Future Generation Computer Systems 78 (2018) 547–557

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Privacy-preserving protocols for secure and reliable data aggregation in IoT-enabled Smart Metering systems

Samet Tonyali [a,*], Kemal Akkaya [a], Nico Saputro [a], A. Selcuk Uluagac [a], Mehrdad Nojoumian [b]

[a] The Department of Electrical and Computer Engineering, Florida International University, Miami, FL, 33174, USA

[b] The Department of Computer & Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL, 33431, USA

ARTICLE INFO

Article history:
Received 1 October 2016
Received in revised form
25 February 2017
Accepted 17 April 2017
Available online 24 April 2017

Keywords:
Internet of things
Privacy-preserving protocols
Reliable data aggregation
Secure multiparty computation
Fully homomorphic encryption
Smart Metering systems

ABSTRACT

As the Internet of Things (IoT) gets more pervasive, its areas of usage expands. Smart Metering systems is such an IoT-enabled technology that enables convenient and high frequency data collection compared to existing metering systems. However, such a frequent data collection puts the consumers' privacy in risk as it helps expose the consumers' daily habits. Secure in-network data aggregation can be used to both preserve consumers' privacy and reduce the packet traffic due to high frequency metering data. The privacy can be provided by performing the aggregation on concealed metering data. Fully homomorphic encryption (FHE) and secure multiparty computation (secure MPC) are the systems that enable performing multiple operations on concealed data. However, both FHE and secure MPC systems have some overhead in terms of data size or message complexity. The overhead is compounded in the IoT-enabled networks such as Smart Grid (SG) Advanced Metering Infrastructure (AMI). In this paper, we propose new protocols to adapt FHE and secure MPC to be deployed in SG AMI networks that are formed using wireless mesh networks. The proposed protocols conceal the smart meters' (SMs) reading data by encrypting it (FHE) or computing its shares on a randomly generated polynomial (secure MPC). The encrypted data/computed shares are aggregated at some certain aggregator SM(s) up to the gateway of the network in a hierarchical manner without revealing the readings' actual value. To assess their performance, we conducted extensive experiments using the ns-3 network simulator. The simulation results indicate that the secure MPC-based protocol can be a viable privacy-preserving data aggregation mechanism since it not only reduces the overhead with respect to FHE but also almost matches the performance of the Paillier cryptosystem when it is used within a proper sized AMI network.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Automatic Meter Reading (AMR) systems collect consumption, diagnostic and status data [1] 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 to better manage the power demand, reduce CO_2 emissions, and ensure reliability [2,3], 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" [4,5]. However, AMR systems are far from providing the required data to implement smart functions such as demand-load matching, demand response, dynamic pricing, etc. [6].

The necessity of such an infrastructure brings the Internet of Things [7] 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

[*] Corresponding author.
E-mail addresses: stonyali@fiu.edu (S. Tonyali), kakkaya@fiu.edu (K. Akkaya), nsapu002@fiu.edu (N. Saputro), auluagac@fiu.edu (A.S. Uluagac), mnojoumian@fau.edu (M. Nojoumian).

http://dx.doi.org/10.1016/j.future.2017.04.031
0167-739X/© 2017 Elsevier B.V. All rights reserved.

on a network infrastructure that connects SMs and the UC, typically via a wireless mesh-based network, referred to as AMI Network in the rest of the paper.

Collection and storage of such fine-grained data, however, raises the issue of privacy for the consumers who have to use the SMs daily [8,9]. Specifically, the collected consumption data can be analyzed using load monitoring techniques to infer activities of the consumers [10]. 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 [11]. 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 [9,12]. 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 [2]. 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 [14–19] suggested using partially homomorphic encryption (PHE) [20], fully homomorphic encryption (FHE) [21] 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 AMI networks. 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 paper 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 IEEE 802.11s-based [23] SG AMI networks.

Our contributions are three-fold. (1) For the adaptation of FHE systems in AMI networks, 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 paper 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 paper, 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 [24] 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 the paper is organized as follows. In the next section, we summarize the related work. In Section 3, we provide some background on PHE, FHE, secure MPC, the network and attack models, and define the problem. Section 4 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. In Section 6, we assess the performance of the proposed approaches. Finally, Section 7 concludes the paper.

2. Related work

This section gives the related work under three subsections. The subsections discuss the related work on data aggregation in SG, TCP modifications for SG, and homomorphic systems, respectively.

2.1. Data aggregation in smart grid

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. [25] used Paillier cryptosystem to provide in-network data aggregation while protecting user consumer privacy. Li and Luo [26] 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 [27] proposed a decentralized security framework for data aggregation and access control in SGs. Consumers' private data are encrypted by using homomorphic encryption. In [28], 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. [29] 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 AMI networks 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.

2.2. *TCP modifications for smart grid*

There are a number of works which investigated the TCP performance for SGs. For instance, the work in [30] 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 [30], Khalifa et al. [31] 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.

2.3. *Homomorphic systems*

2.3.1. *Partially homomorphic encryption*

PHE has attracted most of the researchers' attention studying SG privacy preserving [32]. Among many PHE cryptosystems, Paillier [20] is widely proposed for data aggregation in SG thanks to its addition property, smaller message expansion factor compared to others, and security features [9]. There are many SG privacy preserving aggregation applications based on Paillier [25,33,34]. In [25], the aggregation is performed at each level of a tree topology whereas the other applications perform the aggregation only at the gateway.

Ozgur et al. [35,36] 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 paper 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.

2.3.2. *Fully homomorphic encryption*

Gentry proposed the first FHE system using ideal lattices in 2009 [21]. 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 [37] 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 [38]. Besides these efforts, Gentry and Halevi [39] 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. [40] presented a working implementation of the Smart-Vercauteren scheme [37]. Brakerski et al. [41] 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 [42]. 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.

2.4. *Secure multiparty computation-based protocols*

There have been a few studies using secure MPC-based protocols to perform data aggregation in SG. These protocols can be implemented with different cryptographic schemes in order to make data aggregation private and secure. For instance, Rottondi et al. [17–19] 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. [43,44] 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. [45] 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 AMI applications. A ring does not apply to AMI networks. 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. 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.

3.1. 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 paper, 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 [20] 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.

$$D_{S_K}((E_{P_K}(m_1)\,x\,E_{P_K}(m_2)) \mod n^2) = (m_1 + m_2) \mod n, \qquad (1)$$

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 [40].

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.

3.2. 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.

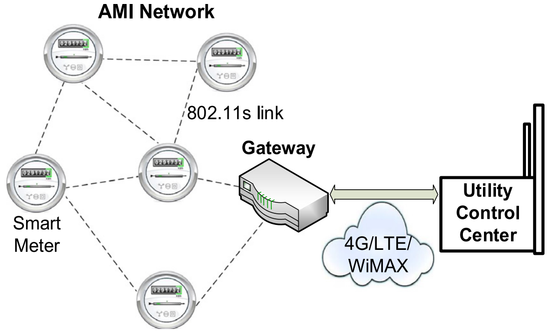

Fig. 1. AMI mesh network of smart meters implemented using IEEE 802.11s.

Shamir's Secret Sharing (SSS) [46] 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 $m \in \{1, \ldots, n\}$. The constant term of $g(x)$ is the aggregation of all individual n private secrets.

3.3. 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 [47–49]. The gateway collects all the SM readings and sends them to the UC using a wide area network connection such as WiMAX or LTE [50]. A sample AMI network based on IEEE 802.11s is given in Fig. 1.

3.4. 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 paper can be defined as follows: "Devise network protocols that will help adapt FHE and secure MPC for deployment in AMI networks. In addition, assess their performance with respect to PHE solutions in a realistic network to understand the overhead of achieving comprehensive privacy".

3.5. 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.

4. FHE scheme for AMI networks

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.

4.1. The complexity of Smart-Vercauteren addition and multiplication operations

As mentioned, we use an implementation of Smart-Vercauteren scheme [40] which is an FHE system. In this work, we extended [40] 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.

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 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 [51] having four 64-bit ARM Cortex-A53 processors at 1.2 GHz with 1 GB RAM using Raspbian OS. The results are given in Tables 1 and 2. As shown in Table 1, multiplication suffers from excessive processing times. Even for two operands, its processing time is more than 10 min. 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 paper, 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.

Table 1
Delay comparison of addition and multiplication.

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

Table 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

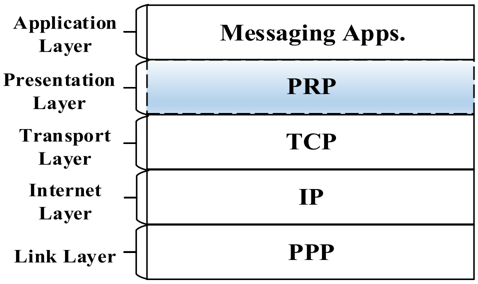

Fig. 2. Placement of the *PRP* in protocol stack.

4.2. 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.

4.2.1. 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*.

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. 2. The

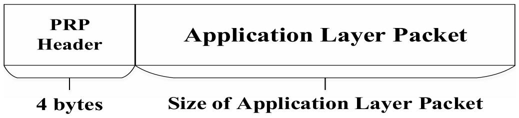

Fig. 3. An illustration of a *PRP* packet.

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*)

```
buffer ← bufferMap.RetrieveBuffer(from)
if buffer == null then
    header ← segment.GetPRPHeader()
    buffer ← CreateBuffer(header.GetPacketSize())
end if
residualBytes ← buffer.Add(segment)
if buffer.IsFull() then
    appPacket ← CreateAppPacket(buffer)
    ReportUpperLayer(appPacket)
    bufferMap.RemoveBuffer(from)
    if residualBytes.Size() $\neq$ 0 then
        resSegment ← CreateSegment(residualBytes)
        Receive(resSegment, from)
    end if
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. 3. 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.

4.2.2. 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 *segments* 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. Adapting secure MPC for AMI networks

As mentioned in Section 3.2, 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 AMI networks. 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 [52]. 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 [53], which is the most commonly used key-exchange protocol can be used to share the secret keys.

We give an overview of the protocol we used in our work in Fig. 4. 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 $j \in \{1, \ldots, n\}/i$. These values are the shares that would be computed by the other meters. Now, we have n points: $\{(0, r_i), (x_1, f_1(x_i)), \ldots, (x_n, f_n(x_i))\}/(x_i, f_i(x_i))$. 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 (2):

$$l_i(x) = \prod_{\substack{j=1 \\ j\neq i}}^{n} \frac{x - x_j}{x_i - x_j}. \tag{2}$$

Hence, the polynomial $F_i(X)$ can be derived as in Formula (3):

$$F_i(X) = \sum_{j=1}^{n} F_i(X_j) \cdot l_j(X). \tag{3}$$

From Formula (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.

5.1. Hierarchical secure MPC in AMI networks

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 meter. The meters compute their total share (F_i in Fig. 4) 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). This process goes on up until

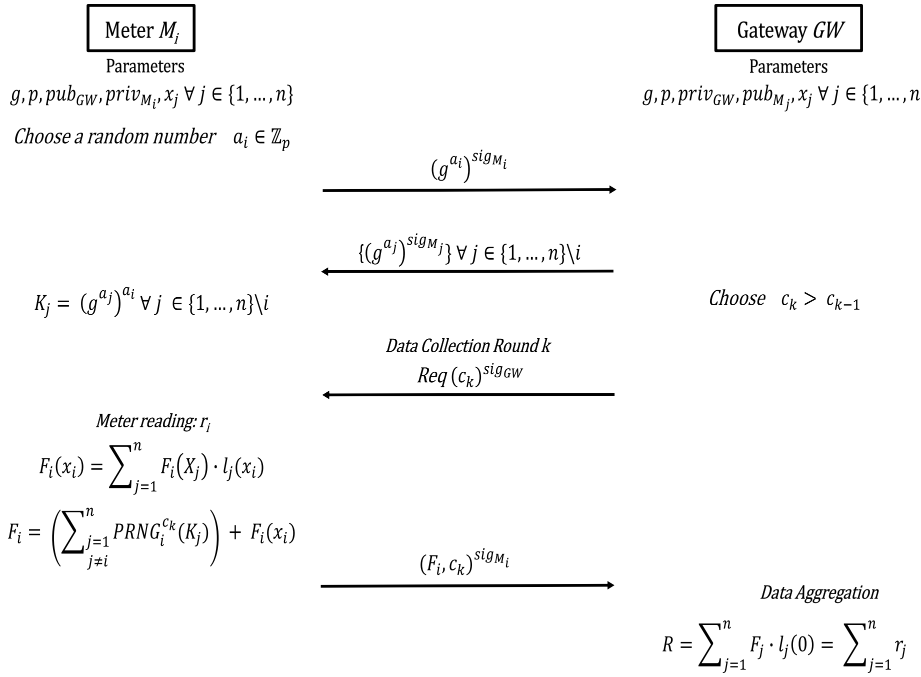
Meter M_i
Parameters
$g, p, pub_{GW}, priv_{M_i}, x_j \; \forall j \in \{1, \ldots, n\}$
Choose a random number $a_i \in \mathbb{Z}_p$
Gateway GW
Parameters
$g, p, priv_{GW}, pub_{M_j}, x_j \; \forall j \in \{1, \ldots, n\}$
$(g^{a_i})^{sig_{M_i}}$
$\{(g^{a_j})^{sig_{M_j}}\} \; \forall j \in \{1, \ldots, n\} \backslash i$
$K_j = (g^{a_j})^{a_i} \; \forall j \in \{1, \ldots, n\} \backslash i$
Choose $c_k > c_{k-1}$
Data Collection Round k
$Req\,(c_k)^{sig_{GW}}$
Meter reading: r_i
$F_i(x_i) = \sum_{j=1}^{n} F_i(X_j) \cdot l_j(x_i)$
$F_i = \left(\sum_{\substack{j=1 \\ j \neq i}}^{n} PRNG_i^{c_k}(K_j)\right) + F_i(x_i)$
$(F_i, c_k)^{sig_{M_i}}$
Data Aggregation
$R = \sum_{j=1}^{n} F_j \cdot l_j(0) = \sum_{j=1}^{n} r_j$

Fig. 4. Overview of the secure MPC-based protocol we used in this work.

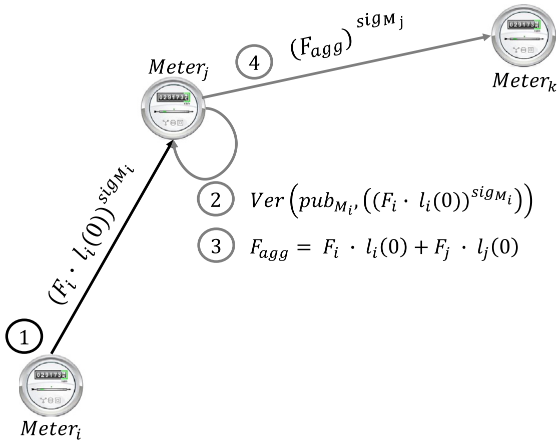
$Meter_j$
4
$(F_{agg})^{sig_{M_j}}$
$Meter_k$
$(F_i \cdot l_i(0))^{sig_{M_i}}$
1
2
$Ver\left(pub_{M_i}, ((F_i \cdot l_i(0))^{sig_{M_i}})\right)$
3
$F_{agg} = F_i \cdot l_i(0) + F_j \cdot l_j(0)$
$Meter_i$

Fig. 5. A simple example for hierarchical secure MPC of a parent meter with one child meter.

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 [54] can be implemented by applying $PRNG(\cdot)$ function twice consecutively followed by a degree reduction [55]. For the sake of a fair comparison with FHE and PHE, we have not implemented and discussed the multiplication operation in this paper.

6. Performance evaluation

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

6.1. Security analysis

In this section, we evaluate our proposed protocols based on the security goals listed in Section 3.5.

***Security Goal* 1**: Let $m_i \forall i \in \{1, 2, \ldots, n\}$ 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

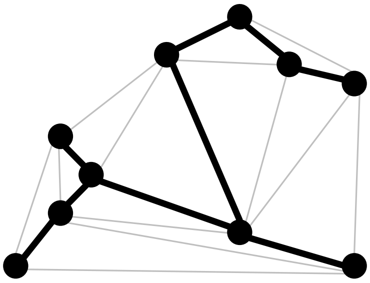

Fig. 6. A minimum spanning tree of a mesh network shown as black links and nodes.

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.

$\{\langle Enc_{PK_{UC}}(m_i), TS\rangle, Sig_{SK_i}(\langle Enc_{PK_{UC}}(m_i), TS\rangle)\}.$

6.2. Experimental setup

We assessed the performance of our protocols using network simulator ns-3 [24], 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 60 s [56] 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 **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 [57].

There are two types of data aggregation mechanisms defined for SG AMI networks [14]. 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 [58] as illustrated in Fig. 6. 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 [59]. 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.

6.3. 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.

6.4. 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.

6.4.1. Packet delivery ratio

As mentioned before, we give the PDR only for EtoE mechanism. As shown in Fig. 7, 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.

6.4.2. 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.

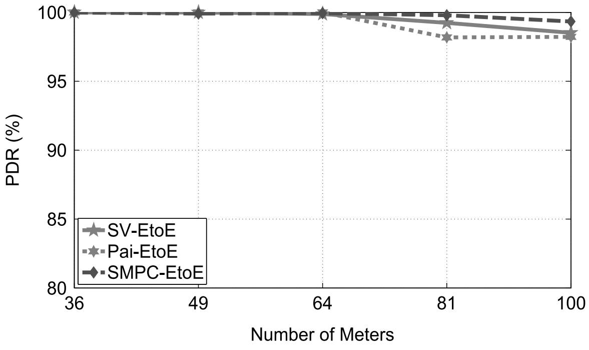

Fig. 7. The EtoE PDR values at different number of nodes.

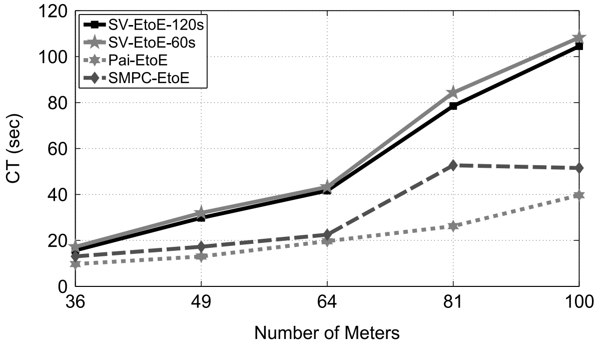

Fig. 10. The EtoE CT values at different number of nodes.

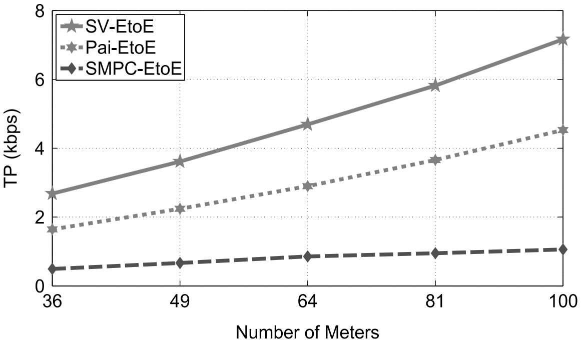

Fig. 8. The EtoE TP values at different number of nodes.

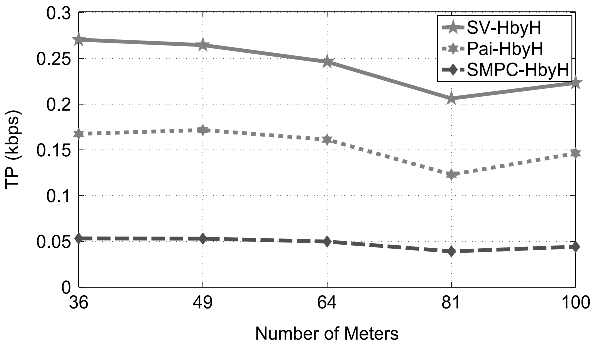

Fig. 9. The HbyH TP values at different number of nodes.

We give throughput figures for both EtoE and HbyH mechanisms in Figs. 8 and 9, 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 8, 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. 9. 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. 8. This order stems from the same reasons mentioned above for the EtoE TP values.

6.4.3. *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 Figs. 10 and 11, 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 60 s for **SV** approach. Therefore, we ran another simulation in which meter readings are collected at every 120 s 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 120 s, which makes **SV** approach feasible. Both in EtoE and HbyH, 60 s and 120 s approaches show a very similar tendency because the meters experience the same delay since they try to send their readings at the same time.

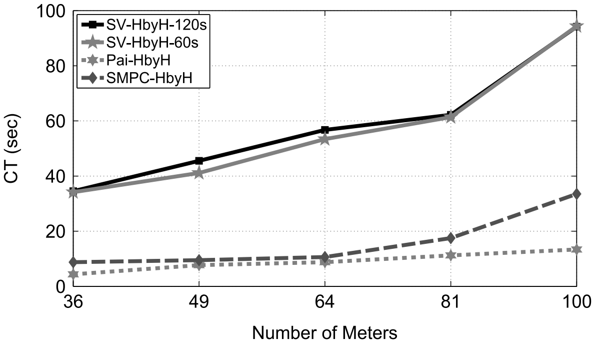

Fig. 11. The HbyH CT values at different number of nodes.

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.

7. Conclusion

In this paper, we tackled the problem of reliable and privacy-preserving in-network data aggregation in IEEE 802.11s-based SG AMI networks. 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 consumers' 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.

Acknowledgment

This work is supported by US National Science Foundation under the grant number CNS-1550313.

References

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

[2] N. Saputro, K. Akkaya, S. Uludag, A survey of routing protocols for smart grid communications, Comput. Netw. 56 (11) (2012) 2742–2771.

[3] W. Wang, Y. Xu, M. Khanna, A survey on the communication architectures in smart grid, Comput. Netw. 55 (15) (2011) 3604–3629.

[4] M.H. Cintuglu, H. Martin, O.A. Mohammed, Real-time implementation of multiagent-based game theory reverse auction model for microgrid market operation, IEEE Trans. Smart Grid 6 (2) (2015) 1064–1072.

[5] M.H. Cintuglu, T. Youssef, O.A. Mohammed, Development and application of a real-time testbed for multiagent system interoperability: A case study on hierarchical microgrid control, IEEE Trans. Smart Grid (2011).

[6] Smart grid functions. URL https://www.smartgrid.gov/files/definition_of_functions.pdf.

[7] L. Atzori, A. Iera, G. Morabito, The Internet of things: A survey, Comput. Netw. 54 (15) (2010) 2787–2805.

[8] J. Gao, Y. Xiao, J. Liu, W. Liang, C. Chen, A survey of communication/networking in smart grids, Future Gener. Comput. Syst. 28 (2) (2012) 391–404.

[9] N. Saputro, K. Akkaya, On preserving user privacy in smart grid advanced metering infrastructure applications, Secur. Commun. Netw. 7 (1) (2014) 206–220.

[10] G.W. Hart, Nonintrusive appliance load monitoring, Proc. IEEE 80 (12) (1992) 1870–1891.

[11] X. Fang, S. Misra, G. Xue, D. Yang, Managing smart grid information in the cloud: opportunities, model, and applications, IEEE Netw. 26 (4) (2012) 32–38.

[12] 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, IEEE Internet Things J. 3 (5) (2016) 709–719.

[13] Stop smart meters. URL http://stopsmartmeters.org.

[14] N. Saputro, 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.

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

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

[18] C. Rottondi, G. Verticale, C. Krauss, Distributed privacy-preserving aggregation of metering data in smart grids, IEEE J. Sel. Areas Commun. 31 (7) (2013) 1342–1354.

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

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

[21] C. Gentry, A fully homomorphic encryption scheme (Ph.D. thesis), Stanford University, 2009.

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

[23] G.R. Hiertz, D. Denteneer, S. Max, R. Taori, J. Cardona, L. Berlemann, B. Walke, Ieee 802.11 s: the wlan mesh standard, IEEE Wirel. Commun. 17 (1) (2010) 104–111.

[24] ns 3, ns-3: network simulator 3, Release 3.24.1, 2016. URL http://www.nsnam.org/.

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

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

[27] S. Ruj, A. Nayak, A decentralized security framework for data aggregation and access control in smart grids, IEEE Trans. Smart Grid 4 (1) (2013) 196–205.

[28] Z. Lu, Y. Wen, Distributed algorithm for tree-structured data aggregation service placement in smart grid, IEEE Syst. J. 8 (2) (2014) 553–561.

[29] M. Ambrosin, H. Hosseini, K. Mandal, M. Conti, 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.

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

[31] T. Khalifa, A. Abdrabou, K. Naik, M. Alsabaan, A. Nayak, N. Goel, Split-and aggregated-transmission control protocol (sa-tcp) for smart power grid, IEEE Trans. Smart Grid 5 (1) (2014) 381–391.

[32] C. Fontaine, F. Galand, A survey of homomorphic encryption for nonspecialists, EURASIP J. Inf. Secur. 2007 (1) (2007) 1–10.

[33] R. Lu, X. Liang, X. Li, X. Lin, X. Shen, Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications, IEEE Trans. Parallel Distrib. Syst. 23 (9) (2012) 1621–1631.

[34] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, X. Shen, Eppdr: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid, IEEE Trans. Parallel Distrib. Syst. 25 (8) (2014) 2053–2064.

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

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

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

[38] D. Stehlé, R. Steinfeld, Faster fully homomorphic encryption, in: Advances in Cryptology, Springer, 2010, pp. 377–394.

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

[40] H. Perl, M. Brenner, 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.

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

[42] Y. Zhang, J.-L. Chen, Wide-area scada system with distributed security framework, J. Commun. Netw. 14 (6) (2012) 597–605.

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

[44] C. Thoma, T. Cui, F. Franchetti, Privacy preserving smart metering system based retail level electricity market, 2013.

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

[46] A. Shamir, How to share a secret, Commun. ACM 22 (11) (1979) 612–613.

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

[48] Cyan technology ami solutions - smart electricity metering. URL http://www.cyantechnology.com/smart-electricity-metering/.

[49] Trilliant ami solutions - smart metering. URL http://trilliantinc.com/solutions/metering.

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

[51] S. Monk, Raspberry Pi Cookbook: Software and Hardware Problems and Solutions, O'Reilly Media, Inc., 2016.

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

[53] W. Diffie, M.E. Hellman, New directions in cryptography, IEEE Trans. Inform. Theory 22 (6) (1976) 644–654.

[54] R. Gennaro, M.O. Rabin, T. Rabin, Simplified vss and fast-track multiparty computations with applications to threshold cryptography, in: Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing, ACM, 1998, pp. 101–111.

[55] M. Ben-Or, S. Goldwasser, 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.

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

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

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

[59] G. Locke, P. Gallagher, Fips pub 186-3: Digital signature standard (dss), Fed. Inf. Process. Stand. Publ. 3 (2009) 26–30.

Samet Tonyali is currently a Graduate Research Assistant in the Department of Electrical & Computer Engineering, Florida International University, USA and pursuing his Ph.D. degree in the same department. He received the B.S. degree and the M.S. degree in Computer Engineering from Marmara University, Istanbul, TURKEY in 2011 and 2013, respectively.

Kemal Akkaya is an associate professor in the Department of Electrical and Computer Engineering at Florida International University. He received his Ph.D. in Computer Science from University of Maryland Baltimore County in 2005 and joined the department of Computer Science at Southern Illinois University (SIU) as an assistant professor. Dr. Akkaya was an associate professor at SIU from 2011 to 2014. He was also a visiting professor at The George Washington University in Fall 2013. His current research interests include security and privacy, energy aware routing, topology control, and quality of service issues in a variety of wireless networks such as sensor networks, multimedia sensor networks, smart-grid communication networks and vehicular networks. Dr. Akkaya is a senior member of IEEE. He is the area editor of Elsevier Ad Hoc Network Journal and serves on the editorial board of IEEE Communication Surveys and Tutorials. He has served as the guest editor for Journal of High Speed Networks, Computer Communications Journal, Elsevier Ad Hoc Networks Journal and in the TPC of many leading wireless networking conferences including IEEE ICC, Globecom, LCN and WCNC. He has published over 90 papers in peer reviewed journal and conferences. He has received "Top Cited" article award from Elsevier in 2010.

Nico Saputro received the Ph.D. degree from Florida International University in 2016. He is a senior lecturer at the Department of Informatics at Parahyangan Catholic University, Bandung, Indonesia. Currently, he is a visiting PostDoctoral Associate at the Department of Electrical and Computer Engineering, Florida International University Miami, FL. His research interests include security and privacy, communication protocols for Smart Grid, and internet of things. He is a member of IEEE.

A. Selcuk Uluagac is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Florida International University, where he leads the Cyber–Physical Systems Security Lab (CSL). He has served as a member of the research faculty as a Senior Research Engineer in the School of Electrical and Computer Engineering at The Georgia Institute of Technology and prior to Georgia Tech, he was a Senior Research Engineer at Symantec.

Mehrdad Nojoumian is currently an Assistant Professor in the Department of Computer & Electrical Engineering at Florida Atlantic University. My research interests lie on applied cryptography, security and privacy, trust management, game theory and interdisciplinary research on the intersection of computer and social sciences such as economics and psychology.