EPSDA: Energy Efficient Privacy preserving Secure Data Aggregation for Wireless Sensor Networks

Joyce Jose[1], M Princy[2] and Josna Jose[3]

Post Graduate Scholar[1], Lecturer[2], Post Graduate Scholar[3],
Dept. Information Technology, Karunya University, Coimbatore, India

joycejose1990@gmail.com[1], princym@karunya.edu[2], josnajose1990@gmail.com[3]

Abstract

The privacy preserving data aggregation protocols in wireless sensor networks have many applications in security critical areas, since it hides individual nodes' data from adversaries. The existing hop by hop and shuffling based privacy preserving protocols does not provide an energy efficient, accurate and secure data aggregation result in base station, due to the energy consuming decryption at the aggregator node, reveals large amount of privacy protected information to adversaries, thereby it looses end to end confidentiality of data and allows the adversary to generate inaccurate results. The proposed privacy preserving protocol named EPSDA overcomes the problems in the existing scheme by performing aggregation on encrypted data, based on privacy homomorphic property of the encryption algorithm however there is a chance of replying the old information to the network. The EPSDA protocol prevents the replay attack by achieving data freshness during aggregation, this increases the accuracy of the aggregated result by performing the aggregation on encrypted data and reduced number of transmissions. It also guarantees the integrity of the aggregated result at base station and the authentication of the data during non delayed aggregation. Our main aim is to provide an energy efficient and secure data aggregation scheme, which guarantees the privacy, authenticity and freshness of individual sensed data as well as the confidentiality, accuracy and integrity of aggregated data.

Keywords: *Wireless Sensor Networks; Data Privacy; Data confidentiality; Data authentication; End to end encrypted data aggregation; Privacy homomorphism; Accuracy; Data integrity; Data aggregation; Hop by hop encrypted data aggregation; Data freshness*

1. Introduction

An event based Wireless Sensor Network (WSN) [1] consists of many numbers of resource constrained sensor nodes, it cooperatively collects information from the deployed area and pass information to the Base Station (BS), through a wireless medium in a multi hop manner. The WSN's are mainly used for security applications such as battlefield surveillance. However, it is used in civilian application areas, including habitat monitoring, home automation, health care applications, traffic control etc.

The dense deployment of the resource constrained sensor nodes in a particular area sense same data. The communication takes a large part of sensors limited energy, transfering redundant data through network is not energy efficient. The data aggregation [2] process combines data from different sensor nodes based on aggregation functions (SUM, AVG, MIN, MAX, HISTOGRAM *etc*.), by avoiding redundant data, achieves high bandwidth and energy efficiency, which in turn increases the network lifetime.

The WSN's in a security critical application requires to keep privacy of individual sensing data from adversaries. The hop by hop based privacy preserving protocols present in the WSN's can ensure the secrecy of individual data, however, it does not guarantee an accurate and robust aggregated results to the BS in an energy efficient manner due to the power consuming decryption at the aggregator node. The decryption at the aggregator node will give a chance to adversary to modify the aggregated result, the end to end confidentiality of data is lost. It is difficult to prevent node compromise attack in WSN's, hence a system which does not decrypt data at the aggregator eliminates the energy wasted in decryption, reduces the node compromising frequency aswell. The proposed privacy preserving protocol provide such a system that allows aggregation on encrypted data rather than plain data, reduces energy consumption by avoiding decryption at intermediate nodes. Thus, EPSDA provides a precise result to the BS by avoiding the decryption at the aggregator node, reduces number of transmissions aswell. Key distribution protocol of EPSDA prevents replay attack by generating a new encryption key for every new session and ensures data freshness.

Ensuring data authentication and aggregated data integrity are important in WSN's that is transmitted along a wireless communication medium. The secret key and ID pair of each sensor node provide the identity and the aggregated MAC corresponds to the aggregated data gives the guarantee for the received aggregated data to BS. The homomorphism MAC [3] is used for checking the integrity of the aggregated data. The homomorphism MAC has a property that is given below, where a and b are data of two nodes.

$$MAC(a+b) = MAC(a) \oplus MAC(b)$$

The passing of count value (which represent the number of nodes contributed in an aggregated result) supports a new aggregation function AVG in addition to SUM function in the existing shuffling based privacy preserving technique (EEHA) [4]. Thus, EPSDA provides secure data aggregation, which guarantees data authenticity, data freshness, data privacy, end to end confidentiality, aggregation accuracy and data integrity with minimal energy consumption.

The rest of the paper is organized as follows. Section 2 describes the related works. Section 3 presents the system model and design objectives. The Section 4 demonstrate the key distribution in EPSDA scheme, Section 5 describes the EPSDA scheme. In Section 6, we evaluate the performance of EPSDA scheme. The Section 7 concludes the work.

2. Related Works

The privacy preserving protocols are divided into two types, homogeneous and heterogeneous protocols based on the type of nodes in the sensor network. In homogeneous protocols, all sensor nodes in the network have the same resources and the intermediate node performs sensing and aggregation, then forward the aggregated result to BS.i.e, all the sensor node can play the role of aggregator. However in heterogeneous protocols, the intermediate nodes are considered as a special node and it can perform all the task of a normal sensor node except sensing.

Two types of secure aggregation protocols are presented in WSN's, these are end to end encrypted data aggregation [5, 6] and hop by hop aggregated encrypted data aggregation [5, 6]. In an end to end encrypted data aggregation, the aggregation is performed on encrypted data rather than plain data. There is only one decryption at the sink. In hop by hop encrypted data aggregation, the decryption is done on each hop during the transmission of data from sender to the receiver. Thereby loosing the end to end confidentiality between sender and receiver.

The homogeneous and heterogeneous protocols are of four types, these are perturbation, privacy homomorphism, shuffling and hybrid. For achieving secrecy of data using a perturbation technique, each node customizes its data into some form using its encryption key and public or private seed generated by randomization technique [7]. The CPDA [8] is the example of perturbation technique.

The privacy homomorphism technique saves the energy wasted for decryption at the intermediate node by allowing arithmetic operation on encrypted data as well as in network data aggregation. There is only one decryption at sink node. The ESPDA [9], CDA [10], EAED [11] and RCDA [12] are examples of privacy homomorphism technique.

In ESPDA (Energy-Efficient Secure Pattern Based Data Aggregation for Wireless Sensor Networks), the energy efficiency is achieved using the pattern code sent during data aggregation instead of actual data and the end to end confidentiality is achieved using the encryption key shared between the source node and sink. The symmetric cryptographic algorithms are used in ESPDA, so the resource consumption is less. In addition it provides message authentication and data freshness during data aggregation.

The CDA (Concealed Data Aggregation) uses the DF approach [13] for the end to end data aggregation, reduce the processing burden on hop by hop encrypted data aggregation. In CDA, all the sensor nodes share a common secret key with the BS, compromising of one node reveals the security information of another node which looses the privacy of other nodes. But in EAED (Efficient Aggregation of Encrypted Data in Wireless Sensor Networks), each node shares a distinct key with the BS this achieves privacy against other nodes however it is not scalable, since the BS has to keep the distinct secret key of all nodes in the network.

The RCDA (Recoverable Concealed Data Aggregation for Data Integrity in Wireless Sensor Networks) is based on elliptic curve cryptography achieves end to end confidentiality by using the additive privacy homomorphism technique. It can use any number of aggregation functions in BS by recovering the individual sensing data from the aggregated data at the BS. In addition, it guarantees the integrity and authenticity of sensed data by using the aggregated signature scheme.

The hybrid data aggregation technique uses more than one privacy preserving technique to conceal the individual sensed data. The PIA (Privacy Preserving Integrity Assured Data Aggregation) [14] is an example of the hybrid data aggregation technique, guarantees the integrity of the sensed data in an energy efficient manner. For the integrity protection and privacy preserving data aggregation, the PIA proposed four symmetric key solutions for the single aggregator model. The first solution combines the homomorphism and MAC to construct an authenticated encryption scheme. It supports two aggregation functions such as average and standard deviation. The second solution preserves the privacy of distribution of data by using the Order Preserving Encryption Scheme (OPES) [15] and it verifies the integrity of comparison based data aggregation. The third solution is used for distributed integrity verification by using the Secure Hierarchical in networking Aggregation (SHIA) [16] scheme and it supports any number of aggregation functions. The fourth solution improves the privacy and integrity of the third solution by constructing a logical aggregation tree within the aggregtion node and it supports decomposable functions such as mean, standard deviation, count, MIN/MAX.

In the shuffling based privacy preserving data aggregation scheme, each sensor node slices its data into k number of pieces. One of the piece is kept on the node itself and the remaining $k-1$ slices are encrypted and send to $k-1$ neighbour nodes within h hops

(for a dense network, *h=1*). The SMART [8], EEHA [4] and iPDA [17] are examples of shuffling technique.

The SMART (Slice Mix AggRegaTe) scheme provides hop by hop based additive agregation for preserving the privacy of individual sensing data in a tree topology. In this technique, each node slices its data into *k* number of pieces to conceal the individual sensed data from other nodes or adversaries. The communication overhead increases with increase the number of slices. However it does not cause any extra computational overhead. The EEHA (Energy Efficient and High Accuracy secure data aggregation) scheme overcomes the communication burden of SMART technique by applying the slicing operation on leaf nodes without affecting the privacy of other sensor nodes. This scheme provides a secure data aggregation scheme without leaking private data of other nodes and without introducing any extra overhead on power limited sensors. The iPDA (integrity protecting Private Data Aggregation) technique guarantees the privacy and integrity of individual sensed data. The data privacy is achieved using the slicing and assembling technique and the integrity is achieved using the disjoint aggregation tree.

3. System Model and Design Objectives

3.1. Network Model

The sensor network consists of a large number of resource constraint sensors to cooperatively perform a specific task. The aggregation is performed on the aggregation tree routed at the BS. The sensor network consists of three types of nodes, leaf node, intermediate node or aggregator, base station (BS) or sink. The leaf node is responsible for slicing operation to achieve the privacy preserved data aggregation. It also performs both the sensing and aggregation and then forwards the aggregated result to its parent. The intermediate node is responsible to perform both the sensing, aggregation of both data and MAC, after that it forward both the aggregated data and MAC to upper aggregator or to the sink. The sink is responsible for the processing of an aggregated result received from the network and generates the information reflecting the target field. In addition, it checks the integrity of the aggregated result obtained from the network.

3.2. Attack Model

The transmission of data from sensor node to the BS without adopting any secure method leads to the leakage of large amounts of data to adversaries. So they can easily deploy their own sensors and inject false data into the network. One method to protect the sensor node from the intruder attack is that each sensor node should be initialized with a unique identifier and a secret key before it is deployed. The identifier helps to prove that the sensor node is a legitimate one and the secret key is used for data authentication.

In node compromising attack, the adversaries get the access to the authorized nodes through physical means of tampering the device, the adversaries get access to the secret key and the identifier can inject false data or modify, forge or discard the received data from another node as a trusted node in the network. There are more possibilities to change the aggregation result obtained from the sensor network, it looses the percentage of aggregation accuracy.

In an eavesdropping attack, the attacker tries to overhear the transmission channel to get the sensitive information. The eavesdroppers are of two types: inside and outside eavesdroppers. The inside eavesdroppers are intruder or compromising node, they can get the

private data destined for other nodes. The privacy preserved data aggregation approach prevents them from recovering the private sensed data of individual sensor nodes. The outside eavesdroppers can be prevented by using the encryption technique.

3.3. Assumptions and Design Objectives

Assume that the BS is equipped with tamper resistant hardware and is trustworthy with enough resources. The sensor nodes are not equipped with tamper resistant hardware and it has limited resources, adversaries can compromise sensor nodes except the sink node.

Our data aggregation scheme should satisfy the following criteria.

3.1.1. Data Privacy: To use the WSN in secure applications, the privacy of the data must be guaranteed during the transmission of data. The privacy preserving data aggregation ensures that the data sensed by each sensor node should be known only to itself, no matter how many nodes have been compromised. The privacy is achieved using the slicing operation at the leaf nodes.

3.1.2. Data Confidentiality: It ensures that the partially or fully aggregated data is known only to the sink, no matter how many nodes have been compromised. The data confideniality is achieved using end to end encrypted data aggregation.

3.1.3. Data Authentication: It gives assurance to the BS that the data it received from an authenticated node not from an intruder. The data aggregation is achieved using the ID and secret key of each sensor node.

3.1.4. Energy Efficiency: The sensor nodes are battery limited, hence the data aggregation protocol should be energy efficient while it achieves all the security requirements. In WSN's, transmission takes much more energy than computation, by reducing the number of transmissions, the communication overhead can reduce and by reducing the number of power consuming decryption at each node during data aggregation will reduce the computational overhead. It thereby improves the energy efficiency of sensor nodes.

3.1.5. Aggregation Accuracy: The final decision at the sink is based on the aggregation result obtained from the BS. The reduced number of transmissions and the avoiding of decryption at sensor nodes during data aggregation improve the accuracy of the data.

3.1.6. Data Integrity: It guarantees that the data has not been altered during the transmission of data from the sensor node to the BS. The integrity of the aggregated result at BS is checked by comparing the MAC aggregate received from the sensor network with the MAC geneated from the aggregation result obtained in the BS.

3.1.7. Data Freshness: It provides protection from replay attack. Data freshness is achieved using the changing encryption keys during each session.

4. Key Distribution in EPSDA

In a resource constrained nature of sensor nodes, the asymmetric cryptographic algorithms are unsuitable for WSN. It needs more resources than symmetric cryptographic algorithms. Here we describe the energy efficient and secure key

distribution in EPSDA. It is similar to the ESPDA [9] key distribution used in cluster topology.

During the manufacturing phase, each node is assigned with a common secret key (K), node specific key (K_i) and a unique ID_i.

K	K_i	ID_i

Figure 1. Keys Stored in a Node

The sink is assigned with a common secret key (K), a session key (K_s), ID and secret key (ID_i-K_i) pairs of all nodes in the network before deployment.

K	K_s	ID_i-K_i

Figure 2. Secret Information Stored in Sink

Whenever a node wants to send data to sink, it first requests the sink to generate session key (K_s), encrypt it using the common secret key (K) and broadcast to the network. Upon receiving the encrypted sesion key, each node decrypts it using the common secret key (K) it has and calculates the encryption key (K_{ei}) by XOR ing the node specific key with the session key it received from the sink. *i.e.*, $K_{ei} = K_i \oplus K_s$

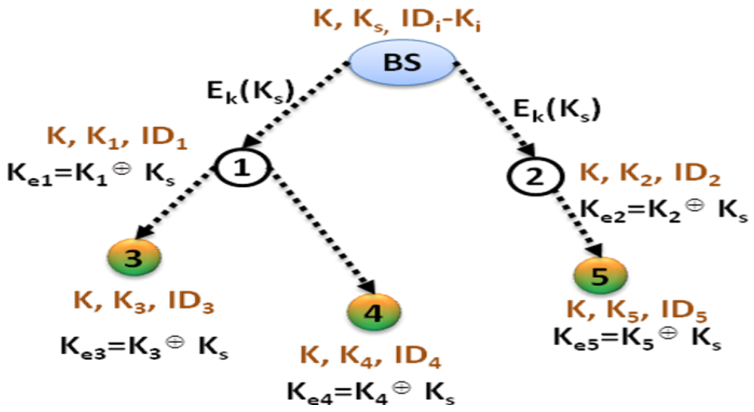

Figure 3. Key Distribution in EPSDA

The BS keeps the ID and secret key (ID_i-K_i) pairs of all nodes in the network, whenever the BS receives the aggregated encrypted result, it first determines the secret key (K_i) corresponding to the node ID's, generates the decryption key (K_{ei}) by XOR ing it with session key transmitted by sink to the network, thereby it eliminates the use of asymmetric cryptography for encryption and decryption. Confidentiality is achieved by encrypting the data with varying encryption key for each session, provides authentication and data freshness.

5. Energy Efficient Privacy Preserving Secure Data Aggregation

In this section, we describe the details of our proposed Energy efficient and Privacy preserving Secure Data Aggregation (EPSDA) scheme. There are five steps *i.e.*, aggregation tree construction, slicing, mixing, aggregation and verification.

5.1. Aggregation Tree Construction

The aggregation tree is routed at the BS. It is the join of all paths from the sensor node to BS. The aggregation tree is constructed using the TAG [18] protocol. Whenever a leaf node receives the request from the BS, it forward its sensed data to its parent within a single hop distance. Upon receiving the data from the child nodes, the aggregator performs aggregation on the received data from its child nodes and forward the aggregated result to the upper aggregator or sink for further processing.

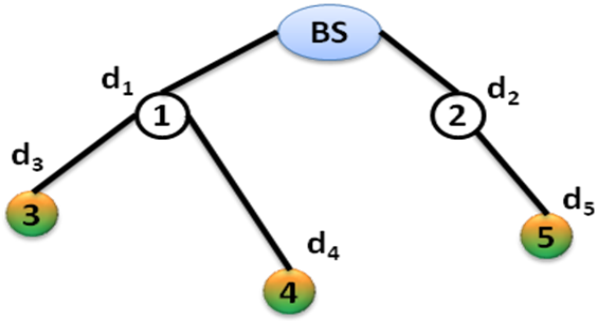

Figure 4. Aggregation Tree Construction

5.2. Slicing

This step is similar to the step 2 of EEHA scheme. Here the slicing operation is done on the leaf node for preserving the privacy of sensitive data. In order to check the integrity of the aggregated result at the sink, each node calculates the MAC of the sensed data using the encryption key of it and then kept in the node before slicing. During slicing, each leaf node slices its data into *m* number of pieces and encrypts all the *m* slices using the encryption key of the node. Among the *m* slices, one is kept in the node itself and the remaining *m-1* slices are appended with ID and send to *m-1* neighbor nodes within a single hop distance except the encrypted slices to its parent. The encrypted slice to its parent is appended with one of the encrypted slices kept in the node.

The encryption, aggregation and decryption process are done using the privacy homomorphism technique. The CMT [11] algorithm is used to achieve end to end encrypted data aggregation.

$$\text{Encryption } C_{ij} = (\sum_{i,j=1}^{N} (d_{ij} + K_{ei})) \bmod M$$

Where d_{ij} is the data slice from node i to node j and $d_{ij}=0$, when there is no data transfer between node i to node j. The K_{ei} represent the encryption key of node i.

$$\text{Aggregation } d_{iA} = (\sum_{i,j=1}^{N} C_{ij}) \bmod M$$

$$\text{Decryption } f_R = (\sum_{i,j=1}^{N} C_{ij} - \sum_{i=1}^{N} K_{ei}) \bmod M$$

Suppose, the M is smaller than the sum of all sensed data and the encryption key, then the sink fails to reproduce the real sum, instead it produces a smaller number than M, in order to avoid this problem take M as large $M=n*t$, where n is the total number of nodes and $t=max(d_i)$, *i.e.*, maximal sensed data.

Figure 5 shows the slicing step in EPSDA. The leaf node slices its data into *m* pieces (*m=3*) and encrypts all the *m* pieces using the encryption key of it. One of the encrypted slices is kept in the node itself and the remaining *m-1* slices are sent to *m-1* neighbor nodes except the slice to its parent.

Suppose, d_i represent the sensed data of the node *i* and d_{ij} represent the data slice sent from node *i* to node *j*. The d_{ii} represent the slices sent to itself. Then,

$$d_i = \sum_{i,j=1}^{N} d_{ij}$$

Where $d_{ij}=0$, when there is no data from node *i* to node *j*.

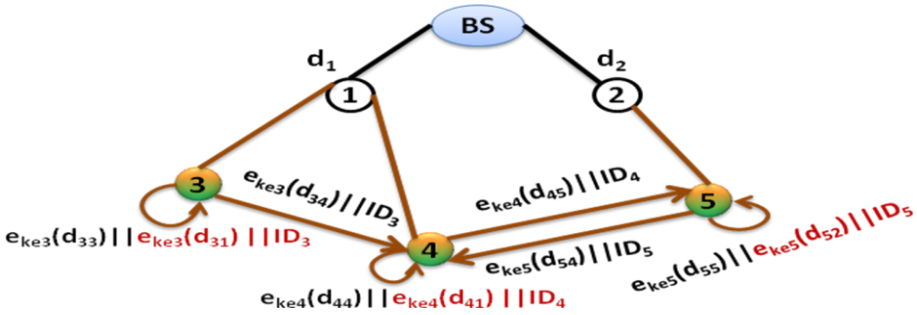

Figure 5. Slicing

5.3. Mixing

After the slicing operation, each node waits for a certain time to guarantee that all slices are received by corresponding node. During mixing, each leaf node sum up the encrypted slices received from other neighbor nodes with one of encrypted slice kept on the node during slicing.

One of the main disadvantages of privacy homomorphism technique is that, it can only be used for the SUM aggregation function. In order to overcome this, a count is appended with the transmission, at the sink node, it can use AVG aggregation function also.

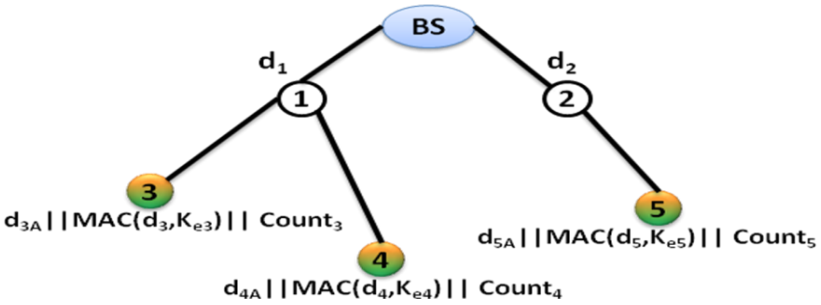

Figure 6. Mixing

Here,

$d_{3A}=E_{Ke3}(d_{33}) \| E_{Ke3}(d_{31}) \| ID_3$

$d_{4A}=E_{Ke4}(d_{44}) + E_{Ke3}(d_{34}) + E_{Ke5}(d_{54}) \| ID_3 \| ID_5 \| E_{Ke4}(d_{41}) \| ID_4$

$d_{5A}=E_{Ke5}(d_{55}) + E_{Ke4}(d_{45}) \| ID_4 \| E_{Ke5}(d_{52}) \| ID_5$

5.4. Aggregation

During aggregation, each leaf node sends the aggregated data and encrypted slice to its parent if any, the MAC of the original sensed data, a count with a value one and its node ID to its parent.

Aggregated encrypted data	Encrypted slice to its parent	MAC	Count

Figure 6. Packet During the Transmission of a Mixed Result from Leaf Node

Each intermediate node calculates the MAC of its sensed data using its encryption key and aggregate it with the MAC it received from the child nodes using privacy homomorphism technique. The CMT [11] approach is used for the aggregation of MAC.

$$MAC_{Agg} = (\sum_{i=1}^{N} MAC(d_i, K_{ei})) \bmod M$$

Each intermediate node then encrypts its sensed data and aggregate it with the aggregated data it received from the child nodes and also add if any encrypted slice appended in the aggregated result from the leaf node. It also sums up the count value it received from the child node. After that increment the count value by one and append the count value and ID to the partially aggregated encrypted data. Then send to the upper aggregator or sink.

Aggregated encrypted data	Aggregated MAC	Count

Figure 7. Packet During the Transmission of Aggregation Results from Intermediate Node

Each intermediate node takes more time than its child nodes, it calculate the time interval difference Δt and then calculate the time out t_i of each node. When t_i elapses, the partially aggregated result is sent to the upper aggregator. The aggregation result and MAC aggregates goes level by level until it reaches the BS. The BS keeps an ID and secret key of all the nodes in the network, it decrypts the received aggregated encrypted result using the sum of the decryption key generated by XOR ing the session key with the node specific secret key, identified from the ID attached to the received data.

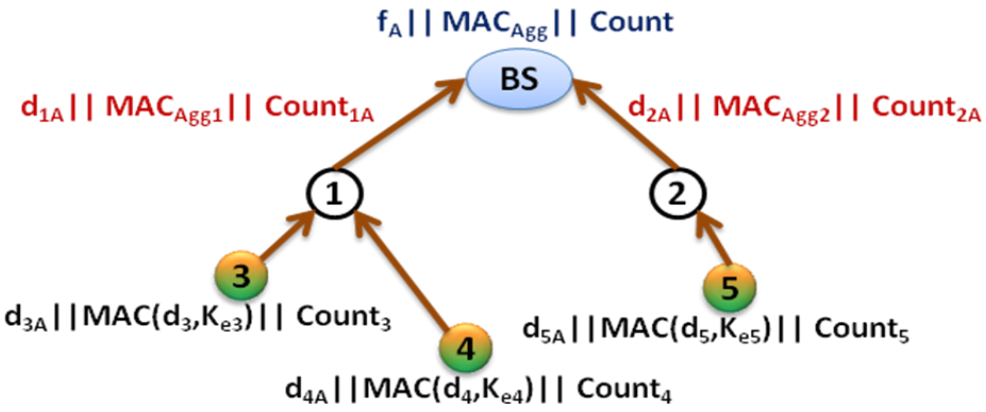

Figure 8. Aggregation

Here,

$d_{3A} = d_{3A} || ID_3$

$d_{4A} = d_{4A} || ID_4$

$d_{5A} = d_{5A} || ID_5$

$d_{1A} = d_{3A} + d_{4A} + E_{Ke1}(d_1) + E_{Ke3}(d_{31}) + E_{Ke4}(d_{41}) || ID_3 || ID_4 || ID_1$

$d_{2A} = d_{5A} + E_{Ke2}(d_2) + E_{Ke5}(d_{52}) || ID_5 || ID_2$

$f_A = d_{1A} + d_{2A}$

$Count_{1A} = Count_3 + Count_4 + 1$

$Count_{2A} = Count_5 + 1$

$Count = Count_{1A} + Count_{2A}$

$MAC_{Agg1} = MAC(d_3, K_{e3}) + MAC(d_4, K_{e4}) + MAC(d_1, K_{e1})$

$MAC_{Agg2} = MAC(d_5, K_{e5}) + MAC(d_2, K_{e2})$

$MAC_{Agg} = MAC_{Agg1} + MAC_{Agg2}$

5.5. Verification

After decrypting the aggregated encrypted result, the BS verifies the integrity of the aggregated result by generating the MAC of the aggregated result using the sum of all decryption keys generated by XOR ing the node specific key with the session key transmitted to the network. Then check the aggregated MAC received from the network with the MAC generated from the decrypted aggregated result. If too much difference is found, then the sink concluded that the modification occurred during aggregation. Then BS rejects the aggregation result.

Table 1. EPSDA Algorithm

1. Construct an aggregation tree using TAG protocol
2. The BS broadcasts the encrypted session key K_s by using the common secret key (K) in the network.
3. Upon receiving the encrypted session key, each node decrypts it using the common secret key K and then generate the encryption key by XOR ing the session key (K_s) with node specific secret key (K_i).
   i.e, $K_{ei} = K_i \oplus K_s$
4. Set the time interval difference Δt in an aggregation tree between the child node and its parent. Then each node computes its time out t_i .
5. **if** isLeafNode **then**
   1) Generate the MAC of the sensed data using the encryption key (K_{ei}).
   2) Set the count value to one for finding the number of nodes participated in aggregation.
   3) Slices its data into *m* number of pieces.
   4) Encrypt all the slices using the encryption key of it.
   5) One of the *m* encrypted slice is kept in the node itself and the remaining *m-1* encrypted slices are appended with ID and sent to *m-1* neighbor nodes except the encrypte slices to its parent.

6) The encrypted slice to its parent is appended with the encrypted slice kept in the node.
7) **if** t_i elapsed, **then**
a) Each leaf node performs mixing by summing all the encrypted slices to itself with the encrypted slice Kept in the node and then appends, if there is any encrypted slice to its parent is appended with the encrypted slice kept in the node and the ID of the node. Then transmit the packet.
Mixed result||encrypted slices to its parent with ID||MAC||Count||ID
8) **end if**
6. **end if**
7. **if** isIntermediateNode **then**
1) Generate the MAC of its sensed data and aggregate it with the MAC it received from its child node using privacy homomorphism technique.
2) Sum up the count value received from the child node. Then increment by one and set the count value.
3) Encrypt its sensed data using an encryption key of it.
4) **While** t_i not elapsed then **do**
a) Sum up all the aggregated data received from its child nodes with its own encrypted data and also add if any encrypted slices appended in the aggregated result from the leaf nodes.
5) **end while**
6) Append the partially aggregated result with its ID, MAC aggregate and count. Then send to upper aggregator or sink.
8. **end if**
9. **If** isSink **then**
1) Decrypt the aggregated encrypted data by using the sum of the decryption key generated by XOR ing the session key with a node specific key while seeing the node ID.
2) Generate the MAC on the decrypted aggregated result by using the sum of the decryption key generated by XOR ing the session key with node specific key while first seeing the ID of the node.
3) Compare the MAC generated on the aggregated result obtained at the sink with the MAC aggregate received from the network during data aggregation. If too much difference is obtained, then discard the aggregation result.
10. **end if**

6. Simulation Study and Performance Analysis

6.1. Simulation Settings

The EEHA and EPSDA schemes are implemented in ns_2. We considered a wireless sensor network with 15 sensor nodes, these are randomly deployed over an area of 1000 m × 500 m. One of the node is taken as the sink and the remaining nodes form a tree routed at the sink node. The parameters of the simulation are transmission power=0.660W, receiving power=0.395W, idle power=0.395W. Each node is assigned with 100J energy. We evaluated the communication overhead, computational overhead, accuracy, security and energy consumption of EPSDA with EEHA scheme.

6.2. Performance Analysis

6.2.1. Communication Overhead: In WSN's, the communication takes more energy than computation. Total number of communication occurred from each node during data aggregation is taken as the metric. Figure 9 shows the communication overhead of EPSDA and EEHA under different time intervals with *m=3*. The evaluation shows that bandwidth consumption of EEHA is more than EPSDA. In EEHA, the number of transmissions at leaf node is three, for achieving the privacy preserving data aggregation, two transmissions for slicing and one for data aggregation. Other nodes have only one transmission for data aggregation. But in EPSDA, the number of transmissions at leaf node is reduced by transmitting the encrypted slice to its parent during the transmission of an aggregation result from leaf node, other intermediate nodes have only one transmission for data aggregation. The reduced number of transmissions leads to the reduced communication overhead of EPSDA.

Consider the aggregation tree in Figure 3.4, take *m=3* (number of slices), assume that slicing start from left most leaf node and in the *m-1* neighbour nodes, one is considered as the parent node. Then,

Communication overhead in EEHA=(nl × 3)+(ni × 1)
Communication overhead in PEPPDA= (nl × 2) + (ni × 1)

Where,

nl→Number of leaf nodes
ni→Number of intermediate nodes

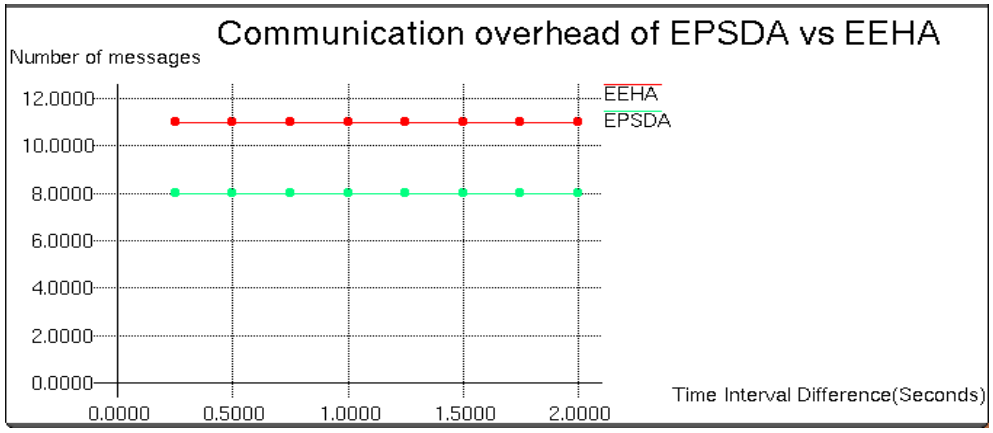

Figure 9. Communication overhead of EPSDA vs EEHA

6.2.2. Accuracy: The accuracy is defined as the ratio between the sum of sensed data reached at BS and the actual sum of sensed data. Figure 10 shows the accuracy of EPSDA and EEHA (*m=3*) with respect to the different time interval.

For a transmission with no collision, congestion and no error in the packet, accuracy is one. The accuracy is increased with increase in time interval. With longer time interval, fewer chances to collude with each other and get more chances to deliver within the time. In EPSDA, the number of data transmission is less than EEHA and therefore the chance of collision is reduced, thereby packet loss is reduced. Hence the chance of getting accurate results at sink is more. The other reason is the reduced number of decryption in the network,

reduces the frequency of node compromising attack. It reduces the modification of data during data aggregation, leads to the improvement of aggregation accuracy.

$$\text{Aggregation accuracy} = \frac{\text{Collected summation of data through aggregation}}{\text{Actual sum of data}}$$

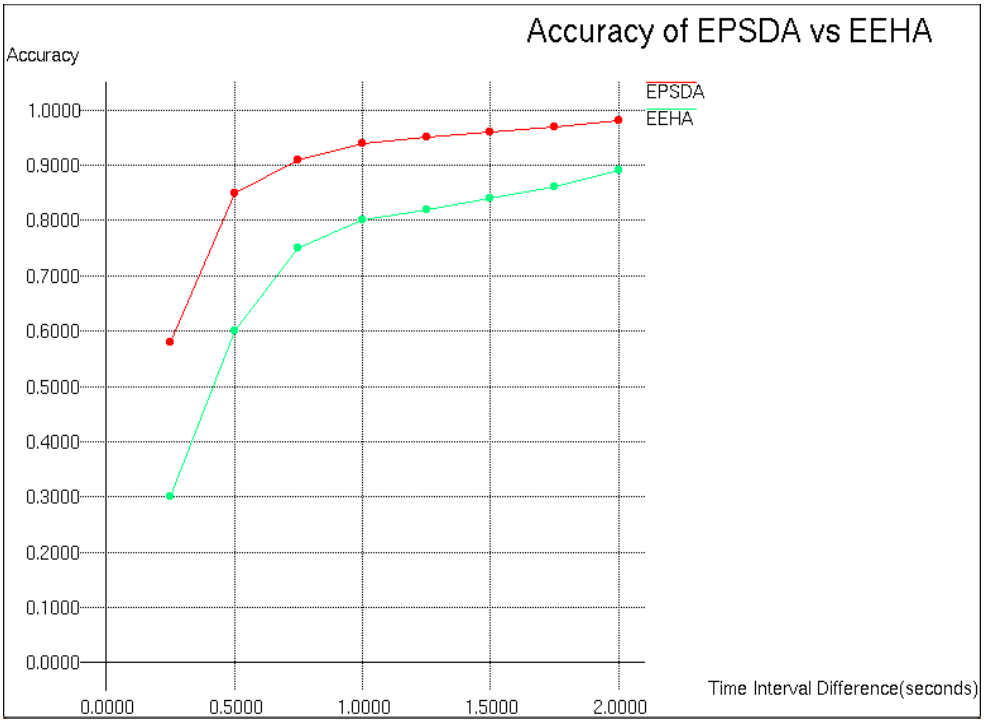

Figure 10. Accuracy of EPSDA vs EEHA

6.2.3. Computational Overhead: The computational overhead is defined as the overhead of a processor to perform power consuming arithmetic operations such as encryption and decryption. The EEHA scheme uses hop by hop encrypted data aggregation, the power consuming encryption and decryption done on each hop during the transmission of data from sensor node to the BS increases the computational overhead. This can be overcome in EPSDA by using the end to end encrypted data aggregation. It allows arithmetic operations on encrypted data, it needs only one decryption at sink node.

Figure 11 shows the computational overhead of EPSDA vs EEHA. Consider the aggregation tree in Figure 4, take $m=3$, in EEHA, each leaf node perform three encryptions and zero or more decryptions (zero number of decryption for the starting node of slicing and two decryptions for second last node). Each intermediate node performs one encryption and decryption operation. However in EPSDA, each leaf node performs three encryptions, and intermediate node performs one encryption operation. There is only one decryption operation at the sink node, by reducing the number of decryption operation, the computational overhead can minimize.

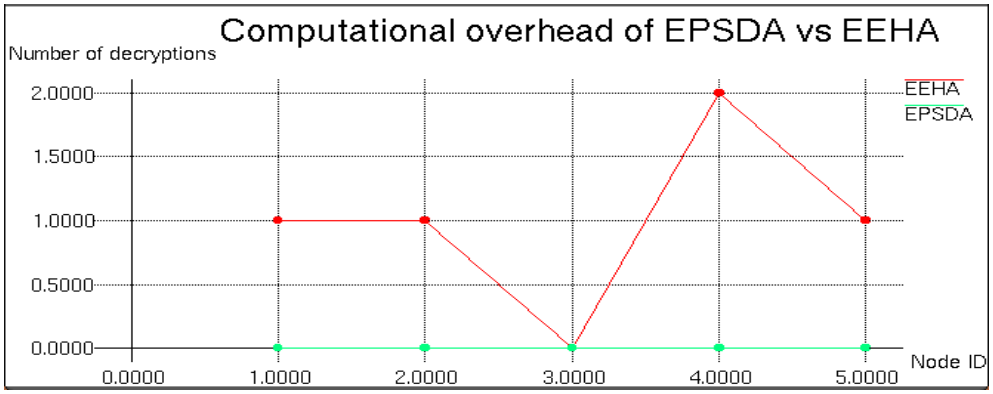

Figure 11. Computational Overhead of EPSDA vs EEHA

6.2.4. Energy Consumption: The Figure 12 shows the percentage of left energy of a node in both schemes. From the figure, we can conclude that the EEHA scheme consumes more energy than EPSDA, since the communication and computational overhead of EEHA is more than EPSDA. The decreased number of transmissions and the decreased delay of using end to end encrypted data aggregation leads to decreased communication overhead. The reduced number of power consuming decryption operation leads to reduce the computational overhead thereby improves the energy efficiency of EPSDA. Hence improves the network lifetime of battery limited sensors.

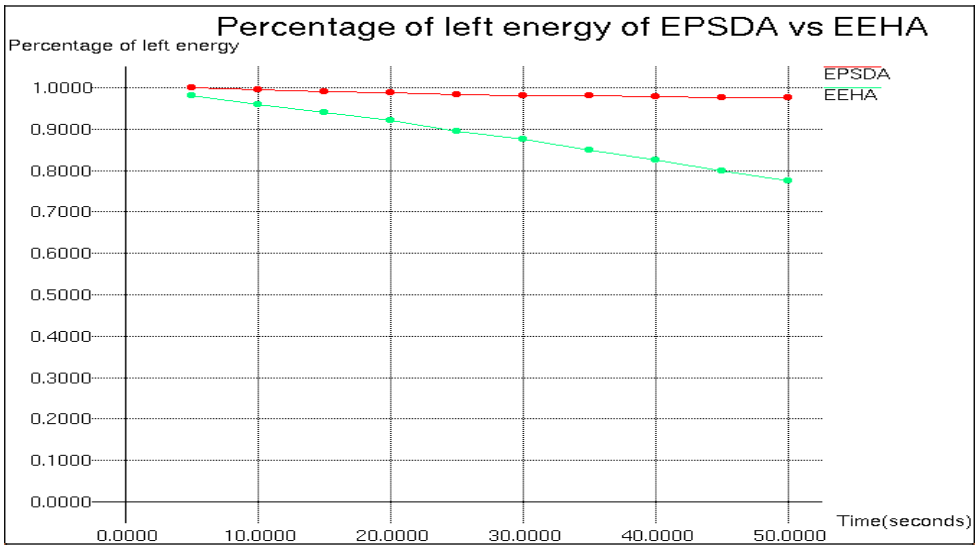

Figure 12. Energy Consumption of EPSDA vs EEHA

6.2.5. Security: The EPSDA scheme provides a secure data aggregation, which guarantees end to end confidentiality, data freshness, data privacy and message authentication, it is suitable for security critical application. The EEHA scheme only provides privacy preservation with high communication and computational overhead, it is more suitable for applications that have relatively loose requirements of security. The EPSDA scheme uses end to end encrypted data aggregation, it reduces the frequency of node compromise attack and

provides end to end confidentiality. Each node in the sensor network shares different key with the BS. Thus the data is protected from adversaries as well as trusted participated nodes. The EPSDA scheme provides a secure key distribution mechanism, which guarantees the data freshness. The encryption key is different for each session, by encrypting the data with the changing encryption key prevent the attackers from replying with old information. The EEHA uses hop by hop encrypted data aggregation, it looses end to end confidentiality and give more chance to adversaries for compromising. It increases the transmission delay, which in turn increase the chance to modify data. The authenticity mechanism of EPSDA avoids the intruder, ensures the integrity of the aggregated data using the MAC generated during data aggregation.

Table 1. Evaluation of EPSDA vs EEHA

Metrics	EEHA	EPSDA (Our scheme)
Performance Evaluation		
Communication Overhead	Low	High
Computational Overhead	Low	High
Energy Efficiency	Low	High
Network Lifetime	Low	High
Bandwidth Efficiency	Low	High
Chance of collision	More	Less
Aggregation accuracy	Low	High
Delay	More	Less
Security Evaluation		
End to end data confidentiality	No	Yes
Data Authentication	No	Yes
Data Privacy	Yes	Yes
Data Freshness	No	Yes
Data Integrity	No	Yes
Overall data security	Less	More

7. Conclusion

The proposed end to end encrypted privacy preserving data aggregation scheme (EPSDA) provides an energy efficient and secure solution for WSN's, by avoiding the power consuming decryption at the aggregator and the reduced number of transmission than existing hop by hop privacy preserving system. The privacy homomorphism based encryption algorithm is the basis of EPSDA scheme, which allows to perform aggregation on encrypted data, thereby reducing the frequency of node compromising attack. The EPSDA guarantees all the security requirements such as data privacy, message authentication, data freshness, data confidentiality, accuracy, data integrity with minimal communication and computational overhead. Thus the EPSDA scheme provides a time critical solution for security critical applications.

References

[1] V. Pandey and A. N. Chand, "A review on data aggregation techniques in wireless sensor network", Journal of Elecronics & Electrical Engineering, vol. 1, no. 2, **(2010)**, pp. 01-08.
[2] N. S. Patil and P. R. Patil, "Data Aggregation in wireless sensor network", IEEE International Conference on Computational Intelligence and Computing Research, **(2010)**.
[3] S. Peter, K. Piotrowski and P. Langendoerfer, "On Concealed Data Aggregation for Wireless Sensor Networks", Consumer Communications and Networking Conference, IEEE CCNC, **(2007)**.
[4] H. Li, K. Lin and K. Li, "Energy-efficient and high-accuracy secure data aggregation in wireless sensor networks", Computer Communication, vol. 34, **(2011)**, pp. 591-597.
[5] S. Peter, D. Westhoff and C. Castelluccia, "A Survey on the Encryption of Convergecast-Traffic with In-Network Processing", IEEE Transactions on Dependable and Secure Computing, vol. 7, **(2010)**.
[6] J. Jose, J. Jose and F. Jose, "A Survey on Secure Data Aggregation Protocols in Wireless Sensor Networks", International Journal of Computer Applications, vol. 55, no. 18, **(2012)**.
[7] H. Kargupta, Q. W. S. Datta and K. Sivakumar, "On the privacy preserving properties of random data perturbation techniques", Proceedings of the IEEE International Conference on Data Mining, Melbourne, FL, USA, **(2003)** November 19-22, pp. 99-106.
[8] W. He, X. Liu, H. Nguyen, K. Nahrstedt and T. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks", Proceedings of 26th IEEE International Conference on Computer Communications (Infocom 2007), Anchorage, Alaska, USA, **(2007)** May, pp. 2045-2053.
[9] H. Cam, S. Ozdemir, P. Nair, D. Muthuavinashiappan and H. Ozgur Sanli, "Energy-Efficient Secure Pattern Based Data Aggregation for Wireless Sensor Networks", **(2009)**.
[10] J. Gyro, D. Westhoff and M. Schneider, "CDA: Concealed Data Aggregation for reverse multicast traffic in Wireless Sensor Networks", Proc.40th International Conference on Communications, IEEE ICC, **(2005)** May.
[11] C. Castelluccia, E. Mykletun and G. Tsudik, "Efficient aggregation of encrypted data in wireless sensor networks", Proceedings of the 2nd Annual International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, MobiQuitous, San Diego, CA, USA, **(2005)** July 17-21, pp. 109-117.
[12] C.-M. Chen, Y.-H. Lin, Y.-C. Lin, H.-M. Sun, "RCDA: RecoverableConcealed Data Aggregation for Data Integrity in Wireless Sensor Networks", IEEE Transactions on Parrallel and Distributed Systems, vol. 23, no. 4, **(2012)** April.
[13] J. Domingo-Ferrer, "A provably secure additive and multiplicative privacy homomorphism", Proceedings of the 5th International Conference on Information Security, Sao Paulo, Brazil, **(2002)** September 30-October 2, pp. 471-483.
[14] G. Taban and V. D. Gligor, "Privacy-preserving integrity-assured data aggregation in sensor networks", Proceeding of International Symposium on Secure Computing, SecureCom, Vancouver, Canada, **(2009)** August 29-31, pp. 168-175.
[15] R. Agrawal, J. Kiernan, R. Srikant and Y. Xu, "Order preserving encryption for numeric data", Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data, Paris, France, **(2004)** June 13-18, pp. 563-574.
[16] H. Chan, A. Perrig and D. Song, "Secure hierarchical in-network aggregation in sensor networks", Proceedings of the 13th ACM Conference on Computer and Communications Security, Alexandria, VA, USA, pp. 278-287, **(2006)** October 30-November 3.
[17] W. He, X. Liu, H. Nguyen, K. Nahrstedt and T. Abdelzaher, "iPDA: An Integrity –Protecting Private Data Aggregation Scheme for Wireless Sensor Networks", IEEE MILCOM, **(2008)** November, pp. 1-7.
[18] S. Madden, M. J. Franklin and J. M. Hellerstein, "TAG: A Tiny Aggregation service for Ad-Hoc Sensor Networks", OSDI, **(2002)**.

Authors

Joyce Jose received the Bachelor of Technology (B Tech) degree in Computer Science & Engineering from Viswajyothi College of Engineering & Technology, Vazhakulam, Kerala, India in the year 2011, and Master of Technology (M Tech) in Network and Internet Engineering from karunya University, Coimbatore, Tamilnadu, India in 2013. She has published 2 International Journals and 2 International IEEE Conferences papers. Her research interest is in the area of privacy preserving data aggregation in Wireless Sensor Networks.

M. Princy pursued her M Tech degree specialized in Network and Internet Engineering, in the year 2011, her research interest is in the area of enhancing power efficiency in Sensor Networks ,also has presented papers in various national and international conferences,published a papers in International Journal. Her project is on sleep scheduling algorithm for power efficiency in Sensor Networks and guiding projects in Power efficient aggregation and routing in Sensor Networks and planned to pursue her doctorate in the same.

Josna Jose received the Bachelor of Technology (B Tech) degree in Computer Science & Engineering from Viswajyothi College of Engineering & Technology, Vazhakulam, Kerala, India in the year 2011, and Master of Technology (M Tech) in Network and Internet Engineering from karunya University, Coimbatore, Tamilnadu, India in 2013. She has published 2 International Journals and 2 International IEEE Conferences papers. Her research interest is in the area of secure data aggregation in Wireless Sensor Networks.