Secure Aggregation of Exact Sum Queries with Integrity Protection for Wireless Sensor Networks

Heta K. Desai
Department of Computer Engineering
S V National Institute of Technology
Surat,India.

Devesh C. Jinwala,Ph.D
Department of Computer Engineering
S V National Institute of Technology
Surat,India.

ABSTRACT
The in-network processing consists of data aggregation operations e.g. Summation, averaging, min-max value, variance etc. Data aggregation reduces the number of message transmitted to the Wireless Sensor Network(WSN) by calculating the sub aggregation results at the intermediate nodes. Furthermore the sensor nodes are deployed in open and unsafe environments, so the security of sensed and aggregated data is crucial. This situation necessitates the investigation of the Secure Data Aggregation (SDA) protocol. Many protocols have been proposed concerning finding answers for SUM queries in WSNs but most of them either offer only approximate answers for SUM queries or do not live up to all the security requirements. The focus of the research in this paper is to investigate a SDA protocol which satisfies all the security requirements viz. Confidentiality, Privacy, Authentication, Integrity and Freshness of the query result and also returns the exact answers for SUM queries (as well as their derivatives, e.g., COUNT, AVG, etc). The protocol applies homomorphic encryption that promises the privacy as well as confidentiality of data sent by sensor nodes with in-network aggregation. The protocol achieves integrity of sensed data by means of secret sharing scheme, message authentication code (MAC) and data diffusion methods. Proposed scheme satisfies all the essential security requirements for secure in-network aggregation. This scheme does not impose extra overhead in communication. Therefore, the scheme is best desirable for resource-restrain WSNs.

General Terms
Security in Wireless sensor network, Integrity protection for secure data aggregation, Data Aggregation

Keywords
Security, Wireless Sensor Networks, Secure Data Aggregation.

1. INTRODUCTION
Recent advances in embedded systems and wireless communication technologies have empowered the use of wireless sensor networks (WSN) in a large number of applications, e.g., military field surveillance, environment monitoring, etc. Sensor networks should apply the promise of ease for large-scale and real-time data processing in complex surroundings. A sensor node is constrained in terms of bandwidth for communication, reserved energy and capabilities for computation. Energy is a most important resource in WSN. In WSNs nodes has limited computation capabilities and tight resource capacities. The problem of energy efficiency needs to be undertaken at all degrees of the wireless sensor network to accomplish good length of network lifetime. It is encountered that transmitting one bit from one sensor to another might consume as much power as executing large number(thousands) of instructions [1]. Therefore, the way should be picked up to sustain the wireless sensor network lifetime to abridge the sensor energy consumption during transmission of data.

Quality of sensed data is important in terms of getting precise results for the queries disseminate to the sensor networks. One way to improve quality of data and fault tolerance is by data redundancy. But this will induce significant amount of energy consumption overhead as well as large number of collisions. One solution for getting quality data without high energy consumption is aggregating sensed data, with use of mathematical functions: such as SUM, MAX, MIN, AVG, etc. This will return only final aggregation results instead of many raw sensed data and so reduces the energy consume as well as collisions of packets. In-network data aggregation can significantly reduce the number of bytes transmitted, and consequently improve the energy efficiency and sustain the wireless network lifetime [2]. Intermediate nodes called aggregators will compute sub results for the query and sends them to upper level aggregator nodes and finally to the base station. Basestation will compute final aggregation result using sub aggregate results.

Many times sensor nodes are deployed in remote and hostile environments where there is a threat of injection of false information (packets), node capturing or forge aggregation values into a network. Hence security of data is very crucial for most of the sensor network applications, such as security monitoring, target tracking, etc..

For security critical applications, hop-by-hop communication has the drawback that the data must be decrypted and re-encrypted on every aggregator node which causes the security related operations to be implemented at each node. To secure the data aggregation in WSNs is significant if any adversarial attacks harm the network by compromising aggregator nodes. Many Secure data aggregation schemes have been proposed based on hop-by-hop data aggregation and end-to-end data aggregation. But hop-by-hop aggregation protocols provide weaker data confidentiality than end-to-end aggregation protocols. Hence, we focus on end-to-end data aggregation in which decryption of data is performed only at basestation rather than at all aggregator nodes. Using Privacy homomorphism, we can achieve end-to-end security by performing aggregation operation directly on cipher text.

In this paper we introduce a protocol for Secure Data Aggregation (SDA) named Secure In-network Integrity Protected Aggregation of Sum Queries (SISQ) which will return quality results for SUM queries (as well as their

derivatives, e.g., COUNT, AVG, etc).This is end-to-end data aggregation protocol. We focus on maintaining integrity and authentication of data while providing freshness for the sensed data to detect the false data injection. Confidentiality of data is achieved by using additive homomorphic encryption scheme mentioned in the SIES [7] and CMT[17]. Hash-based Message Authentication Code (HMAC) and Message Authentication Code(MAC) with symmetric key encryption is used for providing integrity of data. In the proposed approach symmetric key encryption is applied because public key cryptography is not efficient on resource constrained nodes as it involve large number of keys shared between nodes [2]. Our approach achieves integrity of SUM result using secret sharing and data diffusion mechanisms which is thoroughly explained in section 4.

We considered tree based topology for SISQ. We moved forward by assuming that topology has already been established and each sensor node as well as aggregator node knows its parent aggregator node. At the beginning base-station distributes keys to sensor nodes. These keys are necessary for performing homomorphic encryption, calculating MAC and diffusion values. Sensor nodes concatenate sensed data with secret and diffusion seed, then perform homomorphic encryption and send it to parent aggregator node. Each sensor node transmits pair of encrypted values. Both values in the pair contain same sensor reading but they are diffused differently. The query emitted to the base station is for calculation of SUM of sensed data. Each sensor node also sends MAC generated with key known to it and its parent aggregator node which is used to verify integrity and authenticity of received data. At last when base station receives final pair of aggregated data it will decrypt data and retrieve SUM and secret values from it. Basestation is capable of calculating the secret share and diffusion value of each sensor node as it has distributed the parameters (keys used to generate the secret share) to generate secret share and diffusion values. Basestation compares the received secret with the calculated one, it also compares received result SUM from both aggregated values and if both matches then it verifies the authenticity and hence integrity of SUM results. This delayed authentication is useful to detect aggregator node capturing.

The rest of the paper is structured as follows. Section 2 describes related works in the area of secure data aggregation with integrity. Section 3 presents security requirements for SDA. Section 4 describes our final complete protocol SISQ. Security analysis of our protocol is described in section 5.In section 6 simulations results of our protocol under TinyOS are elaborated. Section 7 represents conclusion.

2. SECURE DATA AGGREGATION

In this section we give a short survey on former work done on secure data aggregation with security requirements integrity, authentication and confidentiality.

2.1 Protocols for Data Authentication and Integrity

In protocol proposed by authors of [8] Basestation (BS) can regain all sensing data even though these data has been aggregated. The authors named this property as "recoverable." The design has been extrapolated and adopted on both homogeneous and heterogeneous wireless sensor networks. It is based on end-to-end SDA security mechanism and composed of four phases: Setup, Encrypt-Sign, Aggregate and Verify. In setup phase, for each sensor s_{ni}, the BS generates pairs (ps_{ni}, rs_{ni}) based on Boneh et al.'s scheme. All keys are being inserted in sensor nodes and BS keeps its keys with itself. In encryption phase sensor node encrypts its data with the key provided to it. Sensor nodes also generates signature. Finally every sensor node generates ciphertext as a pair of encryption of sensed data and signature and sends it to their parent aggregator node. When parent aggregator node receives data from all child sensor nodes it performs aggregation operation on both cipher text and signature, and send the pair of aggregated data and summation of signature to the base station. Upon receiving pair base station can recover and verify each sensing data by performing some reverse manipulations. But the generation of this reverse function is quite complex and may introduce overhead to the whole verification phase.

In [18], authors have proposed secure data aggregation protocol which provides control integrity and which is based on a two-hop verification mechanism of data integrity. This scheme avoids referring to the basestation for data integrity verification procedure. It also adopts Hop-by-hop mechanism for data aggregation. In protocol each sensor node in WSN shares the key with two predecessor nodes (parent and grandparent). MAC is calculated using shared key. This way integrity of the data is preserved and any modification is detected by the grandparent which then instantly blocks the altered data at the compromised node. So it avoids transmitting forged data to the sink and hence prevents other data getting infected. However in case if two consecutive parent nodes have been compromised than this protocol will not give any sureness about aggregation results.

The authors of [9] have proposed Integrity preserving aggregation Protocol (E2IPAP) for tree-based sensor networks. It provides a new approach for result-checking and reduces communication overhead. It is based on hop-by-hop mechanism for SDA protocol is being divided into three phases viz. 1) Query dissemination phase, 2) Data aggregation phase and 3) Result-checking phase. In query dissemination phase Aggregation tree with BS at root is generated after query was being issued to the WSN. In data aggregation phase each sensor node sends data-commitment tuple containing sensor reading and count to its parent node in the aggregation tree. On receiving the tuple intermediate node performs aggregation function and sends the partial result to high-level aggregator node until the BS. In result checking phase BS broadcasts final aggregation result and final commitment to the network. Then each sensor node sends an authentication code back to BS after verifying that its data value as was added into the aggregation result, and the complement of its commitment was added into the aggregate commitment. If data value is different then it denies the aggregation result.

In [10], Esam Mlaih and Salah A. Aly have come with the approach which provide secure hop-by-hop aggregation of end-to-end concealed data in wireless sensor networks. The protocol is a blend of flexible data aggregation as in hop-by-hop Protocols and optimal data confidentiality as in end-to-end protocols. The main component of the proposed protocol is a process named data diffusion. It performs aggregation using diffusion function and generates MAC to assure integrity. In [10], the SIA: secure information aggregation in sensor networks is proposed. It uses Merkel hash tree for providing data authentication and so integrity. It is based on aggregate-commit-prove mechanism which means aggregators help computing aggregation of sensor nodes' raw data and reply to the home server with the aggregation result together with a commitment to the collection of data. The home server and the aggregators then perform efficient

Interactive proofs such that the home server will be able to verify the correctness of the results (or detect cheating with High probability).

Authors of [7], have proposed Secure and Efficient In-Network Processing of Exact SUM Queries (SIES). Which is aim to find SUM aggregation value from tree based structure. In this approach secret value (using the key shared with BS) is being generated by sensor node and it is merged with sensor reading in a single packet. Every aggregator node performs additive homomorphic encryption on encrypted data received from their child sensor nodes. When BS receives final aggregation value it decrypts packet and find aggregated result and sum of the secret generated by all sensor nodes. Basestation possess all the parameters to generate the secret (as it is the distributor of parameters) of all sensor node so. It generates secrets and finds the sum of secret and approves the integrity of received data by comparing it with received secret value. This approach proves integrity with delayed authentication. Our protocol is based on this approach but it improves upon it by providing integrity protection between neighboring nodes. In our protocol we have also used data diffusion mechanism to improve the probability of detecting forge aggregated results.

In literature many approaches [7, 11-17] for providing authentication and data integrity is proposed. Some provides perfect authentication but they either impose extra overhead for computation or provides delayed authentication. Many provide integrity protection with some assumptions with privacy homomorphism.

3. SECURITY REQUIREMENTS

Wireless sensor network is a peculiar type of network that demands security requirements of typical computer network along with its unique requirements suited solely to it. The key security properties for secure in-network aggregation in WSNs are report in this section.

3.1 Data Confidentiality

In wireless sensor networks, data confidentiality ensures that secrecy of sensed data is never disclosed to unauthorized parties and it is the most important issue in mission critical applications. Authors of [18] state that a sensor node should not leak its readings to neighboring nodes public sensor information, such as sensor identities and public keys, should also be encrypted to some extent to protect against traffic analysis attacks. The standard approach for keeping sensitive data secret is to encrypt the data with a secret key that only intended receivers possess, hence achieving confidentiality. Data aggregation protocols providing hop-by-hop SDA must decrypt the sensor data to perform data aggregation and encrypt the aggregated data before transmitting it [2]. This not only results in delay and energy consumption but also prevents end-to-end data confidentiality. Privacy homomorphism is the way to assure data confidentiality in end-to-end SDA. However, in hop-by-hop SDA protocols, imposing confidentiality becomes a challenging task.

3.2 Data Integrity

Employing confidentiality doesn't mean that data is safe. An adversary with high computational capabilities can alter the data. A malicious node or even due to unreliable communication channels, data may be altered with or without the presence of an intruder. Thus, message authentication codes or cyclic codes are used to prevent data integrity. Data aggregation itself results in alterations of data. Therefore it is not possible to have end-to-end integrity check when data aggregation is employed [4]. Moreover, if a data aggregator is compromised, then it may corrupt sensor data during data aggregation and the base station has no way of checking the integrity of this aggregated sensor data. Hence, data integrity preservation is attaining focus in building a fully secure data aggregation mechanism.

3.3 Authentication

With the data integrity, authentication ensures that the communicating node is the one that it claims to be. Due to shared wireless medium an adversary can not only modify data packets but also can change a packet stream by injecting fabricated packets [18]. Source authentication enables a sensor node to ensure the identity of the peer node it is communicating with. Without source authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes. A compromised node may send data to its data aggregator under several fake identities so that the integrity of the aggregated data is corrupted. Faking multiple sensor node identities is called Sybil attack and it poses significant threat to data aggregation protocols [4].

3.4 Data Freshness

Informally, data freshness suggests that the data is recent, and it ensures that no old messages have been replayed. Data freshness protects data aggregation schemes against replay attacks by ensuring that the transmitted data is recent [4].

3.5 Availability

This requirements ensures that the services of a WSN should be available always even in presence of an internal or external attacks such as a denial of service attack (dos). In addition to dos attacks, excessive communication or computation may exhaust battery charge of a sensor node. Wireless sensor networks are deployed with high node redundancy to tolerate such availability losses. Since data aggregators collect the data from number of sensor nodes and sends the aggregated data to the base station, availability of data aggregators is more important than regular sensor nodes [4].

4. PROPOSED PROTOCOL

4.1 Preliminaries

Privacy Homomorphism offers end-to-end concealment of data. A privacy homomorphism is an encryption function that allows direct computation on encrypted data. Suppose Enc denotes encryption and Dec denotes decryption. Also here let + refers to addition operation and x refers to multiplication operation, over the set S (set of sensor readings). If private key and public key of basestation are K_{pr} and K_{pu} , respectively.

An encryption operation is called additively homomorphic if,

$m_1+m_2 = d_{kpr} (e_{kpu} (m_1) + e_{kpu} (m_2))$; $m_1,m_2 \in S$

And it is called multiplicatively homomorphic if,

$m_1*m_2 = d_{kpr} (e_{kpu} (m_1) * e_{kpu} (m_2))$; $m_1,m_2 \in S$

Since, additive and multiplicative operations on encrypted data are supported by homomorphic cryptographic functions, aggregator nodes can perform addition and multiplication based data aggregation. Many homomorphic encryption operation are proposed viz. CMT[17], paillier, Elgamal, RSA. We are using additive homomorphic encryption schemes mentioned in [8] and [17].

The scheme is as follows, Assume that p is a prime, mi (i.e. message to be encrypted)< p, and K 0, $ki < p$ two secret keys. The encryption and decryption operations are defined as below,

(1) Encryption1: $ci = \mathcal{E}(mi,K, ki, p) = K \cdot mi + ki \bmod p$,
Encryption 2: $ci = \mathcal{E}(mi, ki, p) = mi + ki \bmod p$,

(2) Decryption1: $mi = \mathcal{D}(ci,K^{-1}, ki, p)$
$=(ci - ki) \cdot K^{-1} \bmod p$
Decryption1: $mi = \mathcal{D}(ci,ki, p) = (ci - ki) \bmod p$

Here K^{-1} is a multiplicative inverse of K modulo p. Here K^{-1} always exists since p is a prime. It can be observed that these encryption functions are secure in an information theoretic sense. We can say so because lacking knowledge about key k, the encrypted data preserves no information about plaintext data m.

Query Template

We considered exact SUM queries of the form:
"SELECT **SUM** FROM **Network** WHERE Time Epoch **T**"
To get value of COUNT as a SUM, in the protocol we can send the data from sensor as 1 if sensor reading is greater than zero. Also it is obvious that SUM and COUNT values can be combined to get answer of other queries, e.g., the average can be computed as AVG = SUM/COUNT. In the same way other queries like standard deviation and variance can also derived from SUM and COUNT values.

System Architecture

For simplicity we considered that sensors are being organized in to tree topology as depicted in Figure 1. The tree is rooted at a basestation. Leaf sensor nodes performs task of sensing. The intermediate nodes of tree are aggregator nodes.

Table 1. Notations used in protocol description

Notation	Meaning
N	Number of Sensor Nodes
S	Sensor node
BS	Basestation
Agg	Aggregator node
K_{all}	Key known to Bs and every source
K_c	Key known to every node in n WSN
$< K_{1i}, K_{2i} >$	Key pair known to BS and $\mathcal{S}i$
P	Public prime modulus
R	Random Modulus shared between BS and S_i
T	Time epoch of query
$Sec_{i,t}$	Secret share generated by SNi at epoch T
$SR_{i,t}$	I^{th} Sensor node reading at epoch T
$< D_{1i,t}, D_{2i,t} >$	Packets generated by i^{th} sensor node at epoch T
$<ED_{1i,t}, ED_{2i,t} >$	Encrypted pair of Data $D_{1i,t}, D_{2i,t}$
$< eag_1,eag_2 >$	Encrypted data pair at aggregator node
Sec_t	Secret verifiable by BS at epoch T
Dif_{sum}	Sum of all diffusion values generated by node S_i
SUM	SUM result at epoch T
MAC(D,K)	Message Authentication code for data D with key k
HMAC1(D,K)	Hash based MAC implemented with SHA-1

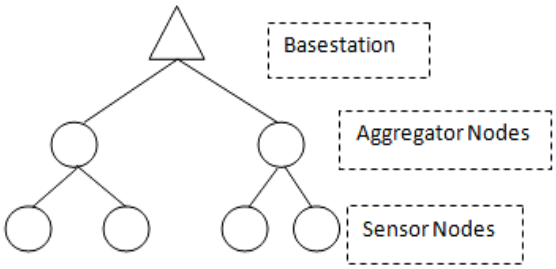

Fig 1: Tree topology for our protocol

Aggregator nodes receive encrypted data from child sensor nodes /aggregator nodes and perform aggregation operation on them and send them to the parent aggregator nodes. Basestation aggregates received data and decrypt them. Basestation accepts the SUM result only if integrity if data is not violated. Description of symbol used in protocol is given in Table 1.

Protocol (SISQ):

Our protocol is broadly divided into four phases: Key Distribution, Packet Generation, Aggregation and Integrity Checking and SUM result generation. Key distribution is performed only once after topology had been setup. Each phase is described below.

4.2 Key Distribution

Before deployment each sensor node is preloaded with the key K_{all} which is used in encryption. Each node (sensor node and aggregator node) is also preloaded with the prime number P and Key K_c which is being used at time of encryption and MAC generation respectively. However, if required basestation can generate and broadcast new prime P and keys K_{all}, K_c after some predefined period. . After topology is being setup, basestation generates different key pairs $<k_{1i},k_{2i}>$ for each sensor node (S_i) in a WSN. Basestation also generates random modulus R (R < P), which is use to generate diffusion values. The prime number P and all the keys are having appropriate size to lessen the probability of random guessing.BS manually registers key pairs and modulus R to each sensor node(S) in a network.

$$BS \rightarrow Si : < <k1i,k2i> ,R> \quad (1)$$

4.3 Packet Generation

On completion of key distribution phase basestation will emit query (SUM query of format depicted in template) to the WSN. After receiving SUM query request, when sensor nodes are ready to supply sensor readings they first generates secret share $Sec_{i,t}$,. The secret share is generated with the key k1i for sensor node Si and time epoch T using HMAC1 (k_{1i},T).Note that query received at basestation contains time epoch with it. It then concatenate sensor reading $sr_{i,T}$ with the secret share $Sec_{i,t}$ as described in eq. (2). Hence packet $D1_{i,t}$ to be encrypted will now holds the sensor reading and the secret share within it.

$$D1_{i,t} = sr_{i,t} \parallel Sec_{i,t} \quad (2)$$

Now sensor node uses eq. (3) to generate diffusion value. This diffusion value is then added with sensor reading as given in eq. (4).

$$Dif_{i,t} = HMAC1(k_{2i},t) \bmod R : R < P \quad (3)$$

$$D2_{i,t} = sr_{i,t} + Dif_{i,t} \quad (4)$$

Sensor node then performs additive homomorphic encryption on the packet $D1_{i,t}$ and it also perform homomorphic encryption on packet $d2_{i,t}$. Encryption operations are described by eq. (5) and (6).

$ED_{1i,t} = E(D1_{i,t}, K_{1i}, K_{all}) = K_{all} * D1_{i,t} + K_{1i} \bmod P$ (5)

$ED_{2i,t} = E(D2_{i,t}, k_{2i,} Q) = D2_{i,t} + K_{2i} \bmod P$ (6)

Then S_i calculates MAC value (eq. (7)) of encrypted packets ($ED_{1i,t} \| ED_{2i,t}$) using the key K_c which is preloaded in to the every node in WSN before deployment. Then it sends the encrypted packet and MAC to parent aggregator node.

$S_i \rightarrow Agg$: MAC ($ED_{1i,} \| ED_{2i,t}, K_c$) , $ED_{1i,t}, ED_{2i,t}$ (7)

4.4 Aggregation

For our protocol follows the properties listed bellow for aggregation function:

- AGR(M1,M2)=AGR(M2,M1) : aggregate function is commutative
- AGR (M1, M2, M3) = AGR (M1, A (M2, M3)): aggregate can be calculated using sub aggregate values.

Parent aggregator node receives the message containing MAC value and pair of encrypted data. It first calculates the MAC of received data and compares it with the received MAC. Note that aggregator node can calculate MAC because key K_c is preloaded in it before deployment. If both are equal then it wait for the messages from other child sensor nodes. When aggregator node receives authenticated messages from all of child sensor nodes it perform aggregation operation by adding received encrypted messages using modulo prime P as mentioned in eq. (8) and (9). Modulo prime P is also preloaded into every node in a network before deployment.

$$Eag_1 = \sum_{i=1}^{j} ED_{1i,t} \bmod P \quad (8)$$

$$Eag_2 = \sum_{i=1}^{j} ED_{2i,t} \bmod P \quad (9)$$

An Aggregator node also generates MAC of aggregated data pair with key K_c and sends us these data to the parent aggregator node (or to the base station if it is a child node of a base station).

$Agg \rightarrow BS$: Eag_1, $Eag_{2,}$ MAC ($Eag_1 \| Eag_2, K_c$) (10)

4.5 Integrity Checking and SUM Result Generation

After receiving messages from all child aggregator nodes basestation first performs aggregation operation on them and calculate final pair of aggregated packets which hold the SUM result. Aggregation procedure follows same steps as described in eq. (8) and (9).

Now as part of integrity verification basestation calculate secret value SECT by adding secret share of each sensor node(eq. (11)).It also calculate SUM of diffusion values of each sensor node (eq. (12)). Note that the base station is a distributor of key pair <k1i,k2i>, hence BS can calculate secret share and diffusion values of each sensor node.

$$SEC_T = \sum_{i=1}^{N} HMAC1(K_{1i}, T) \quad (11)$$

$$Dif_{sum} = \sum_{i=1}^{N} (HMAC1(K_{2i}, T) \bmod R) \quad (12)$$

In order to decrypt aggregated packet BS generate SUM of keys K_{1i} and SUM of keys k_{2i}. Decryption is performed as described in eq. (15) and (16).

$$K_{1sum} = \sum_{i=1}^{N} K1i \quad (13)$$

$$K_{2sum} = \sum_{i=1}^{N} K2i \quad (14)$$

$D_1 = D(Eag_1, K^{-1}, K_{1sum}) = (Eag_1 - K_{1sum}) * K^{-1} \bmod P$ (15)

$D_2 = D(Eag_2, K_{2sum}) = (Eag_2 - K_{2sum}) \bmod P$ (16)

Decrypted packet D_1 and D_2 contains final aggregation SUM result, but they are diffused with other values. Hence to retrieve original SUM result BS deducts Dif_{sum} value from decrypted data D_2. BS performs bit shift operation on data D_1 to retrieve SUM value. Note that in D_1 first k (k bit sensor reading) bits contains SUM result and last L-k (L bit packet D_1) bit contains SUM of secret shares.

$$D_1 = SUM \| \sum_{i=1}^{N} SEC_{i,t} \quad (17)$$

$$D_2 = SUM + \sum_{i=1}^{N} Dif_{i,t} \quad (18)$$

$$\sum_{i=1}^{N} SEC_{i,t} = \text{last L-k bits of } D_1 \quad (19)$$

SUM: $S1 = D_2 - Dif_{sum}$, SUM: S2= k right bit shift D_1 (20)

Now that BS posses all the information required for integrity verification it compares sum of secret SEC_T it has calculated with received one.

$$\left.\begin{array}{l} \text{If } \left(\sum_{i=1}^{N} SEC_{i,t} = SECT \quad \text{AND} \quad \text{If } (S1 = S2) \right) \\ \text{Then: Accept SUM} \\ \text{Else: SUM result is not authenticated:} \\ \quad \text{Reject Packet} \end{array}\right\} \quad (21)$$

BS also compares received SUM results (eq. (21)), if both (sum of secrets and received SUM values) are equal then BS accepts the SUM result as authenticated. Otherwise it doubts the result and rejects it.

Figure 2 depicts the working of SISQ using example in which 9 nodes tree topology is considered.

5. SECURITY ANALYSIS

We present various attacks and their effect on our protocol in this section.

Protection against eavesdropping (Data Confidentiality):

Consider passive attack on described protocol in which an attacker tries to eavesdrop. Eavesdropping doesn't disclose any aggregate or reading values to attacker as all messages are encrypted using additive homomorphic encryption.

Protection against replay attack (Data Freshness):

Considering an active attack wherein the attacker tries to replay the packets previously captured. As we described earlier that each message contains a secret and diffusion values which are calculated using keys shared with base station and a time epoch which protects replay attacks. Hence our protocol provides data freshness.

Detection of aggregator node capturing (Integrity protection and authentication):

Considering attacker takes over an aggregator node and generates forged aggregate value instead of actual aggregation. This kind of misbehavior can be detected by the basestation as each pair of encrypted packets generated by sensor node contains secret share and diffusion values. If an attacker wants to alter the packet it has received from its child sensor node, he has to calculate correct secret share value for first encrypted packet in pair. Also second sensor reading is diffused with the value known to sensor node and BS only. So for including forge packets attacker must know secret share and diffusion values of each sensor node. As both packets are diffused differently, if attacker alters the packet without knowing diffusion values, the BS will detect it.

Hence, our protocol provides detection of change in original sensor readings. However, detection of exact compromised aggregator node is not provided by our protocol. Extending our protocol with merkel hash tree or with attestation method proposed by authors of [10] for detection of exact captured node will protect against aggregator node capturing.

aggregator node he has to recalculate MAC for the packet. However, the key used to calculate the MAC is not known to intruder it cannot alter the packet. In case if the packet is altered by an intruder aggregator node can detect the alteration because MAC of received packet and MAC calculated by aggregator node do not match.

If an intruder (attacker) is aware about how packets are diffused before encryption than it can alter readings of sensor node. Hence though integrity of sensor nodes readings is violated than also this protocol does not provide any mechanism to protect or at least detect this kind of attack.

Sensor node capturing:

We do not consider a case where an attacker captures a node and tries to forge its own reading contribution. The reason for not considering such attack is that cryptography does not prevent such attacks. Further an attacker can launch passive attack to change sensor reading thus an attacker do not need to capture any node. For example if sensors are sensing temperature value then an attacker can simply put a lighted candle to change sensor reading contribution. Hence our protocol does not provide protection against capturing of sensor nodes. Also in literature we didn't encounter any efficient method for sensor node capturing.

Node Failures:

An important point to be considered is node failures, i.e., Situations where either a source does not produce a packet or an aggregator fails to combine its children's massage in a time epoch. In this situation the failed node must be reported to the basestation. However, BS must also manually check the corresponding node, since a compromised node may falsely report the failure.

However our protocol does not provide detection and protection against Denial of service.

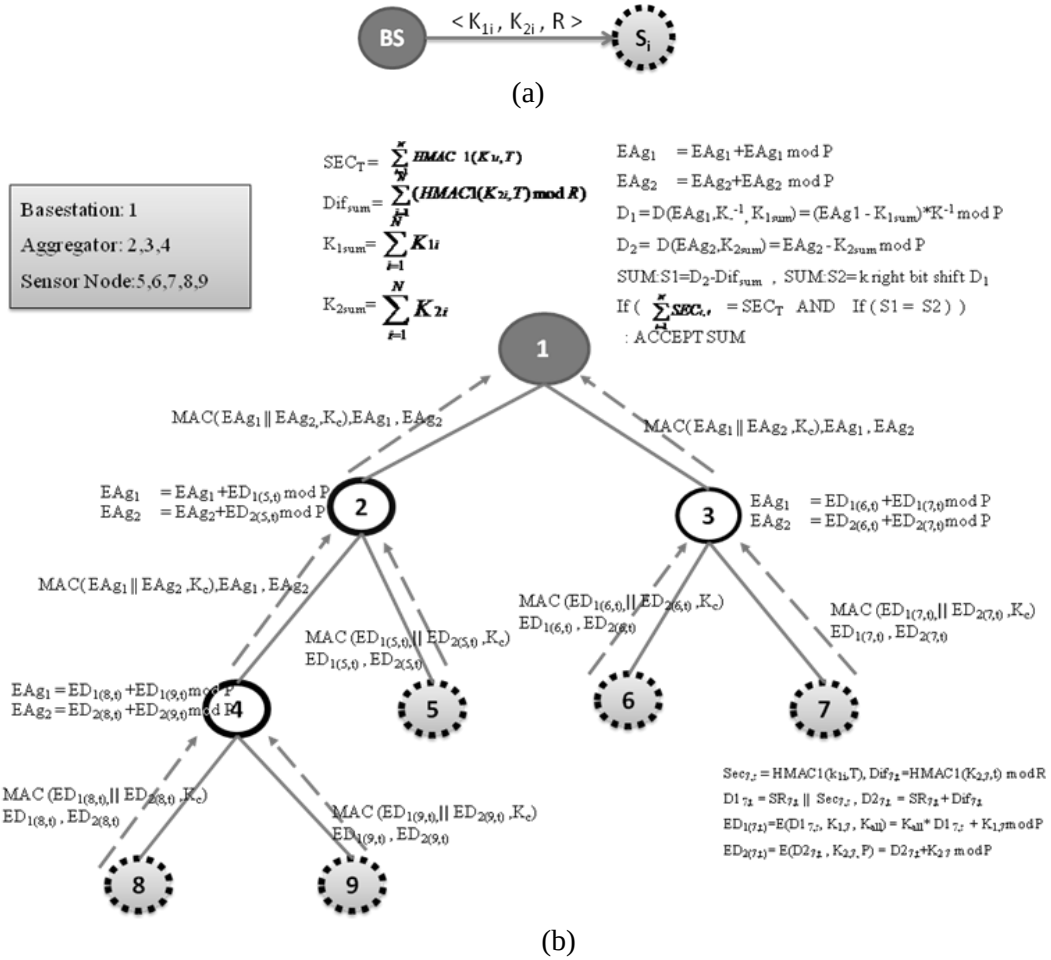

Fig 2: (a) key distribution phase (b) SUM aggregauon with integrity verification

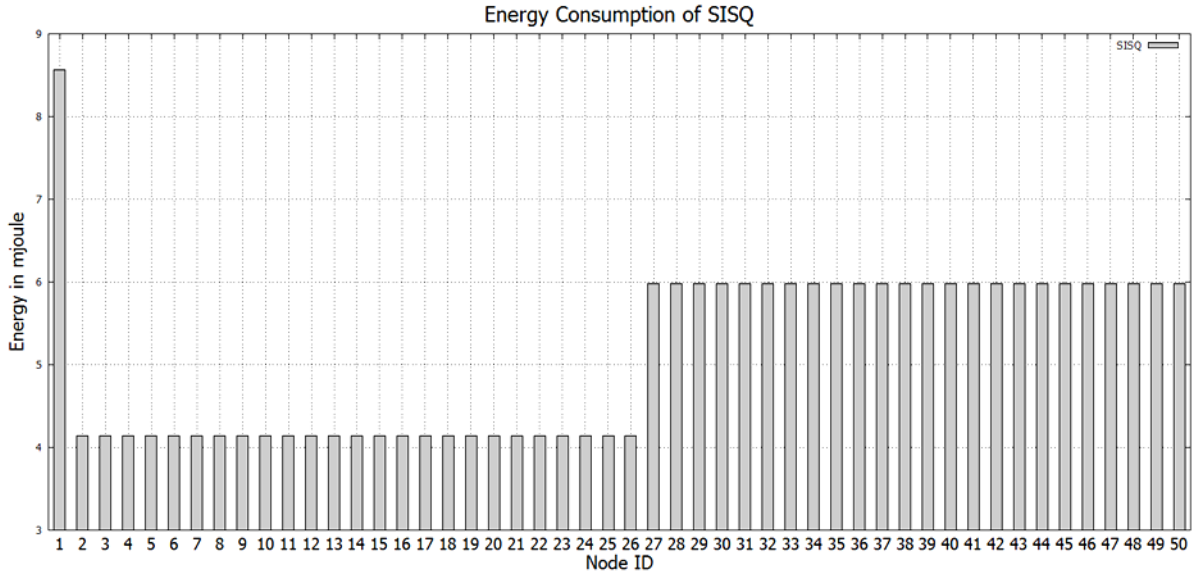

Fig 3: Energy consumption for 50 nodes in simulation

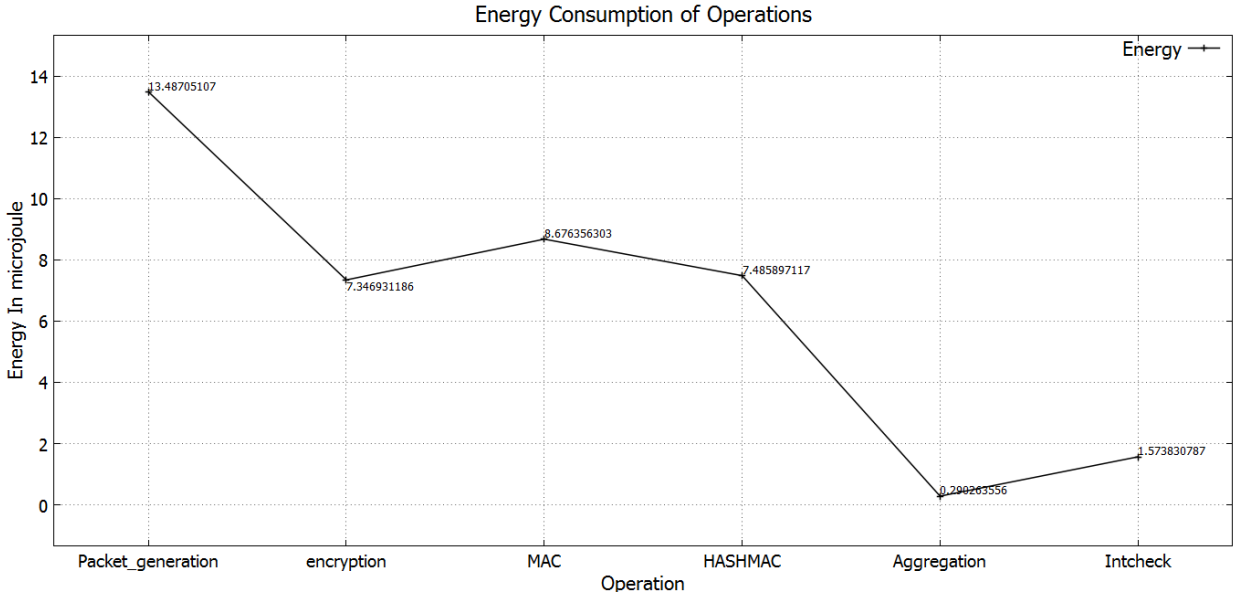

Fig 4: Energy consumption in micro joule for Key distribution, Packet generation and encryption at sensor node, MAC and HMAC generation, aggregation and Integrity verification

6. IMPLEMENTATION DETAILS AND SIMULATION RESULTS

We implemented the proposed framework in tinyos2.1 with TOSSIM for MicaZ and Mica2 motes. We used additive homomorphic encryptions as described in section 3. HMAC-SHA1 is applied to generate secret share. AES-CMAC [25] is used for MAC generation. In this section we present simulation results for our implementation. The memory requirement for our implementation is given in Table 2. MicaZ and Mica2 both motes requires almost same memory because both has 8-bit AVR microcontroller.

We collected energy consumption of MicaZ motes by using avroraz [26] simulator. Avroraz is an extension of Avrora [27]. Figure 3 presents the energy consumption of 50 nodes in simulation. In simulation of 50 nodes for SISQ, from node 27 to 51 are sensor nodes and nodes 2 to 26 are aggregator nodes. Node 1 is a basestation.

Table 2. Memory Requirements of Motes

Mote	SISQ	
	RAM(Bytes)	ROM(Bytes)
MicaZ	1566	39800
Mica2	1724	39532

The nodes with less energy consumption are aggregator node. Aggregator node consumes less energy because they perform only aggregation and MAC related functionality (i.e. Addition and MAC generation). Sensor node consumes more energy because they perform large number of computations on sensor reading and also generated MAC. However energy consumption of basestation is very high the reason is it performs large number of computations. Energy consumes by sensor node and basestation are of essence because it is

energy outcome of operations involved in integrity verification.

Figure 4 depicts the energy consumption in modules viz. Key distribution, Packet generation and encryption at sensor node, MAC and HMAC generation, aggregation and Integrity checking.

Packet generation takes more energy compare to other operations because it involves calculation of secret share, diffusion value and concatenation of these values with sensor reading, while in integrity checking phase we have considered comparison of sum of secret share and SUM result received from pair of aggregated data. Hence Energy consumption is less for Intcheck operation. Table 3. Describes Energy consumption in micro joule for Packet generation, Aggregation and Integrity checking operations for SISQ.

Table 3. Energy Consumption of operations in SISQ

Operation	Energy in micro joule
Packet_generation	13.48705107
Aggregation	0.290263556
Intcheck	1.573830787

7. CONCLUSION

In-network integrity protected processing of SUM queries (as well as their derivatives, e.g., COUNT, AVG, etc.). SISQ is the solution that offers exact query answers, satisfying all the necessary security properties of the targeted model, i.e., data confidentiality, integrity, authentication, and freshness. Every sensor node generates two differently diffused packets for same sensor reading in order to provide minimal chance of replacing authenticated sensor readings. The operation uses security mechanisms those are lightweight, leading to very small bandwidth consumption for all parties involved, and a very low CPU cost because they entail a small number of inexpensive cryptographic operations (hashes and modular additions/multiplications). Hence, SISQ is a powerful security tool for resource-constrained sensor networks. We confirm our performance claims through a detailed analytical and experimental evaluation.

8. REFERENCES

[1] Jia, Guo., Jian'an, Fang., and Xuemin, Chen,2011. Survey on secure data aggregation for wireless sensor networks. In IEEE International Conference on Service Operations, Logistics, and Informatics (SOLI),138-143.

[2] Suat, Ozdemir., and Yang, Xiao., "Secure data aggregation in wireless sensor networks: A comprehensive overview",Journals on Computer networks ,Vol.53, Issue.12,2022–2037,2009.

[3] Akyildiz, F., Su, W., Subramaniam, S. Y., and Cayirci E.,"Wireless sensor networks a survey",Journal of Computer Networks,393 – 422,2002.

[4] Ramesh, Rajagopalan., and Pramod, K. Varshney.,2006.Data aggregation techniques in sensor networks: A survey. In Communication Surveys Tutorials IEEE, Volume 8, 48–63.

[5] Nath, S., Yu, H., and Chan, H., 2009. Secure outsourced aggregation via one way chains. In Proceedings of ACM SIGMOD International Conference on Management of data, 31-44.

[6] Boppana, R.V., and Pengjun, Pan., 2009.A comparison of secure data aggregation schemes for wireless sensor networks. In International Conference on High Performance Computing, 179-188.

[7] Papadopoulos, Stavros., Kiayias, A., and Papadias, D., 2011.Secure and efficient in-network processing of exact SUM queries.In Proceedings of IEEE 27th International Conference on Data Engineering,517-528.

[8] Chien-Ming, Chen., Yue-Hsun, Lin., Ya-Ching, Lin., and Hung-Min, Sun., 2012. RCDA: Recoverable Concealed Data Aggregation for Data Integrity in Wireless Sensor Networks. In IEEE Transactions On Parallel And Distributed Systems, Volume 23,727-734.

[9] Zhu, Liehuang., and Li, Meng.,2011.An Energy Efficient and Integrity-Preserving Aggregation Protocol in Wireless Sensor Networks. In Performance Computing and Communications Conference IEEE 30th International, 1-6.

[10] Mlaih, Esam., and Aly, Salah A.,2008. Secure Hop-by-Hop Aggregation of End-to-End Concealed Data.In IEEE Wireless Sensor Networks INFOCOM Workshops, 1-6.

[11] Przydatek, Bartosz., Song, Dawn, and Perrig, Adrian., 2003.SIA: Secure Information Aggregation in Sensor Networks. In ACM Proceedings of the 1st international conference on Embedded networked sensor systems, 255 – 265.

[12] Jadia, P., and Mathuria, A., 2004.Efficient secure aggregation in sensor networks. In High Performance Computing, 40-49.

[13] Xiaoyan, Wang., Jie, Li., Xiaoning, Peng., and Beiji, Zou., 2011. Secure And Efficient Data Aggregation For Wireless Sensor Networks.In IEEE Seventh vehicular Technology Conference Fall , 1-5.

[14] Lingxuan, Hu., and David, Evans., 2003. Secure Aggregation for Wireless Networks. In ACM Proceedings of theSymposium on Applications and the Internet Workshops, 384-391.

[15] Ozdemir, Suat., 2007. Secure and reliable data aggregation for wireless sensor networks. In ACM Proceedings of the 4th international conference on Ubiquitous computing systems, 102-109.

[16] Poornima, A., and S.,Amberker, B., 2010.SEEDA: Secure End-to-End data Aggregation in Wireless sensor networks. In IEEE Seventh International Conference on Wireless and Optical Communication Network, 1-5.

[17] Castelluccia, C., Mykletyn, E., and Tsudik, G., 2005. Efficient aggregation of encrypted data in wireless sensor networks. In the Second Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services MOBIQUITOUS, 109-117.

[18] Bagaa, M., Lasla, N., Ouadjaout, A., and Challal, Y., 2007. SEDAN: Secure and Efficient protocol for Data Aggregation in wireless sensor Networks. In IEEE Conference on Local Computer Networks, 1053-1060.

[19] Mahimkar, A., 2004. SecureDav: A secure data aggregation and verification protocol for sensor networks.In Proceedings of the IEEE Global Telecommunications Conference, 2175-2179.

[20] Wenliang, Du., Jing, Deng., Yunghsiang, S. Han., and Pramod, K. Varshney., 2003. A witness based approach for data fusion assurance in wireless sensor networks. In

Proceedings of the IEEE Global Telecommunications Conference, 1435-1439.

[21] Ozdemir, Suat.,".Functional reputation based reliable data aggregation and transmission for wireless sensor networks",Journal of Computer Communications, November, Volume 31, Issue 17, 3941-3953,2008.

[22] Yi, Yang., Xinran, Wang., Sencun, Zhu., and Guohong, Cao.,2006.SDAP: A secure hop-by-Hop data aggregation protocol for sensor networks.In Proceedings of the 7th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc '06), 356-367.

[23] Rabindra, Bista., Kyoung-Jin, Jo., and Jae-Woo, Chang.,2009. A new approach to secure aggregation of private data in wireless sensor networks.In Proceedings of the 8th IEEE International Conference on Dependable, Autonomic and Secure Computing, 394–399.

[24] Kui, Wu., Dennis, Dreef., Bo, Sun., and Yang, Xiao.,"Secure data aggregation without persistent cryptographic operations in wireless sensor networks",Journal of Security Issues in Sensor and Ad Hoc Networks, Volume 5, Issue 1, 100-111,2007.

[25] The AES-CMAC Algorithm, [Online] Available :http://tools.ietf.org/html/rfc4493.

[26] Alberola, R de Paz., and Pesch, D., 2008. AvroraZ: extending Avrora with an IEEE 802.15. 4 compliant radio chip model. In Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks, 43-50.

[27] Titzer, B. L., Lee, D. K.,and Palsberg, J., 2005. Avrora: Scalable sensor network simulation with precise timing. In Fourth International Symposium on Information Processing in Sensor Networks, Article No. 67.

[28] Abduvaliev, A., Sungyoung, Lee., and Young-Koo, Lee., 2009.Simple hash based message authentication scheme for wireless sensor networks. In 9th International Symposium on Communications and Information Technology, 982-986.