A Reliable Data Aggregation Mechanism with Homomorphic Encryption in Smart Grid AMI Networks

Samet Tonyali, Kemal Akkaya, Nico Saputro, and A. Selcuk Uluagac

Department of Electrical & Computer Engineering, Florida International University, Miami, FL 33174 USA
Email: {stony002, kakkaya, nsapu002, suluagac}@fiu.edu

***Abstract*—One of the most common methods to preserve consumers' private data is using secure in-network data aggregation. The security can be provided through the emerging fully (FHE) or partial (PHE) homomorphic encryption techniques. However, an FHE aggregation scheme generates significantly big-size data when compared to traditional encryption methods. The overhead is compounded in hierarchical networks such as Smart Grid Advanced Metering Infrastructure (AMI) as data packets are routed towards the core of the AMI networking infrastructure from the smart meters. In this paper, we first investigate the feasibility and performance of FHE aggregation in AMI networks utilizing the reliable data transport protocol, TCP. Then, we introduce the packet reassembly problem. To address this challenge, we propose a novel packet reassembly mechanism for TCP. We evaluated the effectiveness of our proposed mechanism using both PHE and FHE-based aggregation approaches in AMI in terms throughput and end-to-end delay on an 802.11s-based wireless mesh network by using the ns-3 network simulator. The results indicate significant gains in terms of delay and bandwidth usage with the proposed mechanism.**

I. Introduction

Advanced Metering Infrastructure (AMI) applications are used to collect fine-grained power consumption data via smart meters in Smart Grid [1]. With the AMI networks, the utility companies are able to monitor power demands over even short periods to provide both more accurate billing and dynamic pricing efficiently utilizing the decrease in peak demand [2].

While AMI networks enable collection of data at frequent intervals, such a fine-grained level poses several privacy concerns as discussed extensively in the literature [2], [3]. Specifically, the data can be analyzed using load monitoring techniques to capture the behavior of the consumers. Due to such threats, the research community proposed several solutions to address the issue [3].

One of the widely used techniques that hides individual meter readings is data aggregation [4]. The idea is to perform the aggregation in the network when the data are transmitted towards the gateway. Each intermediate smart meter performs such an aggregation as they have access to actual data. However, this exposes the data of a particular smart meter to another intermediate meter. To solve this problem, several studies suggested using of the nascent Partially Homomorphic Encryption (PHE) or Fully Homomorphic Encryption (FHE) that have the ability to perform certain arithmetic operations on the encrypted data in a privacy-preserving fashion [4]. In this way, intermediate meters can perform aggregation without having access to the actual data. Indeed, although FHE systems [5] are becoming more popular since they allow any type of operations on the data, giving flexibility to the applications to perform different computations for their needs [6], FHE systems suffer from generating large size ciphertexts in addition to longer computational times to encrypt and decrypt data [7]. In particular, the size of the ciphertexts becomes a problem when aggregation is performed within a network with hierarchical transfer patterns. Smart meter data collection in AMI is such an example where there is only one gateway collecting all the meter data. In such a case, when FHE is used for performing in-network data aggregation, the packet sizes would grow dramatically when packet reaches close to the gateway. Therefore, there is a need to assess the feasibility of FHE aggregation under different conditions.

However, this is not the only problem regarding aggregation. Using aggregation creates another problem when the packets are to be transmitted via TCP that is crucial for reliable services. The problem is due to dynamic window size applied by TCP. Specifically, when TCP employs a dynamic window size, which fluctuates based on network conditions, the sent segments will also vary in size. This poses a packet reassembly problem at the receiver side because the receiving meter needs to reassemble packets from each child meter to perform data aggregation. As such, the receiving end does not know how many bytes each packet consists of. This is because the size of data packets to be received can vary from meter to meter due to different number of child meters. Note that this problem is specific to TCP and does not occur with User Datagram Protocol (UDP) since UDP does not employ a streaming mechanism for packet transfer. While there are some works which use UDP for secure data aggregation in different applications [8], we do not prefer UDP since it does not provide reliability guarantees when meter data is transmitted, which is crucial for Smart Grid applications.

The solution to this problem necessitates additional information that will let the receiving side know the actual packet sizes from each child meter. To this end, in this paper we propose adding a presentation layer above the transport layer in order to include packet size information at the sender side.

978-1-4673-9292-1/16/$31.00 ©2016 IEEE
Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on May 07,2023 at 19:02:20 UTC from IEEE Xplore. Restrictions apply.

We implemented the proposed approach in an IEEE 802.11s-based wireless mesh AMI network consisting of smart meters and a data collector gateway. IEEE 802.11s [9] is the standard for creating a multi-hop mesh among smart meters using IEEE 802.11 as the MAC layer. We used the widely used ns-3 [10] network simulator which has a built-in implementation of IEEE 802.11s standard. The simulation results indicated that the proposed approach enables the realization of FHE-based data aggregation using TCP with improved performance in terms of data delay and used bandwidth.

This paper is organized as follows. In Section II, we summarize the related work. Section III provides the preliminaries about the topics. Section IV investigates the feasibility of FHE aggregation. We presents the details of the proposed approach in Section V. In Section VI, we assess the performance of the proposed approach. Finally, Section VII concludes the paper.

II. Related Work

A. Data Aggregation in Smart Grid

Data aggregation is utilized in Smart Grids to reduce packet traffic in the network and eventually minimize the number of dropped packets. Data from different sources are collected at aggregator meters and the collected data is forwarded to the utility server. In most cases, aggregator meters perform some arithmetic operations on the collected data before they are transmitted to the utility server. In order to preserve consumer privacy, several works made use of homomorphic encryption and homomorphic arithmetic operations.

For instance, Li et al. [11] used Paillier cryptosystem to provide in-network data aggregation while protecting user consumer privacy. Li and Luo [12] 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 [13] proposed a decentralized security framework for data aggregation and access control in Smart Grids. Customers' private data are concealed by using homomorphic encryption. In [14], the authors focused on finding the optimal placement for the data aggregation service, which minimizes the cost of in-network processing.

While these approaches considered different aspects of data aggregation, none of them provided a reliable aggregation utilizing a TCP-based network, where in-network operations are performed. Therefore, the reliable aggregation with privacy-preserving encryption problem was not studied before. However, our approach would be complimentary to these approaches as it will allow others to work under TCP especially if the data sizes are larger.

B. TCP Modifications for Smart Grid

There are a number of works which investigated the TCP performance for SGs. For instance, the work in [15] proposes a scalable protocol that can handle both security and reliability using a TCP-friendly congestion control scheme. While the implementations are done in ns-2 simulator, there is no focus of data aggregation. Due to similar motivations of the work in [15], Khalifa et al. [16] 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 performed at intermediate nodes. Moreover, the work does not provide privacy-preserving encryption features on data. The goal is to reduce the used bandwidth. Our approach is still needed if the scheme would perform in-network data aggregation in a privacy-preserving fashion.

C. Homomorphic Systems

PHE has attracted attention of most of the researchers studying on SG privacy preserving [17]. Among many PHE cryptosystems, Paillier [18] is widely proposed for SG thanks to its addition property, smallest message expansion factor compared to others, and security features [3]. There are many SG privacy preserving aggregation applications based on Paillier [19]–[21]. In [19], the aggregation is performed at each level of a tree topology whereas the other applications perform the aggregation only at the gateway.

FHE research on the other hand has started to accelerate after its proposal by Gentry in 2009 [5]. Eventually, working implementations in the form of *somewhat homomorphic* such as the one by Perl et al. [6], which is an implementation of the Smart-Vercauteren scheme has been available to researchers. These actual implementations paved the way for different applications. Yet, the studies that targeted SG applications are still limited. For instance, [22] utilized a somewhat FHE for wide-area supervisory control and data acquisition (SCADA) security. [7] used FHE implementation for assessing its performance in a wireless mesh based AMI network. Our work in this paper will help accelerate these SG applications that potentially use TCP for their communications.

III. Preliminaries

A. Network Model

We assume an AMI network which consists of smart meters and a gateway that can communicate with a utility company. The smart meters send periodic power consumption data to a local gateway using IEEE 802.11s mesh standard through wireless communication. The gateway collects all the smart meter data and sends them to the utility company using a wide are network connection such as LTE. A sample AMI network based on IEEE 802.11s is given in Fig. 1.

Each smart meter acts as a parent node for multiple smart meters to forward their data to the upstream nodes that are determined by the routing protocol of IEEE 802.11s standard. We assume that smart meter waits to collect readings from all the child smart meters and then aggregates them (i.e., adds) to create a single packet. This aggregated data is passed to the upstream neighbor. Due to reliability concerns, TCP is used at the transport layer.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on May 07,2023 at 19:02:20 UTC from IEEE Xplore. Restrictions apply.

B. Overview of Homomorphic Encryption Systems

Homomorphic property is a property that enables a cryptosystem to perform a set of operations on the ciphertext 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.

$$D_{S_K}(E_{P_K}(m_1) \square E_{P_K}(m_2)) = m_1 \triangle m_2 \quad (1)$$

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

In this paper, we make use of two types of homomorphic encryption cryptosystem: partially homomorphic encryption and fully homomorphic encryption. A partially homomorphic cryptosystem is able to perform either addition or multiplication while the FHE enables one to perform both operations on the ciphertext.

C. Partially Homomorphic - Paillier Cryptosystem

Paillier cryptosystem [18] 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.

$$D_{S_K}(E_{P_K}(m_1) \cdot E_{P_K}(m_2)) = m_1 + m_2 \quad (2)$$

D. Fully Homomorphic - Smart-Vercauteren Scheme

Smart-Vercauteren (SV) scheme consists of key generation, encryption, decryption, addition/multiplication and recryption functions [6].

As more operations are performed on a ciphertext noise is accumulated in the ciphertext. The recryption function removes noise in the ciphertext without decrypting it and the cleartext is kept unchanged. The function utilizes a 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.

SV scheme is a member of public-key cryptography family, so it generates a public-secret (private) key pair. 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.

The keys are generated considering three-parameter tuple, which is called "key geometry": The number of bits ($|B|$), the number of *shares* (S_1) and the number of *cells* (S_2) [6].

Algorithm 1 $FHE\_Multiply(encrOp1, encrOp2, pk)$

```
for all encrBit2 ∈ encrOp2 do
    for all encrBit1 ∈ encrOp1 do
        tempOp ← fhe_mul(encrBit1, encrBit2, pk)
    end for
    result ← FHE_Add(result, tempOperand, pk)
    for all encrBit ∈ result do
        fhe_recrypt(encrBit, pk)
    end for
end for
```

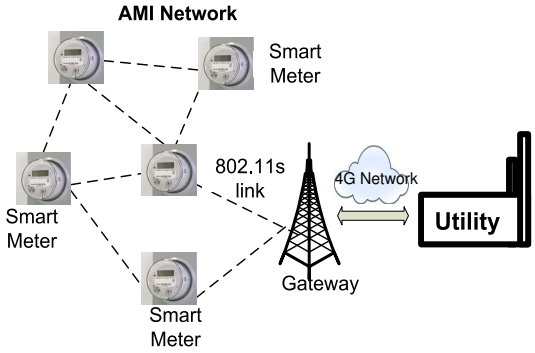

Fig. 1: Hierarchical AMI network of smart meters implemented using IEEE 802.11s.

IV. The Feasibility of Fully Homomorphic Addition and Multiplication Operations

Secure in-network data aggregation is carried out by using either homomorphic addition or homomorphic multiplication operation. While partially homomorphic systems have been widely implemented and tested, fully homomorphic systems have not been used for aggregation as mentioned, SV scheme is both additive and multiplicative. In [6], both addition and multiplication operations are defined on a single bit. In this paper, we extended them so that the operations can be performed on multiple-bit numbers without losing the ability to perform decryption. Due to space constraints we give a pseudocode of only multiplication operation in Algorithm 1.

We assessed the feasibility of SV homomorphic addition and multiplication operations for 16-bit operands. We performed sequential homomorphic operations on encrypted data and assessed the time and storage complexity. The tests are performed on a machine with Intel Xeon CPU E5 at 2.3 GHz. with 32 GB RAM using Ubuntu 14.04 OS. The results are given in Table I and II. As shown in Table I, multiplication suffers from excessive processing times. Even for two operands, its processing time is more than one minute. For the generated data size, we observed that addition generates much 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, SV fully homomorphic multiplication processing times in the order of minutes which may not be applicable to smart meter data collection. Thus, for the rest of the paper, we focus on SV fully homomorphic addition. We analyze its performance when used in an AMI network under TCP.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on May 07,2023 at 19:02:20 UTC from IEEE Xplore. Restrictions apply.

TABLE I: Delay comparison of addition and multiplication

# of operands	*Delay (sec)*	
	Addition	*Multiplication*
2	0.49	63.547
3	0.835	160.598
4	1.336	290.88
5	2.342	455.693

TABLE II: Data size comparison of addition and multiplication

# of operands	*Data Size (bits)*	
	Addition	*Multiplication*
2	52237	101556
3	55348	153626
4	58353	206014
5	61486	258403

V. 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.

A. The Packet Reassembly Problem under TCP

Secure data aggregation can be transmitted with either UDP or TCP. Given the critical nature of the smart meter data, one should use TCP in order to ensure reliability. Nonetheless, when data aggregation is implemented with TCP, 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 its child meters may send different size packets due to aggregation performed. We call this problem the packet reassembly problem.

In order to overcome this problem, we propose a new protocol which tells the receiver meter the size of the packet it receives. We develop this new protocol above the TCP layer, referred to as presentation layer as shown in Fig. 2. The proposed *Packet Reassembly Protocol (PRP)* enables an aggregator meter to reassemble a packet from its segments. The protocol adds an 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.

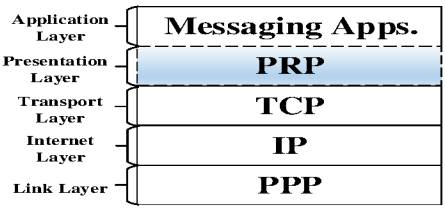

Fig. 2: Placement of the PRP in protocol stack

Fig. 3: An illustration of a PRP packet

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 4 bytes and it includes the size of the application layer packet. Even if a packet is exposed to TCP segmentation, the first segment is received first by the receiver meter because the TCP guarantees ordered delivery of a stream of bytes. Thus, a meter will be able to know how many bytes the packet actually consists of by using the header information in this first segment it received.

B. 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 any packet in the receive buffer. We provide a pseudocode for only *Receive* function in Algorithm 2 because Send function is straightforward.

Algorithm 2 $Receive(segment, from)$

```
buffer ← bufferList.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)
    bufferList.RemoveBuffer(from)
    if residualBytes ≠ 0 then
        resSegment ← CreateSegment(residualBytes)
        Receive(resSegment, from)
    end if
end if
```

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on May 07,2023 at 19:02:20 UTC from IEEE Xplore. Restrictions apply.

VI. Performance Evaluation

A. Experimental Setup

We assessed the performance of our new protocol using network simulator ns-3 [10], which has an implementation of IEEE 802.11s mesh networking. We created random multi-hop network topologies of size **N**, where **N** $\in$ (25, 36, 49, 64, 81). For each topology, a mesh node acts as the gateway/data collector and (**N**-1) mesh nodes act as smart meters that send their reports to the gateway periodically at every 60s [23]. The data size generated at the smart meters is assumed to be 16 bits, large enough to hold the power readings. 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.

B. Baselines and Performance Metrics

For performance evaluation, we used the following metrics:

- *Throughput*: The total amount of data received at the transport layer by the gateway per second.
- *Average Completion Time*: The average elapsed time for crypto operations and receiving all the power readings from all smart meters at the gateway in one round.

We tested the performance of our new protocol on both FHE and PHE systems. For the FHE, we used the implementation of [7] and for PHE we used Paillier cryptosystem [18]. SV uses the key geometry of (384/8/5), Paillier uses 64 bit keys.

To compare with our approach, we implemented another approach where all the aggregations are performed at the gateway. This means there is no in-network aggregation, but all the received packets will be aggregated using PHE or FHE at the gateway. This is simply referred to as *forwarding* where the meters encrypt their readings and send them to the gateway. In the figures, the approaches are represented as *PHE with Forwarding (PHE-F)* and *FHE with Forwarding (FHE-F)*.

For our in-network aggregation, we used a minimum spanning tree (MST) algorithm rooted at the gateway to form the aggregation network tree of a random network topology and configured the parent-child relations of each smart meter based on this aggregation network tree. Each meter sends its encrypted power reading to its parent. The parent meter aggregates all the readings of its child meter(s), and then sends the aggregated result to the parent of this parent meter. This process repeats at each aggregator meter until the gateway aggregates the aggregated result of its child meter(s). In the figures, the approaches are represented as *PHE with Aggregation (PHE-A)* and *FHE with Aggregation (FHE-A)*.

C. Performance Results

1) Throughput: First, we investigated the throughput performance to analyze the bandwidth usage of the approaches. The goal is to use less channel bandwidth to accommodate other types of traffic.

The throughput, as shown in Fig. 4, increases for all approaches with the increased network size. This is due to the increase in the number of packets received by the gateway as the number of meters in the network increases. Forwarding approaches generate more throughput as expected because aggregation process decreases the total amount of data sent to the gateway.

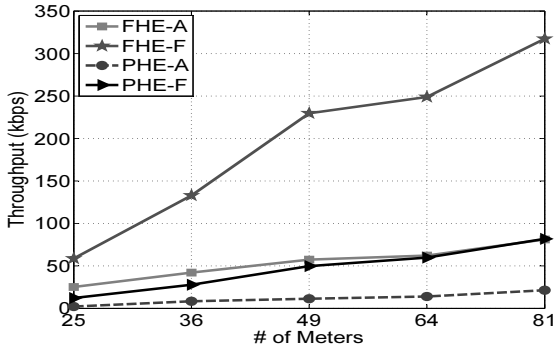

Fig. 4: Throughput for various number of nodes

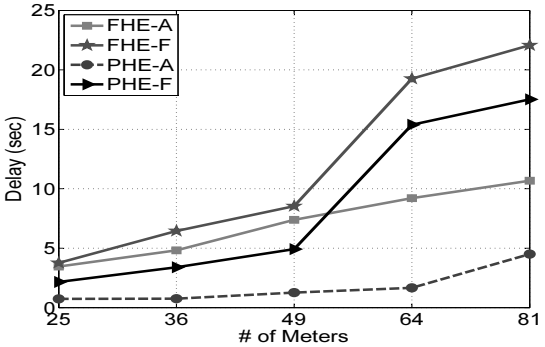

Fig. 5: Avg. Completion Time for various number of nodes

The bandwidth gain in the case of FHE-A is very significant (i.e., around three times reduction) compared to FHE-F which indicates the advantage of aggregation approach for FHE systems. In particular, when the network scales, the performance gap increases further. This is due to the fact that FHE generates very large-size packets that are accumulated towards the gateway if no in-network aggregation is performed. In particular, towards the meters that are close to the gateway, there will be a lot of traffic (funneling effect) due to increased data size and accordingly the number of TCP segments. FHE significantly decreases this effect by performing aggregation.

2) Completion Time: The most interesting metric we investigated was the average completion time of readings from all meters in the network. This is because the proposed approach performs computationally costly data aggregation and also introduces another layer that may induce additional delay.

As seen in Fig. 5, the time required for data collection increases as the number of meters in the network increases since the packet traffic in the network increases. This also causes an increase in carrier sense waitings for the nodes at the data link layer.

From the same figure, we observed that our approach does not bring any additional delay to the completion time despite the delay introduced by FHE or PHE aggregation.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on May 07,2023 at 19:02:20 UTC from IEEE Xplore. Restrictions apply.

On the contrary, it significantly reduces the end-to-end delay for both FHE and PHE about 30% and 40% respectively since it reduces the number of TCP segments to be transmitted. Specifically, channel contention is minimized among the meters, which reduces end-to-end data delay and thus the completion time. In the case of FHE-F and PHE-F, performing all the aggregations at the gateway becomes a major bottleneck in terms of processing delay and thus the completion time significantly increases.

There are other interesting observations. For instance, we observe that PHE-A benefits from the delay reduction much more than that of FHE-A. We speculate that this is due to increased computation time for PHE data aggregation. In case of FHE, only addition is performed and thus the processing time increases linearly with the increased operand count. However, this is not the case in PHE since the operation performed for aggregation is multiplication. The processing time increases much more with increased operand count in PHE-F and thus eliminating this processing time at the gateway by using PHE-A helps reducing the completion time significantly.

Furthermore, it can also be seen a steep increase from 49-Node topologies to 64-Node topologies. This indicates that there is a threshold network size that significantly affects the delay in the network due to increased interference and contention.

Finally, the results show that the the whole data collection is completed in a time less than the half of the frequency of data collection (e.g., 60 sec) which makes it feasible to be used in practical AMI applications.

VII. Conclusion

In this paper, we investigated the use of FHE for reliable data aggregation in Smart Grid AMI networks. We first assessed its overhead in implementing data aggregation in an AMI network using both addition and multiplication operations. We then investigated a new problem called packet reassembly that occurs during the use of TCP for secure data aggregation in Smart Grid AMI applications due to varying TCP segment sizes. Then we addressed this problem by proposing a new presentation layer that adds a header including the packet size to application layer packets before they are passed to the transport layer.

We evaluated the proposed scheme for FHE and PHE using ns-3 simulator. The results showed that the scheme not only enables realization of TCP but also uses bandwidth efficiently compared to baseline approaches. In average, we found that the proposed aggregation approaches reduces data completion time around 30% for FHE and 40% for PHE.

VIII. Acknowledgement

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

References

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

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

[3] 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.

[4] 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

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

[6] H. Perl, M. Brenner, and M. Smith, "Poster: An implementation of the fully homomorphic smart-vercauteren crypto-system," in *18th ACM Conference on Computer and communications security (ACM CCS)*, October 2011, pp. 837 – 840.

[7] S. Tonyali, N. Saputro, and K. Akkaya, "Assessing the feasibility of fully homomorphic encryption for smart grid ami networks," in *The Seventh International Conference on Ubiquitous and Future Networks 2015 - ICUFN2015*, 2015.

[8] A. Kothari, M. Panchal, and R. Jain, "Hybrid spread spectrum based smart meter network using fast frequency hopping," *International Journal of Computer Applications*, vol. 100, no. 2, pp. 39–42, 2014.

[9] 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.

[10] "Network simulator - ns - 3," *http://www.isi.edu/nsnam/ns/index.html*.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] C. Fontaine and F. Galand, "A survey of homomorphic encryption for nonspecialists," *EURASIP J. Inf. Secur.*, vol. 2007, pp. 15:1–15:15, Jan. 2007. [Online]. Available: http://dx.doi.org/10.1155/2007/13801

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the 17th international conference on Theory and application of cryptographic techniques*, ser. EUROCRYPT'99, 1999, pp. 223–238.

[19] 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*, oct. 2010, pp. 327 –332.

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

[21] 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, Aug 2014.

[22] Y. Zhang, J.-L. Chen *et al.*, "Wide-area scada system with distributed security framework," 2012.

[23] 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.

Authorized licensed use limited to: FLORIDA INTERNATIONAL UNIVERSITY. Downloaded on May 07,2023 at 19:02:20 UTC from IEEE Xplore. Restrictions apply.