Abstract

Objectives: A reliable and effective data communication is required without imposing overheads in terms of energy, communication costs, processing and the network latency. Data aggregation is the process which provides capabilities like In-Network processing, reducing the communication overhead, eliminating redundant packet transmissions and increases the lifetime of the network. Methods: A method for implementing secure data aggregation using Lightweight Cryptographic primitives is proposed which is suitable for resource constrained networks. Security services provided include authentication, integrity and confidentiality. A Lightweight Cryptography algorithm SPECK is used for encryption and decryption of the aggregated data. Instead of using the same key repeatedly an unique key is generated for every session. Findings: By implementing security primitives it is possible to provide basic services that are required to make the established network immune to attacks. The proposed method is analysed in terms of execution time, memory usage and throughput and provides better results in comparison to existing standard encryption algorithms. Application/Improvement: It is very well suited for application in which the data needs to be processed periodically and continuously. The future scope of work includes analyzing the scalability and communication overhead for large network of nodes and Homomorphic evaluation of the algorithm to deliver end-to-end security services.

Keywords: Lightweight cryptography, SPECK, Secure Data Aggregation, Wireless Sensor Network

1. Introduction

A WSN is formed by the collection of sensor nodes, which acquire information from real-time environment. WSNs provide reliable and efficient data gathering methods. WSN consist of devices which have resource restrictions with respect to power, memory, processing ability, transmission range and lifetime. The primary characteristics of WSN include the cooperation of multiple devices to sense, control and make decisions based on the environmental conditions. The participating devices may perform various roles like sensing, routing, gateway and sink in order to achieve the desired functionality. WSN provides capabilities like cooperative processing, distributed computing, data fusion, data aggregation, remote monitoring and control and surveillance. Specific applications focus either one or a more of these capabilities in order to achieve the desired objectives. It is a verified fact that the energy consumption during communication of data is significantly greater than processing it. So any amount of energy savings that can be achieved by optimizing the communication process greatly increases the overall lifetime of the system. Data aggregation mainly focuses on reducing the number of packets that are transmitted. It is a process in which a faction of nodes sends their data to a specific designated node which performs aggregation function. The nature of aggregation function may be sum, mean, average, modulus value of a predefined value or any simple arithmetic/algebraic function. Data aggregation process provides significant energy savings by decreasing the amount of energy expended for communication by slightly increasing the amount of computation. This trade-off pays rich dividends since energy spent on processing is much lesser than energy spent in communication. The data aggregation technique to be used also depends on the network topology, computational capabilities, energy

*Author for correspondence
Secure Data Aggregation for Wireless Sensor Network using Lightweight Cryptography

level and type of application\(^2\). The accuracy of aggregated data is an essential parameter which presents challenges because of the transient behaviour of the wireless communication medium and also due to the overwhelming power of the attackers who find every possible way to tamper with the network and data. The security services required by WSN may include integrity, confidentiality, availability, non-repudiation and authentication which varies depending upon the application under design. Implementing data security may add extra data packets whereas data aggregation primarily aims at reducing the amount of packets communicated across the system. This contrasting nature makes it a challenging task to implement security for data aggregation\(^3\). In general the aggregation process depends upon the structure of the network also. The network structure may be cluster, ring, tree-based and flat. The cluster head or the aggregator node selection depends upon the parameters like coverage, residual power of the node, relative position of the node from the Base Station (BS) and other factors specific to the application. The level of security necessary and the security services required is also based upon the demands of the application.

The primitive works on secure data aggregation focussed primarily on data integrity alone. A method which uses designated nodes known as witnesses to monitor the aggregation process is proposed\(^4\). An aggregation method using delay for authentication and aggregation when a rogue node is present in the network\(^5\). The above mentioned methods focus primarily on prevention of stealthy attacks. A method in which the confidentiality of data is also ensured by using the concept of Concealed Data Aggregation (CDA)\(^6\) in which the aggregator nodes perform Homomorphic Encryption directly over cipher texts received from sensor nodes. This method ensures end-to-end data confidentiality. A method which uses separate keys for individual nodes ensures both integrity and confidentiality\(^7\). Hierarchical level based aggregation method using dynamic key for each level is implemented to ensure security against malicious nodes\(^8\). Integrity and authenticity of concealed data aggregation is improved using Mykletun encryption method\(^9\). A technique which investigate the application of Elliptic Curve Cryptography (ECC) and verify its Homomorphic nature is proposed\(^10\). The results conclude that ECC provides a reasonably higher level of security than the RSA algorithm for reduced key size. A method uses a scrambling technique enabling transitional nodes to perform verification alongside aggregation process\(^11\). A protocol which uses Elliptic Curve El Gamal (ECEG) and signatures while the aggregation process is done\(^12\). Individual data identification by using separate keys for each node increases the communication and computation bandwidth. A modified scheme is proposed\(^13\) in which the signatures are also included in the aggregation process. The authors propose a method for preserving integrity of CDA scheme\(^14\), in which multiple public keys can be used to reconstruct the plaintext. A method is proposed in which data aggregation is performed at the node level itself\(^15\). The method aims at reducing network congestion and increases data accuracy and integrity. A witness based slice and mix data aggregation approach in which the aggregation process is parallel done by a node designated as witness node\(^16\). A scheme which makes use of combined benefits of symmetric and asymmetric algorithms to implement secure data aggregation is implemented\(^17\). The scheme offers data integrity, confidentiality, versatility, scalability and portability. The method uses the faster computation advantage of symmetric cryptographic techniques and key generation techniques of asymmetric algorithms to achieve the desired security services. Use of Homomorphic encryption and HMAC ensures Data integrity and confidentiality.

2. Background

2.1 Lightweight Cryptography

Lightweight cryptographic protocols are primarily for application in which the constituent devices have resource limitations. Their major application areas include end-to-end communications applications and lower resource device applications. The term lightweight refers to the reduced Gate Equivalents (GE), i.e., the quantity of logic gates required to realize the logic in hardware, overall memory usage size and execution time. They provide adequate security, but does not always take into account the security-efficiency trade-offs. The selection of cryptographic algorithm depends on the requirements of security and the limitations of the device over which the application is developed. From Figure 1 we can deduce that security is explicitly measured in terms of key size, performance with number of rounds for execution. The cost factor is determined by the type of architecture which is used by the processing unit, which can either be serial or parallel. The type of architecture also has an impact on the performance. Higher the key length and more the number of rounds the desired security and
same round function as described for the original cipher implementation. The key schedule process is used to generate keys for individual rounds. HMAC (Hash based Message Authentication Codes) and KDF (Key Derivative Functions) used are SHA-512 and standard NIST KDF functions respectively.

3. Proposed Method

The proposed method is implemented in a WSN model with three hierarchical levels. They are the sensor node, the aggregator node and the BS. The sensors are grouped into clusters and a designated node acts as the cluster head, which is the aggregator node. The data obtained from the sensor node is processed before it is transmitted to the aggregator node. This is done so that the aggregator node doesn’t have to spend any resources for data processing. Once data from all nodes in the cluster is received the aggregator node starts the aggregation process. The aggregated data is then encrypted using the Lightweight cryptographic algorithm SPECK which is symmetric block cipher based security algorithm. A dynamic key is generated for each round of data processing. This is done to ensure that the encrypted data varies for each transmission period. The nomenclature of the terms used in the proposed algorithm is given as follows:

Let $S_1, S_2, \ldots, S_n$ represent sensors in the cluster.

Let $d_1, d_2, \ldots, d_n$ represent the data captured by the nodes.

$K_{SNAGG}$ be the key shared between the sensor nodes and the aggregator node.

$D_1, D_2, \ldots, D_n$ represent the data encrypted using $K_{SNAGG}$.

$D_{agg} = \text{AGGREGATE} (D_1, D_2, \ldots, D_n)$

$K_{AGGBS}$ represents the key used by the aggregator node for encrypting the aggregated data.

$K_{BSAGG}$ represents the key used by the BS for decryption.

$K_{HMAC}$ represents the key used for hashing.

$IV$ - Initialisation Vector for key generation

$C_{BS}$ - Secret key shared between Aggregator node and BS.

$N_i$ - Counter Value

$AGG_{ID}$ - Aggregator Node ID.

$ED_{agg}$ - Encrypted Data aggregate value

$MAC_{agg}$ - HMAC value computed by the aggregator node

$MAC_{BS}$ - HMAC value computed by the BS

Figure 1. Trade-offs in design for lightweight cryptography

Performance can be achieved at the cost of increased execution time. Generally high performance is directly linked with increased cost. So a balance has to be achieved between cost, security and performance which is purely application specific.

2.2 SPECK Algorithm

In June 2013, NSA (National Security Agency) launched a family of lightweight cryptographic algorithm and termed it as SPECK\textsuperscript{18}. SPECK and its sibling SIMON are typically the same implementation but differ in the way they are optimized for. While the former is crafted for high performance on software, SIMON is optimized for high performance on hardware. SPECK belongs to the category of ARX (AND-ROTATE-XOR) cipher. The major highlight of SPECK algorithm is that it supports multiple sizes for data block and key length. A block is always made of 2 words, and words have varying lengths ranging from 16, 24, 32, 48 up to 64 bits. The chosen key may be constituted with 2, 3 or 4 words. The block size ranges from 32 to 128 bits and the length of key varies from 64 to 256 bits. Usually any variant of SPECK can be represented as SPECK 128/192 in which 128 denotes block size and 192 denotes the key length. The functions in each round includes typically 2 rotations, addition of word in the right to the left, performing XOR operation of the selected key with the left word, and then XORing the resulting left word to the right word. This function is applicable for encryption process. The function for decryption is the inverse round function where modular subtraction is used in place of modular addition. The key schedule process also makes use of the same round function as described for the original cipher implementation. The key schedule process is used to generate keys for individual rounds. HMAC (Hash based Message Authentication Codes) and KDF (Key Derivative Functions) used are SHA-512 and standard NIST KDF functions respectively.
The proposed method can be decomposed into three phases i) Data Collection Phase, ii) Secure Aggregation Phase and iii) Verification and Decryption Phase. The various actions involved in each phase can be found as:

3.1 Data Collection Phase
All sensor nodes collect real time data from the environment. The rate at which the data collection is performed is defined at the time of network deployment. Encryption of the collected data is performed using the key common to the sensor node and the aggregator and then transmitted to the aggregator node.

3.2 Secure Aggregation Phase
The aggregator node waits till the data from all the member nodes reaches it. Then the aggregation of collected data is performed. The nature of aggregation function can be generally used methods like Summation, Average and Median of all the values. The session key used for SPECK and HMAC are generated in the next step. The combination of SPECK and HMAC provides the security for the aggregated data. The encrypted aggregated data is then transmitted to the BS. Algorithm 1 represents the sequence of actions performed by the aggregator node.

Algorithm 1: Secure Aggregation Process

Input: D1, D2, …… Dn, CBS, IVi, Ni, AGG ID

Output: EDagg, MACAGG

1. Create a random IVi ∈ [1, n-1]
2. Compute KaggBS = KDF (IV, CBS, Ni, AGG ID)
3. Compute KHMAC = KDF (IVi, CBS, Ni, AGG ID)
4. Compute Dagg
5. Compute EDagg using KaggBS
6. Compute MACAGG = HMAC (EDagg, KHMAC)

3.3 Verification and Decryption Phase
BS receives the data from the aggregator node and generates the session key for HMAC verification and decryption. HMAC of the received data is computed and it is compared with the received HMAC. Upon successful verification, decryption is performed to get the aggregated data. If the HMAC verification fails, the received data is not processed any further. Algorithm 2 depicts the same.

Algorithm 2: Verify and Decrypt

Input: EDagg, MACAGG, CBS, IVi, Ni, AGG ID

Output: Dagg, MACBS

4. Results and Discussion
The proposed method is implemented on Arduino UNO board which is a AtMega328 microcontroller. For experimental analysis the ZigBee network configuration for devices are done using X-CTU utility software in which, 2 devices are configured as end devices (sensor nodes), one device as router (aggregator node) and one device as coordinator (BS). The output is viewed by connecting the ZigBee adaptor board to a PC or Laptop with the coordinator device placed in it. Figure 2 shows an experimental setup with 3 nodes.

4.1 Execution Time
The analysis of execution time for implementing the algorithm in the chosen hardware is done. Keeping in mind the severe resource constraints of WSN faster execution time means lesser processing overhead and increased efficiency. In Figure 3 the execution times of our proposed algorithm is compared with the standard AES algorithm with 128, 192 and 256 bit key size implementations. The proposed scheme although has a higher time for key setup the time for encryption and decryption is very less compared to AES algorithm.

4.2 Memory usage and Throughput
The efficiency of a lightweight cryptography is measured in terms of GE, memory usage and throughput. In WSN
The security services provided include data integrity, confidentiality. Data integrity is ensured using HMAC and the confidentiality is provided using the Lightweight cryptographic primitives. It is very well suited for application in which the data needs to be processed periodically and continuously. The security implementation doesn’t impose any limitation to the data aggregate function. The proposed method achieves security services while performing aggregation process also. The future scope of work includes analyzing the scalability and communication overhead for large network of nodes and Homomorphic evaluation of the algorithm to deliver end-to-end security services.

6. Acknowledgement

The authors wish to express their sincere thanks to the Department of Science & Technology, New Delhi, India (Project ID: SR/FST/ETI-371/2014). The authors also thank SASTRA University, Thanjavur, India for extending the infrastructural support to carry out this work.

7. References

1. Akkaya K, Demirbas M, Aygun RS. the impact of data aggregation on the performance of wireless sensor networks. Wireless Communication and Mobile Computing. 2008 Feb; 8(2):171–93.
2. Sasirekha S, Swamynathan S. A comparative study and analysis of data aggregation techniques in WSN. Indian Journal of Science and Technology. 2015 Oct; 8(26):1–10.
3. Ozdemir S, Xiao Y. Secure data aggregation in wireless sensor networks: A comprehensive overview. Computer Networks. 2009 Aug; 53(12):2022–37.
4. A Witness-Based Approach For Data Fusion Assurance In Wireless Sensor Networks. Available from: http://www.powershow.com/view/1450b7-NDhlZ/A_Witness_Based_Appearance_For_Data_Fusion_Assurance_In_Wireless_Sensor_Networks_W_Du_J_Deng_Y_Han_P_Varshney_powerpoint_ppt_presentation. Date Accessed: 19/04/2004.
5. Secure aggregation for wireless networks. Available from: http://www.cs.fsu.edu/~yasinsac/group/slides/polite5.pdf. Date Accessed: 3/03/2003.
6. Westhoff D, Girao J, Acharya M. Concealed Data Aggregation For Reverse Multicast Traffic In Sensor Networks: Encryption, Key Distribution, And Routing Adaptation. IEEE Transactions on Mobile Computing. 2006 Oct; 5(10):1417–31.
7. Efficient Aggregation of Encrypted Data in Wireless Sensor Networks. Available from: http://archive.cone.informatik.
uni-freiburg.de/lehre/seminar/adhoc-w09/submissions/DataAggInWSN.pdf. Date Accessed: 30/01/2010.
8. Saranya S, Bharathi N. Hierarchical Approach to Secure Data Aggregation in Wireless Sensor Networks. Indian Journal of Science and Technology. 2015 Jul; 8(14):1–6.
9. Raj GS, Thanjaivadivel M, Viswanathan M, Bindhu N. Efficient Sensing of Data when Aggregated with Integrity and Authenticity. Indian Journal of Science and Technology. 2016 Jan; 9(3):1–5.
10. Public key based cryptoschemes for data concealment in wireless sensor networks. Available from: http://ieeexplore.ieee.org/document/4024506/. Date Accessed: 11/06/2006.
11. Zhu WT, Gao F, Xiang Y. A Secure And Efficient Data Aggregation Scheme For Wireless Sensor Networks. Concurrency and Computation, Practise and Experience. 2011 Aug; 23(12):1414–30.
12. An Efficient And Verifiable Concealed Data Aggregation Scheme In Wireless Sensor Networks. Available from: http://ieeexplore.ieee.org/document/4595533/. Date Accessed: 29/07/2008.
13. Boneh D, Gentry C, Lynn B, Shacham H. Aggregate and verifiably encrypted signatures from bilinear maps. EUROCRYPT’03 Proceedings of the 22nd International Conference on Theory and Applications of Cryptographic Techniques. 2003 May. p. 416–32.
14. Ozdemir S, Xiao Y. Integrity Protecting Hierarchical Concealed Data Aggregation For Wireless Sensor Networks. Computer Networks. 2011 Jun; 55(8):1735–46.
15. Sabrina Sicari A, Luigi Alfredo Grieco B, Gennaro Boggia B, Coen-Porisini A. DyDAP: A Dynamic Data Aggregation Scheme For Privacy Aware Wireless Sensor Networks. The Journal of Systems and Software. 2012 Jan; 85(1):152–66.
16. Selvin V, Kumar M. Tree based Efficient and High Accuracy Data Aggregation for Wireless Sensor Networks. International Conference on Modelling Optimisation and Computing. Procedia Engineering. 2012 Sep; 38:3833–9.
17. Merad Boudia OR, Senouci SM, Feham M. A Novel Secure Aggregation Scheme for Wireless Sensor Networks using Stateful Public Key Cryptography. Ad Hoc Networks. 2015 Sep; 32:98–113.
18. The Simon and Speck Families of Lightweight Block Ciphers. Available from: https://www.cryptolux.org/images/a/a0/Beaulieu-DAC2015.pdf. Date Accessed: 06/2015.