ADA: Authenticated Data Aggregation in Wireless Sensor Networks

E. G. Prathima
Department of Computer Science and Engineering, University Visvesvaraya College of Engineering, Bangalore University, Bangalore-560001

Shiv Prakash T.
Vijaya Vittala Institute of Technology, Bangalore, Karnataka

Venugopal K. R.
University Visvesvaraya College of Engineering, Bangalore University, Bangalore-560001

S. S. Iyengar
Florida International University
Florida, USA

L. M. Patnaik
INSA, National Institute of Advanced Studies, Indian Institute of Science Campus, Bangalore, India

ABSTRACT
Wireless Sensor Networks are vulnerable to communication failures and security attacks. It is quite challenging to provide security to data aggregation. This paper proposes Authenticated Data Aggregation for Wireless Sensor Networks, where the nodes organize themselves into tiers around the sink. Message Authentication Code (MAC) is generated and transmitted along with the synopsis to ensure integrity. All nodes in the network store the same key that is used for rekeying operation during each round to generate MAC. Thus ADA ensures data freshness and integrity at a communication cost of O(1). Simulation results show that the proposed ADA protocol results in high security, low energy consumption and low communication cost compared to the state-of-the art protocol.

Keywords
Data aggregation, Synopsis, Tiers, WSN.

1. INTRODUCTION
Wireless Sensor Networks (WSNs) consists of a set of sensors that are severely constrained in resources such as energy, bandwidth, memory and computational capability. Each sensor node senses the physical environment, process the reading and communicates its observation to the Base Station either directly or through multi-hop communication. The highest energy consuming activity in a sensor node is transmission. Therefore as the number of transmissions increases, the network lifetime decreases.

In-network aggregation techniques were introduced that combines partial results at intermediate nodes by which there is a significant reduction in the number of messages communicated resulting in comparatively lesser energy consumption per node and increase the network lifetime. In-network aggregation can be either cluster-based or tree-based. In Cluster-based aggregation, the sensor nodes are grouped into clusters with cluster-head that performs in-network aggregation. In Tree-based aggregation an aggregation tree is constructed where non-leaf nodes in the aggregation tree perform in-network aggregation.

In the aggregation techniques mentioned above if the aggregator node fails, the data from entire cluster or subtree will become unavailable for aggregation. Multi-path communication based techniques were introduced in which a node can have more than one parent in the aggregation hierarchy. But multipath based communication results in message duplication where same data will be aggregated multiple times. In case of duplicate-sensitive aggregates, such as Count and Sum, the individual readings and partial results sent along the multiple paths results in overcounting. Two approaches were designed to address overcounting problem in multi-path aggregation, synopsis diffusion and summation sketch.

The mode of communication for sensor nodes is broadcast by nature and they are generally deployed in open environment. Due to this reason, WSNs are vulnerable to various types of security attacks. Many types of attacks can be launched on in-network aggregation such as compromising a node to affect aggregated results, impersonating a node, replaying an outdated message. Hence authenticating data and sender of data is important while performing aggregation.

Motivation: The cryptographic algorithms require higher computation capacity and require messages to be encrypted and decrypted at each end. On the other hand data aggregation functions are applied on the plain text. Hence securing the data aggregation process in an energy efficient manner is challenging.

Contribution: This paper proposes Authenticated Data Aggregation (ADA) combines the concept of adaptive rings with TDMA and pairwise verification with rekeying.

Organization: This paper is organized as follows: Section II reviews various data aggregation techniques. Section III describes the synopsis diffusion framework. Section IV defines the problem and describes the system model. Section V presents ADA. Section VI discusses the simulation results and performance analysis. Section VII concludes the paper.

2. LITERATURE SURVEY

2.1 Data Routing and Aggregation Techniques:
Intanagonwiwat *et al.*, [1] have designed a stable Directed Diffusion for distributed sensor networks where a query is transformed to interest and then diffused to nodes in different regions. These nodes propagate the data in the opposite direction of interest. Handziski *et al.*, [2] explored the effect of directed diffusion on sensor network with passive clustering that can significantly reduce the required energy while improving delay and delivery rate. Dargahi *et al.*, [3]

enhanced the Directed Diffusion based on nodes' credit resulting in energy efficiency, reliability and supports load distribution.

Considine *et al.*, [4] investigated the use of approximate in-network aggregation for computing duplicate sensitive aggregates by combining duplicate-insensitive *sketches* with multipath-routing techniques. The sketches generated are compressed using run-length encoding and reduces the space requirement by 30%. Fan and Chen, [5], [6] proposed linear counting sketches for multipath routing based in-network aggregation. The Scalable Counting (SC) sketch and its variant adaptive scalable counting (ASC) sketch presented in [7] can produce duplicate-insensitive synopsis and at the same time suppress data transmissions insignificant to aggregate computation. This algorithm performs in-network aggregation with much less space requirement than [6].

Nath *et al.*, [8], [9] presented synopsis diffusion, a general framework to overcome double-counting problem where best effort, multipath routing schemes called rings is used together with order and duplicate insensitive (ODI) synopsis. The implicit acknowledgement mechanism enables synopsis diffusion adapt to dynamic message loss condition.

Different energy efficient routing techniques are presented in [10], [11], [12] and [13].

2.2 Secured data aggregation:

Garofalakis et al., [14] derived proof sketches which provide verifiable approximations for a broad class of distributed queries. It combines Flajolet-Martin (FM) sketches with and authentication manifests resulting in low false negative rate. The algorithm is robust as the adversary must compromise the aggregators near the root of the topology to get near the worst case bounds undetected.

Nath et al., [15] developed Secure Outsourced Aggregation (SECOA) for aggregation by untrusted third party service providers based on unified use of one way chain and support a wide range of aggregation functions. The proposed framework detects malicious aggregators without communicating with sensors and incurs low additional communication and computational overheads. Yang et al., [16] have designed a Secure Hop-by-hop Data Aggregation Protocol (SDAP) that uses a probabilistic grouping to partition the aggregation tree into subtrees of similar size. A commit-based hop by hop aggregation is performed to generate group aggregate and is verified by the base station. The protocol effectively defends against both count and value changing attacks.

Chen and Yu, [17] proposed Verifiable Minimum with Audit Trail (VMAT), which relies only on symmetric key cryptography. VMAT guarantees either the correct aggregation result or revokes some key held by the adversary. Papadopoulos et al., [18] developed Secure In-network processing of Exact Sum queries (SIES) that provides both integrity and confidentiality through a combination of homomorphic encryption and secret sharing. The variance and standard deviation queries require larger plain texts and keys resulting in performance degradation.

3. BACKGROUND WORK

3.1 Introduction to Synopsis Diffusion

The nodes organize into adaptive rings around the sink as the query propagates through the network. It is named adaptive rings since each node creates their neighbor list during each query dissemination phase and hence in the neighbor list *failed* nodes are not added. A node that is *i* hops away from base station is considered to be in ring L_i. A node in ring *i* has multiple parents in ring *i-1* and multiple children in ring *i+1*. When all nodes in outermost ring have received the query, the second phase stars. The aggregation process starts from outermost ring. Each node *X* in the outermost ring computes synopsis which is a bit-vector generated using Probabilistic Counting with Stochastic Averaging (PCSA) algorithm proposed by Flajolet-Martin [22]. The synopsis generated by using *SynGen()* function, LS[X] is then broadcasted.

When a node *Y* at level L_i receives the synopsis from a node *X* in level L_{i-1}, it performs aggregation by applying *SynFuse()* function as shown below:

$$FS_Y = LS_Y \mid FS_{X_1} \mid FS_{X_2} \mid ... \mid FS_{X_c}$$

Where FS_Y is the fused synopsis of the node *Y*, LS_Y is the synopsis generated at node *Y* corresponding to its data V_Y and *c* represents the number of children of node Y. The node Y then broadcasts the fused synopsis FS_Y. This process is repeated until all the aggregated synopses reach the base station. A node broadcasts its synopsis multiple times to provide better resilience against communication failure.

When base station receives synopsis from all its children, the base station applies synopsis fusion function on all received synopses. The final synopsis obtained is a bit-vector that is represented by the regular expression, $1^{z-l}0[0, 1]^{l-z}$ where *z* is the index of leftmost (least significant) 0-bit in the final synopsis. Finally, the base station evaluates the synopsis for count query as $2^z/0.7735$ and for Sum query as 2^z.

3.2 Secured Data Aggregation

Roy et al., [19] presented a data aggregation protocol for *sum* and *count* aggregates that secures the original synopsis diffusion protocol by sending Message Authentication Code (MAC)s to the base station with partial results computed at each level in the hierarchy. The base station can detect the presence of false subaggregates by verifying these MACs. In [20] a verification algorithm is presented to secure the synopsis diffusion technique that generates *k* MACs authenticating the each of *k* rightmost 1 bits in the fused synopsis of node X. Later they have proposed a two phase verification algorithm [21], in which a node transmits MAC for each of the '1' bit it is contributing. Phase II of the algorithm is initiated only if the base station is not able to verify the index of at least one '1' bit it received in the final aggregated synopsis.

This approach incurs more communication overhead. In both the cases, a node *X* transmits MAC authenticating index of i^{th} rightmost 1 bit which may be generated at *X* itself or may be received from any one of its children. The problem with this approach is that, since node *X* does not verify the MAC received from any of its children, it is possible that *X* may generate a genuine MAC for a falsified *1* bit and transmit it along with the synopsis and the attack remains undetected.

4. PROBLEM DEFINITION AND SYSTEM MODEL

4.1 Problem Definition

Given a sensor network *G*, with *N* sensor nodes and a query *Q* issued from the base station, compute duplicate sensitive aggregate corresponding to the query *Q* on demand, while removing contributions from the malicious nodes, M at a reduced communication and computation overhead.

Objectives:
1) Reduce malicious contribution.
2) Reduce communication cost and increase network lifetime.

4.2 System Model and Assumptions

4.2.1 Network Model:

The Sensor Network consists of N homogeneous sensors with a configuration similar to that of MicaZ or Telos in their communication and computation capabilities. The sensor network is organized into 2D grid of size N X N in which sensor nodes are placed on grid points and base station is placed at center of the grid as shown in Figure 1. The sensor nodes send their data to the sink through multihop transmission. All the nodes in the network are assumed to be synchronized. Each node in the network has same initial energy E_0. It is assumed that each node has exactly eight neighbors. The communication range R of each sensor node is chosen to be $\sqrt{2}$ so as to have 8 neighbours.

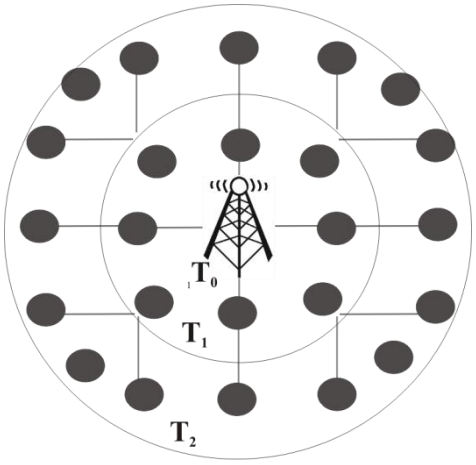

Fig 1: Network Deployment

4.2.1.1 Adaptive tier

ADA uses extended version of adaptive rings topology called adaptive tier. This architecture allows a node at tier *i* to have parents in same tier in addition to parents in previous tier. The aggregation time is divided into n−1 mini-slots, where n is the maximum number of neighbors a node can have in previous tier. All nodes whose *id* is odd transmit in odd numbered mini slots and all nodes whose *id* is even number transmit in even numbered slots. The nodes that have not yet transmitted their synopsis, *i.e.*, with unexpired timer, aggregate the received data from neighbors in the same level before transmitting. The adaptive tier ensures that data of each node is aggregated by at least 3 neighbors and hence it is more resilient to failures than adaptive rings.

4.2.2 Attack Model

It is assumed the sink cannot be compromised whereas all other sensor nodes are assumed to be vulnerable to attacks. ADA algorithm tries to address mainly two types of attacks, replay attack and false data injection.

4.2.2.1 Replay attack

Replay attack affects data freshness. Here a compromised node retransmits a genuine synopsis packet that has been generated during one of the previous epochs in place of current synopsis packet.

4.2.2.2 False data injection

In this type of attack, a compromised node tries to introduce false contribution in its aggregated data. If the data sensed by sensors are transmitted as bit vector (such as synopsis or sketches), a compromised node may either inflate (where bit with value '0' is changed to '1') or deflate (bit with value '1' is converted to '0') either in its own synopsis or aggregated synopsis.

4.2.3 Security Model

Every node in the topology broadcasts its data to all its neighbors at previous level. Due to the broadcast nature of communication, establishing a pairwise key is not well suited for this type of communication. Because, if pairwise key is used, multiple unicast messages must be exchanged between each pair of neighbors to achieve multipath routing and results in increased communication overhead. So establishing pairwise key is not suitable solution to multipath routing.

A second option is to have a group key shared between node *X* and its parent. This again requires multiple unicast messages to be communicated between node *X* and its neighbors to agree upon a common key. This approach also ends up in increased communication overhead.

It is assumed that every node *X* is pre-loaded with one master key *K* which is generated at the base station.

If a node *X* wishes to compute MAC for synopsis either self-generated or received from its neighbors, it performs a re-keying operation to generate KX computed as follows:

$$K_X = f_K(id,e) = (K \parallel id) \oplus e$$

A new key is generated for every round synopsis is to be transmitted. This key, K_X is used for computing the Message Authentication Code. Only the master key, *K* is stored at all the sensor nodes. The key of each node K_X is generated in the MAC generation procedure. Even if an attacker gets the master key, a genuine MAC authenticating fake data cannot be generated.

5. THE ADA ALGORITHM

ADA aims at allowing the base station to obtain the approximate estimate of the aggregate while keeping the computational, communication and memory overhead minimal. For achieving this ADA uses pull based architecture for data collection where the sink pulls data from the sensor nodes using a query. The algorithm comprises of three phases:

1) Query generation and propagation: Query propagates through network and nodes organize into adaptive tier
2) Synopsis generation and aggregation: Synopsis is generated and aggregated and
3) Evaluation.

The generation of synopsis using primitive polynomials is presented in [22]. ADA can be used for both persistent as well as single shot queries.

5.1 Query generation and propagation

In this phase the Base Station generates two random integers; a random integer g and a prime number p. These two random integers ensure data freshness and are used for exchanging pairwise keys. The base station generates a query packet containing the fields: < ***Q, g, p, t, e, T*** > where Q represents the type of query (count, sum or average), t represents time of query generation and e represent the interval after which subsequent aggregated data packets are expected. Base station is the only node at tier 0, hence when query packet is generated at the base station, it sets T to 0. When the query packet reaches a node *X* for the first time, it sets a timer for synopsis generation. The node *X* then increments the *T* field in the packet by *1* and set its own tier to *T+1*. The node also

stores all information related to the query locally. Then it replaces the *id* field in packet by its own *id* and rebroadcasts the packet. Here limited flooding technique is used to reduce the number of packets in transit. A node *X* rebroadcasts query packet QP_j for query *j*, when of the two following conditions hold:

(i) QP_j is received for first time at *X*
(ii) QP_j is already received but the node *X* belongs to a lower tier, i.e., *T* of has been reduced to *T-1*. On receipt of query packet, each sensor node updates its active neighbor list. In addition the node sets a timer inversely proportional to its tier, i.e., timer of leaf nodes expire first and timer of nodes at *tier1* expires last.

Table 1. List of notations used

Notation	Meaning
Id, X, Y	Identifier of sensor node
N	Number of sensor nodes in the network
R	Communication range of sensor node
T_i	i^{th} Tier around the Sink
V_X	Reading of sensor node *X*
Q	Type of query from Sink Sum, Count and Average
g	Random number generated at the base station
p	Prime number generated at the base station
LS_X	Local synopsis generated at the node X
FS_X	Fused synopsis of the node X
LS[i]	i^{th} bit of synopsis Local Synopsis
len	Length of the synopsis
Z	Index of the least significant 0 bit in the synopsis
D_i	Data of sensor node *i*
K	Key preloaded in all sensors
K_i	Key of the sensor node *i*
M_I	MAC corresponding to bit at index I

The two random integers *p* and *g* allows a node X to differentiate query packet QP_j from the previous query packet QP_{j-1} corresponding to queries Q_i and Q_{i-1} respectively and hence ensures data freshness. This process is repeated until the query packet reaches all nodes in the network. The resultant topology formed is the *adaptive tier*, presented in Section IV.

5.2 Synopsis Generation and Aggregation

5.2.1 Synopsis generation

When the timer for synopsis generation at node *X* expires, it generates reading v_X corresponding to the type of query as discussed in section II and resets the data generation timer. Then node *X* generates its local synopsis. *Primitive polynomials modulo 2* with coefficients 0 or 1, is used as an alternative to hash function to generate random bit positions, corresponding *synGen()* function is given in Function 1. The advantage of using *primitive polynomials modulo 2* as hash function in comparison to PCSA based hash function is twofold: 1) Since it uses bitwise XOR and shift operations, computation cost is low 2) It does not require arrays for the computation in comparison to PCSA based hash function, which uses two arrays of size 64 and hence it incurs very low memory overhead. SynGen() function works differently for *Sum* and *Count* queries as shown in Function 1.

Function1: Function to generate synopsis

Function: SynGen(*Id*, V_{id}, l*en*)
Compute qtime as t + (e * *round*)
if Query = "Count" **then**
  Set *rseed* to *Id* ⊕ *qtime*
  Initialize *i* to *0*
  while i < len **do**
    Perform bitwise XOR on the bits of *rseed* that correspond to the selected polynomial of order *len*
    Store the result in *newbit*
    Perform 1 bit left shift on *rseed*
    Reset *rseed* as *rseed* ⊕ *newbit*
    if *newbit = 1* **then**
      Set LS[i] to 1
      Return LS_{id}
    else
      Increment *i* by 1
else if Query = "Sum" **then**
  Set n1 equal to then number of 1 bits in the reading, V_{id}
  Set *rseed* to *Id* . V_{id} ⊕ *qtime*
  Initialize *i* to *0*, *j* to *0*
  while *i* < *n1* **do**
    while *j* < *len* **do**
      Perform bitwise XOR on the bits of *rseed* that correspond to the selected polynomial of order *len*
      Store the result in *newbit*
      Perform 1 bit left shift on *rseed*
      Reset *rseed* as *rseed* ⊕ *newbit*
      if *newbit = 1* **then**
        Set LS[j] to 1
        Return LS_{id}
      else
        Increment *j* by 1
    Increment *i* by 1

5.2.1.1 Count Query

Synopsis for count query is simple. As discussed above the *hash()* function implemented using *primitive polynomials modulo 2* and *CountSyn()* function invokes *hash()* function repeatedly until it returns 1. If i^{th} invocation of *hash(id, len)* returns 1, then i^{th} bit of its local synopsis LS_{id} is set to 1 as in original synopsis diffusion.

Example: Let *id* = 960 and let the polynomial selected is 14**x+**5x+3**x+x+**1. When *hash(id, len)* is invoked for the first time, it performs bitwise XOR of 14^{th}, 5^{th}, 3^{rd}, 1^{st} and 0^{th} bit. Since bits in all the corresponding positions are 0, the *hash(id, len)* returns 0 as result. In this case, a single left shift is performed on *seed*. We can see that on 6^{th} invocation of *hash(id, len)*, i.e, after 5 left shift operations, the function returns 1. Hence the synopsis generation function sets the fifth bit to 1.

5.2.1.2 Sum Query

To generate Synopsis for Sum query, node X executes the *CountSyn()* function *b* number of times and sets, where *b* is the number of 1 bits in reading measured by X. The local synopsis LS_X has *b* bits set to *1*. Let V_{max} represent the

maximum value of count. Then the number of nodes contributing to ith bit of synopsis is equal to $V_{max} / 2^i$. Let c represent the number of consecutive 1 bits in the synopsis, then c = z - 1, where z is the index of least significant 0 bit. $E(c) = \log_2(V_{max})$.

Algorithm 1: Authenticated Data Aggregation

Input: Query from user
Output: Aggregated result corresponding to the query A_Q
begin

PHASE I: Query generation and propagation
Generate random number *g* and prime *p* at Sink
Generate and broadcast query packet $QPacket_t$ with fields <Q, g, p, t, e, T, *Id*>
if $QPacket_t$ is received by sensor node, *X* **then**
  if *T* in $QPacket_t < T_X$, tier of node *X* **then**
    if $QPacket_t$ is received for the first time **then**
      Store fields of $QPacket_t$ in node *X*
      Set synopsis generation timer
  Set T_X to *T + 1*
  Increment *T* field of $QPacket_t$ by 1
  Set *Id* field of $QPacket_t$ to *X*
  Set aggregation timer α 1/T
Add *X* to active neighbour list
PHASE II: Synopsis generation and aggregation
if synopsis generation timer fires **then**
  Call SynGen(*Id*, V_X, *len*) LS_X
  Initialize FS_X to LS_X
for each DPacket received **do**
  if T_Y of DPacket received from $Y \geq T_X$ of node *X* ***then***
    Call SynGen(*Id*, V_Y, *len*) to generate synopsis of *Y*
    if received synopsis of Y = generated synopsis at *X* **then**
      Generate MAC for left most 0 bit in the received synopsis
      i**f** received MAC = generated MAC **the**n
      Aggregate synopsis in the received *DPacket*
    else
      Drop the *DPacket*
  else
    Drop the *DPacket*
if aggregation timer fires **then**
  Generate MAC for left most 0 bit
  Create *DPacket* containing fused synopsis
  Broadcast fused synopsis
PHASE III: Evaluation
  Find index of least significant 0 bit, Z
  if Q = "Count" t**hen**
    Compute $A_Q = 2^{Z-1}/0.7335$
  if Q = "Sum" t**hen**
    Compute $A_Q = 2^{Z-1}$
  Return A_Q

5.2.2 Synopsis Aggregation

When any non-leaf node X at T_i receives a packet from its neighbor at level $T_{i+1,}$ it first generates the synopsis corresponding to the reading. If the received synopsis and generated synopsis match then, the node regenerates the MAC for the received synopsis using the MAC generation algorithm discussed above. If the generated MAC agrees with the received MAC, then X aggregates the data received from Y with its own as $FS_X = FS_X \mid FS_Y$; where | indicates bitwise OR operation. When timer of X expires, X generates its fused synopsis and then generates a MAC authenticating the least significant 0 bit it is contributing. It then broadcasts its fused synopsis along with the MAC and its reading corresponding to query Q_j to P_x. Evaluation phase is performed at the Sink node. When the sink node receives a data packet from a node at tier 1 say Y, it generates a MAC authenticating the index of least significant 0 bit using the MAC generation procedure discussed above. It then verifies the received MACs with the generated ones and if they match then the synopsis in received packet is fused at the sink as mentioned in Data generation and aggregation above.

5.2.2.1 MAC generation

MAC generation procedure takes <id, V, K, FS_{id}, T, g, p, t, e> as input. It first generates a key for this data collection round using the common key shared by all nodes using a function similar to Diffie-Hellman Key exchange protocol

Key = $((K \parallel id) \oplus qtime \oplus g)^{L+L-1}$ mod p where *g* and *p* are random numbers transmitted along with query from sink.

*qtime = t + (e * round)*; *t* and *e* are received at each node along with query and round represents the number of intervals lapsed after receiving the query Q in synopsis generation and aggregation. T_i is tier of sending node. Once the Key is computed, a MAC are generated using cryptographic hash function authenticating least significant '0' bit it is contributing. The MAC thus generated is then grouped into 4 byte chunks and then a bitwise XOR operation is performed on each of the 4 byte chunk to obtain the final MAC of size 4 bytes. For example, Let *M* be the 128 bit MAC generated, then divide *M* into blocks of size 4 bytes say *m1*, *m2*, *m3*, *m4* and recompute MAC as: M = m1 ⊕ m2 ⊕ m3 ⊕ m4

The reason for generating the MAC for the least significant 0 bit is that the final value of synopsis depends on the expected index of the least significant '0' bit E(Z). If an inflation attack is launched at any bit position (index) i, it does not affect the value of final approximate computed at base station as long as i < Z.

5.3 Evaluation

The evaluation phase is similar to the synEval() function given under section 3.1.Once the synopsis of all eight neighbors are aggregated, the sink evaluates the result of count query as $2^z/0.7735$, where z is the index of least significant 0 bit. The result of average query is evaluated as sum/count.

6. RESULTS AND ANALYSIS

This section presents a detailed analysis of the simulation results performed on NS2 simulator. The basic network size used consists of 900 sensor nodes placed in a *grid* topology. The sink is placed at the center of the grid as shown in Figure 3. The node density is 4 nodes / m^2. During each data collection round every sensor generates its reading, which is a random uniform integer within range 0 to 250. Various parameters considered for simulation include:

1) **Network Size:** The simulations are performed by varying the network size from 10x10 to 50x50.

2) **Average Energy Dissipated per node:** This parameter tells how much energy is consumed in micro Joules during each data collection round. The energy consumption unit is

micro Joules because of the dense deployment and lower transmission power.

3) **Average Packet Size:** Average packet size is represented in bytes and is a measure of communication cost, because communication cost is proportional to the size of packet sent.

4) **Root Mean Square Error:** Is a measure of deviation of computed result at sink from expected value and is computed using the formula:

$$\text{RMSError} = 1/V \sqrt{(1/r \sum_{i=1}^{r} (V_i - V)^2)}$$

where Vi is the value of result computed at the sink during round i and V is the value of expected result at the sink. The closer the value of RMS error to 0, the accurate is the computed aggregate.

6.1 Basic Comparison

Performance of ADA is compared with that of Synopsis Diffusion algorithm (referred as SynDiff) presented by Nath *et al.*, in [9] and two phase verification algorithm (referred as SDA-2PV) proposed by Roy *et al.*, [21] which computes the exact aggregate even in presence of falsified subaggregate attack. SDA-2PV, tries to provide security to original synopsis diffusion algorithm SynDiff where the nodes run SynDiff and SDA-2PV simultaneously. In SDA-2PV and SynDiff a node first generates a synopsis using original Synopsis Diffusion algorithm described in [9], where a node *X* computes *m* synopsis representing its reading V_x during each data collection round.

6.2 Communication Cost

The cost of communication increases with the size of data transmitted. The main aim of ADA algorithm is to keep the communication cost to a minimum. The Figure 2 shows a comparison of number of bytes set per node during each data collection round and the network size. Both SynFiff[9] and SDA-2PV[21] uses adaptive rings topology where the synopsis is retransmitted multiple times. In the simulation we have restricted the number of retransmissions to 2. The SDA-2PV incurs highest communication cost due to the two reasons: 1) It uses adaptive rings and therefore the synopsis is retransmitted and 2) it generates *k* 4 byte MACs for each of the *k* one bit in the synopsis.

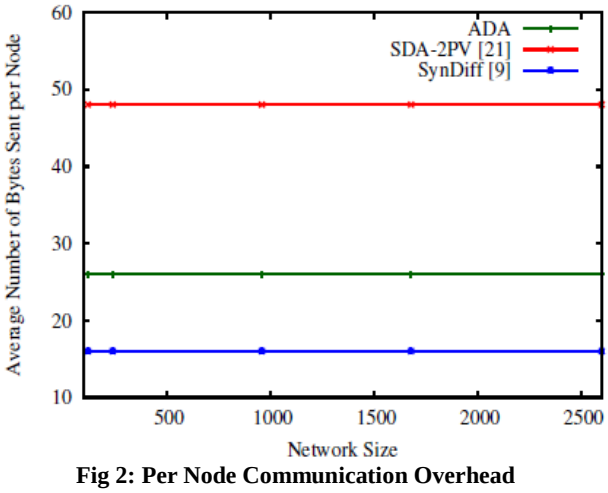

Fig 2: Per Node Communication Overhead

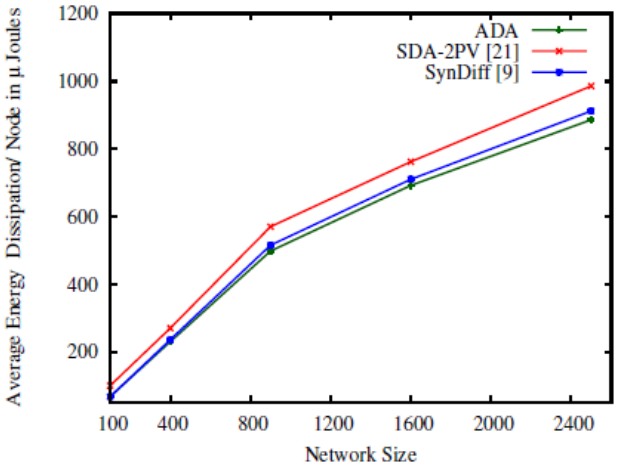

Fig 3: Energy Consumption per Data Collection Round in the Absence of attack

6.3 Energy Consumption per Data Collection Round

The main source of energy loss in sensor nodes is data communication. More precisely transmission consumes more energy in comparison with reception. Figure 3 shows average energy expended in transmission without any attack. The average energy dissipation of ADA is least among the three algorithms due to adaptive tier. The adaptive tier uses TDMA where nodes adjacent nodes transmit their data in alternate time slots. But in adaptive rings, all the nodes in same ring transmit multiple times simultaneously to provide resilience to communication failure. This retransmission increase the communication cost and hence the energy consumed. SynDiff consumes least energy in comparison to ADA and SDA-2PV. The smaller the size of synopsis packet, lesser is the energy consumption. The energy consumed is uniform throughout its operation.

6.4 Impact of Inflation Attack on Final Aggregate Computed

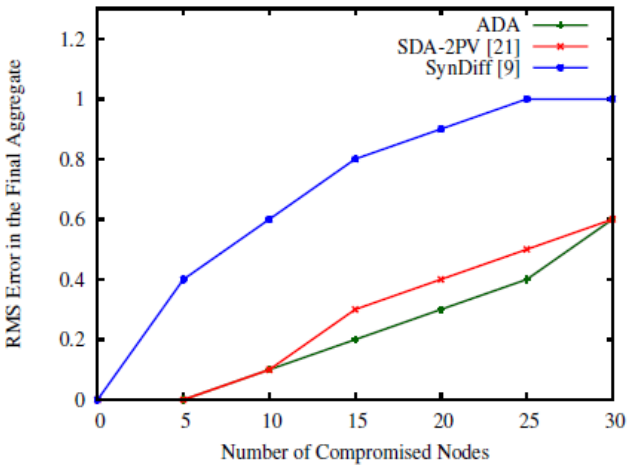

Fig 4: Impact of Inflation Attack over Aggregated Result

Figure 4 shows the impact of the percentage of compromised nodes over the Root Mean Square (RMS) error. As the number of compromised nodes increases, the RMS Error increases. The lower the value of RMS Error, the better is the performance of the algorithm. Out of the three algorithms, Syndiff is most susceptible to inflation attack. In ADA since the MAC send by each node are verified by its parent node, RMS Error is less when compared to SynDiff. But as the percentage of compromised node increases the performance deteriorates. When compared to SDA-2PV, the ADA provides almost equal security at lesser communication cost by performing a double verification at each node.

6.5 Impact of Compromised Nodes on Number of Bytes Sent per Node

To analyse the impact of compromised nodes on communication overhead, the average number of bytes sent was analysed per node during each data collection round as shown in Figure. 5. The average number of bytes sent per node in ADA and SynDiff are constant and does not increase with increase in number of compromised nodes. In case of SDA-2PV, for each 1 bit the node is contributing, Index of the 1 bit and MAC authenticating the 1 bit is transmitted. Hence when inflation attack is launched the number of 1 bits transmitted increases and hence the number of Indices and MACs resulting in an overall increase in average number of bytes sent per node.

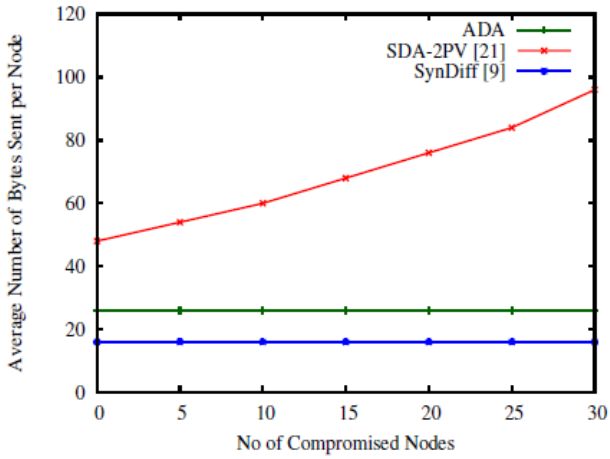

Fig 5: Impact of Compromised Nodes over Number of Bytes Sent

7. CONCLUSIONS

The synopsis diffusion framework is robust to communication failure. ADA uses a modified version of synopsis diffusion framework called adaptive tier that utilizes combines TDMA with adaptive rings [9], [21]. It can withstand node compromises to a great extent by using both synopsis verification and Message Authentication Code to ensure data integrity. It incurs low communication and computation overhead and low energy consumption resulting in enhanced lifetime of the WSNs.

8. REFERENCES

[1] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, pp. 2– 16, 2003.

[2] V. Handziski, A. K¨opke, H. Karl, C. Frank, and W. Drytkiewicz, "Improving the Energy Efficiency of Directed Diffusion using Passive Clustering," *Wireless Sensor Networks*, pp. 172–187, 2004.

[3] F. Dargahi, A. Rahmani, and S. Jabehdari, "Nodes' Credit Based Directed Diffusion for Wireless Sensor Networks," *International Journal of Grid and Distributed Computing*, vol. 1, no. 1, pp. 39–47, 2008.

[4] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate Aggregation Techniques for Sensor Databases," *20th International Conference on Data Engineering*, pp. 449–460, 2004.

[5] Y.-C. Fan and A. L. Chen, "Efficient and Robust Sensor Data Aggregation using Linear Counting Sketches," *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*, pp. 1–12, 2008.

[6] Y.-C. Fan and A. L. Chen, "Efficient and Robust Schemes for Sensor Data Aggregation based on Linear Counting," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 11, pp. 1675–1691, 2010.

[7] Y.-C. Fan and A. L. Chen, "Energy Efficient Schemes for Accuracy-Guaranteed Sensor Data Aggregation using Scalable Counting," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 8, pp. 1463–1477, 2012.

[8] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks," *Proceedings of International Conference on Embedded Network Sensor Systems (SenSys)*, pp. 250–262, 2004.

[9] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis Diffusion for Robust Aggregation in Sensor Networks," *ACM Transactions on Sensor Networks (TOSN)*, vol. 4, no. 2, p. 7, 2008.

[10] S. Tarannum, B. Aravinda, L. Nalini, K. Venugopal, and L. Patnaik, "Routing protocol for lifetime maximization of wireless sensor networks," *International Conference on Advanced Computing and Communications (ADCOM)*, pp. 401–406, 2006.

[11] S. Manjula, C. Abhilash, K. Shaila, K. Venugopal, and L. Patnaik, "Performance of AODV Routing Protocol using Group and Entity Mobility Models in Wireless Sensor Networks," *Proceedings of the International MultiConference of Engineersand Computer Scientists*, vol. 2, pp. 1212–1217, 2008.

[12] A. Kanavalli, D. Sserubiri, P. D. Shenoy, K. Venugopal, and L. Patnaik, "A Flat Routing Protocol for Sensor Networks," *Proceeding of International Conference on Methods and Models in Computer Science (ICM2CS)*, pp. 1–5, 2009.

[13] U. Prathap, D. P. Shenoy, K. Venugopal, and L. Patnaik, "Wireless Sensor Networks Applications and Routing Protocols: Survey and Research Challenges," *International Symposium on Cloud and Services Computing (ISCOS)*, pp. 49–56, 2012.

[14] M. Garofalakis, J. M. Hellerstein, and P. Maniatis, "Proof Sketches:Verifiable In-Network Aggregation," *IEEE 23rd International Conference on Data Engineering (ICDE)*, pp. 996–1005, 2007.

[15] S. Nath, H. Y, and H. Chan, "Secure Outsourced Aggregation via One-way Chains," *ACM SIGMOD International Conference on Management of data, SIGMOD'09*, pp. 31–44, 2009.

[16] Y. Yang, X. Wang, S. Zhu, and G. Cao, "Sdap: A Secure HopbyHop Data Aggregation Protocol for Sensor Networks," *ACM Transactions on Information and System Security (TISSEC)*, vol. 11, no. 4, pp. 18–43, 2008.

[17] B. Chen and H. Yu, "Secure Aggregation with Malicious Node Revocation in Sensor Networks," *31st International Conference on Distributed Computing Systems (ICDCS)*, pp. 581–592, 2011.

[18] S. Papadopoulos, A. Kiayias, and D. Papadias, "Exact Innetwork Aggregation with Integrity and Confidentiality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 10, pp. 1760–1773, 2012.

[19] S. Roy, S. Setia, and S. Jajodia, "Attack-Resilient Hierarchical Data Aggregation in Sensor Networks," *Proceedings of the Fourth ACM Workshop on Security of Ad hoc and Sensor Networks*, pp. 71–82, 2006.

[20] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure Data Aggregation in Wireless Sensor Networks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1040–1052, 2012.

[21] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure Data Aggregation in Wireless Sensor Networks: Filtering out the Attackers Impact," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 4, pp. 681–694, 2014.

[22] P. Flajolet and G. N. Martin, "Probabilistic Counting Algorithms for Data Base Applications," *Journal of computer and system sciences*, vol. 31, no. 2, pp. 182–209, 1985.

[23] E G Prathima, T Shiva Prakash, K R Venugopal, S S Iyengar and L M Patnaik, "SADA: Secure Approximate Data Aggregation in Wireless Sensor Networks," *Proceedings of IEEE International Conference of Data Science and Engineering (ICDSE)*, pp. 46-51, 23-25 August 2016.

9. AUTHOR PROFILE

E G Prathima is pursuing her Ph.D in the Department 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.

Dr. Shiv Prakash T, Professor, Department of Computer Science and Engineering at Vijaya Vittala Institute of Technology, Bangalore, India. He obtained his Ph.D, M.S and B.E in Computer Science and Engineering from Bangalore University, Bangalore. He has over ten years of IT experience in the field of Embedded Systems and Digital Multimedia. His research areas include Wireless Sensor Networks, Computer Vision, Embedded Linux and Digital Multimedia.

Venugopal K R is currently the Principal, University Visvesvaraya College of Engineering, Bangalore University, Bangalore. 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 is a Fellow of IEEE. He has received ACM distinguished educator award. 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.

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.