A Survey of the Privacy Homomorphism in Wireless Sensor Networks

Ankit Chandra
Computer Engineering Department
Sardar Vallabhbhai National Institute of Technology Ichchanath, Surat-395007, Gujarat

Chintan Choksi
Computer Engineering Department
Sarvajanik College of Engineering and Technology
Athwalines, Surat - 395001 Gujarat

Manish Wadhwani
Computer Engineering Department
Sardar Vallabhbhai National Institute of Technology Ichchanath, Surat-395007, Gujarat

Nimit Shah
Computer Engineering Department
Sardar Vallabhbhai National Institute of Technology Ichchanath, Surat-395007, Gujarat

Vedarth Desai
Computer Engineering Department
Sardar Vallabhbhai National Institute of Technology Ichchanath, Surat-395007, Gujarat

ABSTRACT
The applications of the Wireless Sensor Networks (WSNs) comprising of resource constrained sensor nodes are increasing day by day. However, the pervasive environments in which the WSNs are deployed and the criticality of the available resources therein make the applicability of the security protocols therein, non-trivial. Amongst various panaceas pursued in research, one of the attractive ones is using privacy homomorphism based secure data aggregation. Indeed one can find several algorithms based on either Symmetric Key Cryptography or Asymmetric Key Cryptography in the literature that supports either additive or multiplicative homomorphic encryption. In this paper, we attempt to survey the existing algorithms with a view to highlight the characteristics of the same. However not limiting ourselves to only theoretical review of the existing literature, we also implement the algorithms. Our work principally focuses only on the support for confidentiality and privacy, the solutions for supporting message integrity and entity authentication are beyond the purview of our survey in this paper.

General Terms:
Survey

Keywords:
Wireless Sensor Networks, Secure Data Aggregation, Privacy Homomorphism

1. INTRODUCTION
In Wireless Sensor Networks, large numbers of sensor nodes are deployed to perform the application-related sensing task. But due to small size of these nodes and the severe environmental conditions in which they are normally deployed, they are bound by lot of constraints like energy, memory, computational capability and communicational constraints.

Wireless Sensor Networks are potentially going to be used in military, environmental monitoring, health monitoring, and home appliances applications among others[3]. Many of these applications being critical in nature, it is highly necessary that security and privacy of the data travelling through the WSNs is maintained[11]. However, bearing in mind the numerous constraints applied on WSNs, it becomes quite difficult for researchers to devise a highly efficient security protocol that keeps the network data totally secure and at the same time utilizes minimal amount of network resources. Hence, in WSNs, instead of traditional route-centric protocols, data-centric protocols supporting data aggregation feature are used. The data-centric multi-hop communication is based on the premise of pre-processing the sensed data at the intermediate sensor nodes, using typical data aggregation operators, before communicating a single data packet towards the base station[8]. Thus, significant reduction in the communication costs is obtained due to the reduced number of packets sent.

In order to provide security and privacy to the network data, we tried to implement simple secure data aggregation techniques in WSNs. During this implementation of secure data aggregation,

we found some drawbacks in the hop by hop security technique.In order to overcome the limitations, we moved towards another approach i.e. Privacy Homomorphism, allowing aggregation to be performed on cipher-texts eliminating the need for the aggregator node to perform decryption every time that it receives the data.

There are several privacy homomorphic encryption algorithms that have been proposed in the literature till date[21][13][9][1][22][19][2]. The performance of this algorithms needed to be surveyed on different metrics in order to identify the scenarios where this algorithms would be useful. We carried out this survey to understand the usefulness of different algorithms in different scenarios.We compared the privacy homomorphism algorithms over parameters such as throughput, memory usage and CPU cycles. Instead of theoretical study, all the comparisons are based on our implementations of the algorithms.

There have been previous attempts at surveying the privacy homomorphism algorithms in Wireless Sensor networks [12][15][18]. The results of all these papers have considered the practical implementation of the algorithms. The authors have not mentioned any attempt at trying to optimize the implementation of the algorithms for WSNs. [15][18] show the performance of the algorithms as a function of bits of data being worked on. These papers do not compare the RAM and ROM requirements of the implementations. While comparing the algorithms on highly resource constrained environments, the RAM and ROM requirements of the algorithms also needs to be taken into account. We have made an effort to reduce the number of computations by using extended Euclidean algorithm[4] and Chinese remainder theorem[5]

Organization of the rest of the paper: In section 2, we have provided a brief understanding of the privacy homomorphism concept and various privacy homomorphism algorithms that we have implemented. The implementation methodology is discussed in section 3, followed by results of comparison and analysis in section 4. In the end, we conclude by mentioning future directions in this particular field in section 5.

2. THEORETICAL BACKGROUND

Secure Data Aggregation : Data aggregation is an essential data-processing primitive in sensor networks. Sensor nodes forward data towards the sink. Sensor nodes closer to the sink receive data from nodes further away, they aggregate the information into concise digests.The aggregated data is encrypted using Privacy Homomorphic algorithms. This enables end-to-end security. Thus implements confidentiality and integrity to the data being transferred. This results in significant energy savings over having each node forward their respective readings directly to the sink.

Privacy Homomorphism is an encryption transformation that allows direct computation on encrypted data.It is a form of encryption which allows specific types of computations to be carried out on ciphertext and obtain an encrypted result which is the ciphertext of the result of operations performed on the plaintext. For instance, one person could add two encrypted numbers and then another person could decrypt the result, without either of them being able to find the value of the individual numbers. Privacy homomorphism is either additive or multiplicative or both. A Partial Homomorphic function has either additive or multiplicative property, whereas a fully homomorphic function has both multiplicative and additive property.

Eg: If E() is an transformation, E(x) and E(y) are two transformed values if E(x + y) = E(x) + E(y) then E() is said to be Homomorphic transformation.

The algorithms implemented were Benaloh, Elgamal, Paillier, RSA, GoldwasserMicali, Okamoto Uchiyama and Domingo Ferrer.Enlisted below are the details about the encryption algorithms viz parameters for encryption and decryption, public key, private key, encryption and decryption functions

Benaloh[2]:
Parameters: Prime numbers p, q and integers y, r
Public key: $n = p*q$, y, r.
Secret key: (p, q)
Encryption: Ciphertext $c = y^m u^r \pmod n$
Decryption: Compute m such that $(y^{-m} c \pmod n) \in Encr(0)$ for m'= 0, 1, 2,... until r−1.
$z \in Encr(0)$ if $z^{(p-1)(q-1)/r} \pmod n = 1$
Acronym:BE

Elgamal[7]:
Parameter: Prime number p, generator g, message m, random integer x in range 0 to p−1
Public key: (p, g, $g^x \bmod p$) and Secret key is x.
Encryption: Message m is represented in range 0 to p−1, a random integer y in range 1 to p−2. Compute cipher text $c_1 = g^y \bmod p$ and $c_2 = m^{-1} (g^x)^y \bmod p$.
Decryption: Compute message $m=(c_1^{-x})*c2$
Acronym:EL

Paillier[17]:
Parameter: Prime numbers p, q. $n = p*q$.
Encryption: Ciphertext $c=g^m r^n \bmod n^2$
Decryption: Compute message $m=L(c^{\lambda} \bmod n^2)*\mu \bmod n$
Acronym:PA

RSA[20]:
Parameters: Prime numbers p and q with similar bit length.
$n = p*q$. $\varphi(n) = (p-1)*(q-1)$.
Public Key: Pair of (n,e). e is selected random such that $1<e<\varphi(n)$ and gcd(e, $\varphi(n)$) = 1.
Private Key: Pair of(n,d). d is calculated as $d \equiv Inv(e) \pmod{\varphi(n)}$.

Encryption: Ciphertext $c=m^e \pmod n$
Decryption: Message $m=c^d \pmod n$.
Acronym:RS

Goldwasser Micali[10]:
Parameters:Prime numbers p, q
Encryption: Encode message m as a string of bits (m_1, ..., m_n).Ciphertext for each bit m_i is $C_i = y^2 x^{mi} \pmod N$
Decryption: For each i in (c_1 , ..., c_n), determine whether the value c_i is a quadratic residue. If so, $m_i=0$, otherwise $m_i=1$.
Acronym:GM

Okamoto Uchiyama[16]:
Parameters:Prime numbers p, q. Select integer g such that $g^p = 1 \pmod{p^2}$.
Public key:$n=p^2q$, g, h
Private key:(p, q)
Encryption: Plaintext $m \in 2^k$. Select $g \in_R Z_n$,
such that element $g_p = g^{p-1} \pmod{p^2}$ has order p and set $h=g^n \pmod$

n).
Ciphertext $c=g^m h^r (\text{mod} n)$
Decryption:$c'=c^{p-1}(\text{mod} p^2)$. Compute $m=L(c')L(g_p)^{-1}$ (modp) where $L(x) = (x-1)/p$
Acronym:OU

Domingo Ferrer[6]:
Parameters: integer d >= 2, large integer m
Secret Key: Pair of (r, m'). $r \in Z_m$ such that r^{-1} mod m exists. And a small divisor $m' > 1$ of m.
Encryption:Randomly split $a \in Z_m$ into d parts $a_1 .. a_d$ such that
$\Sigma_{i=1}^{d}(a_i) \text{ mod } m' = a$
$C = [c_1, ..., c_d] = [a_1 r \text{ mod } m, a_2 r^2 \text{ mod } m, ..., a_d r^d \text{mod } m]$
Decryption :Compute the scalar product of the j^{th} coordinate by r^{-j} mod m
$\Sigma_{i=1}^{d}(a^i) \text{ mod m}$ to get a.
$a = (c_1 r^{-1} \text{ mod } m + c_2 r^{-2} \text{ mod } m + ... + c_d r^{-d} \text{ mod } m) \text{ mod } m.$
Acronym:DF

3. IMPLEMENTATION METHODOLOGY

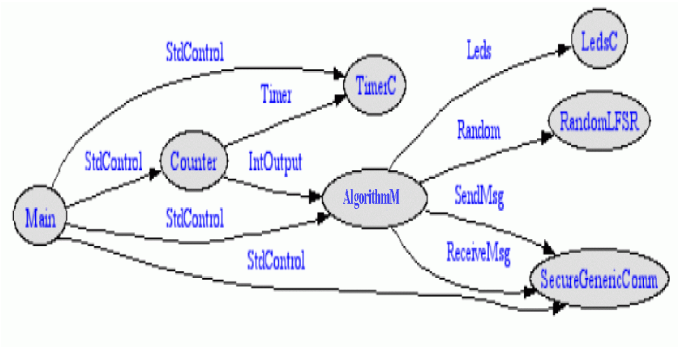

Fig. 1. Implementation Diagram

In this section, we describe our experimental setup including the tools, the test application and the metrics that we use for evaluation. We devise an application AlgorithmM in the TinyOS 1.x operating environment[7] using nesC[17] as our implementing language. We implement our application AlgorithmM to support all the homomorphic encryption algorithms discussed in theoretical background. We use TOSSIM[14] as the simulator to simulate the algorithms. TOSSIM captures TinyOS behavior at very low level and cannot provide exact information of CPU energy consumption and hence it does not model the power consumption.Hence, for energy and CPU cycle analysis, we use Avrora[10], an instruction level event simulator. Using results obtained from TOSSIM and Avrora, we compare performance of all the privacy homomorphic encryption algorithms.

Every application in TinyOS is a collection of modules, configurations and interfaces. AlgorithmM application is doing encryption of plain data and decryption of encrypted data using various algorithms discussed earlier. Figure 1 shows the part of a component graph of the test application that we create for each privacy homomorphic encryption algorithm.

We have kept the comparison as unbiased as possible by deciding the encryption and decryption key and the message to be encrypted, keeping the requirements of all the algorithms in mind.

4. PERFORMANCE RESULTS AND ANALYSIS

The seven privacy homomorphic algorithms are implemented in the AlgorithmM module. We survey these based on the different metrics viz. storage requirements (RAM and ROM), throughput in terms of bits/sec, energy in joule and CPU cycles. As we know that sensor nodes of WSNs are working in the resource constraint environment, the algorithm employed in it must be carefully design to save its energy and increase lifetime of WSNs. Hence, we use above mentioned metrics that are directly affecting the life time of the sensor nodes to compare the performance of privacy homomorphic encryption algorithms. In this section,we show our experimental results for these algorithms based on the above-mentioned metrics.

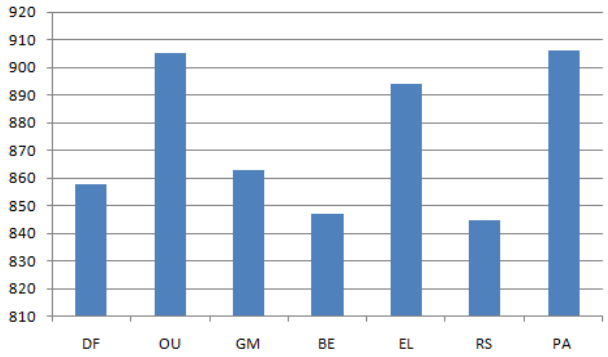

Fig. 2. RAM Usage(Bytes)

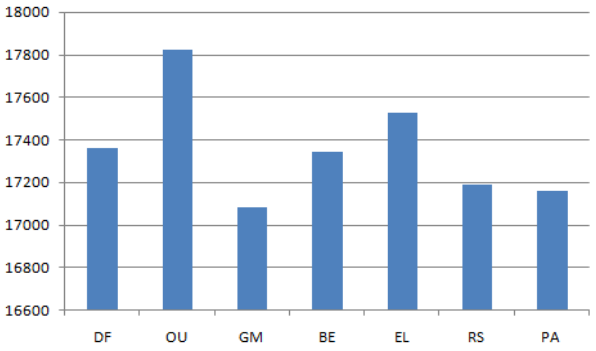

Fig. 3. ROM Usage(Bytes)

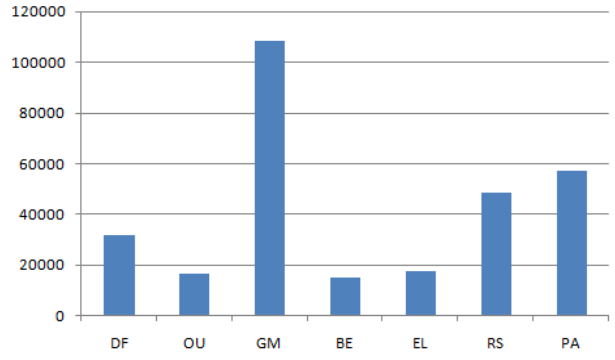

Fig. 4. CPU Cycles

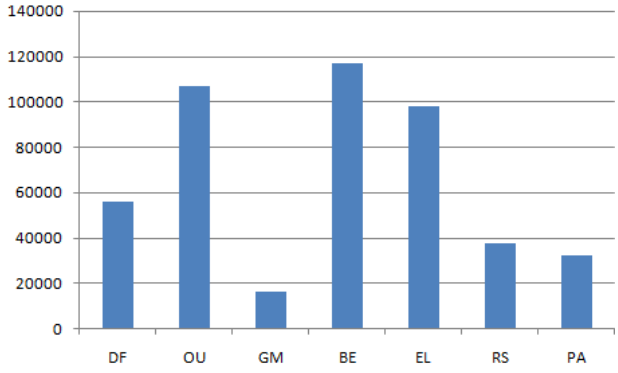

Fig. 5. Throughput(Bits/Sec)

We calculate throughput using following formula.
Throughput = (Message size in bit * 8 MHz)/(Total CPU Cycles)
8 MHz is the clock speed of mica2 mote

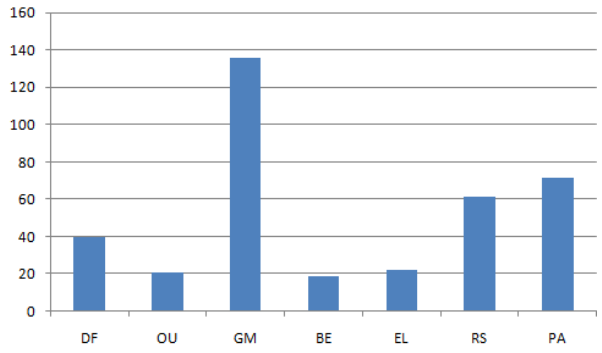

Fig. 6.
Energy(µJoule)

Energy = (no.of cycles) $*1.254125*10^{-9}$

5. CONCLUSION

In this paper we have surveyed the performance of privacy homomorphism algorithms after applying optimizations. Our paper shows that BE is fastest in encryption and decryption but the cipher text value is larger than plain text and hence requires more energy in transmission or requires more energy in compression[4].OU should be our next choice considering the resource constrained environment. Hence, if speed is our requirement, then we should go with BE while if energy usage is to be considered, then we should go with OU.

6. FUTURE WORK

Authenticating the message is as important as keeping it confidential. Since all the sensor nodes in the network have same security key, current authentication technique only makes sure that the message is sent from within the network and not whether a particular node has sent it. We will be working on providing a better authentication mechanism.

7. REFERENCES

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. Computer networks, 38(4):393–422, 2002.

[2] J. Benaloh. Dense probabilistic encryption. In Proceedings of the Workshop on Selected Areas of Cryptography, pages 120–128, 1994.

[3] C. Castelluccia, A.C.F. Chan, E. Mykletun, and G. Tsudik. Efficient and provably secure aggregation of encrypted data in wireless sensor networks. ACM Transactions on Sensor Networks (TOSN), 5(3):20, 2009.

[4] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. Introduction to algorithms. MIT press, 2001.

[5] C. Ding, D. Pei, and A. Salomaa. Chinese remainder theorem. World Scientific, 1996.

[6] J. Domingo-Ferrer. A provably secure additive and multiplicative privacy homomorphism*. Information Security, pages 471–483, 2002.

[7] T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In Advances in Cryptology, pages 10–18. Springer, 1985.

[8] D. Gay, P. Levis, R. Von Behren, M. Welsh, E. Brewer, and D. Culler. The nesc language: A holistic approach to networked embedded systems. In Acm Sigplan Notices, vol- ume 38, pages 1–11. ACM, 2003.

[9] C. Gentry. Fully homomorphic encryption using ideal lattices. In Proceedings of the 41st annual ACM symposium on Theory of computing, pages 169–178. ACM, 2009.

[10] S. Goldwasser and S. Micali. Probabilistic encryption. Journal of computer and system sciences, 28(2):270–299, 1984.

[11] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sen- sors. ACM Sigplan Notices, 35(11):93–104, 2000.

[12] V. Jariwala and D. Jinwala. Evaluating homomorphic encryption algorithms for privacy in wireless sensor networks. International Journal of Advancements in Computing Technology, 3(6), 2011.

[13] V. Jariwala and DC Jinwala. Evaluating galois counter mode in link layer security architecture for wireless sensor networks. International Journal of Network Security and Its Applications, 2(4):55–65, 2010.

[14] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: Accurate and scalable simulation of entire tinyos applications. In Proceedings of the 1st international conference on Embedded networked sensor systems, pages 126–137. ACM, 2003.

[15] E. Mykletun, J. Girao, and D. Westhoff. Public key based cryptoschemes for data concealment in wireless sensor networks. In Communications, 2006. ICC'06. IEEE International Conference on, volume 5, pages 2288–2295. IEEE, 2006.

[16] T. Okamoto and S. Uchiyama. A new public-key cryptosystem as secure as factoring. Advances in CryptologyEUROCRYPT'98, pages 308–318, 1998.

[17] P. Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Advances in CryptologyEUROCRYPT99, pages 223–238. Springer, 1999.

[18] S. Peter, D. Westhoff, and C. Castelluccia. A survey on the encryption of convergecast traffic with in-network processing. Dependable and Secure Computing, IEEE Transactions on, 7(1):20–34, 2010.

[19] R. Rajagopalan and P.K. Varshney. Data-aggregation techniques in sensor networks: a survey. Communications Surveys & Tutorials, IEEE, 8(4):48–63, 2006.

[20] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 26(1):96–99, 1983.

[21] B.L. Titzer, D.K. Lee, and J. Palsberg. Avrora: Scalable sensor network simulation with precise timing. In Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on, pages 477–482. IEEE, 2005.

[22] H. Zhi, L. San-Yang, and Q. Xiao-Gang. Overview of routing in dynamic wireless sensor networks. International Journal of Digital Content Technology and its Applications (JDCTA), AICIT Publications, 4(4):199–206, 2010.