SDACQ: Secure Data Aggregation for Coexisting Queries in Wireless Sensor Networks

E G Prathima[†], Venugopal K R[†], S S Iyengar[††] and L M Patnaik[†††]

[†]University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India
[††]Florida International University, USA
[†††] National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India

Summary
Wireless Sensor Network consists of sensor nodes that are constrained in energy and other resources and is vulnerable to security attacks since the inherent nature of communication is broadcast. In order to reduce the energy consumption it is necessary to optimize the number of packets transmitted. In addition the data has to be encrypted to withstand security attacks. We propose Secure Data Aggregation for Coexisting Queries (SDACQ) in Wireless Sensor Networks that allows parallel coexisting aggregate queries from the sink to be disseminated in an authenticated manner and aggregate the data belonging to coexisting queries into a single packet. The cluster heads aggregate the encrypted data from sensor nodes using additively homomorphic encryption. Thus SDACQ provides secure data aggregation by combining authenticated query propagation with homomorphic encryption at low energy consumption. Simulation results shows that SDACQ provides better performance than other state of the art algorithms.

Key words:
Coexisting queries, Data Aggregation, Homomorphic encryption, Sink authentication WSN.

1. Introduction

WSNs comprise of a number of sensor nodes that senses its environment and communicates its data to the base station using multihop communication. The Sensor nodes are severely constrained in battery power, computation and communication capacity and memory. Data aggregation significantly reduces the number of messages transmitted and thereby reduces the overall energy consumption.

Data aggregation can be either centralized where the base station (sink) performs aggregation or in-network where the intermediary nodes perform aggregation. The aggregation can be based on cluster, tree or multi-path topology. In clustered aggregation, sensor nodes are grouped into clusters with one cluster head per cluster and the cluster-head performs aggregation. In the tree based aggregation, a minimum spanning tree rooted at sink is constructed where all the non-leaf nodes perform aggregation. In multi-path aggregation every node has more than one parent in the aggregation hierarchy and all the intermediary nodes perform aggregation.

The inherent nature of communication in any WSN is broadcast and it is usually deployed in open environment. Hence the sensor nodes are vulnerable to various types of attacks. An opponent can easily gain control over aggregator node to manipulate the aggregated result or gain data of all the sensor nodes under the compromised aggregator. Thus Securing data gathering and aggregation in an energy efficient manner is a primary concern in sensor networks. The concept of Concealed Data Aggregation (CDA) was introduced in which each sensor node transmits the encrypted data and the aggregator node performs additive or multiplicative operation on the encrypted data. Thus, even if cluster heads or aggregator nodes are compromised, it cannot manipulate the aggregated result. All CDA techniques use privacy homomorphism. In [1] Concealed Data Aggregation Scheme for Multiple Applications in WSNs (CDAMA) is proposed to aggregate data from multiple applications into a single packet but incurs higher communication and computation overhead.

Data aggregation can be categorized as push-based and pull-based. In push-based aggregation, the sensors push the aggregated data to the sink on detection of an event. On the other hand in pull-based aggregation the users of the WSNs issue queries through a special gateway node called the base station or the Sink. The sink node injects queries into the network. Each sensor node that has data to satisfy the query sends its data to its designated aggregator and it is forwarded to another aggregator or sink. Many algorithms are proposed for query optimization, routing and processing. The algorithms proposed in [2]–[9] handle single aggregate query. Some applications require multiple aggregate queries to be processed simultaneously. Algorithms [10]–[13] are designed to handle multiple aggregate queries in sensor networks by removing replicated data that is common to different queries.

Contribution: SDACQ integrate additively homomorphic encryption with multi-query processing where data

Manuscript received April 5, 2017
Manuscript revised April 20, 2017

belonging to different coexisting queries are encrypted using Elliptic Curve Cryptography and then the ciphertexts are aggregated by point addition. The contributions of the paper include:

1) Authenticated query dissemination and data aggregation

2) Secure Aggregation of data of different queries into a single packet using additively homomorphic encryption.

3) Low energy consumption and enhanced lifetime of WSNs.

Organization: This paper is organized as follows: Section II presents a literature survey. Section III describes the preliminaries. Section IV defines the problem and describes the system model. Section V presents SDACQ. Section VI discusses the simulation results and performance analysis. Section VII contains the conclusions.

2. Literature Review

2.1 Data Aggregation for Multiple Coexisting Queries:

Different algorithms were proposed to process, optimize and disseminate multiple coexisting queries. Some algorithms preprocess queries [3], [4], [11], [12], before disseminating it into the network and then merge the results of sub-queries to before handing over the result to the corresponding user. The aggregation query is then routed to the sensor network using power cost incurred [5] or by dividing the query region into cells [6]. Some algorithms perform query processing by constructing separate routing tree for each query [7]. Niedermayer et al., [8] addressed the problem of computing exact quantiles in hierarchical WSNs by employing a b-ary search. In [9], multi query optimization is being performed based on the shortest path, trust and energy efficient query processing. The data aggregation is classified into Single query data aggregation and inter-query data aggregation. The inter-query data aggregation saves energy as redundancy between queries increases [10]. Some algorithms perform query optimization in two tier. The tier-one happens at base station where redundant

queries are removed and and the second tier of optimization happens in-network where redundant data are removed [13]. Query and data encoding is used for securely processing multiple range queries [14].

2.2 Concealed Data Aggregation:

Concealed data aggregation techniques allow the execution of aggregate function on encrypted data. The CDA techniques use homomorphic encryption. Different CDA techniques are analyzed in [15], [17], [18]. A group signature generation algorithm is presented in [16] that generate verifiably encrypted signature. Different CDA techniques are presented in [19], [20], [21], [22] and [25]. In [23] an algorithm that can withstand passive and active attack is presented. The algorithm presented in [24] generates aggregated cipher-text, signature pair that allows the base station to obtain individual sensor's reading.

3. Background Work

Handling of multiple aggregate queries in Sensor Networks is a challenging task. Sensor Networks are vulnerable to security attacks due to the broadcast nature of communication. This section gives a brief insight into the two background works that lay a foundation for the development of SDACQ: 1) SafeQ and 2) CDAMA.

3.1 SafeQ

Chen et al. [2] proposed SafeQ protocol that allows the sink and sensor nodes to exchange data in privacy and integrity preserving manner in a two-tier sensor network architecture. The two-tier architecture uses mobile sinks as storage nodes between sensor nodes and the sink. In SafeQ both data and queries are encoded. The base station encodes the query and transmits the encoded query to the storage nodes. In a similar manner the sensor nodes encode their data collected over a period of time and send the encoded data to their respective storage nodes. SafeQ uses neighborhood chaining technique by which the Sink node is able to verify if the result of the query contains any false contribution. SafeQ is designed for answering range queries.

3.2. CDAMA: Concealed Data Aggregation Scheme for Multiple Applications in Wireless Sensor Networks

Lin *et al.* [1] proposed concealed data aggregation scheme for aggregating data from multiple applications into a single ciphertext. The base station extracts application specific data from the final aggregated ciphertext received. The sensor nodes are grouped into clusters. Each cluster may have sensor nodes running different applications. Each Sensor node encrypts the application specific data and sends to its respective cluster-head. The cluster-head aggregates the data without decrypting the ciphertexts recieved from its cluster members and transmits to the base station. The base station decrypts the application specific data.

4. Problem Definition and System Model

4.1 Problem Definition

The users of Wireless Sensor Networks normally are far away from the WSNs and interact through the Base Station (Gateway) by injecting queries into the Sensor Networks. Multiple users may inject queries into the network through sink. Normally, the Sink broadcasts query into the network, which is propagated throughout the network in a multi-hop communication mode. An attacker snooping into the network can get the desired information by just listening to the communication or can pretend as the Base Station and inject queries. Hence, the query must be transmitted in a secured manner.

There can be multiple sum based queries co-existing in the network. Sending the data related to each query in separate packets involves high communication overhead. As discussed under Section II and Section III, many algorithms exists for optimizing and processing aggregate multiple queries. Transmission of aggregated data related to multiple coexisting queries in separate packet may lead to larger energy depletion and hence reduce the overall network lifetime.

Motivation: Distributed Sensor Network has coexisting queries. If separate packets are transmitted for each query, the communication overhead increases, resulting in increase in overall energy consumption of the network. Securely aggregating data belonging to coexisting queries into a single packet is a challenging task.

4.2 System Model and Assumptions

This section discusses the network model and attack model and the assumptions made in designing the algorithm.

4.2.1 Network Model: The Wireless Sensor Network consists of N sensor nodes randomly deployed in the network. The data generated at sensor nodes are transmitted to the base station using multihop communication. The sensor network is assumed to be heterogeneous consisting of high capacity sensors and low capacity sensors. The high capacity sensors act as cluster-heads. The cluster heads perform aggregation and forward the aggregated data to either another cluster head nearer to base station or to the base station itself. Figure 1 shows a sample wireless sensor network with four clusters (represented by circles) each having a highend sensor acting as cluster-head and four or five low-end sensor nodes. The radius of the circles representing clusters is equal to the communication range of the clusterheads. The circle with the base station at the center is the top level of hierarchy. The cluster-heads CH1 and CH2, that falls within the communication range of the base station send their aggregated data directly to the base station. CH3 falls within the communication range of CH1 and hence chooses CH1 as its aggregator and CH4 can communicate only with CH3 and hence CH3 becomes aggregator for CH4. The clustering hierarchy is represented using dotted line.

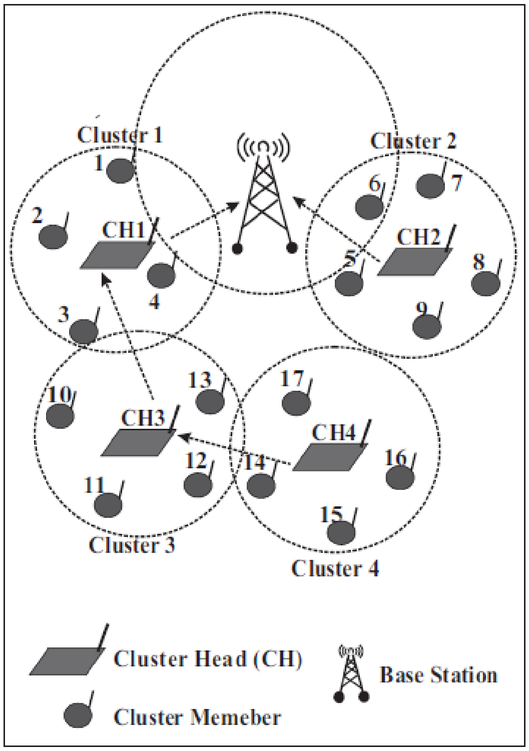


Fig. 1 Network deployment.

4.2.2 Query Model: Multiple sum based queries may coexist in the Sensor Network. Multiple sum based queries are assumed to coexist in the WSN. The following scenario shows an example for the query model:

- User1 issues the query ”SELECT SUM(TEMP) FROM SENSORS WHERE TEMP BETWEEN 70 AND 100”
- User2 issues the query ”SELECT AVG(TEMP) FROM SENSORS”
- SDACQUser3 issues the query “SELECT COUNT(TEMP) FROM SENSORS WHERE TEMP BETWEEN 50 AND 100”

The query model assumes that all the coexisting queries to be SUM based queries so that additively homomorphic encryption can be applied. Query model also assumes that different queries may expect different types of data.

4.2.3 Attack Model: An algorithm intended to provide security must provide Confidentiality, Integrity and Authenticity as the basic requirement. An attacker may launch various types of attacks to break confidentiality, integrity and authenticity.

4.2.3.1 Attacks against Confidentiality: An attacker tries to gain access to key by launching one of the following attacks such as known plaintext attack, chosen ciphertext attack and chosen plaintext attack. Once the attacker gains the key, the aggregated data can be decrypted.

4.2.3.2 Attack on Integrity: The attacker successfully compromises one or more aggregator nodes sensor nodes. The compromised node may either drop some data or may change the aggregated result with the intention of propagating false aggregate to the base station. eg: replay attack

4.2.3.3 Attack on Authenticity: There are two types of attacks that can form threat against authenticity; (i) The attacker pretends to be base station and injects query into the network (ii) The attacker pretends to be a genuine sensor node or aggregator and injects false data into the network.

We assume that a compromised sensor node may try to violate either integrity or authenticity of data. In public key based cryptosystems even if the attacker gains the encryption key, it cannot be used for decryption. But this key can be used to generate a false ciphertext in a valid format and thus violate integrity.

5. The SDACQ Algorithm

Goal: The main goal of the algorithm is to disseminate queries into the sensor network in an authenticated manner and aggregate data belonging to different queries into a single packet but still allowing the base station to retrieve the query specific aggregate. Table II contains the list of notations used in SDACQ algorithm.

5.1. Algorithm

The SDACQ algorithm has three parts out of which two are performed at the base station.

5.1.1 Query Generation and Dissemination: The user's query is transformed at the base station to a format that can be understood by the sensor nodes. Each query is uniquely identified by a query identifier.

Table 1: List of notations used

Notation	**Meaning**
Q_i	Query with identifier i
Q	The query message
SK_i, PK_i	Private and Public keys for query Q_i
O	Order of the Elliptic Curve
a_1, a_2, a_3...	Prime numbers
G_i	Generator for query Q_i
β	Maximum plaintext boundary
β_i	Plaintext boundary of query Q_i
η	Average number of sensors contributing to each query
γ	A point with order $\alpha + 1$
P_i	A point with order a_i
δ	Transmission delay incurred by the packet
C_T	Time when packet was created
R_T	Time of packet reception
τ	Delay threshold
R_1 and R_2	Random numbers between 0 to _-*1*
C_I , C_J	Ciphertext of node I, J respectively
K_I	Key of sensor node *I* shared with its cluster head for signature generation
K	Key generated during each data collection round for authenticating base station
ζ_I	Signature generated by a node I
ζ_{I-J}	Signature of sensor node I generated by node J
V_I	Reading of the sensor node *I*
H_{K_I} ()	Cryptographic hash function that generates signature using key K_I

The format of query message referred as Q is given below:

< Query, Predicate, Duration, Period >

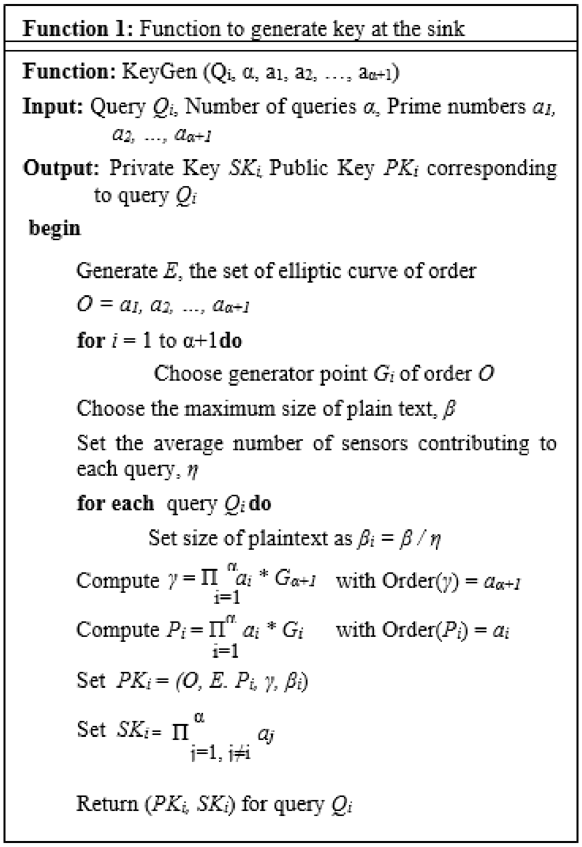
Where Query specifies the type of query (COUNT, SUM, AVERAGE, MEDIAN, STANDARD DEVIATION etc.), and Predicate specifies the query predicate. Duration is the duration of query i.e, how long the query lasts and Period tells the regular intervals at which data is expected at the base station. For example if the duration is 2 hours and period is 15 minutes, the sensor nodes transmit their sensed reading after every 15 minutes period to the base station for next 2 hours. Each query is uniquely identified using an identifier, Q_i. The base station generates the public key (PK_{Q_i}), private key (SK_{Q_i}) pair corresponding to each query Q_i as shown in Function 1.

Authenticated Query Dissemination: All the sensor nodes in the network and the Base Station share a common key (*K*). The Base Station first performs a re-keying operation as: $(K) = (K) \oplus TS$ where *TS* is the query generation time and ⊕ represents bitwise XOR operation. During each data collection round, the authentication key is regenerated. This ensures that an attacker cannot inject a false query into the network. Then the Base Station generates signature (ζ_{BS}) using the key (K) ,its ID and Q_i as shown below :

$$\zeta_{BS} = H_{K_A}(ID, Q_i)$$

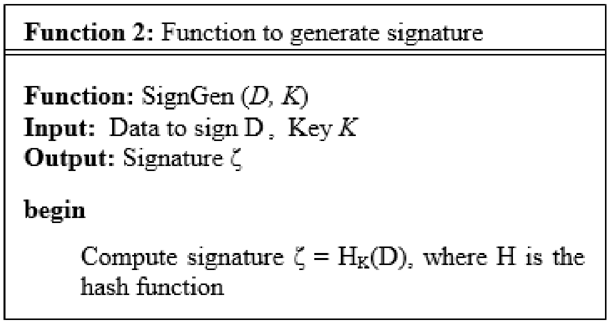
$H_{(K)}(ID,Q_i) = h((K) \parallel ID \parallel Q_i))$ where ∥ represents the concatenation operation. The signature is used to authenticate the query origin. Once the keys and signature are generated, the Base Station sends a query message consisting of the following fields:

$< Q_i, Q, PK_{Q_i}, \zeta_{BS}, TS >$



Function 1: Function to generate key at the sink

```
Function: KeyGen ($Q_i, \alpha, a_1, a_2, \ldots, a_{\alpha+1}$)
Input: Query $Q_i$, Number of queries $\alpha$, Prime numbers $a_1$,
        $a_2, \ldots, a_{\alpha+1}$
Output: Private Key $SK_i$, Public Key $PK_i$ corresponding
        to query $Q_i$
begin
    Generate $E$, the set of elliptic curve of order
    $O = a_1, a_2, \ldots, a_{\alpha+1}$
    for $i$ = 1 to $\alpha$+1do
        Choose generator point $G_i$ of order $O$
    Choose the maximum size of plain text, $\beta$
    Set the average number of sensors contributing to
    each query, $\eta$
    for each query $Q_i$ do
        Set size of plaintext as $\beta_i = \beta / \eta$
    Compute $\gamma = \prod_{i=1}^{\alpha} a_i * G_{\alpha+1}$   with Order($\gamma$) = $a_{\alpha+1}$
    Compute $P_i = \prod_{i=1}^{\alpha} a_i * G_i$   with Order($P_i$) = $a_i$
    Set $PK_i = (O, E. P_i, \gamma, \beta_i)$
    Set $SK_i = \prod_{j=1, j \neq i}^{\alpha} a_j$
    Return ($PK_i$, $SK_i$) for query $Q_i$
```



Function 2: Function to generate signature

```
Function: SignGen ($D$, $K$)
Input: Data to sign D , Key $K$
Output: Signature $\zeta$
begin
    Compute signature $\zeta = H_K(D)$, where H is the
    hash function
```

After successful verification, the cluster-heads (aggregators) rebroadcast the query. Only the cluster-head performs the verification and hence a non cluster-head node waits for its intended cluster-head to rebroadcast the query even if it has received the query from other cluster-heads or the base station. This process is repeated until all the cluster-heads in lowest level of hierarchy receives the query.

5.1.2 Data Generation and Aggregation: On receipt of query from respective cluster-heads, each sensor node sets a timer inversely proportional to its level in the aggregation hierarchy for data transmission. The node first generates its reading and checks whether it has data corresponding to the query to contribute. If a sensor node say A, has data to contribute for this query Qi, it encrypts the data using public key generated at base station for QID, viz., PKQID using elliptic-curve encryption. The procedure for generating ciphertext of A, CA is given in Function 2.

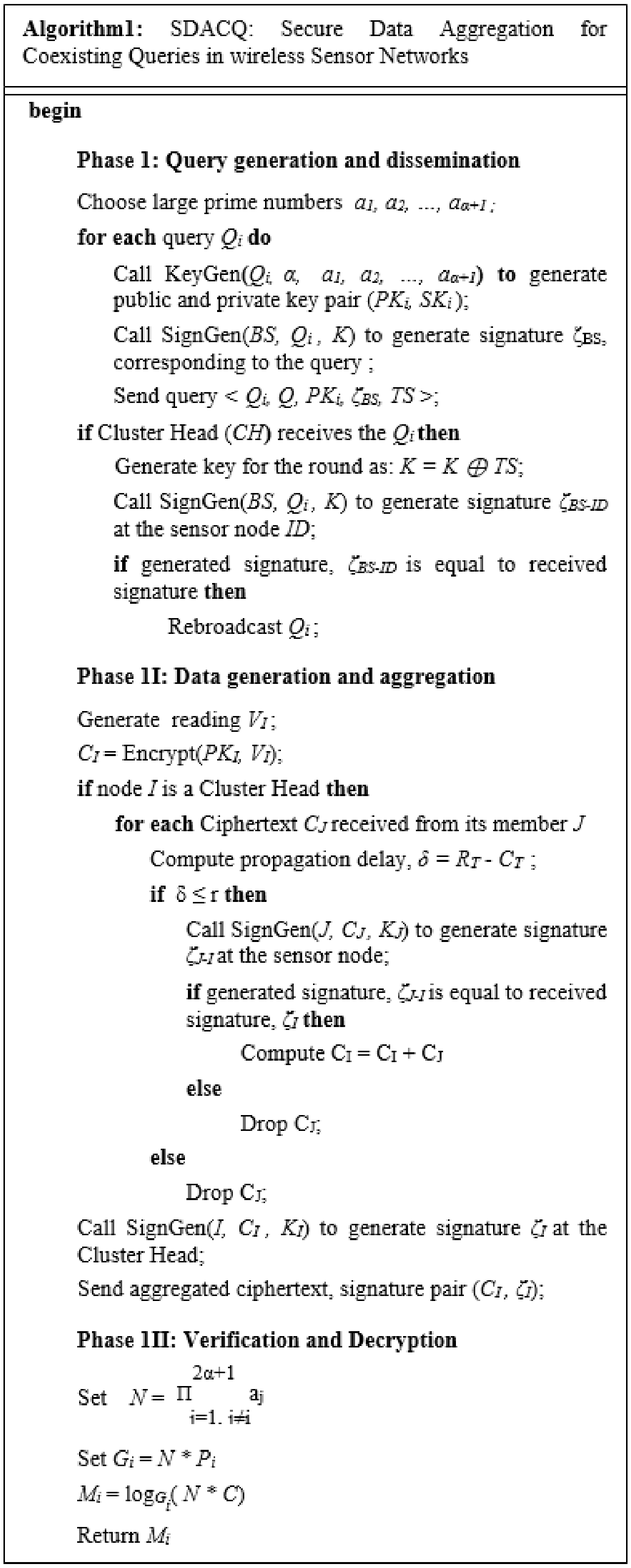
If a node has data corresponding to multiple coexisting aggregate queries, it encrypts the same data multiple times using the public key of the respective query. In that case even if it is non-aggregator node, aggregates the independent ciphertexts generated, corresponding to multiple coexisting queries.

Function 3: Function to encrypt data

```
Function: Encrypt ($PK_i$, $V_j$)
Input: Public key $PK_i$ of Query $Q_i$, Data of sensor $j$,
       $V_j$
Output: Ciphertext $C_I$
begin
    if size of sensor data is between $0$ to $\beta_i$ then
        Select two random numbers $R_1$ and $R_2$
        between 0 to $\eta$-1
    Compute C = $PK_{i+\alpha} + R_1 * \gamma$
    Generate ciphertext $C_I = C + V_j * PK_i + R_2 * \gamma$
```

At the time of network establishment, a secure communication channel is established between each node and its respective aggregator using elliptic curve Diffie-Hellman key exchange. Each low-end sensor node A that is non-aggregator, generates a signature to authenticate itself and its ciphertext using the key (K_A) shared between itself and its cluster-head CH as shown in Function 2. The sensor node A then sends its ciphertext and signature together in a message to its respective cluster-head, say CH.

Once the data transmission timer expires, each sensor transmits its message consisting of (C_A, ζ_A). When a



```
Algorithm1: SDACQ: Secure Data Aggregation for Coexisting Queries in wireless Sensor Networks

begin
    Phase 1: Query generation and dissemination
    Choose large prime numbers $a_1, a_2, ..., a_{\alpha+1}$;
    for each query $Q_i$ do
        Call KeyGen($Q_i, \alpha, a_1, a_2, ..., a_{\alpha+1}$) to generate public and private key pair ($PK_i, SK_i$);
        Call SignGen($BS, Q_i, K$) to generate signature $\zeta_{BS}$, corresponding to the query ;
        Send query < $Q_i, Q, PK_i, \zeta_{BS}, TS$ >;
    if Cluster Head ($CH$) receives the $Q_i$ then
        Generate key for the round as: $K = K \oplus TS$;
        Call SignGen($BS, Q_i, K$) to generate signature $\zeta_{BS\text{-}ID}$ at the sensor node $ID$;
        if generated signature, $\zeta_{BS\text{-}ID}$ is equal to received signature then
            Rebroadcast $Q_i$ ;

    Phase 1I: Data generation and aggregation
    Generate reading $V_I$;
    $C_I$ = Encrypt($PK_I, V_I$);
    if node $I$ is a Cluster Head then
        for each Ciphertext $C_J$ received from its member $J$
            Compute propagation delay, $\delta = R_T - C_T$ ;
            if $\delta \leq r$ then
                Call SignGen($J, C_J, K_J$) to generate signature $\zeta_{J\text{-}I}$ at the sensor node;
                if generated signature, $\zeta_{J\text{-}I}$ is equal to received signature, $\zeta_J$ then
                    Compute $C_I = C_I + C_J$
                else
                    Drop $C_J$;
            else
                Drop $C_J$;
    Call SignGen($I, C_I, K_I$) to generate signature $\zeta_I$ at the Cluster Head;
    Send aggregated ciphertext, signature pair ($C_I, \zeta_I$);

    Phase 1II: Verification and Decryption
    Set $N = \prod_{i=1, i \neq i}^{2\alpha+1} a_j$
    Set $G_i = N * P_i$
    $M_i = \log_{G_i}(N * C)$
    Return $M_i$
```

cluster-head *CH* receives this message from a neighbour *A*, it first verifies the signature by generating signature and comparing it with the received signature. If verification is successful, the cluster-head aggregates the received ciphertext with its own as follows: $C_{CH} = C_{CH} + C_A$; Where C_{CH} is the aggregated Ciphertext of the Cluster head and C_A is the ciphertext recieved from its member A. When the data transmission timer of the cluster head *CH* expires, it generates a signature authenticating itself and its ciphertext and sends its ciphertext, signature pair (C_{CH}, ζ_{CH}) in a message.

5.1.3 Verification and Decryption: When base station receives the (C_{CH}, ζ_{CH}) message, it first verifies the signature as discussed before. Then it aggregates the received ciphertexts. Finally it decrypts the aggregated ciphertext using the private key corresponding to query by applying Pollard's λ method. The SDACQ algorithm is given in Algorithm 1. A portion of the work is published in [26].

5.2 Analysis of SDACQ

5.2.1 Confidentiality: SDACQ uses privacy homomorphic encryption to ensure confidentiality as discussed under Section III. Since privacy homomorphism is used, the intermediary sensors do not decrypt the data for aggregation. Hence, confidentiality of data is not compromised. The intermediary nodes perform aggregation on the encrypted data.

5.2.2 Integrity: In order to launch an attack to violate integrity, the attacker modifies the cipher-text by (i) dropping some ciphertexts received from its cluster members or by (ii) adding a valid ciphertext during aggregation.

Case (i): SDACQ cannot prevent a compromised node from dropping ciphertexts received from its cluster members during aggregation. Since the data is encrypted and aggregation is performed on encrypted data, the cluster members cannot verify if its contribution is aggregated or not by the cluster-head.

Case (ii): The cluster-head can add a format valid false cipher-text which is either created by it or received by its cluster members during one of the previous data collection while performing aggregation. A cluster head cannot add a ciphertext created in previous data collection round after signature generation, because SDACQ necessitates each node to send a signature authenticating itself and its ciphertext. The procedure for signature generation uses different keys for different rounds. So even if a sensor node has same reading in two different epochs, its signature need not be same and hence a replay attack launched cannot not be successful. A node can add a false ciphertext during aggregation and the attack may remain undetected.

5.2.3 Authenticity: There are two types of nodes to be authenticated, the base station and the sensor nodes that can be either a cluster member or cluster head.

5.2.3.1 Authenticating the Base Station: SDACQ uses signature to authenticate the query message being broadcast from the base station. No malicious node can pretend to be base station and inject a query in the network. The Cluster heads share a secret key with the base station and during each data collection round, a new key is generated from the previous one. Each cluster head verifies the signature in the received query to ensure that it is from base station and not from a compromised node. Suppose if the attacker gains the key and generates a signature pretending to be base station and injects a false query message into the network. In SDACQ, the base station performs a rekeying operation each time a query is transmitted and same rekeying operation is performed at cluster-heads during signature verification. Hence the attacker cannot generate a valid signature even if it is successful in capturing the key.

A compromised sensor node can launch a replay attack by retransmitting the entire query message that is generated in a previous time period. But the time of generation of query does not match with the clock time in sensor node. A cluster head discards a query that is received beyond a time threshold. The time threshold τ is function of number of hops the specific cluster head is away from Base Station. Hence launching a replay attack to initiate the data collection process cannot be successful.

5.2.3.2 Authenticating the non-Base Station node: SDACQ necessitates all nodes in the network to be authenticated. All members of cluster share a key with their respective cluster-heads as discussed in discussion of SDACQ. Members of cluster sign their message using the key. Whenever, the cluster-head receives data from its cluster members, it first verifies the signature. Suppose a malicious node tries to inject false data then it must change its own reading to inject false data. A genuine ciphertext is be generated corresponding to the false reading. It is difficult to detect such attacks.

6. Results and Analysis

Simulations are performed on NS2 simulator. The metrics considered for comparison are 1) delay, 2) energy consumption and 3) packet drop ratio with respect to network size, number of simultaneous queries and % of compromised nodes. To compare the performance of SDACQ two state-of-the-art algorithms, Lin et al.'s CDAMA [1] and Chen et al.'s SafeQ [2] are implemented. For attaining uniformity in simulation while implementing SafeQ, the data centers are organized in multiple levels of hierarchy. So when query is issued from the base station, it is transmitted to all data centers using multi-hop communication between data centers. The sensor nodes send their encrypted data only to the data centers. Each data center collects data from ten sensor nodes.

6.1 Impact of Network Size on Delay

The network size is varied from 100 nodes to 500 nodes and the variation in delay is analysed. The graph in Figure 2 shows the impact of network size on delay. It can be seen that SafeQ [2]incurs least delay due to the presence of storage nodes that sent result of a query to the base station directly. SDACQ incurs slightly higher delay than. CDAMA due to the signature verification performed at all cluster heads.

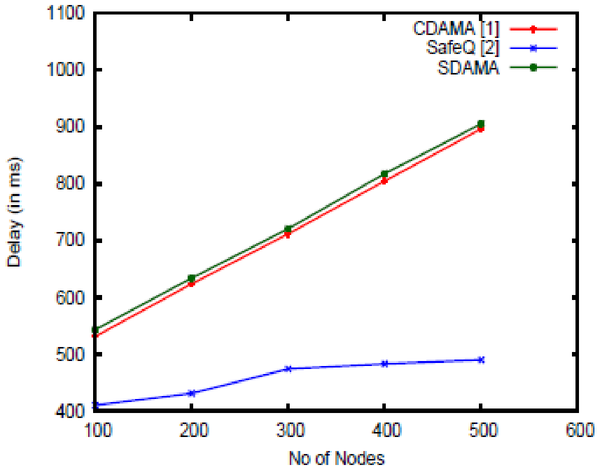


Fig. 2 Delay vs. Network size

6.2 Impact of Network Size on Overall Energy Consumption

To analyse the impact of network size on total energy consumption, the network size is varied from 100 to 500 nodes with total number of parallel queries set to 2. Figure 3 shows a comparison of the performance of SafeQ [2] and SDACQ. The highest energy consuming task of a sensor node is communication. For example, let three different sum-based queries are issued from sink. If the queries are non-overlapping and have no common data, any query optimization technique processes it as independent aggregate queries. Different data packets are transmitted to

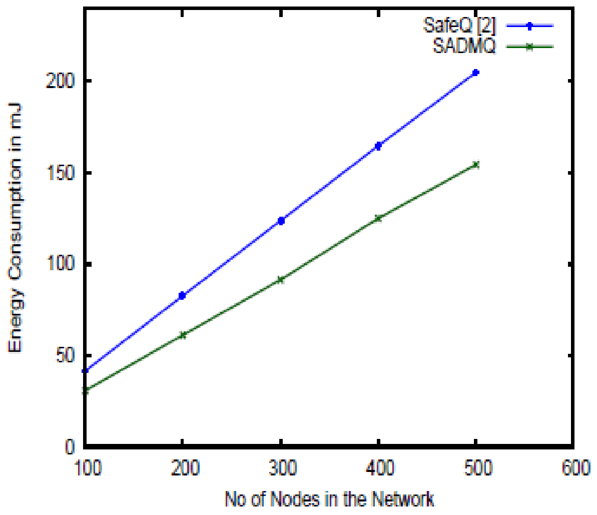


Fig. 3 Average energy consumed vs. Network size

the Sink corresponding to each query. Hence in SafeQ, for each query separate messages will be sent to the base station from the storage node. But in SDACQ the data belonging to different independent queries are aggregated into single packet and hence results in low communication overhead and better energy saving.

6.3. Impact of Number of Queries on Energy Consumption

To analyse the impact of number of queries on the average energy consumption per node, the number of simultaneous queries was varied from 1 to 5, by using a network consisting of 300 nodes. The comparison of both the algorithms is depicted in Figure 4. It can be seen that the SDACQ with aggregation of data from different queries can significantly reduce the energy consumption when compared with SafeQ that transmit data belonging to individual simultaneous query in separate packets.

6.4. Impact of Attack on Packet Delivery Ratio

The resilience of the algorithm against Sink impersonation and false query injection attacks is shown in Figure 5. In CDAMA an aggregator fuses all data received irrespective of whether it is unauthenticated or old data and hence the aggregated cipher text may include false contribution. Similarly in SafeQ, though the query is encoded, it cannot detect a replay attack and hence transmits the result back to sink. On the other hand, SDACQ is able to detect the false query injection and replay attack. It drops all such packets that are not received within time threshold or does not pass signature verification. We can see that the packet drop ratio increases with increase in % of compromised nodes. This shows that SDACQ provides more resistance to replay attack and false data injection attack. A compromised node cannot launch denial of service attack in SDACQ.

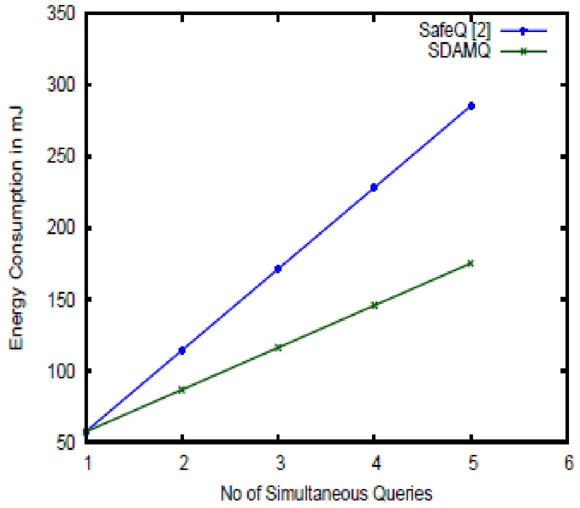


Fig. 4 Energy consumed vs. Number of simultaneous queries

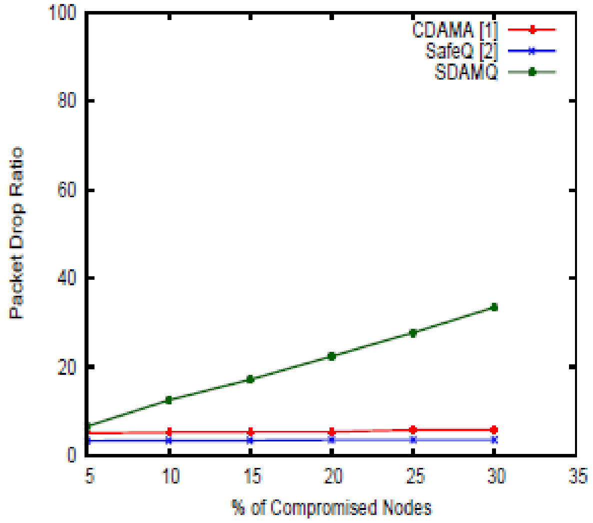


Fig. 5 Packet delivery ratio vs. % of compromised nodes

7. Conclusions

This paper proposes SDACQ: Secured Data Aggregation for Coexisting Queries in Wireless Sensor Networks that integrates multi-query aggregation with additively homomorphic encryption. Most of the multi-query aggregation techniques perform aggregation by eliminating redundant data common to multiple queries. SDACQ aggregates data belonging to different queries that can be of different type into a single packet without losing its identity and thereby reducing the overall energy consumption. SDACQ performs authenticated query dissemination by which no false query is injected into the network. The performance analysis shows that SDACQ identifies replay attack and does not aggregate malicious contributions. SDACQ authenticates all sensor nodes and hence incurs a small delay.

References

[1] Y.-H. Lin, S.-Y. Chang, and H.-M. Sun, “CDAMA: Concealed Data Aggregation Scheme for Multiple Applications in Wireless Sensor Networks,” IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 7, pp. 1471–1483, 2013.

[2] F. Chen and A. X. Liu, “SafeQ: Secure and Efficient Query Processing in Sensor Networks,” IEEE INFOCOM, pp. 1–9, 2010.

[3] G. Jiang and G. Cybenko, “Query Routing Optimization in Sensor Communication Networks,” 41st IEEE Conference on Decision and Control, vol. 2, pp. 1999–2004, 2002.

[4] M. Meng, J. Yang, H. Xu, B.-S. Jeong, Y.-K. Lee, S. Lee, and X. Fan, “Query Aggregation in Wireless Sensor Networks,” International Journal of Multimedia and Ubiquitous Engineering, vol. 3, no. 1, pp. 19–26, 2008.

[5] C.-H. Tsai, T.-W. Hsu, M.-S. Pan, and Y.-C. Tseng, “Cross-Layer, Energy-Efficient Design for Supporting Continuous Queries in Wireless Sensor Networks: A Quorum-Based Approach,” Wireless personal communications, vol. 51, no. 3, pp. 411–426, 2009.

[6] J.-J. Kim, I.-S. Shin, Y.-S. Zhang, D.-O. Kim, and K.-J. Han, “Aggregate Queries in Wireless Sensor Networks,” International Journal of Distributed Sensor Networks, 2012.

[7] S. G. Kim and H. S. Park, “Energy-Efficient Dynamic Query Routing Tree Algorithm for Wireless Sensor Networks,” Energy, vol. 3, no. 2, 2012.

[8] J. Niedermayer, M. A. Nascimento, M. Renz, P. Kr¨oger, and H.- P. Kriegel, “Continuous Quantile Query Processing in Wireless Sensor Networks,” 17th International Conference on Extending Database Technology (EDBT), pp. 247–258, 2014.

[9] K. Chakravarthi and V. Bhushan, “Modern Query Optimization Technique (Nature Inspired) for Improving Energy Efficient Data Gathering and Processing in Wireless Sensor Networks,” International Journal of Computer Applications, vol. 132, no. 2, pp. 24–30, 2015.

[10] X. Zhang, X. Yu, and X. Chen, “Inter-Query Data Aggregation in Wireless Sensor Networks,” International Conference on Wireless Communications, Networking and Mobile Computing, vol. 2, pp. 930–933, 2005.

[11] K. C. Lee, W.-C. Lee, B. Zheng, and J. Winter, “Processing Multiple Aggregation Queries in Geo-Sensor Networks,” 11th International Conference on Database Systems for Advanced Applications (DASFAA), pp. 20–34, 2006.

[12] N. Trigoni, A. Guitton, and A. Skordylis, “Routing and Processing Multiple Aggregate Queries in Sensor Networks,” 4th International Conference on Embedded Networked Sensor Systems, pp. 391–392, 2006.

[13] S. Xiang, H. B. Lim, K.-L. Tan, and Y. Zhou, “Two-Tier Multiple Query Optimization for Sensor Networks,” 27th International Conference on Distributed Computing Systems (ICDCS), pp. 39–39, 2007.

[14] R. Tanuja, S. H. Manjula, K. R. Venugopal, and L. M. Patnaik, “Secure and privacy preserving data centric sensor networks with multi-query optimization,” International Journal of Research in Engineering and Technology, vol. 4, no. 1, pp. 247–254, 2015.

[15] S. Peter, K. Piotrowski, and P. Langendoerfer, “On Concealed Data Aggregation for Wireless Sensor Networks,” IEEE Consumer Communications and Networking Conference, 2007.

[16] D. Boneh, C. Gentry, B. Lynn, and H. Shacham, “Aggregate and Verifiably Encrypted Signatures from Bilinear Maps,” Advances in cryptology, EUROCRYPT 2003, pp. 416–432, 2003.

[17] C. Castelluccia, E.Mykletun, and G. Tsudik, “Efficient aggregation of encrypted data in wireless sensor networks,” The Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services (MobiQuitous), pp. 109–117, 2005.

[18] E. Mykletun, J. Girao, and D. Westhoff, “Public Key based Cryptoschemes for Data Concealment in Wireless Sensor Networks,” IEEE International Conference on Communications (ICC), vol. 5, pp. 2288–2295, 2006.

[19] D. Westhoff, J. Girao, and M. Acharya, “Concealed Data Aggregation for Reverse Multicast Traffic in Sensor Networks: Encryption, Key Distribution, and Routing Adaptation,” IEEE Transactions on Mobile Computing, vol. 5, no. 10, pp. 1417–1431, 2006.

[20] C.-F. Chan and C. Castelluccia, “On the Privacy of Concealed Data Aggregation,” Computer Security– ESORICS, pp. 390–405, 2007.

[21] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. Abdelzaher, “PDA: Privacy-Preserving Data Aggregation in Wireless Sensor Networks,” 26th IEEE International Conference on Computer Communications – INFOCOM, pp. 2045–2053, 2007.

[22] S.-I. Huang, S. Shieh, and J. Tygar, “Secure Encrypted-Data Aggregation for Wireless Sensor Networks,” Wireless Networks, vol. 16, no. 4, pp. 915–927, 2010.

[23] P. Yang, Z. Cao, X. Dong, T. Zia et al., “An Efficient Privacy Preserving Data Aggregation Scheme with Constant Communication Overheads for Wireless Sensor Networks,” IEEE Communications Letters, vol. 15, no. 11, pp. 1205–1207, 2011.

[24] C.-M. Chen, Y.-H. Lin, Y.-C. Lin, and H.-M. Sun, “RCDA: Recoverable Concealed Data Aggregation for Data Integrity in Wireless Sensor Networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 23, no. 4, pp. 727–734, 2012.

[25] K.-A. Shim and C.-M. Park, “A Secure Data Aggregation Scheme based on Appropriate Cryptographic Primitives in Heterogeneous Wireless Sensor Networks,” 2014.

[26] E. G. Prathima, Shiv Prakash T., Venugopal K. R., S. S. Iyengar and L. M. Patnaik, “SDAMQ: Secure Data Aggregation for Multiple Queries in Wireless Sensor

Networks," Twelfth International Conference on Communication Networks (ICCN), pp. 283-292, 2016.

E G Prathima is pursuing her Ph.D in the De partment of Computer Science and Engineering at University Visvesvaraya College of Engineering, Bangalore University, Bangalore, India. She obtained her ME in Computer Science and Engineering from Adhiyaman College of Engineering. Her research interest is in the area of Wireless Sensor Networks.

Venugopal K R is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. He obtained his He received his Masters degree in Computer Science and Automation from Indian Institute of Science Bangalore. He was awarded Ph.D in Economics from Bangalore University and Ph.D in Computer Science from Indian Institute of Technology, Madras. He has a distinguished academic career and has degrees in Electronics, Economics, Law, Business Finance, Public Relations, Communications, Industrial Relations, Computer Science and Journalism. He has authored and edited 70 books on Computer Science and Economics, which include Petrodollar and the World Economy, C Aptitude, Mastering C, Microprocessor Programming, Mastering C++ and Digital Circuits and Systems etc., He has filed 101 patents. During his three decades of service at UVCE he has over 600 research papers to his credit. His research interests include Computer Networks, Wireless Sensor Networks, Parallel and Distributed Systems, Digital Signal Processing and Data Mining. He is a Fellow of IEEE. He has received ACM distinguished educator award.

S S Iyengar is currently Ryder Professor, Florida International University, USA. He was Roy Paul Daniels Professor and Chairman of the Computer Science Department of Louisiana state University. He heads the Wireless Sensor Networks Laboratory and the Robotics Research Laboratory at USA. He has been involved with research in High Performance Algorithms, Data Structures, Sensor Fusion and Intelligent Systems, since receiving his Ph.D degree in 1974 from MSU, USA. He is Fellow of IEEE and ACM. He has directed over 40 Ph.D students and 100 post graduate students, many of whom are faculty of Major Universities worldwide or Scientists or Engineers at National Labs/Industries around the world. He has published more than 500 research papers and has authored/co-authored 6 books and edited 7 books. His books are published by John Wiley and Sons, CRC Press, Prentice Hall, Springer Verlang, IEEE Computer Society Press etc. One of his books titled Introduction to Parallel Algorithms has been translated to Chinese.

L M Patnaik is currently working as Adjunct Professor and INSA Senior Scientist, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India. He was a Vice Chancellor, Defense Institute of Advanced Technology, Pune, India and was a Professor since 1986 with the Department of CSA, Indian Institute of Science, Bangalore. During the past 35 years of his service at the Institute he has over 700 research publications in refereed International Journals and refereed International Conference Proceedings. He is a Fellow of all the four leading Science and Engineering Academies in India; Fellow of the IEEE and the Academy of Science for the Developing World. He has received twenty national and international awards; notable among them is the IEEE Technical Achievement Award for his significant contributions to High Performance Computing and Soft Computing. His areas of research interest have been Parallel and Distributed Computing, Mobile Computing, CAD, Soft Computing and Computational Neuroscience.