Performance Evaluation of Smart Grid Data Aggregation via Homomorphic Encryption

Nico Saputro and Kemal Akkaya
Department of Computer Science
Southern Illinois University Carbondale
Carbondale, Illinois - 62901
Email: nico@siu.edu—kemal@cs.siu.edu

***Abstract*—Homomorphic encryption allows arithmetic operations to be performed on ciphertext and gives the same result as if the same arithmetic operation is done on the plaintext. Homomorphic encryption has been touted as one of the promising methods to be employed in Smart Grid (SG) to provide data privacy which is one of the main security concerns in SG. In addition to data privacy, real-time data flow is crucial in SG to provide on-time detection and recovery of possible failures. In this paper, we investigate the overhead of using homomorphic encryption in SG in terms of bandwidth and end-to-end data delay when providing data privacy. Specifically, we compare the latency and data size of end-to-end (ETE) and hop-by-hop (HBH) homomorphic encryption within a network of Smart Meters (SMs). In HBH encryption, at each intermediate node, the received encrypted data from downstream nodes are decrypted first before the aggregation, and then the result is encrypted again for transmission to upstream nodes. On the other hand, the intermediate node in ETE encryption only performs aggregation on ciphertexts for transmission to upstream nodes. We implemented secure data aggregation using Paillier cryptosystem and tested it under various conditions. The experiment results have shown that even though HBH homomorphic encryption has additional computational overhead at intermediate nodes, surprisingly it provides comparable latency and fixed data size passing through the network compared to ETE homomorphic encryption.**

I. Introduction

The modernization of the electricity grid to Smart Grid (SG) that can accommodate future demand growth is on the way. Various digital equipments and communication technologies are integrated to the grid to turn the current one-way flows of electricity into two-way flows in a reliable, efficient, and secure ways [1][2]. Two driving forces to move toward SG are (1) the aging, inadequate, and outdated current electricity grid which need to be improved to meet the future demand challenges; and (2) the benefits of SG in consequence of the improvements in six key value areas: reliability, economics, efficiency, environmental, security, and safety [2]. Hence, generally we will see a reduction of the rate and length of outages as well as the number of disruption due to power quality issues; cheaper electricity bills; better asset utilization and lower operation and maintenance costs; a more environment friendly power grid due to the deployment of electric vehicles and renewable resources; an increase in physical security as well as cyber-security in the whole power grid systems; and a better protection against electricity hazards.

However, due to the complexity of interrelated components on the grid from electricity generation to electricity distribution level and also current enhancement and maturity in each level, the transformations arise at a different pace. The distribution grid is the most obvious one. Tremendous changes, from manual meter reading to one-way communication Automated Meter Reading (AMR), and then two-way communication Advanced Metering Infrastructure (AMI) occur at the distribution grid in recent years. Besides its basic function in data collection for billing purposes, AMI enables many new value-add applications. Via two-way communications, utility companies may collect, measure, and analyze energy consumption data (e.g., grid management, outage notification, and matching production to demand) while consumers can retrieve their historical electricity usage and receive dynamic pricing information.

In contrast to manual data collection for billing purposes that is feasible for monthly meter reading, a much higher frequency (seconds/minutes/hours/daily) of data collection can be provided by AMI for dynamic pricing of power consumption and a variety of applications in addition to billing purposes. Some applications such as electricity demand forecasting, data mining or power generation require statistics of the time series power consumptions at certain area or period of time, rather than an individual consumer's power consumption. Since this may create a lot of traffic and put burden on the underlying communication networks, data aggregation is to be exploited whenever possible to reduce the traffic and data size. This data aggregation can be based on functions such as *sum*, *count*, *average*, etc. which can be computed in the network when the data are traveling. For instance an intermediate meter can collect power data from its neighbors, compute the sum, and transmit only that sum rather than forwarding all the readings from its neighbors.

While data aggregation reduces traffic for SG, there are other concerns in regards to data aggregation. First of all, the privacy of the data should be provided when the data is traveled through the network. This is needed for the home users who do not want to expose their power usage to others. While data can be encrypted during its transmission, data aggregation still requires to access the decrypted data to process it. This will obviously violate the privacy. To overcome this issue, recently homomorphic encryption has been employed

978-1-4673-0437-5/12/$31.00 ©2012 IEEE

[3] [4]. The idea is based on data processing on the encrypted data rather than the plaintext. In this way, an intermediate node will not be able to access the content of the data.

In this paper, we investigate the performance impact of the use of homomorphic encryption on data size and latency. The choice of these metrics stems from two facts. First, SG is expected to deliver huge amount of power, monitoring and other types of data and thus we need to investigate how homomorphic encryption increases or reduces the amount of data to be transmitted. Second, given the real-time requirements of SG to prevent power failures or handle demand response, we need to investigate the ETE delay performance of the data delivered via aggregation. Specifically, we investigate the latency and the data size of ETE secure data aggregation using Paillier cryptosystem [11] which is proposed in [3] [4]. We then compare the results with HBH secure data aggregation that has extra computation overhead due to decryption and encryption process at the intermediate SMs. To the best of our knowledge, this is the first work comparing the performance of homomorphic encryption in ETE and HBH encryption for SG. Performance evaluation results indicates that HBH encryption provides the advantage of comparable latency and less data traffic compared to ETE homomorphic encryption.

The remainder of this paper is organized as follows. Section II briefly discusses some related work. Section III provides background on some homomorphic encryption algorithms. Our assumptions and design are described in Section IV. Section V discusses the experiments and the results. Finally, Section VI concludes the paper and discusses future research.

II. Related Work

The idea of data aggregation has been investigated for years in Wireless Sensor Networks (WSNs) to save energy. A wide variety of data aggregation techniques for WSNs are found in the literature. The classifications of those techniques, their operations and trade offs between different performance measures such as energy consumption, latency, and data accuracy are discussed and summarized in several works [7] [8] [9] [10]. Fundamentally, to maximize energy and bandwidth saving, data aggregation is performed hop by hop to plain data as they are being forwarded to the sink. However, since the plain data may contain sensitive information, many security protocols for WSNs [9], either in HBH or ETE fashion, are proposed to secure the data aggregation. The focus of security in WSNs is to make sure that the adversaries do not involve in the aggregation and thus cannot change the final result of the aggregation function. This is similar to outliers elimination. None of the works mentioned above consider the use of homomorphic encryption.

There is one work in WSNs focusing on the privacy of the data being aggregated. [13] studied concealed data aggregation in WSNs and makes an analytical comparison between ETE encryption and HBH encryption in terms of energy usage. The comparison was between ETE homomorphic encryption, i.e., Domingo Ferrer, and HBH encryption using RC5 block cipher, based on the number of clock cycles of three major operations: encryption, addition, and decryption operation. Even though the addition operation of 10-byte plaintext data only requires 4 clock cycles for RC5 compared to 1452 for Domingo-Ferrer, HBH encryption using RC5 produces higher overhead in terms of energy consumption due to many decryption and encryption operations at the intermediate nodes. For 10 sensor nodes per aggregator node, HBH encryption requires nearly doubled clock cycles compared to ETE encryption. However, this work does not apply to SG for two reasons. First, it does not meet high security requirements in SG since Domingo-Ferrer is insecure and vulnerable to chosen plaintext attacks. Second, since this work was geared for WSNs, its results do not apply to SG where there is no restriction on the energy usage. For instance, use of RC5 or TinyOS does not make sense in our work. In addition, we mainly focused on the ETE data latency which is crucial for SG and was not discussed in [13]. Finally, the results found in this paper is contrary to this work as HBH encryption performance is better than ETE in general in terms of data size and latency.

For SG, there are not many works on secure data aggregation. Recently, [3] proposed an ETE homomorphic encryption using Paillier cryptosystem to secure the information aggregation. At the intermediate node and sink, the aggregation operation is performed by multiplicating all incoming encrypted packets. To obtain the real aggregate value, the sink node decrypts the aggregated ciphertext. Hence, the ETE confidentiality of the information is maintained. While the authors discusses the security properties of their approach, there is no real implementation to assess its performance in terms of ETE delay and data size. We implemented this approach in the performance evaluation section.

Bartoli et al. [5], proposed both ETE and HBH security protection for the information aggregation using two different symmetric keys, a shared key between smart meter and the gateway and pairwise keys between every node and its one-hop neighbors. At the aggregator node, due the fact that packets will be sent to the same destination, the aggregation operation is performed by first eliminating unnecessary overhead from each packet and then concatenating those packets into a single packet. To secure the packet, AES block cipher was used for ETE security and HBH security. The authors focus on lossless data aggregation and investigates the amount of data to be aggregated at intermediate nodes to improve energy savings. The main goal is to look at performance issues of data aggregation whether it be HBH or ETE in terms of energy savings under different channel conditions. We believe that energy is not of concern for a network of smart meters and thus we focus on ETE delay.

III. Homomorphic Cryptosystem

A. Overview

In this section, we provide an overview of homomorphic encryption and summarize the existing techniques. Suppose that m_1 and m_2 be two values of plaintext, homomorphic encryption can be classified as:

TABLE I
HOMOMORPHIC CRYPTOSYSTEMS

Cryptosystem	Homomorphic Operation		Type of Key	Message Expansion Factor	Security Remark
	Additive	Multiplicative			
Paillier [11]	✓		Asymmetric Key	2	semantically secure (IND-CPA)
Okamoto-Uchiyama[15]	✓		Asymmetric Key	3	provable secure equivalent to difficulty of the factorization problem
Naccache-Stern [16]	✓		Asymmetric Key	≥ 4	provable secure under the prime residuosity assumption
RSA [17]		✓	Asymmetric Key	1	not semantically secure
El Gamal [18]		✓	Asymmetric Key	2	semantically secure (IND-CPA)
Domingo-Ferrer [19]	✓	✓	Symmetric Key	2	vulnerable to known plaintext attack
Castelluccia, Mykletun, Tsudik [20]	✓		Symmetric Key	add a small number of bits	provable secure

1) Additively homomorphic encryption. The result of addition operation on plaintext values can be obtained by decrypting the result of multiplication operation on the corresponding encrypted values.
$$m_1 + m_1 = D_{K_{PK}} \left(E_{K_{PUB}} (m_1) \times E_{K_{PUB}} (m_2) \right).$$
2) Multiplicative homomorphic encryption. The result of multiplication operation on plaintext values can be obtained by decrypting the result of multiplication operation on the corresponding encrypted values.
$$m_1 \times m_1 = D_{K_{PK}}(E_{K_{PUB}}(m_1) \times E_{K_{PUB}}(m_2)).$$

Many homomorphic cryptosystems and their variants can be found in the literature. Due to limited space, we only summarize a selection of homomorphic cryptosystems in Table I. A more comprehensive review can be found in [21]. Paillier cryptosystem is selected since it is an additively homomorphic encryption which can provide a sum function, has the lowest expansion factor, and semantically secure against chosen plaintext attack (IND-CPA) which make it attractive for privacy-preserving operations such as electronic voting and ETE encryption. Furthermore, its encryption cost is not too high and has an efficient decryption [21]. We now provide a brief description on this cryptosystem.

B. Paillier Cryptosystem

Basic notations used: $\mathbb{Z}_N$ - set of integers N, $\mathbb{Z}_N^*$ - set of integers coprime to N, and $\mathbb{Z}_{N^2}^*$ - set of integers coprime to N^2. The Paillier cryptosystem has the following algorithms:

1) **Key Generation**
   a) Public keys : (N,g) where N = p.q, gcd(pq, (p-1)(q-1)) = 1, and g a random number $\in \mathbb{Z}_{N^2}^*$, generated either from a set $\mathbb{Z}_{N^2}^*$ where gcd$\left(\frac{g^\lambda \, mod \, N^2 - 1}{N}, N\right)$ = 1 or g = (αN + 1)β^N mod N^2, α and β randomly selected from $\mathbb{Z}_N^*$.
   b) Private keys : (λ,μ) or (p,q) equivalently.
   λ = lcm(p-1,q-1), where lcm represents least common multiple, and a modular multiplicative inverse, μ = $(L(g^\lambda \, mod \, N^2))^{-1}$ mod N, where L(u) = (u-1)/N
   c) Check if $L(g^\lambda \; mod \; N^2)$ and N are co-prime to ensure that N divides the order of g, i.e. $gcd(L(g^\lambda \, mod \, N^2), N)$ = 1
2) **Encryption**
   a) Select a random number: r $\in \mathbb{Z}_N^*$
   b) Encrypt the message m $\in \mathbb{Z}_N$:
   $c = E(m) = g^m r^N \; mod \, N^2$
3) **Decryption**
   Decrypt ciphertext c $\in \mathbb{Z}_{N^2}^*$:
   m = D(c) = $L\left(c^\lambda \; mod \, N^2\right) \; \mu$ mod N

Paillier cryptosystem is non-deterministic because the same message will be encrypted into different ciphers using different random number r.

IV. ASSUMPTIONS AND DESIGN

We make the following assumptions throughout the paper:

1) The network topology is a multilevel network tree (i.e., acyclic) which consist of many SMs and one gateway as the sink node as shown in Figure 1. SMs at the lower level, send their encrypted power consumption data to their parent SM at the upper layer. These intermediate SMs perform either aggregation operation on the ciphertext or decrypt-aggregate-encrypt operation before sending the result to the upper SM or to the sink.

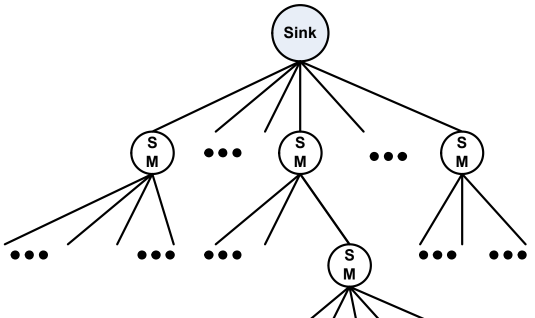

Fig. 1. Multilevel network tree

2) The aggregation tree is known in advance and remains static during the experiments and each downstream node knows the route to its parent.
3) The communication channels are assumed to be perfect and lossless so that there is no packet loss.

4) We use periodic per-hop aggregation [8] where data aggregation is performed using the sum operator and the result is transmitted as soon as the aggregator receives data from all of its children. A timeout is used to avoid the aggregator waiting indefinitely in case some of its children nodes are unable to report. The sink node announces to its first level aggregator nodes about its timeout value. Subsequently, depending on its position in the aggregation tree, an aggregator will adjust its timeout value according to the timeout value of its parent node. The average of the results is computed at the sink node by doing a division on the total sum which is in plain text.
5) To make sure that an aggregator node has received data from all of its downstream nodes, we assume that each aggregator node has a list of identities of its members. This unique identity of each SM, called High Frequency Identity (HFID) is similar to ones described in [12]. Each packet sent from each downstream node consists of this HFID and power consumption data at a certain period of time. The aggregator nodes verify it by using a simple look-up mechanism on the list. If the incoming packet comes from an authorized node, the receiving packet will be included in the aggregation operation. However, we assume that the time spent for source authentication is small and can be ignored in the latency calculation. Moreover, this authentication process has the same effect whether it is in ETE or HBH encryption.
6) The Paillier key generation and distribution were performed before the data reporting operations. All appropriate SMs have their public keys and/or private keys depending on their roles, i.e., SMs that act as aggregator nodes also have private keys.
7) Sink node periodically requests data reporting from each SM and the latency is measured starting from the sink node sending the request and continues until it receives the final average value.
8) We assume that there is one sink node but the results can be easily extended to multiple sink nodes where each sink forms a different tree.

We proposed four-tuples of information with the following format: (OPCODE,HFID, TIMESTAMP, DATA)
OPCODE is used to specify the operation that needs to be performed by each node. Four operation codes are defined and summarized in table II. HFID is the identity of the sender node. This identity is used for source authentication at the intermediate nodes. The unregistered sources are dropped and are not included in the aggregation processes. TIMESTAMP is used for data freshness. If the timestamp of a packet is less then the timestamp currently stored in the node, the operation or the data received will be discarded. DATA represents different information based on OPCODE as defined in the table II.

TABLE II
OPERATION CODE DEFINITION

OPCODE	Type of operation	DATA
K	Public Key distribution to all nodes, initiated by the Sink node	public key: N $\|\|$ g
P	Private Key distribution to the aggregator nodes, initiated by the Sink node	private key: λ $\|\|$ μ
S	Data Request to all nodes, initiated by the Sink node	parent node timeout (see assumption 5)
R	Data Reporting from all nodes to the sink in response to data request operation, generated by smart meters	at the smart meter : encrypted power consumption E(m). at the aggregator node : $\sum_{n=1}^{n}$ E(m_i) $\|\|$ E(n), n=number of nodes involved in the aggregation

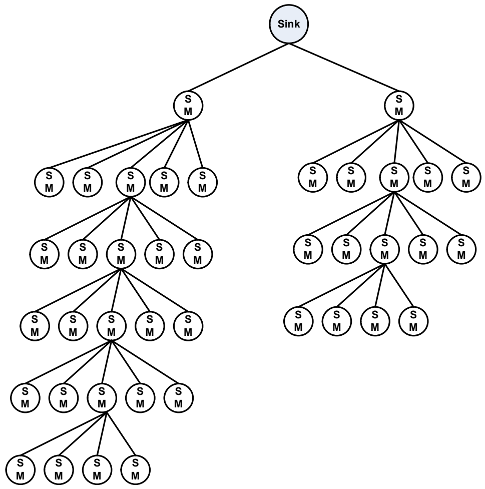

Fig. 2. Network topology with five tree depth level

V. PERFORMANCE EVALUATION

A. Experiment Setup

We used a Java-based simulation to implement the secure data aggregation using Paillier cryptosystem. Sink node announces its maximum timeout value in its periodic query to all SMs. On receiving the request, the aggregator at each depth level adjusts their maximum timeout value with the following simple formula : (maximum sink timeout limit - the tree depth $\times$ minimum delay between depth level). We set the following parameters constant during the experiments : minimum delay between depth level=50ms and power consumption data size= 16 bits.

We performed three experiments to observe the effect of key size, the network tree depth-level, and the number of SMs served per aggregation node on the ETE latency and data size for both ETE and HBH encryption. For the first and third experiments, we used two-level network topology that has 2 aggregators at each level. While the number of SMs per aggregator is fixed at 5 SMs for the first experiment, the third experiment has a varying number of SMs. For the second experiment, we used a total of of 40 SMs with 8 of them also performing as aggregator nodes. We started from a flat network with one tree depth-level, and then increased the tree depth-level by one. Figure 2 shows a sample network topology with five tree depth-level that we used in the second experiment. We used 64 bits key size for the second and third experiments.

In each experiment, each SM randomly generates 16 bits power consumption data. We measured the ETE latency and the average encrypted data size sent from SMs and aggregator nodes for each query. To make a fair comparison, we excluded the data size of the number of nodes at the aggegator nodes and only considered the aggregate power consumption data. After 100 queries, the sink node stops. We calculated the average ETE latency as well as the average data size from these queries. We repeated these steps for both ETE and HBH in a given network topology 20 times.

B. Experiment Results and Discussion

The effect of different key size to latency and data size (i.e., first experiment) are shown in Figure 3 and Table III respectively. The results showed that for a given key size, both ETE and HBH has a similar ETE latency. However, the longer key size will provide better protection at the cost of a significant increase in ETE latency while providing a linear increase in the size of the encrypted message. As shown in Table III, when we multiply the key size with the factor of 2, the encrypted message from the SMs will also double.

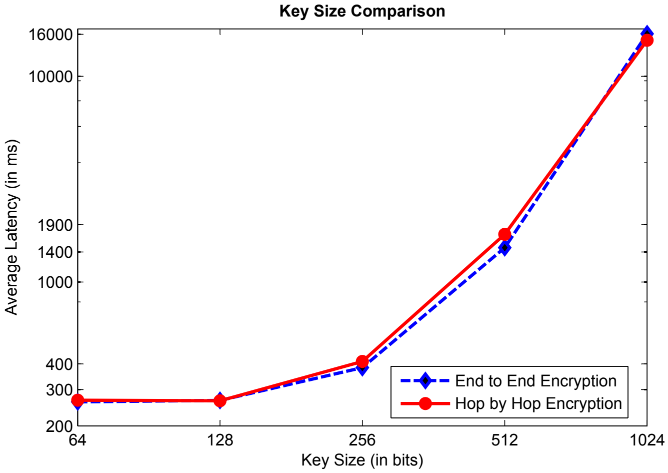

Fig. 3. The effect of key size on end-to-end latency

TABLE III
DATA SIZE COMPARISON WITH DIFFERENT KEY SIZE, SM= SMART METER, AGG = AGGREGATOR

Key Size (in bits)	ETE average encrypted message from (in bytes)			HBH average encrypted message from (in bytes)		
	SM	AGG	% increase	SM	AGG	% increase
64	15.68	132.80	747%	15.71	15.71	0%
128	31.72	269.15	749%	31.70	31.70	0%
256	63.65	540.56	749%	63.64	63.64	0%
512	127.65	1084.56	750%	127.66	127.66	0%
1024	255.60	2172.05	750%	255.70	255.70	0%

Considering the network tree depth level (i.e., second experiment), Figure 4 shows the same thing: In terms of average end-to-end latency, the difference between ETE and HBH encryption is not significant. This result is quite surprising since the aggregator nodes in HBH encryption have more operations (i.e., decryption, addition, and encryption) compared to a single operation (i.e., multiplication) in ETE encryption. This result can be attributed to the fact that the multiplication operation is very expensive and requires a comparable time to the three operations in HBH. However, in terms of bandwidth, while the average data size generated from the aggregator nodes in HBH encryption remained constant, ETE encryption showed a significant increase in the data size with the increase of the network tree depth-level as seen in Table IV. Hence, ETE encryption consumes more bandwidth than HBH encryption.

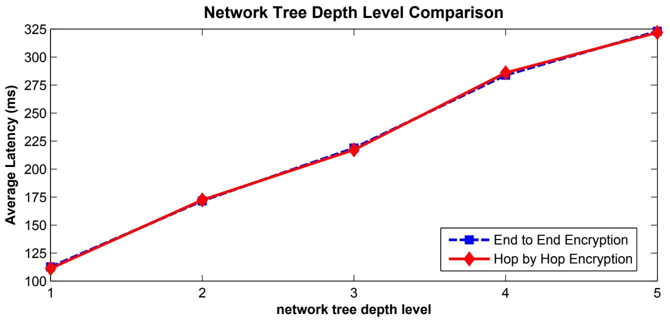

Fig. 4. Latency comparison for different depth-level

TABLE IV
DATA SIZE COMPARISON WITH DIFFERENT NETWORK TREE DEPTH LEVEL

Network tree depth-level	ETE average encrypted message from (in bytes)			HBH average encrypted message from (in bytes)		
	SM	AGG	% increase	SM	AGG	% increase
1	15.77	77.27	390%	15.76	15.76	0%
2	15.84	116.27	634%	15.80	15.80	0%
3	15.76	144.20	815%	15.78	15.77	0%
4	15.82	193.32	1122%	15.76	15.76	0%
5	15.81	202.68	1182%	15.77	15.77	0%

In the third experiment, when we increased the number of SMs per aggregator from 6 to 14, we observed that again the latency difference is not significant. Similarly, the data size is growing with ETE significantly.

Another result that is worth mentioning here is that distributing the aggregation to a bigger number of aggregators in ETE encryption will reduce the average encrypted message size produced by the aggregator. As shown in Table IV, for depth-level 2, there are 8 aggregators where each has 5 SMs (i.e., 40 SMs total) the increase in data size is 634%. However, when we look at Table V, row 3 where there are 10 SMs per aggregator and 4 aggregators are deployed (i.e., 40 SMs total), the increase in data size is 1494%.

Overall, the results have indicated that homomorphic encryption does not necessarily reduce the overhead and provides reduced end-to-end delay. When selection the homomorphic

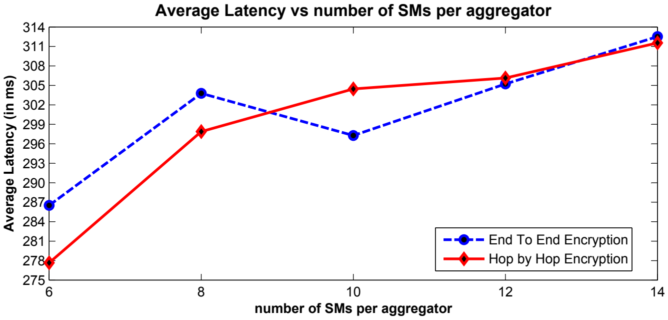

Fig. 5. Latency comparison with different number of SM per aggregator

TABLE V
DATA SIZE COMPARISON FOR DIFFERENT NUMBER OF SMS PER AGGREGATOR

Number of SMs per aggregator	ETE average encrypted message from (in bytes)			HBH average encrypted message from (in bytes)		
	SM	AGG	% increase	SM	AGG	% increase
6	15.67	140.51	797%	15.64	15.65	0%
8	14.89	192.93	1195%	14.89	14.89	0%
10	15.66	249.59	1494%	15.67	15.68	0%
12	15.63	295.76	1793%	15.69	15.69	0%
14	15.68	343.64	2092%	15.67	15.67	0%

cryptosystem, one has to be careful in the cost of the homomorphic operation. Based on the SG application requirements, the choice can consider use of hop-by-hop encryption and handle the privacy issue separately. For instance, the privacy concern about consumer's individual power consumption that will be exposed by hop-by-hop encryption can be addressed by using pseudonyms that are associated with the IDs of smart meters and this association is known only by the sink node as done in [12].

VI. CONCLUSION

In this paper, we have evaluated the performance of Paillier homomorphic cryptosystem in terms of end-to-end latency and data size for its use in SG applications. In end-to-end encryption, data aggregation is performed at the intermediate nodes without any decryption operation so that the data confidentiality is maintained. Furthermore, the sink does not need to do multiple decryption for each data received from its different child nodes to obtain the total values. In hop-by-hop encryption, each intermediate node decrypts, processes and encrypts the data received.

Experimental evaluation has indicated that ensuring data confidentiality and reduction of decryption operations at the sink using Paillier crypto system are at the expense of bigger data size and thus requires higher bandwidth, and longer latency time. The comparison has also shown that the average latency of hop-by-hop encryption is almost same as end-to-end encryption. Therefore, when addressing the underlying privacy in the SG, one has to look at the application requirements in terms of delay and bandwidth.

REFERENCES

[1] National Institute of Standards and Technology (NIST), NIST Framework and Roadmap for Smart Grid Interoperability Standards, Release 1.0, Januari 2010.
[2] The National Energy Technology Laboratory (NETL), Understanding the Benefits of Smart Grid, June 18, 2010.
[3] Fengjun Li and Bo Luo and Peng Liu,Secure Information Aggregation for Smart Grids Using Homomorphic Encryption, First IEEE International Conference on Smart Grid Communications, 2010.
[4] Seo, D., Lee, H., and Perrig, A., Secure and Efficient Capability-based Power Management in the Smart Grid, Parallel and Distributed Processing with Applications Workshops, IEEE International Symposium on, pp. 119-126, 2011 IEEE Ninth International Symposium on Parallel and Distributed Processing with Applications Workshops, 2011
[5] Bartoli, A., Hernandez-Serrano, J.,Soriano, M., Dohler, M.,Kountouris, A.,Barthel, D.,Secure Lossless Aggregation for Smart Grid M2M Networks,First IEEE International Conference on Smart Grid Communications, 2010.
[6] Rajagopalan, R., Varshney, P.K., Data aggregation techniques in sensor networks: a survey, IEEE Communication Surveys Tutorials 8 (4), 2006.
[7] Akkaya, K., Demirbas, M. and Aygun R.S., The Impact of Data Aggregation on the Performance of Wireless Sensor Networks, Wiley Wireless Communications & Mobile Computing (WCMC) Journal, Vol. 8 pp. 171-193, 2008.
[8] Fasolo, E., Rossi, M., Widmer, J., and Zorzi, M., In-network aggregation techniques for wireless sensor networks: a survey, IEEE Wireless Communications, Vol 14 Issue 2, pp 70-87, 2007.
[9] S. Ozdemir, and Y. Xiao, Secure data aggregation in wireless sensor networks: A comprehensive overview, in Computer Network Journal, 2009, Vol 53, pp. 2022-2037.
[10] Peter, S., Westhoff, D., and Castellucia, C., A Survey on the Encryption of Convergecast Traffic with In-Network Processing, IEEE Transactions on Dependable and Secure Computing, Vol. 7., No. 1, 2010.
[11] Pascal Paillier, Public-Key Cryptosystems Based on Composite Degree Residuosity Classes, EUROCRYPT 1999, pp223-238.
[12] Efthymiou, C., and Kalogridis, G., Smart Grid Privacy via Anonymization of Smart Metering Data, First IEEE International Conference on Smart Grid Communications, 2010
[13] Girao, J., Westhoff, D., and Schneider, M., Cda: Concealed data aggregation for reverse multicast traffic in wireless sensor networks, IEEE International Conference on Communications, 2005.
[14] S. Peter, K. Piotrowski, and P. Langendoerfer,On Concealed Data Aggregation for Wireless Sensor Networks, Proc. IEEE Consumer Communications and Networking Conference, pp.192-196, 2007
[15] Okamoto, T., and Uchimaya, S., A new public-key cryptosystem as secure as factoring, Advances in Cryptology - EUROCRYPT'98, pp. 208-218, 1998.
[16] Naccache, D., and Stern. J., A new public key cryptosystem based on higher residues, ACM Conference on Computer and Communication Security, pp 59-66, 1998.
[17] Rivest,R.L., Shamir,A. and Adleman, L. M.,A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM, 21(2):120126, 1978.
[18] Gamal, T.E., A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, CRYPTO 1984, volume 196 of Lecture Notes in Computer Science, pages 1018. Springer, 1984.
[19] J. Doming-Ferrer, A provably secure additive and multiplicative privacy homomorphism, in Proceedings of the Information Security Conference, 2002, pp. 471-483.
[20] C. Castelluccia, E. Mykletun, and G. Tsudik, Efficient aggregation of encrypted data in wireless sensor networks, in Mobile and Ubiquitous Systems: Networking and Services, 2005. MobiQuitous 2005. The Second Annual International Conference on, 2005, pp. 109 117
[21] Fontaine, C., and Galand, F., A survey of homomorphic encryption for nonspecialists. Eurasip Journal of Information Security, 2007(1):115, 2007.