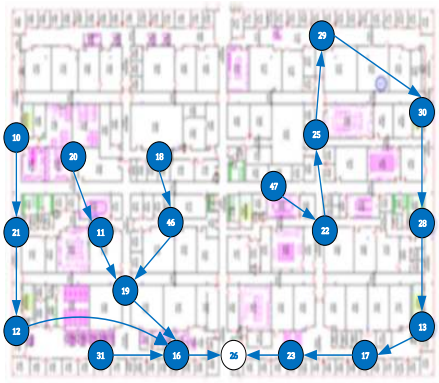
We use the Linux *date* command to synchronize the date and time of all devices in the network for more precise delay measurements.

We created two different spanning trees of the network for the IEEE 802.11s-based mesh network. The first tree was created based on the information of next hop on the path to the gateway in the routing table maintained by HWMP. The second tree was created based on signal strength of links between the neighbor nodes. The more a link gets stronger, the more probable it can be the next hop.

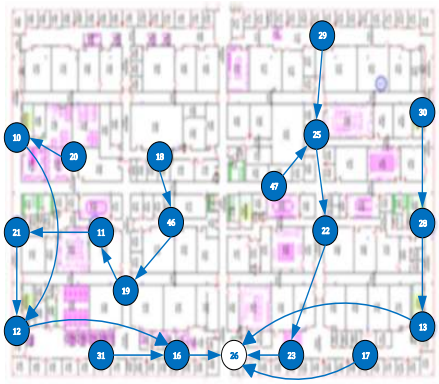
Likewise, signal strength of ZigBee links was used to create a spanning tree for the ZigBee experiments. All of the trees are shown in Fig.s 7.7a, 7.7b and 7.7c, respectively.

The underlying MAC protocol for the IEEE 802.11s is IEEE 802.11g whereas the ZigBee-based testbed is based on IEEE 802.15.4 standard. The Raspi3s are running the Linux 4.9.33 kernel which implements IEEE 802.11-2012 standard [215] while the XBee implements ZigBee standard [191]. To avoid any interference, we ran the WiFi and ZigBee mesh on channel 11 (2.462GHz) and channel 15 (2.425GHz), respectively. The readings from the meters were 16 bits, large enough to hold a real meter reading data.

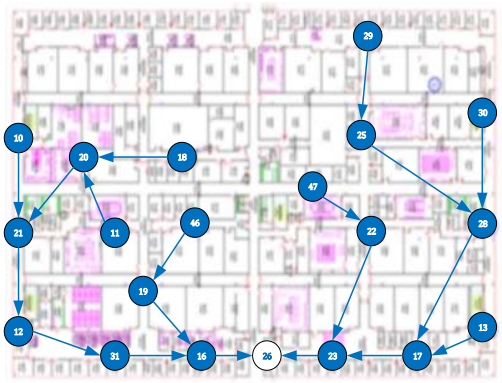
All of required keys, certificates and a CRL are issued and exchanged between the devices before running the protocols. We used OpenSSL v1.1.1 [216] to create the required private keys, certificates and a CRL. The CRL consists of 100 serial numbers other than those of the certificates we created for the experiments. Also, ECDSA was employed to provide authentication since it is an approved signature algorithm by the US NIST [139].



(a) The tree created based on WiFi signal strength.



(b) The tree created based on HWMP routing table.



(c) The tree created based on ZigBee signal strength.

Figure 7.7: The trees used in the experiments.

We implemented all of the protocols in Java, except the FHE-based protocol which is coded in C. The source codes for these protocols will be shared through our AMI testbed's website [217].

We prepared two groups of experiments. In the first group, we set the data collection period to 60 seconds [159] for all protocols except the FHE-based protocol because our preliminary experiments showed that ZigBee needs more than one minute to transmit the encrypted reading efficiently since it allows limited size of data payload due to the constraints of the XBee module. Therefore, we created a second group of experiments in which the data collection period is set to 6 minutes for FHE-based protocol experiments. Since both FHE- and secure MPC-based protocols are privacy-

preserving, we added the secure MPC-based protocol into the second experiment group for performance comparison. We ran the protocols for 30 rounds and presented an average of the test results for statistical significance.

7.3.2 Baselines and Performance Metrics

To assess the performance of privacy-preserving protocols (i.e., MPC- and FHE-based protocols), we have used a number of baselines. The default baseline is the data aggregation on the plaintext readings. In addition, we use 256-bit AES as a second baseline which provides security to a certain extent but not privacy because it requires to decrypt the encrypted readings before the aggregation. In the figures, these approaches are represented with $Plain$, AES, $SMPC$, and FHE, respectively.

The message size of a data packet generated by each method tested in this work is given in Table 7.1. Since the SV scheme encrypts the readings bit-by-bit, the FHE-based protocol transmits them in this format as opposed to the other methods.

Table 7.1: Message overhead for plaintext, 256-bit AES, SV scheme, Secure MPC, and ECDSA signatures.

Message size (in bytes)	
Plaintext	2
256-AES	16
SV scheme (per encrypted bit)	931
Secure MPC	32
ECDSA Signature	132

For performance evaluation, we used the following metrics:

- **Packet Delivery Ratio (PDR):** This is the ratio of the number of data packets received by the gateway to the total number of data packets that are expected to be received by the gateway.

- **Throughput (TP):** This is the average amount of data received by the gateway per second.
- **Completion Time (CT):** This is the elapsed time for gathering all the measurement data from all nodes and aggregating them at the gateway in one round. We measure it at the application layer so that it takes into account the cryptographic operations.

7.3.3 Experiment Results

In this section, we evaluate and discuss the experiment results in terms of packet delivery ratio, throughput and completion time in order to investigate their reliability and efficiency. In the figures, AT, SS and ZB represent the spanning trees created based on HWMP's airtime (AT) metric [218], WiFi signal strength (SS) and ZigBee (ZB) signal strength, respectively. UDP and TCP represent the transport layer protocols UDP and TCP while $ZigBee$ represents ZigBee itself.

Packet Delivery Ratio

EtoE vs HbyH: As shown in Fig. 7.8, all of the protocols except $SMPC$ achieve relatively higher PDR values when they employ $HbyH$ aggregation. This is due to the reduced number of hops to forward a packet to the destination. Also, this decreases the number of packets traveling to the same destination in the network, consequently reduces the packet lost because of the contention during accessing the medium. In contrast, $SMPC$ shows a reverse tendency. We attribute this to the protocol's different way of functioning. Specifically, the participant that computes the secret (the gateway) unicasts a value to each of the other participants in order to reduce the communication overhead due to the share exchanges [17] in traditional secret sharing schemes [149, 219]. This two-way data communication causes extra contention and

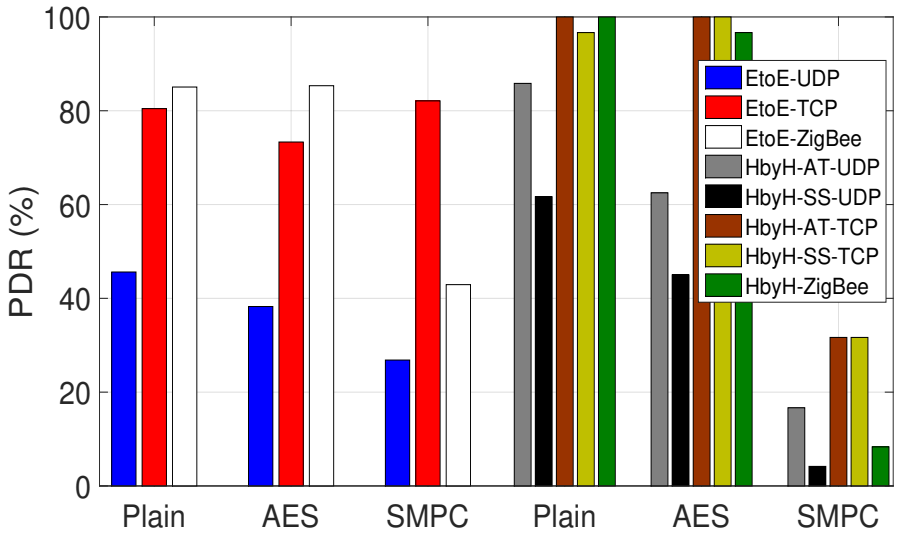


Figure 7.8: The packet delivery ratio for *Plain*, *AES* and $SMPC$ approaches as an indication of the reliability.

results in poor performance. This affects the PDR significantly because a parent meter needs to receive the encrypted readings from all of its child meters in order to aggregate them and report the aggregated value to its parent. If any of the readings cannot be received by the parent, then the aggregation process is stuck at that point and this results in a lower PDR at the gateway. This persists even for the TCP which is a reliable protocol because the number of retransmissions is limited, and the protocol gives up retransmitting the data packet after reaching this limit. Nonetheless, $SMPC$ can achieve more than 80% of PDR with $EtoE - TCP$ aggregation. We attribute this to the direct communication channel between the meters and the gateway. Once the connection is established, data packets can be communicated to the gateway directly. If any of the meters cannot establish the connection, only the reading of this meter cannot be received. However, in $HbyH - TCP$, if any of the parent meters cannot receive the reading from any of its child meters the gateway cannot receive any reading from that branch including this parent meter. Therefore, $SMPC$ would not be a good choice for $HbyH$ aggregation.

AT Tree vs SS Tree: The performance of all approaches shows a similar tendency on different trees. That is, the approaches with $AT-UDP$ always perform better than those with $SS-UDP$. This is due to the fact that the routing table is formed based on the AT metric, while in SS there can be cases where some children access their parents in multiple hops. In general, using TCP makes significantly positive impact on any trees. However, with $SMPC$, the effect of TCP is much significant. Therefore, we can conclude that $SMPC$ should be used with TCP, preferably on AT trees.

802.11s vs ZigBee: As can be seen from the figure, there is a contention between TCP and $ZigBee$. If we exclude $SMPC$ because of its way of functioning, we can say that $ZigBee$ performs at least as good as TCP. This is due to the fact that TCP is a connection-oriented protocol, and that it can neither send nor receive data packets unless a connection is established. Considering that we conducted the experiments in a building with walls and plenty of electrical devices such as workstations, personal computers and the access points for wireless Internet connection, it is inevitable to be exposed to interference and attenuation. $ZigBee$ suffers from interference far less than WiFi does because we operate it on a channel which is not overlapping with the three non-overlapping channels (channels 1, 6 and 11) adopted in the US [220]. However, as we are keen on investigating the performance of $SMPC$, we observed that Zigbee suffers a lot with $SMPC$ as opposed to other approaches. This suggests that IEEE 802.11s should be preferred in $SMPC$ implementations. However, if only security is of concern, Zigbee (with AES) is better.

UDP vs TCP: We have already implicitly discussed the impacts of TCP and UDP. To sum up, UDP achieves the least PDRs in both $EtoE$ and $HbyH$ aggregation since it is inherently connectionless which means that there will be no transport layer retransmissions if any of data packets is lost. TCP outperforms UDP under all

conditions. This can be attributed to the retransmission mechanisms of TCP. If the sender does not receive an ACK within a certain time it retransmits the message to the receiver host.

Throughput (TP)

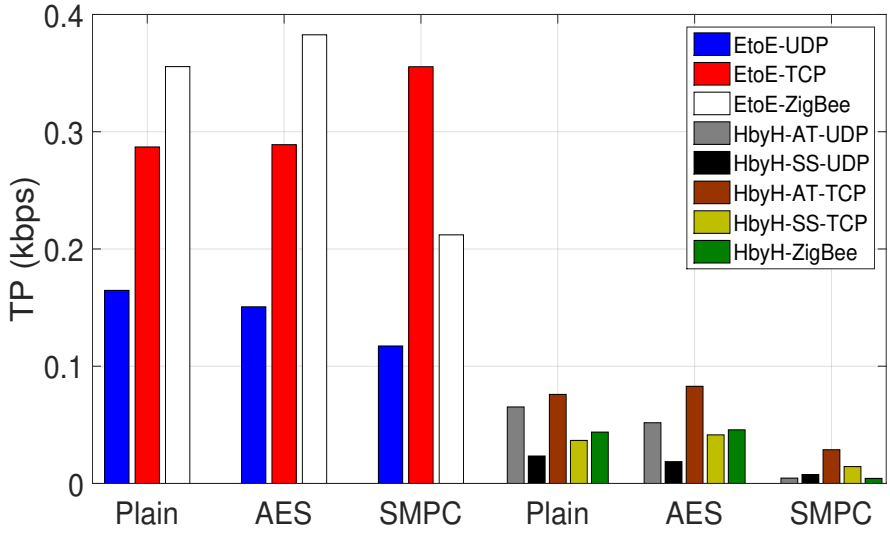


Figure 7.9: The throughput produced at the gateway by $Plain$, AES and $SMPC$ approaches.

EtoE vs HbyH: We investigated the throughput performance to analyze bandwidth usage of the protocols. As can be seen in Fig. 7.9, $EtoE$ aggregation always produces more throughput than $HbyH$ aggregation does. Even those protocols that achieve more than 95% PDR in $HbyH$ aggregation produce two-third less throughput than that was produced in $EtoE$ aggregation. This is due to the fact that the meter readings are aggregated at intermediate meters in $HbyH$ aggregation.

AT Tree vs SS Tree: The approaches produce more TP on AT tree than the TP produced on SS tree. This is due to the PDR achieved by the approaches. As we already discussed the reasons under PDR that also relates to TP, we keep it short in

this subsection. Note that we calculate the TP based on the amount of data received by the gateway during the experiments, and we do not normalize the values.

802.11s vs ZigBee: ZigBee produces more TP where it achieves more PDR than 802.11 approaches, and vice versa. However, the rates are different in TP cases. That is, the gap between Zigbee and 802.11s-TCP is much less now. This can be attributed to the headers added at the Application Support Sublayer (APS) in ZigBee protocol stack [191, 205].

UDP vs TCP: Low TP values of UDP stem from lower PDR. Since TCP provides a reliable communication service it can achieve more message delivery when compared to UDP. The difference between TP values of AES and $SMPC$ approaches is larger than the difference between their PDR values. This can be attributed to the larger size of data packets generated by $SMPC$ approach.

Completion Time (CT)

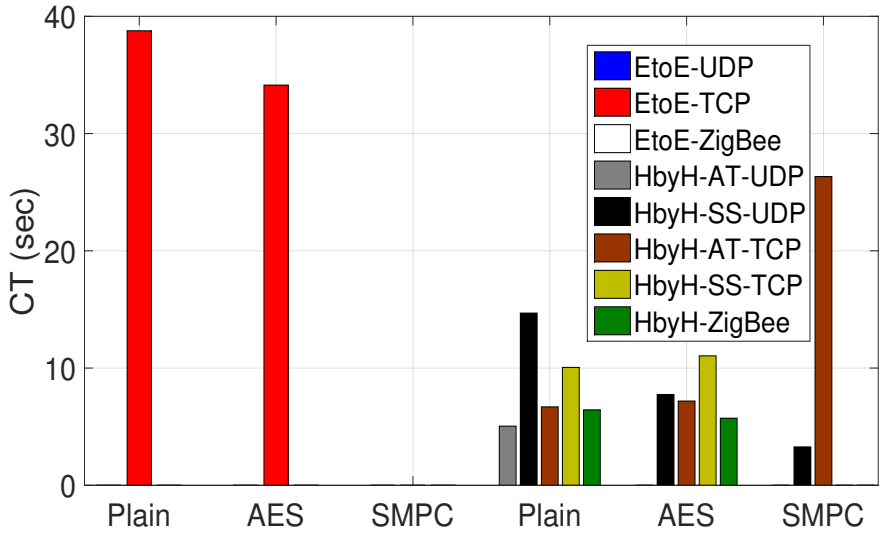


Figure 7.10: The completion time of all meter readings for $Plain$, AES and $SMPC$ approaches.

The last metric we investigated is data collection completion time. Data collection completion includes receiving readings from all of the meters and performing an aggregation operation. Therefore, some of the bars are missing in Fig. 7.10 because the gateway did not receive the readings from all of the meters for even one round.

EtoE vs HbyH: The approaches using $HbyH$ aggregation take less time to complete a data collection round when compared to those using $EtoE$ aggregation. This can be attributed to the lower number of packets traveling throughout the network, thereby reducing the contention during the medium access. This results in shorter backoff times when compared to $EtoE$ aggregation, consequently lower completion time.

AT Tree vs SS Tree: As expected, the approaches need more time to complete a data collection round on SS tree. This is because the data packets sent by some of the child meters need to be forwarded several times before delivering to the parent meter. This is due to the fact that the IEEE 802.11s standard's default routing protocol HWMP which uses airtime metric creates different paths than those we created based on WiFi signal strength.

802.11s vs ZigBee: As can be seen from the figure, it is not possible to compare 802.11s and ZigBee performance in terms of $EtoE$ data aggregation since $SMPC$ was not able to be completed. Thus, we investigated them from the point of $HbyH$ data aggregation. ZigBee requires less time than the others do. We attribute this to the less interference on the channel it operates. However, for $SMPC$ this was not the case. Only 802.11s was able to finish the data collection under $SMPC$. Therefore, while Zigbee is more suitable for Plain data or AES, it is not suitable for $SMPC$.

UDP vs TCP: The protocols running on TCP usually require more time than the others do. This is due to the fact that the TCP needs to establish a connection to the destination before sending data packets. Also, retransmissions add some more extra time. $SMPC$ requires more time when compared to the approaches running on TCP

since the gateway needs to establish separate TCP connections to unicast a value to be used for share computations to all of the meters. Moreover, communication of this value is exposed to a similar contention that occurs in $EtoE$ aggregation.

SMPC vs FHE Experiments

Previous group of experiments have demonstrated the performance of $SMPC$ with respect to other baselines. In this group of experiments, by keeping the same data collection period, we compared the performance of $SMPC$ to FHE-based protocols in terms of the same metrics. Our goal is to observe their pros and cons under different environments.

As mentioned, FHE-based protocol transmits the encrypted meter reading bit-by-bit since even an encrypted bit is very large in size. This results in excessive number of packet transmissions, consequently very high rate in packet losses. While we employed TCP to deal with these packet losses, TCP comes with heavy overhead. Therefore, in addition to TCP, we decided to employ the CoAP for the FHE-based protocol to ensure reliable message delivery with a more lightweight protocol when compared to TCP. We measured and compared its performance with UDP and TCP. CoAP is represented with $CoAP$ in the figures.

We also tested both protocols on the ZigBee tree (ZB) while they are running on the IEEE 802.11s-based mesh network. This is an additional tree (other than AT and SS trees) and we aimed to see if the protocols can perform well on an IEEE 802.11s-based mesh network while the topology is more suitable for ZigBee mesh.

Impact of Data Aggregation Mechanism: The impact of data aggregation mechanism varies based on the metric in question. As can be seen from Fig.s 7.11a and 7.11b, while the FHE-based protocol with $HbyH$ aggregation is comparable to that of $EtoE$ aggregation, $SMPC$-based protocol with $HbyH$ aggregation performs

remarkably worse than that of $EtoE$ aggregation. This is due to the fact that the $SMPC$-based protocol requires the meters to receive a value from the gateway to compute the shares that the other meters would compute for this receiving meter. This makes it more difficult to communicate an aggregated reading to the parent because an intermediate meter cannot aggregate the readings unless it does not receive that value from the gateway even if it already received the readings from all of its child meters.

As shown in Fig.s 7.11c and 7.11d, $HbyH$ data aggregation reduces the bandwidth use regardless of the protocol because when data packets are aggregated the result value does not allocate more resources. This is because the cryptographic operations used in the protocols are based on finite field arithmetic.

Similarly, we can say that $HbyH$ data aggregation on TCP reduces the time required to complete a data collection round from Fig. 7.11e. We attribute this to the extra time required for establishing a connection before sending the data packets. This time increases in $EtoE$ aggregation because all of other meters in the network attempts to establish a connection and causes medium access collisions. This increases backoff waiting times, and consequently the data collection time.

Impact of Tree Topology: As can be seen from the figures in Fig. 7.11, the protocols tend to perform better on WiFi-related trees (AT and SS) when the testbed is built on IEEE 802.11s standard. This can be attributed to the differences between ZigBee and IEEE 802.11s in creating links between neighboring meters. A one-hop neighbor meter in ZigBee mesh can be two or more hops away in IEEE 802.11s mesh. This increases the probability of packet losses when ZB tree is used for data aggregation in a IEEE 802.11s mesh.

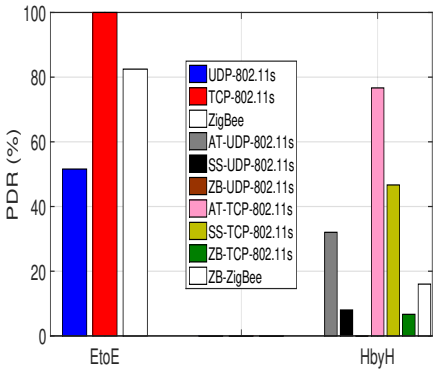
When we compare the performance of the protocols on AT and SS trees, it can be seen that the protocols perform better on AT tree. We attribute this to fact that

AT is formed based on the underlying routing protocol. The next hops in AT tree is the same with the next hops in the HWMP's routing table. When SS tree is used, some of the parent meters are multiple hops far from the sender meter according to the HWMP's routing table.

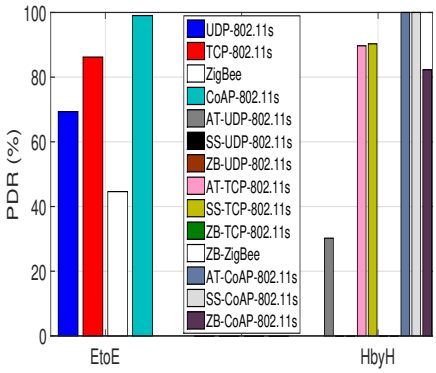
Impact of Transport/Application Protocol: PDR and CT performance should be investigated together in order to evaluate the impact of the transport/application protocols used. TCP and $CoAP$ protocols which both provide reliability for message delivery achieves the most packet delivery ratio, and more importantly they manage to complete at least one data collection round in $EtoE$ data aggregation. In addition, as can be seen in Fig. 7.11f, $CoAP$ can complete at least one data collection round on ZB tree whereas ZigBee cannot complete even once. This is due to the fact that $CoAP$ maintains the state of each transmitted message without establishing a connection with the receiver. Thus, if any of the messages is not delivered this can be detected, and the message is retransmitted by the sender. Moreover, $CoAP$'s congestion control reduces packet drops due to network congestion.

Impact of Mesh Technology: The experiment results indicate that the performance of $ZigBee$ varies based on the size of the concealed meter reading. Hence, The performance of $SMPC$-based protocol on $ZigBee$ is better than that of UDP, but worse than that of TCP while the performance of the FHE-based protocol is the worst when $ZigBee$ mesh is used. This is because in FHE-based protocol, $ZigBee$ needs to fragment an encrypted bit into 5 fragments since it allows to send at most 256-byte payload in a data packet. Moreover, those packets are subject to further fragmentation based on the available buffer [191]. Due to the increased number of fragments to be handled, medium access contention increases dramatically. This results in data loss, consequently lower PDR and poor performance. In general, $Zigbee$

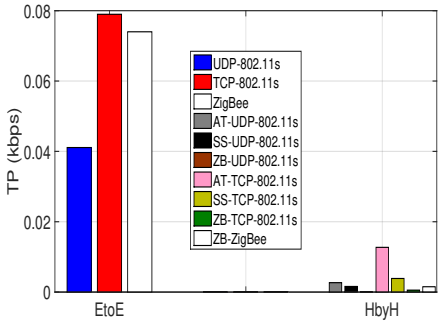
cannot complete any rounds in $SMPC$ and FHE. For 80211.s, the rounds can be completed when using TCP for $SMPC$ and they are much quicker than FHE.



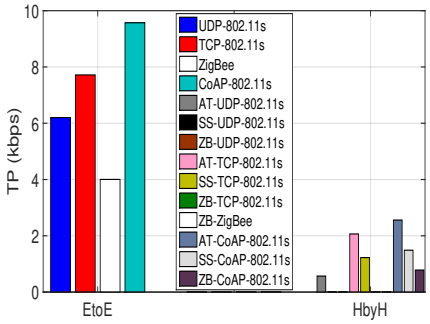
(a) The packet delivery ratio for $SMPC$-based protocol.



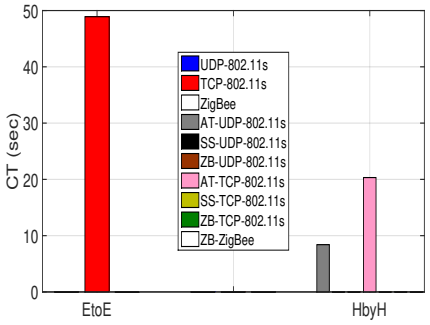
(b) The packet delivery ratio for FHE-based protocol.



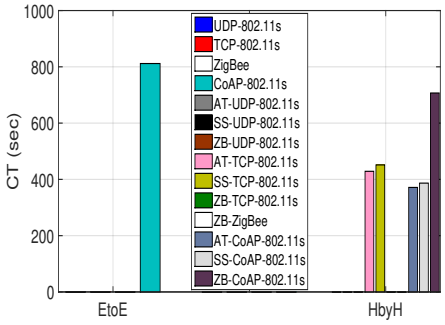
(c) The throughput produced at the gateway by $SMPC$-based protocol.



(d) The throughput produced at the gateway by FHE-based protocol.



(e) The completion time of a data collection round for $SMPC$-based protocol.



(f) The completion time of a data collection round for FHE-based protocol.

Figure 7.11: $SMPC$ vs FHE experiment results.

7.4 Conclusion

In this chapter, we have evaluated the performance of two open-source wireless mesh networking standards, i.e., IEEE 802.11s and ZigBee, under security and privacy requirements of Smart Grid (SG) Advanced Metering Infrastructure (AMI) networks. We built a testbed out of *Raspberry Pi* 3s that implement IEEE 802.11s and ZigBee wireless mesh networking standards in order to mimic the AMI network. We presented a comprehensive explanation about how we built the testbed so that the readers can easily reproduce the same testbed environment.

We employed fully homomorphic encryption (FHE) and secure multiparty computation ($SMPC$) based protocols which are both secure and privacy-preserving along with SSL certificates and Elliptic Curve Digital Signature Algorithm (ECDSA) signatures for authentication. Moreover, we used End-to-End ($EtoE$) and Hop-by-Hop ($HbyH$) data aggregation mechanisms. Furthermore, we integrated the Constrained Application Protocol ($CoAP$) into FHE-based protocol to provide both reliable and lightweight communication.

To begin with, the results for the first group of experiments have demonstrated that $HbyH$ data aggregation mechanism is much suited to be used for the AMI network since it reduces throughput and completion time significantly although it is susceptible to packet losses due to the dependency between tree levels. The protocols running on 802.11s mesh technology is more robust than those using ZigBee although those using ZigBee performs almost equally or outperform those using 802.11s mesh for cases where privacy is not of concern (plain data or AES encryption). Specifically, ZigBee can be a good alternative to 802.11s mesh for the cases where a whole concealed reading can fit into a single ZigBee data packet and $HbyH$ aggregation is employed.

Secondly, we compared the performance of the two privacy-preserving protocols, $SMPC$- and FHE-based protocols. Experiment results have shown that the $SMPC$-

based protocol can be an alternative to FHE-based protocol when the AMI network is built on IEEE 802.11s mesh, and $EtoE$ data aggregation mechanism is employed. On the other hand, the FHE-based protocol with $CoAP$ has shown a remarkable performance with both $EtoE$ and $HbyH$ data aggregation. It can be a viable option if data collection is performed in more than every 6 minutes. We infer that $CoAP$ is the most suitable choice in terms of message delivery and throughput for data-intensive protocols such as the FHE-based protocol at the expense of long data collection completion times.

CHAPTER 8

SECURE AND RELIABLE FIRMWARE UPDATE PROTOCOLS FOR THE AMI NETWORK

One of the major difficulties we had in our testbed work in the previous chapter was in distributing source code of the protocols as we updated them since we were revealing some bugs during the preliminary tests. However, it was not efficient to collect all devices, loading new code and placing them back each time we needed to update the protocols. Therefore, we used *scp* (*secure copy*) which is a file transfer protocol runs over *ssh* (*secure shell*) to distribute the new code to the devices over the wireless mesh network. It was not efficient to unicast the new code either because we needed to change several parameters in the *scp* command each time we needed to update them. Therefore, we wanted to develop a more efficient method for distribution of new code. In this chapter, we investigate this problem in the firmware update concept since the same problem is likely to occur in this concept in the Advanced Metering Infrastructure (AMI) network.

Smart meters (SMs) are comprised of some smaller electronic parts such as NIC (Network Interface Card) [221] and processor [222] which execute some programs called firmware that control, monitor and manipulate the data in the device. The SM firmware is developed by SM vendors and updated in order to fix the bugs detected, improve its functionality and add new functionalities to the device. In addition to SM vendors, utility companies (UCs) may also need to update their service software due to some regulations on the related law as well as bugs and functionality improvement/enhancement. For example, a legislation in Florida state can obligate Duke Energy to update the firmware/software running on the SMs they installed in this state. Also, a firmware update can target legacy SMs of specific brand(s) or model(s). That is, an update can target a subset of the SMs as well as all the SMs in an AMI

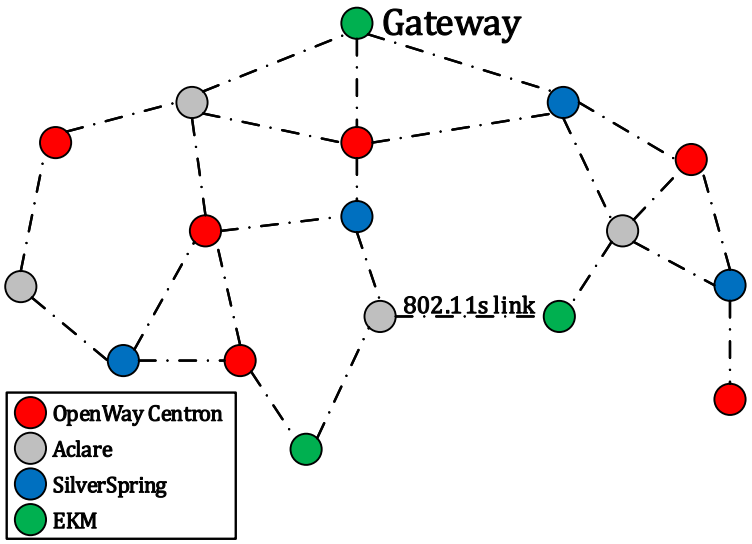


Figure 8.1: An AMI network with different brand SMs

network. For example, the AMI network given in Fig. 8.1 consists of different brand SMs. A firmware update released by SilverSpring targets only blue nodes in the figure. The update should be downloaded and installed by these SMs only.

The firmware update process is initiated by the party that releases the update file (e.g., SM vendor, UC). The party notifies the gateway of the network that there is an update for the SMs. The gateway sends a firmware update request to the SMs that the update targets. Then, the targeted SMs download and install the update [123]. Rather than unicasting the request, it should be multicast to the targeted SMs because unicasting wastes both network bandwidth and processing resources of the device. Also, this needs to be done manually which is cumbersome when there is large number of SMs. However, multicasting requires the SMs to know that there is an update for them beforehand so that they can join a multicast group. If the SMs knew that there is a firmware update for them, then they would download and install it without waiting a request. In addition, multicasting may not be available in wireless environments. Even if it is available, the reliability and security is a big concern. Reliability is due to use of UDP in such environments for reduced overhead

while security refers to confidentiality and authentication. Another alternative is to broadcast the request which causes not only the targeted SMs but also those that are not targeted to download and install the update, which wastes more bandwidth and processing resources.

To overcome this problem, we propose employing ciphertext-policy attribute-based signcryption (CP-ABSC) [34]. CP-ABSC provides not only confidentiality and access control but also data integrity and message authentication. It is based on satisfying an access tree with a set of attributes assigned by attribute authorities [223]. The access tree defines who can grant access to the encrypted data while the attributes are used to decrypt the encrypted data by satisfying the access tree. CP-ABSC signcrypts the data based on the access tree, and the ciphertext can be designcrypted if the set of attributes with which the private key was created satisfies the access tree.

In this work, we propose secure and reliable application layer multicasting (multicast-over-broadcast) protocols based on CP-ABSC for firmware update in the IEEE 802.11s-based AMI network. In general, in these protocols, the firmware provider signcrypts the firmware update file and a firmware update request with an access tree that defines the targeted SMs, and sends them to the gateway of the network. The gateway broadcasts the firmware update request. The SMs that possess the set of attributes that can satisfy the access tree can designcrypt the request and initiate the FTP (File Transfer Protocol) to download the update file.

CP-ABSC fulfills the needs of data integrity and sender authentication. However, the aforementioned protocol cannot guarantee that all SMs in the network receive the firmware update request at least once since broadcasting runs on top of UDP which is an unreliable protocol. Therefore, we introduced an ACK (acknowledgment) mechanism into the protocol. Once the gateway broadcasts the request, it waits an ACK from each of the SMs for a certain amount of time. If it does not, then it

broadcasts a much smaller packet called alarm message that triggers the SMs that have received the request to broadcast a copy of the request. This repeats until the gateway receives at least one ACK from each SM in the network. We call this Broadcast-Alarm (Bcast-Alarm) protocol.

Despite the fact that Bcast-Alarm protocol is both secure and reliable, it generates too many duplicate packets traveling throughout the network. This wastes the network bandwidth and processing resources of the SMs. In addition, signcryption operation increases the size of the update request. It is very likely to increase in the number of packets lost because when the size of a datagram exceeds the maximum transmission unit size it is split into smaller packets. If any of these packets is lost, then the whole datagram is lost. Therefore, in this work, we introduce network coding into Bcast-Alarm protocol and remove the alarm messages in order to reduce the size of application layer packet transmitted at a time. In the new protocol which is called Broadcast-Network Coding (Bcast-NC), the gateway (encoder) divides the request into a certain number of same size data packets, encodes and broadcasts them. The SMs (decoders) buffer the coded packets and decode the update request when they have sufficient amount of innovative coded data. The SMs that have decoded the request send an ACK to the gateway. The gateway keeps broadcasting coded packets until it receives at least one ACK from each of the SMs.

We implemented the proposed protocols under ns-3 [132] and used Kodo [224] for network coding, which is an open source network coding library. We compared their performance with that of unicasting (Ucast) in terms of communication delay, completion time of firmware update process and throughput. The simulation results indicate that Bcast-NC protocol is comparable to Ucast in terms of throughput, and that it utilizes the network bandwidth much more efficiently than Bcast-Alarm

protocol does. However, Bcast-Alarm protocol outperforms Bcast-NC protocol with regards to the completion time.

Our main contributions in this chapter can be summarized as follows.

1. We propose secure and reliable multicasting protocols based on CP-ABSC for firmware update in the IEEE 802.11s-based AMI network.
2. The proposed protocols are implemented under ns-3, which is a very commonly used network simulation tool, and their performance is assessed and compared with that of unicasting.

This chapter is organized as follows. In the next section, we give a background information about CP-ABSC and network coding in Section 8.1. In the following section, we introduce our secure and reliable firmware update protocols. In Section 8.3, the simulation results are presented and discussed. Finally, we summarize our work and conclude the chapter in Section 8.4.

8.1 Preliminaries

8.1.1 Ciphertext-Policy Attribute-Based Signcryption

One of the most known access control methods is to use attribute-based encryption (ABE) methods. The two commonly used ABE methods are key-policy ABE (KP-ABE) [225] and ciphertext-policy ABE (CP-ABE) [226]. KP-ABE differs from CP-ABE in the way of creating private keys and performing encryption/decryption operations. In KP-ABE, private keys are created based on an access tree, and data is encrypted with a set of attributes. A ciphertext which is created with a set of attributes can be decrypted if the set of attributes satisfies the access tree with which the private key was created. In contrast, CP-ABE encrypts the data based on the

access tree, and the ciphertext can be decrypted if the set of attributes with which the private key was created satisfies the access tree.

KP-ABE is not convenient in our case because we define the SMs with the attributes assigned by the attribute authorities while encrypting the data based on the access tree. On the other hand, if KP-ABE is employed it will require to issue a new private key each time the firmware update targets a different group of SMs, i.e., the access tree changes, which introduces a communication overhead. Therefore, CP-ABE fits in firmware update context better than KP-ABE. However, CP-ABE provides confidentially and access control but no data integrity and message authentication. That is, an encrypted firmware update request cannot be checked if it is altered on transit or if it is sent by an unauthenticated party. If the request is processed without an integrity check a piece of malicious code injected during the transmission can damage the device and even brick it. Similarly, a DoS (Denial of Service) attack can be performed against the update provider by sending a plethora of download requests if an authentication mechanism is not employed [227]. To that end, we propose using ciphertext-policy attribute-based signcryption (CP-ABSC) [34] in this chapter since it integrates digital signatures with CP-ABE in order to provide not only confidentiality and access control, but also data integrity and message authentication with an insignificant increase at computational cost, which are required in AMI network communications.

We assumed that each SM has a private key created with 10 attributes and that the update requests are signcrypted with a simple access tree of three attributes as given in Fig. 8.2. For example, a firmware update request signcrypted based on this access tree targets the OpenWay Centron meters running Linux 4.4.39 or 4.1.36 kernel. The structure of the access tree affects the size of the signcrypted data and the computational delay of signcryption/designcryption operations.

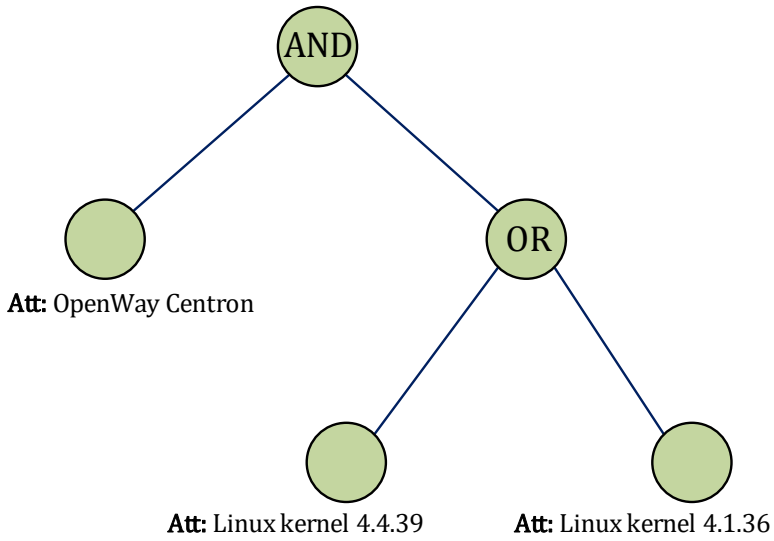


Figure 8.2: An example access control tree

In CP-ABSC, a multicast group is defined based on the access rights of the data of interest. The access rights are specified by the data source and represented with an access tree. The access tree is an access structure containing a set of attributes along with AND and OR relations. Instead of revealing the identity of the destinations, it defines who can access the actual data. The data is signcrypted based on this access tree. The corresponding ciphertext can be designcrypted by any user who has a secret key computed with a set of attributes that can satisfy the access tree.

CP-ABSC consists of the following four primary algorithms [34]:

- ***System Initialization*** This algorithm is executed by a certificate authority (CA). After creating a bilinear mapping $e : G_1 \times G_1 \rightarrow G_2$ where G_1 and G_2 are two cyclic groups of prime order p, it computes the system public parameters denoted by PK and the master key MSK. PK is shared with the users while MSK is kept secret to create a secret key for each user in the system.
- ***Key Generation*** This algorithm is also executed by the CA. It takes MSK and a set of attributes S as input and computes a secret key SK, a signing key K_{sign} and a verification key K_{ver}. SK and K_{sign} are sent to the owner of S, and

K_{ver} is published to the other users to verify any message sent by the owner of S.

- ***SignCryption*** This algorithm is executed by a user who requires any other user who wants to access his data to possess some attributes that can satisfy the access tree specified by himself. It takes a plaintext message M, an access tree T that specifies the access policy of the message M, the public key PK and the signing key K_{sign} as input. Then, it computes and returns a ciphertext along with a relevant signature CT_{sign}.
- ***DeSignCryption*** This algorithm is executed by a user who wants to access the actual message concealed in a signcrypted message CT_{sign}. It takes the signcrypted message CT_{sign}, the private key SK and the verification key K_{ver}. It returns the message M if the verification process is successfully completed.

8.1.2 Network Coding

Network coding (NC) is a technique that increases the network bandwidth efficiency in both wired and wireless networks by eliminating packet header overhead. It can be utilized from the physical to the application layer since it provides error correction as well as encoding/decoding. There are two types of NC: inter-flow NC and intra-flow NC [35]. In this work, we use intra-flow NC since the data packets traveling throughout the network are from the same data flow.

In broadcast-based communication systems, random linear NC (RLNC) is utilized because it does not need a centralized control over the encoding/decoding operations and produces close-to-optimal throughput [228]. In RLNC, the data source divides the data to be broadcast into *generation*s as illustrated in Fig. 8.3. Each generation contains same number of k packets denoted p_i, $i = 1, 2, \cdots, k$, which are d bytes each. A random linear combination of all the packets in a generation is computed

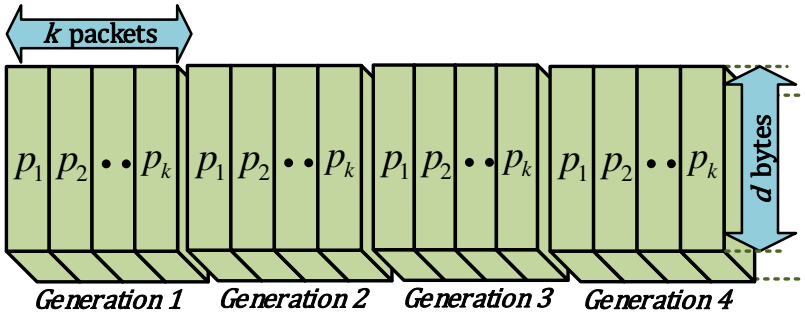


Figure 8.3: Data partitioning before encoding a generation

separately to obtain an encoded packet $x = \sum_{m=1}^{k} \alpha_m p_m$ for each generation, where $\vec{\alpha}$ is the encoding vector. The encoding vector is included in every encoded packet for future decoding purposes. When an encoded packet x is received, the receiver checks if the encoding vector of this packet is linearly dependent with that of all other encoded packets received thus far. If so, this packet is discarded. Otherwise, the packet which is called an innovative packet is stored in the buffer, and the receiver tries to decode all the packets stored in the buffer by performing Gaussian elimination whose computational complexity is $O(k^3)$.

Each α_m in the encoding vector $\vec{\alpha}$ is randomly selected from a Galois field $GF(2^q)$ where q is a positive integer. A Galois field (finite field) consists of finite number of elements and all operations defined in this field are closed, which means that every operation performed on two elements from this field results in an element in the same field. It is critical to choose a finite field for RLNC applications because there is a trade-off between computational cost and efficiency of the coding. That is, the chosen field determines the time required for generating encoded packets and the possibility of generating linearly dependent and consequently useless packets.

8.1.3 Problem Definition

Assuming that the firmware update publishers can determine the SMs in an AMI network, whose firmware need to be updated, they can be notified through the gateway of the network that there is an update. The gateway can unicast a firmware update request to the targeted SMs, which is a costly process in terms of the required bandwidth and computational power. Please note that the gateway gets busy with sending literally the same data several times instead of dealing with waiting jobs in the job queue. Instead, the requests can be multicasted to the SMs. However, wireless mesh standards (e.g. IEEE 802.11s and 802.15.4) do not support IP multicast because they implement routing on MAC addresses instead of IP addresses. Even if they supported IP multicast, this would not be feasible because it requires the targeted SMs to know that there is an update for them beforehand and to send a multicast group join message to the gateway. If the targeted SMs knew that there is an update for them, they would download the update file, so would not need to be notified with an update request.

The firmware update process should be reliable because if a consumer cannot download and install the firmware update that fixes a critical bug, the consumer may suffer from a security vulnerability. If the bug is related to the communication protocols, then it can affect the communication in the whole network since some nodes act as intermediary to relay the data of multiple-hop away nodes in WMNs.

Referring the given information, the problem we address in this chapter can be defined as follows: *Given an IEEE 802.11s-based AMI network with n SMs, update the firmware of a group of SMs without revealing their identities in a secure and reliable way.*

8.1.4 Threat Model and Security Goals

The firmware update requests and the firmware update itself need to be communicated in a secure way. Also, the firmware update file has to be kept private between the publisher (the meter vendor or the UC) and the SMs to be updated because the firmware might be proprietary [229]. We identified the following attacks to the privacy and security of the firmware update process in the AMI and established the associated security goals.

- **Attack 1:** An eavesdropper monitors the communication channel in order to obtain the firmware update file. The eavesdropper can detect some bugs and use them for further attacks; infamize the publisher company or improve/enhance their own firmware if s/he works for a competing company.
- **Security Goal 1:** Conceal the firmware update such that only the SMs that the publisher targets can reveal it.
- **Attack 2:** An attacker in the middle can capture and alter the firmware update file so as to damage the SMs.
- **Security Goal 2:** Provide data integrity to verify the content of the update file.
- **Attack 3:** An attacker impersonates the gateway and sends fabricated firmware update requests to the SMs to initiate a DoS attack against the gateway.
- **Security Goal 3:** Provide sender authentication to verify the sender and contents of the requests.
- **Attack 4:** An eavesdropper captures and replays a legitimate firmware update request to initiate a false download.
- **Security Goal 4:** Identify and discard replayed requests.

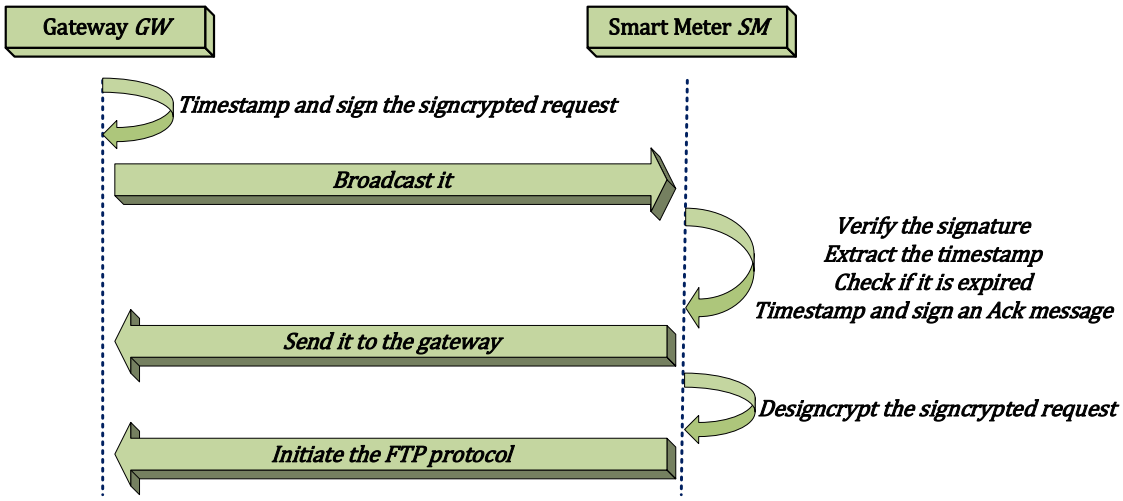


Figure 8.4: Overview of the secure multicasting protocol we proposed in this work

8.2 The Proposed Firmware Update Protocols

In this section, we introduce the two firmware update protocols. We present a general overview of the successful use case in Fig. 8.4 and explain the details of the protocols in the next section. We assume that the certificate authority issued all required keys to the SMs in the network before the protocols are run.

The publisher of the firmware update signcrypts the firmware update file $FWUF$ ($CT_{sign}(FWUF)$) and a firmware update request $FWUReq$ ($CT_{sign}(FWUReq)$) and transmits them to the gateway of the network over the Internet. The publisher signcrypts both data because the firmware is assumed to be proprietary and it should not be revealed even to the gateway if the gateway is not one of the targeted SMs.

While transferring the $CT_{sign}(FWUF)$ a file transfer protocol such as FTP (File Transfer Protocol) should be used since its size is likely much larger than that of an ordinary communication message. Our protocols tackle the part after the communciation of the $CT_{sign}(FWUF)$ and $CT_{sign}(FWUReq)$.

In our protocols, we utilize ECDSA (Elliptic Curve Digital Signature Algorithm) to sign every timestamped packet although CP-ABSC provides data integrity and

message authentication because the integrity and the authenticity of the timestamp TS should be ensured since it can be fabricated or altered in transit.

Algorithm 2 $FWU\_Notify(CT_{sign}(FWUReq))$

1: $FWUREQ \leftarrow Concat(CT_{sign}(FWUReq), TS)$
2: $< FWUREQ, \sigma > \leftarrow Sign(FWUREQ, SK^{GW}_{ECDSA})$
3: $SendTo(< FWUREQ, \sigma >, IPv4(255.255.255.255))$
4: $Schedule(Secs(t), Broadcast\_Alarm)$
5: $done \leftarrow TRUE$
6: **repeat**
7: $< ACK, \sigma', from > \leftarrow Recv(socket)$
8: $Success/Fail \leftarrow Verify(< ACK, \sigma' >, PK^{SM}_{ECDSA})$
9: **if** $Success$ **then**
10: $< ACK, TS' > \leftarrow Split(ACK)$
11: $Yes/No \leftarrow IsExpired(TS')$
12: **if** No **then**
13: $AckMap.Add(from, TS')$
14: **if** $AckMap.Size() == \#SMs$ **then**
15: $Cancel\_Schedule(Broadcast\_Alarm)$
16: $done \leftarrow FALSE$
17: **end if**
18: **end if**
19: **end if**
20: **until** $done$

8.2.1 Broadcast-Alarm Protocol

We assume that the gateway has successfully received the $CT_{sign}(FWUF)$ and $CT_{sign}(FWUReq)$ before running this protocol. A pseudocode for the operations performed by the gateway is given in Alg. 2. The gateway timestamps the $CT_{sign}(FWUReq)$ $(FWUREQ)$, signs it with its secret key SK^{GW}_{ECDSA} generated for ECDSA operations and broadcasts it $(< FWUREQ, \sigma >)$. Then, it begins to wait for a certain amount of time, e.g., t secs to receive at least one ACK from each SM in the network. If it does not, $Broadcast\_Alarm$ function is called. In this function, an Alarm message which is much smaller than the $< FWUREQ, \sigma >$ is created, timestamped, signed

and broadcast. In the end of the function, it is re-scheduled in a way to be called after t secs. This repeats until the gateway receives at least one ACK from each SM in the network. If an SM receives an Alarm message it broadcasts the $< FWUREQ, \sigma >$ if it already received it. Otherwise, it does nothing.

Recv function is non-blocking in the real implementation, but we assume it is blocking in the pseudocode in order to give the idea behind the protocols. In fact, it is called by the lower layer protocol (UDP in our work) when any data packet is received by the socket that is being listened. When an ACK is received its signature σ' is verified. If it succeeds its timestamp TS' is checked if it is expired. If not, it is added to a hash map. If the number of entries in the hash map is equal to the number of SMs in the network the scheduled *Broadcast_Alarm* is canceled.

Algorithm 3 $FWUF\_Download(< FWUREQ, \sigma >)$

1: $Success/Fail \leftarrow Verify(< FWUREQ, \sigma >, PK^{GW}_{ECDSA})$
2: **if** $Success$ **then**
3: $< CT_{sign}(FWUReq), TS > \leftarrow Split(FWUREQ)$
4: $Yes/No \leftarrow IsExpired(TS)$
5: **if** No **then**
6: $ACK \leftarrow Concat(Ack, TS')$
7: $< ACK, \sigma' > \leftarrow Sign(ACK, SK^{SM}_{ECDSA})$
8: $SendTo(< ACK, \sigma' >, IP_{GW})$
9: $FWUReq/Fail \leftarrow DeSignCrypt(CT_{sign}(FWUReq), SK^{SM}_{CP-ABSC}, K^{Publisher}_{ver})$
10: **if** $Verification\ Succeeds$ **then**
11: $Initiate\_FTP()$
12: $Download\_FWUF()$
13: **end if**
14: **end if**
15: **end if**

When an SM receives the $< FWUREQ, \sigma >$ it runs the $FWUF\_Download$ function whose pseudocode is given in Alg. 3. First of all, the signature is verified with the public key of the gateway PK^{GW}_{ECDSA} generated for ECDSA operations. If it suc-

ceeds $FWUREQ$ is split into $CT_{sign}(FWUReq)$ and TS. The TS is checked if the $CT_{sign}(FWUReq)$ was already received before. Otherwise, the SMs may face replay attacks. If it is a new firmware update request, then the SM timestamps an *Ack* message (ACK), signs it with its secret key SK^{SM}_{ECDSA} generated for ECDSA operations and sends it ($< ACK, Signature' >$) to the gateway. Finally, the SM tries to designcrypt the $CT_{sign}(FWUReq)$ with its secret key $SK^{SM}_{CP-ABSC}$ generated for CP-ABSC operations and the verification key of the firmware update publisher $K^{Publisher}_{ver}$. If the process is successfully completed, then the SM initiates the FTP protocol and downloads the $CT_{sign}(FWUF)$.

The download process requires to run another protocol since an update file can be 400KB-2MB in size [121]. The FTP (File Transfer Protocol) is one of the most commonly used protocols to transfer files between two hosts in a network. In our work, we use the FTP protocol to download the firmware update from the gateway.

8.2.2 Broadcast-Network Coding Protocol

Broadcast-Network Coding (Bcast-NC) protocol differs from Bcast-Alarm protocol mostly in the way of broadcasting the signcrypted request and the lack of Alarm messages. In this protocol, the gateway keeps broadcasting encoded packets of the request which are much smaller than the request itself at every t seconds until it receives at least one ACK from each SM in the network.

When an SM receives an encoded packet it checks whether the packet is innovative or not. If not, the packet is discarded. Otherwise, the SM stores the packet in a buffer and performs Gaussian elimination on all the packets stored in the buffer to check if sufficient amount of data is gathered to recover the whole request. If the SM can successfully recover the whole request it follows the steps given in Alg. 3.

8.3 Performance Evaluation

In this section, we analyze the proposed protocols from the security point of view and measure their performance based on some metrics defined in Section 8.3.2.

8.3.1 Security Analysis

We evaluate the proposed protocols based on the security goals listed in Section 8.1.4.

- **Security Goal 1:** The firmware update publisher signcrypts the update file with an access tree that can be satisfied by the attributes of the targeted SMs before sending it to the gateway. This prevents not only the non-targeted SMs but also the eavesdropper from obtaining the actual update file.
- **Security Goal 2:** Since the firmware update file is also signcrypted, during the designcryption operation, the SMs can detect if it is altered.
- **Security Goal 3:** The gateway signs all timestamped messages before transmission. Hence, a fabricated request will be discarded because it will fail in the verification step at the SM.
- **Security Goal 4:** Since all messages are timestamped, the timestamp can be checked to identify a legitimate but replayed request message.

8.3.2 Baselines and Performance Metrics

We compared the performance of the proposed protocols with that of a simple unicast-based protocol (Ucast). In Ucast protocol, the gateway is assumed to know which SMs are targeted. Thus, it unicasts a timestamped and signed firmware update request ($< UcastReq||TS, \sigma >$) to each targeted SM one-by-one and waits an ACK from each of them for a certain amount of time t. If it does not receive an ACK from any one of

the SMs within t sec, then it re-unicasts the request to that SM. This repeats until the gateway receives an ACK from each of the targeted SMs. Since the targeted SMs are known by the gateway, the update request is not encrypted but signed with ECDSA for an authenticated communication.

When an SM receives the request it verifies its signature and checks the timestamp to avoid a replay attack. If these processes do not fail, then the SM establishes a TCP connection to the gateway in order to initiate the FTP protocol to download the firmware update file.

We used the following metrics to measure their performance.

- **Request Delivery Ratio:** This is the ratio of the number of SMs that received at least one firmware update request to the number of SMs that are supposed to receive a firmware update request depending upon the protocol considered.
- **Communication Delay:** This is the average time elapsed between sending and receiving a data packet, which includes transmission and propagation delays.
- **Completion Time:** This is another time-based metric which measures the average time elapsed between the first request message sent by the gateway and the end of the FTP protocol to download the FWUF. It includes the computational delays due to the cryptographic and network coding operations.
- **Communication Throughput:** This is the average amount of communication data (FWUF data excluded) received by each SM per second. We excluded FWUF data in order to observe messaging overhead introduced by each protocol.
- **Total Throughput:** This is the average amount of data (FWUF data included) received by each SM per second.

8.3.3 Experimental Setup

We implemented the baseline and the proposed protocols under the most commonly used network simulator ns-3 [132], which has an implementation of IEEE 802.11s standard. The underlying MAC protocol is IEEE 802.11g.

The experiments were conducted on randomly created topologies containing 36, 49, 64, 81 and 100 nodes (SMs) in an area of 1200mx1200m. This area mimics a realistic neighborhood area network with one gateway which is used to communicate with the UC/SM vendor. The communication range of each SM is set to 100m [101]. The simulations were run with 30 different topologies for 500s and we reported the average of the results from these topologies. At each topology, 10 distinct nodes were randomly chosen as targeted SMs.

All protocols including Ucast were run on top of UDP for the sake of a fair comparison despite Ucast could have been run on top of TCP since it does not use broadcasting.

The values of t parameter were optimized via a set of preliminary simulations. As a result of these simulations, we chose t to be 5 sec and 2 sec for any size of network in Bcast-Alarm (and also Ucast) and Bcast-NC protocols, respectively.

As aforementioned in the previous sections, we used ECDSA to sign all timestamped messages. ECDSA is a signature algorithm approved by the US government [139]. We used the ASN.1 secp128r1 standard curve with SHA1 hashing algorithm, having a key length of 256 bits. ECDSA was also used to sign/verify ACK and Alarm messages since they also do not need to be encrypted but authenticated. We used the ECDSA implementation in crypto++ library [230].

We used a network coding library called Kodo [224] to encode/decode the update requests. We chose k and q to be 12 and 8 [231], respectively.

The update file to be downloaded was assumed to be 2MB [121]. Size of all data/packets and computational delay of the major operations considered in the simulations are given in Tables 8.1 and 8.2, respectively. The operations were run on a Raspberry Pi 3 Model B [232] in order to measure the computational delays.

Table 8.1: Size of data/packets

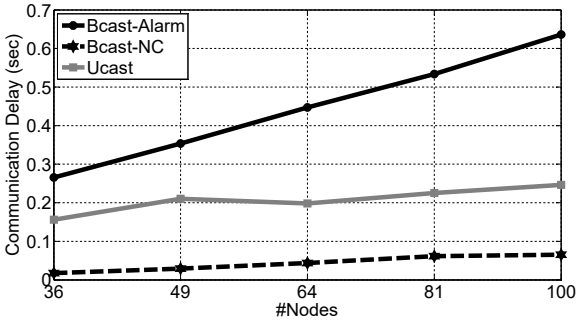
Packet/Data	Size (bytes)
$\boldsymbol{FWUF}$	2097152
$\boldsymbol{CT_{sign}(FWUF)}$	2098692
$\boldsymbol{FWUReq}$	4
$\boldsymbol{CT_{sign}(FWUReq)}$	1540
$\boldsymbol{UcastReq}$	4
$\boldsymbol{Alarm}$	4
$\boldsymbol{Ack}$	4
$\boldsymbol{TS}$	8
$\boldsymbol{Signature}$	32

Table 8.2: Computational delay of the major operations

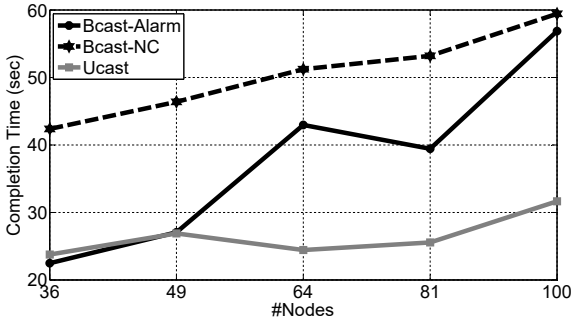
Operation	Delay (ms)
$\boldsymbol{Sign(Ack/Alarm/UcastReq||TS)}$	0.4148
$\boldsymbol{Verify(Signature_{Ack/Alarm/UcastReq||TS})}$	0.8221
$\boldsymbol{Sign(CT_{sign}(FWUReq)||TS)}$	0.4362
$\boldsymbol{Verify(Signature_{CT_{sign}(FWUReq)||TS})}$	0.83
$\boldsymbol{DeSignCrypt(CT_{sign}(FWUReq))}$	23.1566
$\boldsymbol{Download(CT_{sign}(FWUF))}$	14954545

8.3.4 Simulation Results

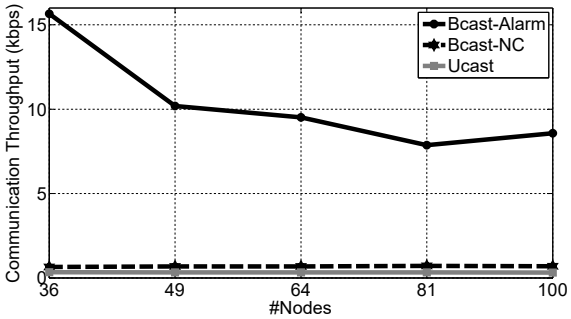
In this section, we compare the performance of the proposed protocols with that of Ucast protocol based on the aforementioned metrics.



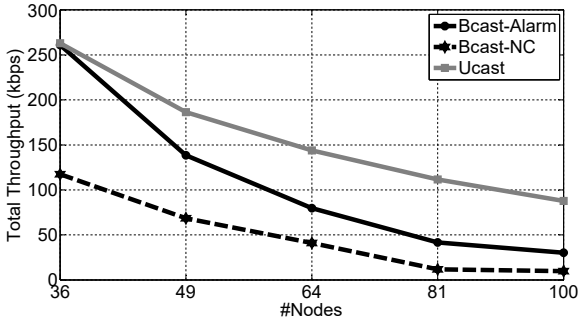
(a) Communication Delay.



(b) Completion Time.



(c) Communication Throughput.



(d) Total Throughput.

Figure 8.5: Broadcast-Alarm vs. Broadcast-Network Coding simulation results.

Request Delivery Ratio

First, we investigate the reliability of the protocols. Since our protocols run on top of UDP which is unreliable, we made use of Alarm messages and network coding along with an acknowledgment mechanism to make them reliable. Thanks to these methods, all of the protocols achieved a 100% request delivery ratio at the expense of communication and computational delay overhead.

Communication Delay

We investigate the effect of the protocols on the communication delay between the SMs. The simulation results are shown in Fig. 8.5a. All the three protocols show an increasing tendency as the network grows. Since Bcast-Alarm and Bcast-NC protocols use broadcasting, the increase in these approaches can be attributed to the

contention at the medium access because the increase in the number of nodes that want to access the medium results in an increase in backoff waiting time. Another factor that raises the delay with the growing network is the probable increase in average distance between the gateway and randomly chosen 10 SMs at each topology. Ucast approach is affected by the latter only because it uses unicasting and there are randomly chosen 10 targeted SMs for all topology sizes.

When compared to Ucast protocol, Bcast-NC protocol takes less time while Bcast-Alarm takes more time. Since unicasting needs to communicate RTS/CTS (request to send/clear to send) messages before sending a frame, broadcasting performs better than unicasting. This is the reason that makes Bcast-NC perform better than Ucast. However, Bcast-Alarm approach suffers from excessive number of packets including Alarm messages and re-broadcast update requests which increase backoff waiting time and consequently the communication delay.

Completion Time

As can be seen in Fig. 8.5b, the values for all the protocols usually show an increasing trend. This is due to the contention during accessing the medium as explained in the previous subsection. However, the performance order among the protocols change in this metric due to the computational delay introduced by each protocol. For example, Bcast-NC requires the most time since it needs to run Gaussian elimination on the buffered encoded messages at every time an encoded message is received whereas Ucast requires the least time because it only verifies the signature on the request.

The fluctuations in Ucast and Bcast-Alarm protocols can be attributed to the positions of the randomly chosen SMs at each topology. Although this case is also present in communication delay for both protocols, it is affected much less than the completion time because we do not consider lost packets in the communication

delay computation. Nonetheless, loss of an update request/Alarm/Ack message whose communication delay is significant increases the completion time as much as the communication delay of this message. The farther a SM is randomly chosen, the more delay the completion time incurs.

Communication Throughput

The throughput was measured to analyze the bandwidth usage of the protocols. The values were computed based on the communication delay of each transmission. As shown in Fig. 8.5c, the values for Bcast-Alarm are very high and decreasing whereas those for Bcast-NC and Ucast protocols are almost fixed for all topology sizes. This is due to the fact that each SM having a copy of the request broadcasts it at each time an Alarm message is received in Bcast-Alarm. The decrease in Bcast-Alarm can be attributed to the increasing number of nodes while the number of targeted SMs is fixed because we are calculating an average for all SMs in the network.

Although the same size request packet is used by both Bcast-Alarm and Bcast-NC protocols, Bcast-NC produces less throughput because it broadcasts the request as encoded but smaller packets at the cost of computational overhead and a small data overhead due to the encoding vector. Also, Bcast-NC does not require to send the whole request when an ACK is not received by the gateway. Instead, the decoder waits for sufficient number of innovative encoded packets which are much smaller than the request itself.

Total Throughput

In total throughput, we took downloading the $CT_{sign}(FWUF)$ into consideration and the values were computed based on the completion time of each individual simulation. The total throughput values are given in Fig. 8.5d. As opposed to communication

throughput, the values fall as the network scales since the number of SMs is fixed for all topology sizes and we computed the total throughput per SM.

When we compare them between each other it can be seen that the order among the protocols is different than the communication throughput. Ucast protocol produces the most average throughput per SM whereas Bcast-NC produces the least. This is due to the fact that the protocols incur different amount of computational delays. Bcast-NC produces the least total throughput since it performs Gaussian elimination each time it receives an encoded packet, which is a computationally expensive operation. It is followed by Bcast-Alarm because it incurs designcryption operation whereas Ucast protocol incurs only signature verification which takes 27 times less time than designcryption operation.

8.4 Conclusion

In this chapter, we have investigated smart meter (SM) firmware update in the IEEE 802.11s-based AMI network. We proposed two secure and reliable multicast protocols based on ciphertext-policy attribute-based signcryption (CP-ABSC): Broadcast-Alarm (Bcast-Alarm) and Broadcast-Network Coding (Bcast-NC). CP-ABSC is employed to provide a secure and flexible SM notification. Bcast-Alarm protocol uses Alarm and ACK messages to provide reliability whereas Bcast-NC reduces bandwidth requirement by making use of network coding. We implemented the protocols and measured their performance in ns-3 network simulator. Simulation results indicate that all of the protocols are reliable by achieving 100% request delivery ratio and that Bcast-NC is more preferable compared to Bcast-Alarm because it consumes less network bandwidth although it takes more time to complete the process and requires more computational power. Since firmware update is not a real-time process and SMs are battery-free devices, these disadvantages can be ignored.

CHAPTER 9

CONCLUDING REMARKS & FUTURE WORK

In this dissertation, we tackled the consumer privacy issues in the IEEE 802.11s-based AMI network communications. In order to prevent the utility companies and eavesdroppers from analyzing the frequently collected power consumption data and thereby making inferences about household/manufacturing activities, we proposed several protocols by employing existing privacy-preserving techniques in the AMI network context. We used data obfuscation for privacy-preserving state estimation. Moreover, to increase the security and efficiency, we proposed a new design with multiple gateways. We implemented and tested different versions of the proposed protocol under the widely used network simulator ns-3 which has a draft implementation of IEEE 802.11s standard. Simulation results showed that the obfuscation approaches do not cause extra end-to-end delay and uses the channel bandwidth efficiently without introducing additional overhead on packet delivery ratio.

To reduce network traffic and conceal the private data irreversibly, we introduced some data aggregation methods into the AMI network data collection mechanism, that can perform arithmetic operations on concealed data. Specifically, we adapted fully homomorphic encryption (FHE) and secure multiparty computation (secure MPC). We tackled a new problem due to excessive fragmentation of FHE packets and proposed a novel solution by adding a new layer above the transport layer. We implemented and tested the proposed protocols under the ns-3 network simulator. Simulation results showed that the FHE-based protocol is not feasible for large-size networks in terms of the time to complete a data collection round, and that the secure MPC-based protocol is much more scalable than the FHE-based protocol in terms of bandwidth usage and average data collection completion time.

We proposed a scalable simulation framework for researchers and other parties that develop applications for the AMI network. We investigated the ns-3 DCE's network protocol stack profoundly and detected the factors that thwart the reliability and scalability. We used a modified version of the Constrained Application Protocol along with five different retransmission timeout (RTO) functions. In addition, we replaced the classical Address Resolution Protocol (ARP) with an efficient and piggybacking-based ARP to improve the proactive path discovery process. Moreover, we updated two critical parameters of the Link Management protocol to improve one-hop communication reliability. Simulation results showed that the proposed changes the reliability and scalability significantly, and that a fixed RTO value adjusted based on the network size can outperform some polynomial functions.

We built a testbed that implements IEEE 802.11s and ZigBee wireless mesh networking standards in order to mimic the AMI network. We presented a comprehensive explanation about how we built the testbed so that the readers can easily reproduce the same testbed environment. We employed FHE- and secure MPC-based protocols which are both secure and privacy-preserving along with SSL certificates and Elliptic Curve Digital Signature Algorithm (ECDSA) signatures for authentication. Moreover, we used End-to-End (EtoE) and Hop-by-Hop (HbyH) data aggregation mechanisms. Furthermore, we integrated the Constrained Application Protocol into the FHE-based protocol to provide both reliable and lightweight communication. By varying these parameters, we conducted a comprehensive performance comparison study. Experiment results indicated that the secure MPC-based protocol can be an alternative to the FHE-based protocol when the AMI network is built on IEEE 802.11s standard, and EtoE data aggregation is employed. On the other hand, the FHE-based protocol with Constrained Application Protocol (CoAP) can be a viable option if data collection is performed in more than every 6 minutes. Also, we inferred that CoAP

is the most suitable choice in terms of message delivery and throughput for data-intensive protocols such as the FHE-based protocol at the expense of long data collection completion times, and that HbyH data aggregation mechanism is much suited to be used for the AMI network since it reduces throughput and completion time significantly. Finally, experiment results showed that the protocols running on 802.11s mesh technology is more robust than those using ZigBee. However, ZigBee can be a good alternative to 802.11s mesh for the cases where a whole concealed reading can fit into a single ZigBee data packet and HbyH aggregation is employed. The testbed is accessible to the educator and researchers at `https://amitestbed.fiu.edu/`.

Finally, we developed two secure and reliable remote firmware update protocols based on ciphertext-policy attribute-based signcryption (CP-ABSC) and network coding. Both protocols were implemented and tested under the ns-3 network simulator. Simulation results showed that the CP-ABSC enables reliable multicast-over-broadcast while the network coding significantly reduces the bandwidth requirement.

Hereinafter, we present several key directions for future research.

- Throughout this dissertation, we used an implementation of the Smart-Vercauteren scheme, libScarab, where we referred to fully homomorphic encryption. Although it was useful for our tests, it suffers from key/ciphertext size and computational delay for multiplication. However, new implementations have been published since then such as HELib and SEAL [233]. These implementations can also be investigated for both privacy-preserving and feasible AMI network communications in further studies.
- One of the biggest problems we encountered while working on the scalable simulation framework was the simulation duration. 400- and 1024-node topologies took almost 2 and 5 days, respectively. Therefore, to distribute the workload, the proposed framework can be moved to clusters and be run in a distributed

manner by means of the standard Message Passing Interface. Also, there were not any applications running other than the periodic data reporting application in this work. In a future work, the frequency of the data reporting can be increased, and new applications can be installed along with the periodic data reporting. Moreover, we tested the framework on a 1024-node grid topology. Running it on a larger and random topology will definitely reveal some new problems. To this end, the framework can be improved such that it can be used for larger and random topologies as well.

- The ns-3 DCE can be used with the Linux network stack as well as its own stack implementation. However, we were unable to use it due to the problems we faced while integrating the PARP, so we used the ns-3 network stack. These problems can be resolved, and the simulations can be conducted on the Linux network stack in order to obtain more realistic results.

BIBLIOGRAPHY

[1] C. H. Liu, B. Yang, and T. Liu, "Efficient naming, addressing and profile services in internet-of-things sensory environments," *Ad Hoc Networks*, vol. 18, pp. 85–101, 2014.

[2] L. Wenpeng, "Advanced metering infrastructure," *Southern Power System Technology*, vol. 3, no. 2, pp. 6–10, 2009.

[3] N. Saputro, K. Akkaya, and S. Uludag, "A survey of routing protocols for smart grid communications," *Computer Networks*, vol. 56, no. 11, pp. 2742 – 2771, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128612001429

[4] W. Wang, Y. Xu, and M. Khanna, "A survey on the communication architectures in smart grid," *Computer Networks*, vol. 55, no. 15, pp. 3604–3629, 2011.

[5] C. Bennett and D. Highfill, "Networking ami smart meters," in *Energy 2030 Conference, 2008. ENERGY 2008. IEEE.* IEEE, 2008, pp. 1–8.

[6] J. Zheng, D. W. Gao, and L. Lin, "Smart meters in smart grid: An overview," in *Green Technologies Conference, 2013 IEEE.* IEEE, 2013, pp. 57–64.

[7] I. Akyildiz and X. Wang, "A survey on wireless mesh networks," *Communications Magazine, IEEE*, vol. 43, no. 9, pp. S23 – S30, sept. 2005.

[8] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer networks*, vol. 47, no. 4, pp. 445–487, 2005.

[9] J. Gao, Y. Xiao, J. Liu, W. Liang, and C. P. Chen, "A survey of communication/networking in smart grids," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 391–404, 2012.

[10] N. Saputro and K. Akkaya, "On preserving user privacy in smart grid advanced metering infrastructure applications," *Security and Communication Networks*, vol. 7, no. 1, pp. 206–220, 2014.

[11] G. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870 –1891, dec 1992.

[12] K. Kursawe, G. Danezis, and M. Kohlweiss, "Privacy-friendly aggregation for the smart-grid," in *Proceedings of the 11th international conference on Privacy enhancing technologies*, ser. PETS'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 175–191. [Online]. Available: http://dl.acm.org/citation.cfm?id=2032162.2032172

[13] "Stop smart meters." [Online]. Available: http://stopsmartmeters.org

[14] S. Tonyali, O. Cakmak, K. Akkaya, M. M. Mahmoud, and I. Guvenc, "Secure data obfuscation scheme to enable privacy-preserving state estimation in smart grid ami networks," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 709–719, 2016.

[15] S. Tonyali, N. Saputro, and K. Akkaya, "Assessing the feasibility of fully homomorphic encryption for smart grid ami networks," in *Ubiquitous and Future Networks (ICUFN), 2015 Seventh International Conference on*. IEEE, 2015, pp. 591–596.

[16] S. Tonyali, K. Akkaya, N. Saputro, and A. S. Uluagac, "A reliable data aggregation mechanism with homomorphic encryption in smart grid ami networks," in *Consumer Communications and Networking Conference (CCNC), 2016 IEEE*. IEEE, 2016, pp. 557–562.

[17] S. Tonyali, K. Akkaya, N. Saputro, A. S. Uluagac, and M. Nojoumian, "Privacy-preserving protocols for secure and reliable data aggregation in iot-enabled smart metering systems," *Future Generation Computer Systems*, 2017.

[18] C. Gentry, "Fully homomorphic encryption using ideal lattices," in *STOC*, vol. 9, 2009, pp. 169–178.

[19] 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*. Springer, 2010, pp. 24–43.

[20] C. Gentry and S. Halevi, "Implementing gentry's fully-homomorphic encryption scheme," in *Advances in Cryptology–EUROCRYPT 2011*. Springer, 2011, pp. 129–148.

[21] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," *SIAM Journal on Computing*, vol. 43, no. 2, pp. 831–871, 2014.

[22] O. Goldreich, "Secure multi-party computation," *Manuscript. Preliminary version*, pp. 86–97, 1998.

[23] M. M. Prabhakaran and A. Sahai, *Secure multi-party computation*. IOS press, 2013, vol. 10.

[24] R. Cramer, I. B. Damgård *et al.*, *Secure multiparty computation*. Cambridge University Press, 2015.

[25] S. Tonyali and K. Akkaya, "A scalable protocol stack for ieee 802.11s-based advanced metering infrastructure networks," in *Consumer Communications and Networking Conference (CCNC), 2018 IEEE*. IEEE, 2018.

[26] U. Ozgur, S. Tonyali, K. Akkaya, and F. Senel, "Comparative evaluation of smart grid ami networks: Performance under privacy," in *Computers and Communication (ISCC), 2016 IEEE Symposium on*. IEEE, 2016, pp. 1134–1136.

[27] U. Ozgur, S. Tonyali, and K. Akkaya, "Testbed and simulation-based evaluation of privacy-preserving algorithms for smart grid ami networks," in *Local Computer Networks Workshops (LCN Workshops), 2016 IEEE 41st Conference on*. IEEE, 2016, pp. 181–186.

[28] S. Tonyali, K. Akkaya, N. Saputro, and A. S. Uluagac, "A reliable data aggregation mechanism with homomorphic encryption in smart grid ami networks," in *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2016, pp. 550–555.

[29] P. Si, H. Ji, and F. R. Yu, "Optimal network selection in heterogeneous wireless multimedia networks," *Wireless Networks*, vol. 16, no. 5, pp. 1277–1288, 2010.

[30] K. Kolderup, "Introducing Bluetooth Mesh Networking," Bluetooth Blog, July 2017. [Online]. Available: https://blog.bluetooth.com/introducing-bluetooth-mesh-networking

[31] S. R. Schach, *Object-oriented and classical software engineering*. Boston: McGraw-Hill Higher Education, 2011.

[32] F. Skopik, Z. Ma, T. Bleier, and H. Grüneis, "A survey on threats and vulnerabilities in smart metering infrastructures," *International Journal of Smart Grid and Clean Energy*, vol. 1, no. 1, pp. 22–28, 2012.

[33] L. AlAbdulkarim and Z. Lukszo, "Information security implementation difficulties in critical infrastructures: Smart metering case," in *Networking, Sensing and Control (ICNSC), 2010 International Conference on*. IEEE, 2010, pp. 715–720.

[34] C. Hu, X. Cheng, Z. Tian, J. Yu, K. Akkaya, and L. Sun, "An attribute-based signcryption scheme to secure attribute-defined multicast communications," in *International Conference on Security and Privacy in Communication Systems*. Springer, 2015, pp. 418–437.

[35] Z. Yang, M. Li, and W. Lou, "A network coding approach to reliable broadcast in wireless mesh networks," *Wireless algorithms, systems, and applications*, pp. 234–243, 2009.

[36] Y. Kim, E. C.-H. Ngai, and M. B. Srivastava, "Cooperative state estimation for preserving privacy of user behaviors in smart grid," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 178–183.

[37] A. Alimardani, F. Therrien, D. Atanackovic, J. Jatskevich, and E. Vaahedi, "Distribution system state estimation based on nonsynchronized smart meters," *Smart Grid, IEEE Transactions on*, vol. 6, no. 6, pp. 2919–2928, 2015.

[38] A. Abdel-Majeed and M. Braun, "Low voltage system state estimation using smart meters," in *Universities Power Engineering Conference (UPEC), 2012 47th International*, Sept 2012, pp. 1–6.

[39] H. Perl, M. Brenner, and M. Smith, "Poster: an implementation of the fully homomorphic smart-vercauteren crypto-system," in *Proceedings of the 18th ACM conference on Computer and communications security*. ACM, 2011, pp. 837–840.

[40] N. P. Smart and F. Vercauteren, "Fully homomorphic encryption with relatively small key and ciphertext sizes," in *Public Key Cryptography–PKC 2010*. Springer, 2010, pp. 420–443.

[41] X. Zhang, C. Xu, C. Jin, R. Xie, and J. Zhao, "Efficient fully homomorphic encryption from rlwe with an extension to a threshold encryption scheme," *Future Generation Computer Systems*, vol. 36, pp. 180–186, 2014.

[42] Z. Brakerski and V. Vaikuntanathan, "Efficient fully homomorphic encryption from (standard) lwe," in *Proceedings of the 2011 IEEE 52nd Annual Symposium*

on Foundations of Computer Science, ser. FOCS '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 97–106. [Online]. Available: http://dx.doi.org/10.1109/FOCS.2011.12

[43] H.-M. Yang, Q. Xia, X.-f. Wang, and D.-h. Tang, "A new somewhat homomorphic encryption scheme over integers," in *Computer Distributed Control and Intelligent Environmental Monitoring (CDCIEM), 2012 International Conference on*. IEEE, 2012, pp. 61–64.

[44] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP Journal on Information Security*, vol. 2007, no. 1, p. 013801, 2007.

[45] Z. Brakerski, C. Gentry, and V. Vaikuntanathan, "(leveled) fully homomorphic encryption without bootstrapping," in *Proceedings of the 3rd Innovations in Theoretical Computer Science Conference*. ACM, 2012, pp. 309–325.

[46] D. Stehlé and R. Steinfeld, "Faster fully homomorphic encryption," in *Advances in Cryptology*. Springer, 2010, pp. 377–394.

[47] C. Gentry, "A fully homomorphic encryption scheme," Ph.D. dissertation, Stanford University, 2009, crypto.stanford.edu/craig.

[48] K. B. Frikken, "Secure multiparty computation," in *Algorithms and theory of computation handbook*. Chapman & Hall/CRC, 2010, pp. 14–14.

[49] K. Balasubramanian and M. Rajakani, "Secure multiparty computation," *Algorithmic Strategies for Solving Complex Problems in Cryptography*, p. 154, 2017.

[50] N. Saputro and K. Akkaya, "Performance evaluation of smart grid data aggregation via homomorphic encryption," in *Wireless Communications and Networking Conference (WCNC), 2012 IEEE*. IEEE, 2012, pp. 2945–2950.

[51] I. . W. Group *et al.*, "Ieee std. 802.15. 4-2006, part 15.4: Wireless medium access control (mac) and physical layer (phy) specifications for low-rate wireless personal area networks (wpans)," 2006.

[52] N. Saputro and K. Akkaya, "On preserving user privacy in smart grid advanced metering infrastructure applications," *Security and Communication Networks*, vol. 7, no. 1, pp. 206–220, 2014.

[53] S. Zeadally, A.-S. K. Pathan, C. Alcaraz, and M. Badra, "Towards privacy protection in smart grid," *Wireless personal communications*, vol. 73, no. 1, pp. 23–50, 2013.

[54] X. Liang, X. Li, R. Lu, X. Lin, and X. Shen, "Udp: Usage-based dynamic pricing with privacy preservation for smart grid," *Smart Grid, IEEE Transactions on*, vol. 4, no. 1, pp. 141–150, 2013.

[55] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "Eppdr: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 25, no. 8, pp. 2053–2064, 2014.

[56] W. Jia, H. Zhu, Z. Cao, X. Dong, and C. Xiao, "Human-factor-aware privacy-preserving aggregation in smart grid," *Systems Journal, IEEE*, vol. 8, no. 2, pp. 598–607, 2014.

[57] Y. Gong, Y. Cai, Y. Guo, and Y. Fang, "A privacy-preserving scheme for incentive-based demand response in the smart grid," *IEEE Transactions on Smart Grid*, 2015.

[58] Y. Kim, E.-H. Ngai, and M. Srivastava, "Cooperative state estimation for preserving privacy of user behaviors in smart grid," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, oct. 2011, pp. 178 –183.

[59] A. Arefi and M.-R. Haghifam, "State estimation in smart power grids," in *Smart Power Grids 2011*. Springer, 2012, pp. 439–478.

[60] S. Mallick, S. Ghoshal, P. Acharjee, S. Thakur *et al.*, "Optimal static state estimation using hybrid particle swarm-differential evolution based optimization," *Energy and Power Engineering*, vol. 5, no. 04, p. 670, 2013.

[61] I. Dzafic, S. Henselmeyer, and H.-T. Neisius, "High performance state estimation for smart grid distribution network operation," in *Innovative Smart Grid Technologies (ISGT), 2011 IEEE PES*. IEEE, 2011, pp. 1–6.

[62] A. Abdel-Majeed and M. Braun, "Low voltage system state estimation using smart meters," in *Universities Power Engineering Conference (UPEC), 2012 47th International*. IEEE, 2012, pp. 1–6.

[63] A. Abdel-Majeed, S. Tenbohlen, D. Schollhorn, and M. Braun, "Development of state estimator for low voltage networks using smart meters measurement data," in *PowerTech (POWERTECH), 2013 IEEE Grenoble*. IEEE, 2013, pp. 1–6.

[64] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 327–332.

[65] F. Li and B. Luo, "Preserving data integrity for smart grid data aggregation," in *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE, 2012, pp. 366–371.

[66] S. Ruj and A. Nayak, "A decentralized security framework for data aggregation and access control in smart grids," *IEEE transactions on smart grid*, vol. 4, no. 1, pp. 196–205, 2013.

[67] Z. Lu and Y. Wen, "Distributed algorithm for tree-structured data aggregation service placement in smart grid," *Systems Journal, IEEE*, vol. 8, no. 2, pp. 553–561, 2014.

[68] M. Ambrosin, H. Hosseini, K. Mandal, M. Conti, and R. Poovendran, "Despicable me(ter): Anonymous and fine-grained metering data reporting with dishonest meters," in *Conference on Communications and Network Security (CNS)*. IEEE, 2016.

[69] Y.-J. Kim, V. Kolesnikov, H. Kim, and M. Thottan, "Sstp: a scalable and secure transport protocol for smart grid data collection," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*. IEEE, 2011, pp. 161–166.

[70] T. Khalifa, A. Abdrabou, K. Naik, M. Alsabaan, A. Nayak, and N. Goel, "Split- and aggregated-transmission control protocol (sa-tcp) for smart power grid," *Smart Grid, IEEE Transactions on*, vol. 5, no. 1, pp. 381–391, 2014.

[71] P. Paillier *et al.*, "Public-key cryptosystems based on composite degree residuosity classes," in *Eurocrypt*, vol. 99. Springer, 1999, pp. 223–238.

[72] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[73] Y. Zhang and J.-L. Chen, "Wide-area scada system with distributed security framework," *Communications and Networks, Journal of*, vol. 14, no. 6, pp. 597–605, 2012.

[74] C. Rottondi, M. Savi, D. Polenghi, G. Verticale, and C. Kraus, "Implementation of a protocol for secure distributed aggregation of smart metering data," in *Smart Grid Technology, Economics and Policies (SG-TEP), 2012 International Conference on*. IEEE, 2012, pp. 1–4.

[75] C. Rottondi, G. Verticale, and C. Krauss, "Distributed privacy-preserving aggregation of metering data in smart grids," *Selected Areas in Communications, IEEE Journal on*, vol. 31, no. 7, pp. 1342–1354, 2013.

[76] C. Rottondi, G. Verticale, and C. Kraus, "Secure distributed data aggregation in the automatic metering infrastructure of smart grids," in *Communications (ICC), 2013 IEEE International Conference on*. IEEE, 2013, pp. 4466–4471.

[77] C. Thoma, T. Cui, and F. Franchetti, "Secure multiparty computation based privacy preserving smart metering system," in *North American Power Symposium (NAPS), 2012*. IEEE, 2012, pp. 1–6.

[78] ——, "Privacy preserving smart metering system based retail level electricity market," 2013.

[79] L. Yang, H. Xue, and F. Li, "Privacy-preserving data sharing in smart grid systems," in *Smart Grid Communications (SmartGridComm), 2014 IEEE International Conference on*. IEEE, 2014, pp. 878–883.

[80] M. Jung, W. Kastner, G. Kienesberger, and M. Leithner, "A comparison of web service technologies for smart meter data exchange," in *Innovative Smart Grid Technologies (ISGT Europe), 2012 3rd IEEE PES International Conference and Exhibition on*. IEEE, 2012, pp. 1–8.

[81] G. Tanganelli, E. Mingozzi, C. Vallati, and C. Cicconetti, "A distributed architecture for discovery and access in the internet of things," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*. IEEE, 2013, pp. 45–46.

[82] J. Kim, F. Filali, and Y.-B. Ko, "A lightweight coap-based software defined networking for resource constrained ami devices," in *Smart Grid Communications (SmartGridComm), 2015 IEEE International Conference on*. IEEE, 2015, pp. 719–724.

[83] Z. Shelby, K. Hartke, and C. Bormann, "The constrained application protocol (coap)," 2014.

[84] A. Gurtov and E. Dashkova, "Computing the retransmission timeout in coap," 2013.

[85] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "Congestion control in reliable coap communication," in *Proceedings of the 16th ACM international conference on Modeling, analysis & simulation of wireless and mobile systems*. ACM, 2013, pp. 365–372.

[86] C. Bormann, A. Betzler, C. Gomez, and I. Demirkol, "Coap simple congestion control/advanced," 2017.

[87] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with cooja," in *Local computer networks, proceedings 2006 31st IEEE conference on*. IEEE, 2006, pp. 641–648.

[88] A. Betzler, C. Gomez, I. Demirkol, and M. Kovatsch, "Congestion control for coap cloud services," in *Emerging Technology and Factory Automation (ETFA), 2014 IEEE*. IEEE, 2014, pp. 1–6.

[89] M. Kovatsch, M. Lanter, and Z. Shelby, "Californium: Scalable cloud services for the internet of things with coap," in *Internet of Things (IOT), 2014 International Conference on the*. IEEE, 2014, pp. 1–6.

[90] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "Flocklab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proceedings of the 12th international conference on Information processing in sensor networks*. ACM, 2013, pp. 153–166.

[91] A. Betzler, C. Gomez, I. Demirkol, and J. Paradells, "Cocoa+: An advanced congestion control mechanism for coap," *Ad Hoc Networks*, vol. 33, pp. 126–139, 2015.

[92] ——, "Coap congestion control for the internet of things," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 154–160, 2016.

[93] R. Bhalerao, S. S. Subramanian, and J. Pasquale, "An analysis and improvement of congestion control in the coap internet-of-things protocol," in *Consumer Communications & Networking Conference (CCNC), 2016 13th IEEE Annual*. IEEE, 2016, pp. 889–894.

[94] I. Järvinen, L. Daniel, and M. Kojo, "Experimental evaluation of alternative congestion control algorithms for constrained application protocol (coap)," in *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*. IEEE, 2015, pp. 453–458.

[95] J. Huang, L. Wang, and C. Chang, "Scalability in wireless mesh networks," in *Wireless Mesh Networking: Architectures, Protocols and Standards*. Auerbach Publications, 2006, pp. 225–261.

[96] S. Srivathsan, N. Balakrishnan, and S. Iyengar, "Scalability in wireless mesh networks," in *Guide to Wireless Mesh Networks*. Springer, 2009, pp. 325–347.

[97] B. Nassereddine, A. Maach, and S. Bennani, "The scalability of the hybrid protocol in wireless mesh network 802.11 s," in *Microwave Symposium (MMS), 2009 Mediterrannean*. IEEE, 2009, pp. 1–7.

[98] ns 2, "ns-2: Network Simulator 2," 2002. [Online]. Available: http://www.isi.edu/nsnam/ns/

[99] N. Saputro and K. Akkaya, "An efficient arp for large-scale ieee 802.11 s-based smart grid networks," in *Local Computer Networks (LCN), 2013 IEEE 38th Conference on*. IEEE, 2013, pp. 723–726.

[100] ——, "An efficient and secure arp for large-scale ieee 802.11 s-based smart grid networks," in *International Conference on Ad Hoc Networks*. Springer, 2013, pp. 214–228.

[101] ——, "Parp-s: A secure piggybacking-based arp for ieee 802.11 s-based smart grid ami networks," *Computer Communications*, vol. 58, pp. 16–28, 2015.

[102] R. G. Garroppo, S. Giordano, and L. Tavanti, "A joint experimental and simulation study of the ieee 802.11 s hwmp protocol and airtime link metric," *International Journal of Communication Systems*, vol. 25, no. 2, pp. 92–110, 2012.

[103] M. Singh, S.-G. Lee, W. K. Tan, and J. H. Lam, "Throughput analysis of wireless mesh network test-bed," *Convergence and Hybrid Information Technology*, pp. 54–61, 2011.

[104] T. Imboden, K. Akkaya, and Z. Moore, "Performance evaluation of wireless mesh networks using ieee 802.11 s and ieee 802.11 n," in *Communications (ICC), 2012 IEEE International Conference on*. IEEE, 2012, pp. 5675–5679.

[105] Y. Takahashi, Y. Owada, H. Okada, and K. Mase, "A wireless mesh network testbed in rural mountain areas," in *Proceedings of the second ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*. ACM, 2007, pp. 91–92.

[106] A. Zimmermann, M. Gunes, M. Wenig, U. Meis, and J. Ritzerfeld, "How to study wireless mesh networks: A hybrid testbed approach," in *Advanced Information Networking and Applications, 2007. AINA'07. 21st International Conference on*. IEEE, 2007, pp. 853–860.

[107] H. Song, B. C. Kim, J. Y. Lee, and H. S. Lee, "Ieee 802.11-based wireless mesh network testbed," in *Mobile and Wireless Communications Summit, 2007. 16th IST*. IEEE, 2007, pp. 1–5.

[108] M. Abolhasan, B. Hagelstein, and J.-P. Wang, "Real-world performance of current proactive multi-hop mesh protocols," in *Communications, 2009. APCC 2009. 15th Asia-Pacific Conference on*. IEEE, 2009, pp. 44–47.

[109] A. Hamidian, C. E. Palazzi, T. Y. Chong, J. M. Navarro, U. Körner, and M. Gerla, "Deployment and evaluation of a wireless mesh network," in *Advances in Mesh Networks, 2009. MESH 2009. Second International Conference on*. IEEE, 2009, pp. 66–72.

[110] J. Eriksson, S. Agarwal, P. Bahl, and J. Padhye, "Feasibility study of mesh networks for all-wireless offices," in *Proceedings of the 4th International Conference on Mobile systems, applications and services*. ACM, 2006, pp. 69–82.

[111] D. Wu, D. Gupta, and P. Mohapatra, "Qurinet: A wide-area wireless mesh testbed for research and experimental evaluations," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1221–1237, 2011.

[112] S. Furrer, W. Schott, H. L. Truong, and B. Weiss, "The ibm wireless sensor networking testbed," in *Testbeds and Research Infrastructures for the Development of Networks and Communities, 2006. TRIDENTCOM 2006. 2nd International Conference on*. IEEE, 2006, pp. 5–pp.

[113] D. Bansal and S. Sofat, "Deployment and evaluation of ieee 802.11 based wireless mesh networks in campus environment," in *Proceedings of the 4th ACM Workshop on Networked Systems for Developing Regions*. ACM, 2010, p. 15.

[114] P. R. Casey, K. E. Tepe, and N. Kar, "Design and implementation of a testbed for ieee 802.15. 4 (zigbee) performance measurements," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, p. 23, 2010.

[115] M. Franceschinis, C. Pastrone, M. A. Spirito, and C. Borean, "On the performance of zigbee pro and zigbee ip in ieee 802.15. 4 networks," in *Wireless and Mobile Computing, Networking and Communications (WiMob), 2013 IEEE 9th International Conference on*. IEEE, 2013, pp. 83–88.

[116] M. D. Abrignani, C. Buratti, D. Dardari, N. El Rachkidy, A. Guitton, F. Martelli, A. Stajkic, and R. Verdone, "The euwin testbed for 802.15. 4/zigbee networks: From the simulation to the real world," in *Wireless Communication Systems (ISWCS 2013), Proceedings of the Tenth International Symposium on*. VDE, 2013, pp. 1–5.

[117] J. Caskey, "Nema sg-ami 1 requirements for smart meter upgradeability," National Electrical Manufacturers Association, Technical Requirements Document, 2009.

[118] E. P. R. Institute, "Nema sg-ami 1 requirements for smart meter upgradeability," Electric Power Research Institute, Technical Guide, Apr 2010.

[119] Y.-j. Kim, D.-e. Oh, J.-m. Ko, Y.-i. Kim, S.-j. Kang, and S.-H. Choi, "A remote firmware upgrade method of nan and han devices to support amis energy services," in *International Conference on Hybrid Information Technology*. Springer, 2011, pp. 303–310.

[120] J. Simmins, "Remote meter firmware update," American Electric Power, Use Case Document, 2011.

[121] M. Kuzlu, M. Pipattanasomporn, and S. Rahman, "Communication network requirements for major smart grid applications in han, nan and wan," *Computer Networks*, vol. 67, pp. 74–88, 2014.

[122] L. Katzir and I. Schwartzman, "Secure firmware updates for smart grid devices," in *Innovative Smart Grid Technologies (ISGT Europe), 2011 2nd IEEE PES International Conference and Exhibition on*. IEEE, 2011, pp. 1–5.

[123] Z. M. Fadlullah, N. Kato, R. Lu, X. Shen, and Y. Nozaki, "Toward secure targeted broadcast in smart grid," *IEEE Communications Magazine*, vol. 50, no. 5, pp. 150–156, 2012.

[124] C. H. Liu, J. Fan, J. W. Branch, and K. K. Leung, "Toward qoi and energy-efficiency in internet-of-things sensory environments," *Emerging Topics in Computing, IEEE Transactions on*, vol. 2, no. 4, pp. 473–487, 2014.

[125] C. H. Liu, J. Fan, P. Hui, J. Crowcroft, and G. Ding, "Qoi-aware energy-efficient participatory crowdsourcing," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3742–3753, 2013.

[126] C. Perera, A. Zaslavsky, C. H. Liu, M. Compton, P. Christen, and D. Georgakopoulos, "Sensor search techniques for sensing as a service architecture for the internet of things," *Sensors Journal, IEEE*, vol. 14, no. 2, pp. 406–420, 2014.

[127] C. Perera, C. H. Liu, S. Jayawardena, and M. Chen, "A survey on internet of things from industrial market perspective," *Access, IEEE*, vol. 2, pp. 1660–1679, 2014.

[128] C. Perera, C. H. Liu, and S. Jayawardena, "The emerging internet of things marketplace from an industrial perspective: A survey," *IEEE Transactions on Emerging Topics in Computing*, vol. 3, no. 4, pp. 585–598, 2015.

[129] N. Saputro, K. Akkaya, and S. Uludag, "A survey of routing protocols for smart grid communications," *Computer Networks*, vol. 56, no. 11, pp. 2742–2771, 2012.

[130] R. Podmore and M. R. Robinson, "The role of simulators for smart grid development," *Smart Grid, IEEE Transactions on*, vol. 1, no. 2, pp. 205–212, 2010.

[131] N. Saputro and K. Akkaya, "PARP-S: A secure piggybacking-based ARP for IEEE 802.11s-based smart grid AMI networks," *Computer Communications*, 2014.

[132] ns 3, "ns-3: network simulator 3," Release 3.24.1, 2016. [Online]. Available: http://www.nsnam.org/

[133] The status of ieee 802.11s standard. [Online]. Available: http://grouper.ieee.org/groups/802/11/Reports/tgsupdate.htm

[134] D. Wu, L. Bao, and C. H. Liu, "Scalable channel allocation and access scheduling for wireless internet-of-things," *Sensors Journal, IEEE*, vol. 13, no. 10, pp. 3596–3604, 2013.

[135] D. Tsolkas, E. Liotou, N. Passas, and L. Merakos, "Lte-a access, core, and protocol architecture for d2d communication," in *Smart Device to Smart Device Communication*. Springer, 2014, pp. 23–40.

[136] G. Poitau, B. Pelletier, G. Pelletier, and D. Pani, "A combined PUSH/PULL service discovery model for LTE direct," in *Vehicular Technology Conference (VTC Fall), 2014 IEEE 80th*. IEEE, 2014, pp. 1–5.

[137] 3GPP, "Technical specification group services and system aspects," 3GPP Technical Report 23.703 v0.5, Release 12, 2014. [Online]. Available: https://www.arib.or.jp/english/html/overview/doc/STD-T63v11_00/5_Appendix/Rel12/23/23703-c00.pdf

[138] W. Dai, "Crypto++ library 5.1-a free c++ class library of cryptographic schemes," *http://www.cryptopp.com/*, 2011.

[139] G. Locke and P. Gallagher, "Fips pub 186-3: Digital signature standard (dss)," *Federal Information Processing Standards Publication*, vol. 3, pp. 186–3, 2009.

[140] V. Gayoso Martnez, L. Hernndez Encinas, and C. Snchez vila, "A survey of the elliptic curve integrated encryption scheme," *Journal of Computer Science and Engineering*, vol. 2, no. 2, 2010.

[141] J. A. Calabro, S. R. Calabro, and P. R. Mich, "Remote automatic meter reading and control system," Aug. 19 1975, uS Patent 3,900,842.

[142] M. H. Cintuglu, H. Martin, and O. A. Mohammed, "Real-time implementation of multiagent-based game theory reverse auction model for microgrid market operation," *IEEE Transactions on Smart Grid*, vol. 6, no. 2, pp. 1064–1072, 2015.

[143] M. H. Cintuglu, T. Youssef, and O. A. Mohammed, "Development and application of a real-time testbed for multiagent system interoperability: A case study on hierarchical microgrid control," *IEEE Transactions on Smart Grid*, 2016.

[144] "Smart grid functions." [Online]. Available: https://www.smartgrid.gov/files/definition_of_functions.pdf

[145] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.

[146] G. W. Hart, "Nonintrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, pp. 1870–1891, 1992.

[147] X. Fang, S. Misra, G. Xue, and D. Yang, "Managing smart grid information in the cloud: opportunities, model, and applications," *Network, IEEE*, vol. 26, no. 4, pp. 32–38, 2012.

[148] G. R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, and B. Walke, "Ieee 802.11 s: the wlan mesh standard," *Wireless Communications, IEEE*, vol. 17, no. 1, pp. 104–111, 2010.

[149] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.

[150] "Cooper industries ami solutions - rf mesh smart grid network." [Online]. Available: http://www.cooperindustries.com/content/public/en/power_systems/solutions/ami.html

[151] "Cyan technology ami solutions - smart electricity metering." [Online]. Available: http://www.cyantechnology.com/smart-electricity-metering/

[152] "Trilliant ami solutions - smart metering." [Online]. Available: http://trilliantinc.com/solutions/metering

[153] V. H. Muntean and M. Otesteanu, "Wimax versus lte-an overview of technical aspects for next generation networks technologies," in *Electronics and Telecommunications (ISETC), 2010 9th International Symposium on.* IEEE, 2010, pp. 225–228.

[154] S. Monk, *Raspberry Pi cookbook: Software and hardware problems and solutions.* " O'Reilly Media, Inc.", 2016.

[155] M. Kirschbaum, T. Plos, and J.-M. Schmidt, "On secure multi-party computation in bandwidth-limited smart-meter systems," in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on.* IEEE, 2013, pp. 230–235.

[156] W. Diffie and M. E. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, 1976.

[157] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified vss and fast-track multiparty computations with applications to threshold cryptography," in *Proceed-*

ings of the seventeenth annual ACM symposium on Principles of distributed computing. ACM, 1998, pp. 101–111.

[158] M. Ben-Or, S. Goldwasser, and A. Wigderson, "Completeness theorems for non-cryptographic fault-tolerant distributed computation," in *Proceedings of the twentieth annual ACM symposium on Theory of computing*. ACM, 1988, pp. 1–10.

[159] A. Beussink, K. Akkaya, I. F. Senturk, and M. M. Mahmoud, "Preserving consumer privacy on ieee 802.11 s-based smart grid ami networks using data obfuscation," in *Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on*. IEEE, 2014, pp. 658–663.

[160] S. Floyd, "Highspeed tcp for large congestion windows," 2003.

[161] S. Rai and S. Sharma, "Determining minimum spanning tree in an undirected weighted graph," in *Computer Engineering and Applications (ICACEA), 2015 International Conference on Advances in*. IEEE, 2015, pp. 637–642.

[162] Y. Zhang, J. Luo, and H. Hu, *Wireless mesh networking: architectures, protocols and standards*. CRC Press, 2006.

[163] N. Saputro, K. Akkaya, and S. Tonyali, "Addressing network interoperability in hybrid ieee 802.11 s/lte smart grid communications," in *Local Computer Networks (LCN), 2016 IEEE 41st Conference on*. IEEE, 2016, pp. 623–626.

[164] H. Mohammed, S. Tonyali, K. Rabieh, M. Mahmoud, and K. Akkaya, "Efficient privacy-preserving data collection scheme for smart grid ami networks," in *Proc. of IEEE Globecom*, 2016.

[165] K. Rabieh, M. Mahmoud, S. Tonyali *et al.*, "Scalable certificate revocation schemes for smart grid ami networks using bloom filters," *IEEE Transactions on Dependable and Secure Computing*, 2015.

[166] K. Akkaya, K. Rabieh, M. Mahmoud, and S. Tonyali, "Customized certificate revocation lists for ieee 802.11 s-based smart grid ami networks," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2366–2374, 2015.

[167] P. Siano, "Demand response and smart grids - a survey," *Renewable and Sustainable Energy Reviews*, vol. 30, pp. 461–478, 2014.

[168] F. Rahimi and A. Ipakchi, "Demand response as a market resource under the smart grid paradigm," *IEEE Transactions on Smart Grid*, vol. 1, no. 1, pp. 82–88, 2010.

[169] M. Allman, V. Paxson, and E. Blanton, "Tcp congestion control," IETF, Tech. Rep., 2009.

[170] H. Tazaki, F. Urbani, and T. Turletti, "Dce cradle: simulate network protocols with real stacks for better realism," in *Proceedings of the 6th International ICST Conference on Simulation Tools and Techniques*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2013, pp. 153–158.

[171] D. Plummer, "Ethernet address resolution protocol: Or converting network protocol addresses to 48. bit ethernet address for transmission on ethernet hardware," 1982.

[172] EventHelix, "ARP Sequence Diagram," 2017. [Online]. Available: http://www.eventhelix.com/RealtimeMantra/Networking/Arp.pdf

[173] A. Joshi, M. Bahr *et al.*, "Hwmp specification," *IEEE P802*, vol. 11, pp. 802–11, 2006.

[174] O. Bergmann, "libcoap: C-implementation of coap," Version 4.1.2, 2017. [Online]. Available: https://libcoap.net/

[175] S. Bu, F. R. Yu, and P. X. Liu, "Dynamic pricing for demand-side management in the smart grid," in *Online Conference on Green Communications (GreenCom), 2011 IEEE*. IEEE, 2011, pp. 47–51.

[176] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, "Tcp performance issues over wireless links," *IEEE communications magazine*, vol. 39, no. 4, pp. 52–58, 2001.

[177] J. P. Lang, "Link management protocol (lmp)," 2005.

[178] O. S. G. Platform, "AMI Use Cases and Business Functions," Reference: IEEE 1471-2000, 2009. [Online]. Available: http://osgug.ucaiug.org/UtiliComm/Shared%20Documents/AMI%20Use%20Cases%20and%20Business%20Functions.doc

[179] U. N. D. of Economic, *World population prospects: The 2006 revision*. United Nations Publications, 2007, vol. 261.

[180] S. Hammer, L. Kamal-Chaoui, A. Robert, and M. Plouin, "Cities and green growth: a conceptual framework," *OECD Regional Development Working Papers*, vol. 2011, no. 8, p. 1, 2011.

[181] P. Lombardi, S. Giordano, H. Farouh, and W. Yousef, "Modelling the smart city performance," *Innovation: The European Journal of Social Science Research*, vol. 25, no. 2, pp. 137–149, 2012.

[182] V. Albino, U. Berardi, and R. M. Dangelico, "Smart cities: Definitions, dimensions, performance, and initiatives," *Journal of Urban Technology*, vol. 22, no. 1, pp. 3–21, 2015.

[183] H. Farhangi, "The path of the smart grid," *IEEE power and energy magazine*, vol. 8, no. 1, 2010.

[184] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *IEEE power and energy magazine*, vol. 3, no. 5, pp. 34–41, 2005.

[185] N. M. G. Strategy, "Advanced metering infrastructure," *US Department of Energy Office of Electricity and Energy Reliability*, 2008.

[186] R. R. Mohassel, A. Fung, F. Mohammadi, and K. Raahemifar, "A survey on advanced metering infrastructure," *International Journal of Electrical Power & Energy Systems*, vol. 63, pp. 473–484, 2014.

[187] SiteSafe, "Fpl ami & distribution automation," RF Exposure Survey, 2011. [Online]. Available: https://www.fpl.com/smart-meters/pdf/site-safe.pdf

[188] FPL, "Smart meters," FPL's Smart Meter Installation, 2017. [Online]. Available: https://www.fpl.com/smart-meters/info.html

[189] S. Spring, "Silver spring networks announces agreement with florida power & light company," Press Release, 2007. [Online]. Available: https://www.silverspringnet.com/article/press-release/silver-spring-networks-announces-agreement-with-florida-power-light-company/

[190] A. Illinois, "Wireless communications and advanced meters," Frequently Asked Questions, 2017. [Online]. Available: https://www.ameren.com/illinois/map/faqs-radio-frequency-advanced-meter

[191] Z. Alliance, "Zigbee specification (document 053474r20)," *ZigBee Alliance: San Ramon, CA, USA*, 2012.

[192] K.-H. Chang and B. Mason, "The ieee 802.15. 4g standard for smart metering utility networks," in *Smart Grid Communications (SmartGridComm), 2012 IEEE Third International Conference on*. IEEE, 2012, pp. 476–480.

[193] U. Ozgur, "Development of a remotely accessible wireless testbed for performance evaluation of ami related protocols," MS Thesis, 2017. [Online]. Available: http://digitalcommons.fiu.edu/cgi/viewcontent.cgi?article=4212&context=etd

[194] "Grid stream - rf router," Product Specifications, 2012. [Online]. Available: https://www.landisgyr.com/webfoo/wp-content/uploads/2012/12/PS_GridstreamRFRouter.pdf

[195] "Grid stream - c6500 rf collector," Product Specifications, 2017. [Online]. Available: https://www.landisgyr.com/webfoo/wp-content/uploads/2017/12/PS_GS-RF-C6500-Collector-20171115.pdf

[196] "Cisco 1000 series connected grid routers data sheet," Data Sheets, 2017. [Online]. Available: https://www.cisco.com/c/en/us/products/collateral/routers/1000-series-connected-grid-routers/datasheet_c78-696278.html

[197] "Openway riva adaptive communications technology," White Paper, 2014. [Online]. Available: https://goo.gl/aDf7zZ

[198] Digi, "XBee/XBee-PRO S2C ZigBee," 2017. [Online]. Available: https://www.digi.com/resources/documentation/digidocs/pdfs/90002002.pdf

[199] Raspberry Pi, "Raspberry Pi 3 Model B," 2017. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[200] Sparkfun, "SparkFun XBee Explorer Dongle," 2017. [Online]. Available: https://www.sparkfun.com/products/11697

[201] M. Burunkaya and T. Pars, "A smart meter design and implementation using zigbee based wireless sensor network in smart grid," in *Electrical and Electronic*

Engineering (ICEEE), 2017 4th International Conference on. IEEE, 2017, pp. 158–162.

[202] R. Technology, "Ralink RT3070 Datasheet," 2008. [Online]. Available: http://www.dingsung.com.cn/attachment.php?id=97&tid=1&filename=1362484609.pdf

[203] "Florida international university advanced metering infrastructure testbed," FIU AMI Testbed Website, 2017. [Online]. Available: https://amitestbed.fiu.edu/

[204] S. Farahani, *ZigBee wireless networks and transceivers*. Newnes, 2011.

[205] D. I. Inc., "Xbee/xbee-pro s2c zigbee rf module user guide," 2016.

[206] J. Wang, *Computer network security: theory and practice*. Springer Publishing Company, Incorporated, 2009.

[207] A. J. Menezes, *Elliptic curve public key cryptosystems*. Springer Science & Business Media, 2012, vol. 234.

[208] D. Cooper, "Internet x. 509 public key infrastructure certificate and certificate revocation list (crl) profile," 2008.

[209] W. Harrington, *Learning Raspbian*. Packt Publishing Ltd, 2015.

[210] Linux Wireless, "Existing Linux Wireless Drivers," 2016. [Online]. Available: https://wireless.wiki.kernel.org/en/users/drivers

[211] D. I. Inc., "Xctu next generation configuration platform for xbee," 1996.

[212] "Rxtx java library," http://rxtx.qbang.org/wiki/index.php/Main_Page, accessed: 2017-10-07.

[213] M. Van Smoorenburg and J. Lahtinen, "Minicom, a friendly menu driven serial communication program," 2006.

[214] K. Rankin, "Hack and: manage multiple servers efficiently," *Linux Journal*, vol. 2009, no. 177, p. 13, 2009.

[215] I. S. Association *et al.*, "802.11-2012-ieee standard for information technology–telecommunications and information exchange between systems local and

metropolitan area networks–specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications," *Retrived from http://standards. ieee. org/about/get/802/802.11. html*, 2012.

[216] E. Käsper, "Fast elliptic curve cryptography in openssl," in *International Conference on Financial Cryptography and Data Security*. Springer, 2011, pp. 27–39.

[217] "Privacy-preserving protocols for the ami network," https://amitestbed.fiu.edu/downloads, accessed: 2017-10-09.

[218] T. Aure and F. Y. Li, "An optimized path-selection using airtime metric in olsr networks: Implementation and testing," in *Wireless Communication Systems. 2008. ISWCS'08. IEEE International Symposium on*. IEEE, 2008, pp. 359–363.

[219] G. Blakley and G. Kabatianskii, "Linear algebra approach to secret sharing schemes," in *Error Control, Cryptology, and Speech Compression*. Springer, 1994, pp. 33–40.

[220] P. Yi, A. Iwayemi, and C. Zhou, "Developing zigbee deployment guideline under wifi interference for smart grid applications," *IEEE transactions on smart grid*, vol. 2, no. 1, pp. 110–120, 2011.

[221] "Silver spring networks - communications module for electricity meters." [Online]. Available: http://www.silverspringnet.com/pdfs/SilverSpring-Datasheet-Communications-Modules.pdf

[222] I. O. CENTRON, "Itron openway centron smart meter," Web Page, 2016. [Online]. Available: https://www.arm.com/markets/embedded/15421.php

[223] M. Chase, "Multi-authority attribute based encryption," *Theory of Cryptography*, pp. 515–534, 2007.

[224] M. V. Pedersen, J. Heide, and F. H. Fitzek, "Kodo: An open and research oriented network coding library," in *International Conference on Research in Networking*. Springer, 2011, pp. 145–152.

[225] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based encryption for fine-grained access control of encrypted data," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 89–98.

[226] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attribute-based encryption," in *Security and Privacy, 2007. SP'07. IEEE Symposium on*. Ieee, 2007, pp. 321–334.

[227] R. Berthier, W. H. Sanders, and H. Khurana, "Intrusion detection for advanced metering infrastructures: Requirements and architectural directions," in *Smart Grid Communications (SmartGridComm), 2010 First IEEE International Conference on*. IEEE, 2010, pp. 350–355.

[228] T. Ho, R. Koetter, M. Medard, D. R. Karger, and M. Effros, "The benefits of coding over routing in a randomized setting," 2003.

[229] F. Skopik and P. D. Smith, *Smart Grid Security: Innovative Solutions for a Modernized Grid*. Syngress, 2015.

[230] W. Dai, "Crypto++ library," 2007.

[231] Z. Yang, M. Li, and W. Lou, "R-code: Network coding-based reliable broadcast in wireless mesh networks," *Ad Hoc Networks*, vol. 9, no. 5, pp. 788–798, 2011.

[232] R. P. Foundation, "Raspberry pi 3 model b," 2016. [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b/

[233] H. Consortium, "Homomorphic encryption standardization," 2017. [Online]. Available: http://homomorphicencryption.org/

.1 Commands for the IEEE 802.11s-based Testbed

A helpful tool *usbutils* and the firmware can be downloaded and installed on Raspi3 by running the following commands.

```
$sudo apt-get update
$sudo apt-get install wireless-tools usbutils
$sudo apt-get install firmware-ralink
```

where $ sign represents the terminal prompt. We use *sudo* command to get root privileges. This is required for kernel-space operations.

iw tool can be installed, and $NetworkManager$ service can be stopped by running the following commands.

```
$sudo apt-get install iw
$sudo killall NetworkManager
```

The network interface can be configured by running the following commands.

```
$sudo iw dev wlan0 interface add IfName type mp
$sudo iw dev IfName set channel 11
$sudo ifconfig IfName IPAdd netmask Mask up
$sudo iw dev IfName mesh join MeshID
```

where IfName, IPAdd, MeshID and Mask represent the interface name, the assigned IP address, the mesh identifier and the subnet mask, respectively.

The ARP cache on RasPi3s can be manipulated by running the following commands.

```
$sudo arp -i IfName -s IPAdd MACAdd
```

where MACAdd represents the MAC address associated with the IPAdd.

VITA

SAMET TONYALI

July 21, 1988	Born, Trabzon, TURKEY
2006-2011	B.Sc., Computer Engineering Marmara University, Istanbul, TURKEY
2011-2013	M.Sc., Computer Engineering Marmara University, Istanbul, TURKEY
2015-2017	Doctoral Candidate, Electrical Engineering Florida International University Miami, Florida, USA
2017	Award, Dissertation Year Fellowship Florida International University Miami, Florida, USA

SELECTED PUBLICATIONS AND PRESENTATIONS

J1) K. Akkaya, K. Rabieh, M. Mahmoud, S. Tonyali, *Customized Certificate Revocation Lists for IEEE 802.11s-based Smart Grid AMI Networks* published in IEEE Transactions on Smart Grid (Sep. 2015).

J2) S. Tonyali, O. Cakmak, K. Akkaya, M.M. Mahmoud, I. Guvenc, *Secure Data Obfuscation Scheme to Enable Privacy-Preserving State Estimation in Smart Grid AMI Networks* published in IEEE Internet of Things Journal (Oct. 2016).

J3) S. Tonyali, K. Akkaya, N. Saputro, A.S. Uluagac, M. Nojoumian, *Privacy-Preserving Protocols for Secure and Reliable Data Aggregation in IoT-Enabled Smart Metering Systems* published in Future Generation Computer Systems Journal (Apr. 2017).

J4) K. Rabieh, M.M. Mahmoud, K. Akkaya, S. Tonyali, *Scalable Certificate Revocation Schemes for Smart Grid AMI Networks Using Bloom Filters* published in IEEE Transactions on Dependable and Secure Computing (Jul. 2017).

J5) S. Tonyali, R. Munoz, , K. Akkaya, U. Ozgur, *A Realistic Performance Evaluation of Privacy-Preserving Protocols for Smart Grid AMI Networks* submitted to Elsevier Journal of Network and Computer Applications (Dec. 2017).

C1) M. Mahmoud, K. Akkaya, K. Rabieh, S. Tonyali, *An Efficient Certificate Revocation Scheme for Large-scale AMI Networks* in Proc. of Performance Computing

and Communications Conference, Dec. 2014.

C2) S. Tonyali, N. Saputro, K. Akkaya, *Assessing the Feasibility of Fully Homomorphic Encryption for Smart Grid AMI Networks* in Proc. of Ubiquitous and Future Networks, Jul. 2015.

C3) S. Tonyali, K. Akkaya, N. Saputro, A.S. Uluagac, *A Reliable Data Aggregation Mechanism with Homomorphic Encryption in Smart Grid AMI Networks* in Proc. of IEEE Consumer Communications & Networking Conference, Jan. 2016.

C4) U. Ozgur, S. Tonyali, K. Akkaya, F. Senel, *Comparative Evaluation of Smart Grid AMI Networks: Performance under Privacy* in Proc. of IEEE Symposium on Computers and Communication, Jun. 2016.

C5) N. Saputro, K. Akkaya, S. Tonyali, *Addressing Network Interoperability in Hybrid IEEE 802.11s/LTE Smart Grid Communications* in Proc. of Local Computer Networks, Nov. 2016.

C6) U. Ozgur, S. Tonyali, K. Akkaya, *Testbed and Simulation-Based Evaluation of Privacy-Preserving Algorithms for Smart Grid AMI Networks* in Proc. of Local Computer Networks Workshops, Nov. 2016.

C7) H. Mohammed, S. Tonyali, K. Rabieh, M. Mahmoud, K. Akkaya, *Efficient Privacy-Preserving Data Collection Scheme for Smart Grid AMI Networks*in Proc. of Global Communications Conference, Dec. 2016.

C8) S. Tonyali, K. Akkaya, N. Saputro, X. Cheng, *An Attribute & Network Coding-Based Secure Multicast Protocol for Firmware Updates in Smart Grid AMI Networks* in Proc. of International Conference on Computer Communication and Networks, Jul. 2017.

C9) S. Tonyali, K. Akkaya, N. Saputro, *An Attribute-based Reliable Multicast-over-Broadcast Protocol for Firmware Updates in Smart Meter Networks* in Proc. of IEEE International Conference on Computer Communications, May 2017.

C10) N. Saputro, S. Tonyali, K. Akkaya, M. Cebe, M. Mahmoud, *Efficient Certificate Verification for Vehicle-to-Grid Communications* in Proc. of International Conference on Future Network Systems and Security, Aug. 2017.

C11) S. Tonyali, K. Akkaya, *A Scalable Protocol Stack for IEEE 802.11s-based Advanced Metering Infrastructure Networks*, in Proc. of IEEE Consumer Communications & Networking Conference, Jan. 2018.