Security of Aggregated Data in Wireless Sensor Network

Surabhi Singhal

***Abstract*— A Wireless Sensor Network can be defined as a group of sensors which are distributed spatially to monitor physical or spatial conditions such as temperature, volcano, fire monitoring, sound, urban sensing, pressure etc. In a large WSN, the data aggregation significantly reduces communication overhead and energy consumption.**

In order to pass data, although data in-network aggregation was used and it reduced the problem of communication overhead and transmission loss but failed in computing double-counting sensitive aggregates at the Base Station. The research community proposed synopsis diffusion to eliminate this problem but it did not helped in securing the network against the problem of attacks caused by the compromised nodes, resulting in the false computation of aggregate. In this paper, synopsis diffusion is being made secure against the attacks by compromised nodes. To do so, an algorithm is being presented which can securely compute aggregates in the presence of such attacks. This algorithm is named as Attack-Resilient algorithm. The attack-resilient algorithm computes the true aggregate by filtering out the contributions of compromised nodes in the aggregation hierarchy.Extensive studies and performance analysis have shown that the proposed algorithm i.e. Attack-Resilient algorithm is more effective and outperforms other existing approaches.

***Index Terms*— attack-resilient, data aggregation, falsified sub-aggregate, in-network aggregation, synopsis diffusion**

I. INTRODUCTION

A Wireless Sensor Network can be defined as a group of sensors which are distributed spatially to monitor physical or spatial conditions such as temperature, volcano, fire monitoring, sound, urban sensing, pressure etc.
In order to pass data from a node to the base station, the nodes transmit their data by forming a multi-hop network, thus passing their data to the base station through the intermediate nodes. But this method was inefficient due to limited battery life and communication overhead.

To resolve this, firstly, TAG i.e." a tiny aggregation service ad hoc sensor networks" [5] and "computing aggregates for ad hoc networks" [6] were implemented. These involved aggregating the intermediate data before passing it to the base station. One of the approaches to implement this was constructing the minimum spanning tree rooted at the base station. The use of multipath routing also helped in reducing the problem of communication and transmission losses.
These effectively were used for various aggregates such as Sum, Count, Average, Min, Max, Standard Deviation and Statistical Moment of any order. Since those aggregates which are duplicate-insensitive such as Min, Max, TAG is very effective. But for duplicate-sensitive aggregates such as Count, Sum, multipath leads to double-counting problem.
Several researchers, then, came with techniques such as "approximate aggregation techniques for sensor databases" [7], "synopsis diffusion for robust aggregation in sensor networks" [8]. The researchers of both [7], [8] used more efficient framework called Synopsis Diffusion.

In this technique, the ring topology was used where a may have multiple parents in the aggregation hierarchy. Also, in order to solve the count duplicity problem, the sensed value of each node or the sub-aggregate value is represented by a duplicate-insensitive bitmap called synopsis.

Although the synopsis diffusion helped in solving the computation of duplicate – sensitive aggregates, but there is a need to make it secure against various challenges posed by the compromised node. A compromised node is a node which exhibits an arbitrary behaviour and may collude with other compromised nodes. These nodes, thus pose a security threat to the wireless network (synopsis diffusion).

A compromised node being distributed uniformly in a network can attack in various possible ways such as message-fabrication,jamming,etc.
In this paper, we are considering a particular attack caused by the compromised node i.e. falsifying the local value or the sub-aggregate value thus causing the BS to calculate incorrect aggregate.
So, in this paper, the researchers are trying to secure the synopsis diffusion by implementing the Attack-resilient computation algorithm, thus making possible for the base station to securely compute the aggregate in the presence of an attack.
Although, previously various algorithms have been introduced such as [12], [13], [19], [21], but they proved to be inefficient for successful computation of aggregates in the presence of an attack. Also, the proposed algorithm does not include the DOS attacks.

A. Falsified sub-aggregate attack

In algorithms [7],[8],during the computation of aggregates , a compromised node X can add a small amount of error in the final estimate of Sum by falsifying its own sub-aggregate. This attack is called as the falsified sub-aggregate attack.

B. Attack-resilient computation algorithm

In order to compute aggregates securely, such as Count and Sum, despite the falsified sub- aggregates attack, an algorithm is being proposed. The name given to this algorithm is called attack-resilient computation algorithm.

Surabhi Singhal is the author of this paper. She has done her B.tech in Computer Science from Galgotias College of Engineering and Technology. This is her first paper.

II. SYNOPSIS DIFFUSION

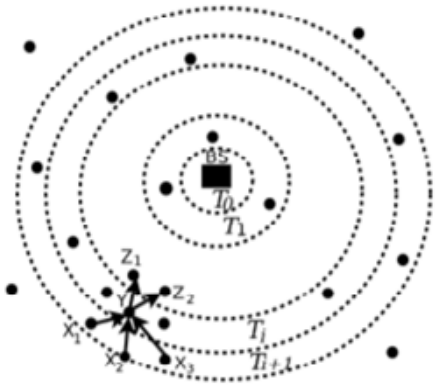

Fig.1 Synopsis Diffusion over a ring topology

The synopsis diffusion uses ring topology as well as multipath routing and helps in calculating duplicate sensitive aggregates even in the presence of the attacks caused by compromised nodes distributed uniformly in the network. The synopsis diffusion is shown in fig.1. Here, when the ring formation phase called 'query distribution phase' starts, nodes form a set of rings around the base station (BS) based on their distance in terms of hops from BS. As can be seen from the figure, T_i denotes the ring consisting of the nodes which are i hops away from BS. After this, the data aggregation as well as transmission starts from the outermost ring to the BS. Each node generates as well as broadcasts its local synopsis.

The synopsis diffusion includes mainly three functions, namely, Synopsis Generation function, SG (v); Synopsis Fusion function, SF (v) and Synopsis Evaluation function, SE (v), where v is the sensor value relevant to the query. Each node generates and broadcasts its local synopsis using SG (v). The SF (v) is used to combine the local data of a node in a ring as well as the data received from the previous ring. This can be explained through the fig.1., where a node in ring T_i, receives data from the nodes in its communication range in ring T_{i+1} and combines it with its own data using the fusion function, SF (v) and then further broadcast this fused synopsis until it reaches the BS where again it combines this received synopsis using SF (v). SE (v) function is used finally to translate the final synopsis to answer the query.

A node X's fused synopsis, B^X, is recursively defined as follows. If X is a leaf node (i.e., X is in the outermost ring), B^X is its local synopsis Q^X. If X is a non- leaf node and suppose it receives synopses B^X_1, B^X_2,...,B^X_d from d child nodes X1, X2, ... , Xd, respectively, then X computes B^X as follows:

$$B^X = Q^X \| B^X_1 \| B^X_2 \| ... \| B^X_d$$

Where || denotes the bitwise OR operator and B^X represents the sub-aggregate of node X, including its descendant nodes.

A. *Assumptions*

It is assumed that BS cannot be compromised. Also compromised nodes are distributed uniformly.
Besides this, each node shares a pair-wise key with BS. Let the key of the node with ID X be denoted as K_X. To authenticate a message to BS, a node X sends a MAC (Message Authentication Code) generated using the key K_X. We further assume that each pair of neighbouring nodes has a pair-wise key to authenticate its mutual communication.

B. *Goal*

The goal of this paper mainly includes two major points: (a) the first goal is to detect if ^B, the synopsis received at BS is the same as the 'true' final synopsis B,(b) the second goal is to compute B from $\hat{B}$, and other received information.
Here, we are considering the Sum aggregate (if not otherwise specified). As Count is a special case of Sum, the algorithm mentioned in this paper is also applicable to the Count aggregate also.
A comparison between the proposed algorithm and previous works is shown later.

C. *The Attack Details*

Since the lowest-order bit z, i.e. '0' in the final synopsis is estimated for the aggregate by the BS, so a compromised node C tends to falsify its data, B^C , in such a way that it would affect the value of z. The node C does so by injecting 1s in one or more bits in positions j, where $z \le j \le \eta$, into B^c which C broadcasts to its parents. Let $\hat{B}^C$ denote the synopsis finally broadcast by node C.

Besides this, the synopsis fusion function is a bitwise Boolean OR so, the fused synopsis computed at any node which is at the higher level than node C on the aggregation hierarchy will contain the false contributions of node C.

The '1' bits which are present in $\hat{B}$ but not in B are considered as false '1's in the rest of this paper.

C can attack by introducing a false '1' at bit j in $B^{\wedge C}$ through any of the following two attacks:

(a) Falsified sub-aggregate attack: in this attack C flips bit j in $B^{\wedge C}$ from '0' to '1', disallowing the local aggregate to justify that '1' in the synopsis $B^{\wedge C}$.

(b) Falsified local value attack: in this attack, C injects a false '1' at bit j in its Q^C . This falsified $Q^{\wedge C}$ thus induces the j bit ib $B^{\wedge C}$ to be '1'.

Fig.2 below shows an example of falsified sub-aggregate attack.

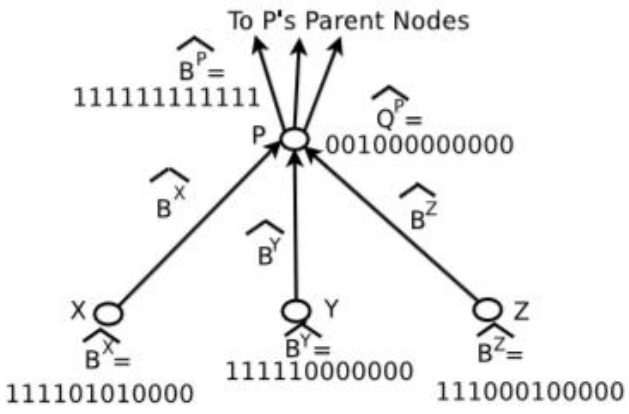

Fig.2. Falsified Sub-aggregate Attack

In the above fig., P node has three child X, Y and Z . Consider P as the compromised node and it receives synopses $B^{\wedge X}$, $B^{\wedge Y}$ and $B^{\wedge Z}$ respectively. The local synopsis of P is Q^P. So, the fused synopsis $B^{\wedge P}$ will be :

$B^{\wedge P} = Q^P \| B^{\wedge X} \| B^{\wedge Y} \| B^{\wedge Z}$.

Let $\hat{}\, R = \hat{}\, z-1$, where $\hat{}\, z$ be the lowest-order bit that is '0' in the received final synopsis $\hat{}\, B$. Also, let $R = z-1$, where z is the lowest-order bit that is '0' in the correct final synopsis B. Then BS's estimate of the aggregate will be larger than the correct estimate by a factor of $2 \hat{}\, R-R$. So, a large amount of error will appear in the final estimate of BS.

III. COMPUTING SUM DESPITE ATTACK

As mentioned, here, the attack-resilient algorithm is explained to compute duplicate-sensitive aggregates in the presence of attacks caused by compromised node (X).The node attacks by inserting one or more 1's in the local value or the sub-aggregate value.

An obvious solution to guard against this attack is as follows. BS broadcasts an aggregation query message containing a random value i.e. Seed which is associated to the current query. After this, the sub-aggregation phase starts in which every node X, besides $B^{\wedge X,}$ also sends a MAC (Message Authentication Code) to the Base Station, thus authenticating its sensed value v_x. Generally, every node uses Seed and its own ID to compute its MAC Node X uses Seed and its own ID to compute its MAC. As a result, BS is able to detect and filter out any false '1' bits inserted in the final synopsis B.

A. *Introducing Message Authentication Code (MAC)*

The Message Authentication Code (MAC) is generated as follows: if X contributes to bits $b_1,b_2,...,b_\zeta$ in its local synopsis Q^X, it generates a MAC, $M = MAC(K_X, L)$, where K_X is the key that node X shares with BS and the content of L is $< X, v_X , b_1, b_2,..., b_\zeta,$ Seed $>$. Each node X sends a message (L', M) where $L' =< X, v_X, b_1, b_2,..., b_\zeta >$ might be needed by BS to regenerate the MAC for the verification. It is observed that this approach is not suitable for a WSN as it requires O(N) MACs to be forwarded to BS. The attack-resilient algorithm presented below also uses similar MACs but reduces the total number of them.

Also, when we say that a message contains MAC, M, it is understood that L' is already present.

A false MAC can be associated either to a false '1' or to a non-false '1' bit. Specifically, a compromised node X can generate a false MAC (in the context of computing the function $MAC(K_X, L)$) in four ways—(i) by using a false L, (ii) by using a false key KX, (iii) by doing both of (i) and (ii) above, or (iv) by simply sending a bogus array of bits. As BS re-executes the MAC generation process for each received MAC, any false MAC will be detected by BS.

B. *Notations*

Let $M_X{}^i$ denotes the MAC, generated by node X, authenticating the i-th bit of its local synopsis Q^X. Note that $M^X{}_i$ is required to be generated only if $Q^X[i]=1$, i.e. there are no MAC for '0' bits. Furthermore, for a particular i, Mi denotes one arbitrary element of the following set: $\{M_i^X \mid Q^X[i] =1\}$, where elements of the set are enumerated with respect to X. As an example, if two nodes X_1 and X_2 set bit i to be '1' in their local synopses, then Mi corresponds to either $M_i{}^X{}_1$ or $M_i{}^X{}_2$. We assume that a node X's message to one of its parents, P, can be lost due to communication failure but it cannot be partially or wrongly received—node-to-node authentication and acknowledgement mechanisms can be used to enforce this property. It implies that if B^X reaches P, all of the MACs sent by X also reach P.

C. *The Main Idea of the Protocol*

Before discussing the attack-resilient protocol, lets take a simpler protocol where each node X forwards one MAC for each of the '1' bits in $B^{\wedge X}$ and BS will verify all of the final synopsis received $B^{\wedge}$. If a compromised node, C injects a false MAC for in few '1' bits. Then, with some probability, these false MACs may get selected at each hop before reaching BS. If for a bit in final synopsis B, say bit i, BS does not receive a valid MAC but only false MACs, then BS cannot determine the real state of bit i. In fact, this can be the consequence of either of the following two scenarios: (i) B[i]=0 and a false MAC has been generated; (ii) a source node (possibly a few hops away from BS) has sent a valid MAC for bit i (B[i]=1, indeed), but this MAC lost the race to false MACs in the random selection procedure en-route BS.

However, we observe that the probability of this 'undecided-ability' problem to arise is not the same for all of the bits. In fact, a false MAC is not equally likely to get selected for all of the bits because the number of source nodes that contribute to a bit (hence, the number of valid MACs) varies with the bit position.

Also, if the number of compromised nodes, t, is small compared to the total number of nodes, N, we expect that BS will receive a valid MAC for the left bits far from bit r, but may not receive a valid MAC for the other bits.

D. *Protocol Details*

An attack- resilient protocol having two phases as follows:

Phase One: run the simple protocol described above. First, BS broadcasts a query message:

BS→→∗:<"PhaseOne","Sum",Seed,η>

where "Phase One" is a flag indicating that phase one is going to begin, and η is the synopsis length.

In this phase, nodes basically execute the original synopsis diffusion algorithm (for the Sum aggregate) with the Seed being used in the hash function in the CoinToss i.e.

```
begin
Q^X[index]=0 ∀index, 1 ≤ index ≤η;
i=1;
while i ≤v_X do
 key_i =< X,i >;
 index = CoinToss(key_i ,η);
Q^X[index]=1;
i = i +1;
end
 return Q^X;
end
```

The nodes also do additional transmission of some MACs. In particular, each node X randomly selects one MAC for each '1' bit in synopses ˆ BX from the MACs received from its child nodes (possibly including X's own MAC). X forwards the selected MACs to its parents. The message broadcast by X to its parent nodes is as follows:

$X \rightarrow *: \hat{}BX, \{Mi | \hat{}BX[i]=1, 1 \leq i \leq \eta\}$

, where ˆ BX represents the fused synopsis at node X, Mi represents a MAC corresponding to ˆ B^x [i]. After all of the MACs have been received by BS, for any '1' bit, say bit ˆ Bi, in the synopses ˆ B for which no valid MAC has been received, BS resets ˆ Bi to '0'. The resulting set of synopses after this filtering process has been performed are denoted by ¯ B, respectively. Now, BS makes an estimate of the expected length of prefix of all '1's, r using ¯ B. Letˆ r be the estimate of r. We observe that there is one factor which could possibly deviate the estimate ˆ r from r: injection of false MACs by the adversary—which can cause BS not receiving any valid MAC for a few '1' bits near bit r in synopsis B. We observe that this factor could contribute to a deviation to the left only (i.e. making ˆr less than r)

Phase Two: BS requests the nodes which contribute to bits i, i > ˆ r, in the synopsis to send back the corresponding MACs. The message sent by BS is as follows:

$BS \rightarrow\rightarrow *: <$"PhaseTwo",ˆr>

where "PhaseTwo" is a flag indicating that phase two is going to begin. After receiving the request from BS, each node X broadcasts to its parents the MACs, $\{Mi \mid \hat{} r < i \leq \eta\}$. Unlike the first phase, now no MAC is dropped by the intermediate nodes, i.e, each node X forwards to X's parents all of the MACs X received from its child nodes. After BS receives the MACs, any bit Bi, i > ˆ r for which a valid MAC is received is set to '1'. The resulting synopsis is denoted by B'

Thus, in particular, it can be proved that BS can correctly infer the values of all of the bits in the synopsis. In other words, we show that when this protocol terminates, BS has already received at least one valid MAC for each '1' bit of the synopsis.

E. *Performance Analysis*

The communication overhead of phase one does not depend on the number of compromised nodes. The worst case per-node communication burden is to forward l MACs, where l is the maximum number of '1's in the synopsis. As per the property of Sum synopsis, we know that l is approximately log2S, S being the Sum. That means the communication overhead per node is O(log2 S). On the other hand, the communication overhead of phase two is determined by how close the estimate ˆ r, obtained in phase one, is to the real value of r.

Furthermore, the probability that B[i]=0 is determined by only the distance of the i-th bit from the r-th bit, where the value of r is $\log_2(\varphi S)$.

IV. RESULTS AND DISCUSSION

A. *Error in Estimate r^*

The performance of the above protocol depends on the looseness of the estimate, r^ as mentioned in the phase 1. Furthermore, the maximum deviation in estimate ˆ r from correct r (which is obtained in phase two) depends on how many compromised nodes participate in the false MAC injection attack during phase one. The analysis mentioned above states that the deviation obeys the following inequality with high probability, $(r - \hat{} r) \leq \log 2\ \varphi t+1$ wheret is the number of compromised nodes. For any particular value of t (0, 25, 50, 100, 200, and 400), simulation of the false MAC injection attack during phase one was done 300 times. We measured (r −ˆ r) for each t, and we observed that (r −ˆ r) was low as expected.

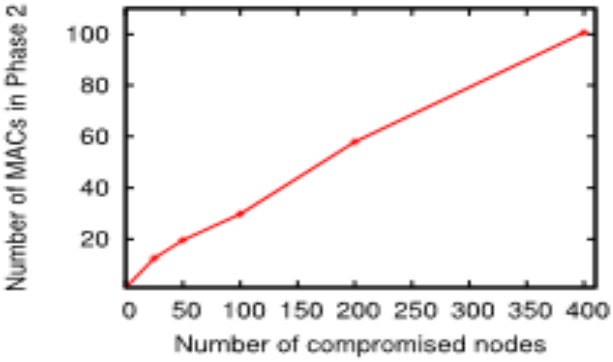

Fig.3. The total no. Of MAC forwarded in Phase two

The above Fig. 3 illustrates how this deviation (r −ˆ r) varies with t. 99% confidence intervals are within ±10% of the reported value.

B. *Worst-Case Communication Overhead*

During phase one a node needs to forward at most η MACs regardless of its position, where η is the length of the synopsis. This overhead cannot be reduced because (in the worst case) the compromised nodes can always inject false MACs for each of the η bits.

On the other hand, in phase two, a node (in the worst case, i.e., near BS) needs to forward O(t) MACs as per the analysis mentioned above, where t is the number of compromised nodes.

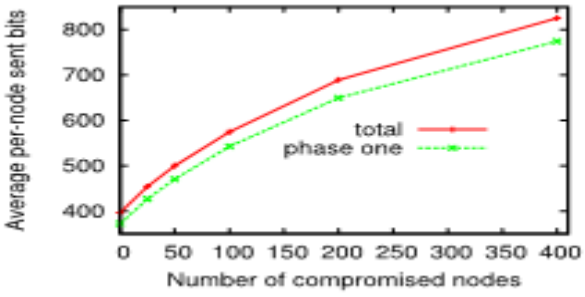

Fig.4. The average per node communication overhead

The above Fig. 4 plots the number of unique MACs sent over the whole network during phase two as a function of t. The 99% confidence intervals are within ±20% of the reported value. We observe that the number of MACs increases linearly with t, which confirms the analysis.

V. CONCLUSION

Firstly, the security issues of in-network aggregation algorithms to compute aggregates such as predicate Count and Sum were discussed. In particular, the falsified sub-aggregate attack launched by a few compromised nodes which can inject arbitrary amount of error in the base station's estimate of the aggregate, were shown. An attack-resilient computation algorithm was explained so as to guarantee the successful computation of the aggregate even in the presence of the attack.

REFERENCES

[1] M. Liu, N. Patwari, and A. Terzis, "Scanning the issue," Proc. IEEE, vol. 98, no. 11, pp. 1804–1807, Apr. 2010.
[2] T. Ko, J. Hyman, E. Graham, M. Hansen, S. Soatto, and D. Estrin, "Embedded imagers: Detecting, localizing, and recognizing objects and events in natural habitats," Proc. IEEE, vol. 98, no. 11, pp. 1934–1946, Nov. 2010.
[3] P. Corke, T. Wark, R. Jurdak, W. Hu, P. Valencia, and D. Moore, "Environmental wireless sensor networks," Proc. IEEE, vol. 98, no. 11, pp. 1903–1917, Nov. 2010.
[4] (2006). James Reserve Microclimate and Video Remote Sensing [Online]. Available: http://research.cens.ucla.edu/ projects/2006/terrestrial/microclimate/defau%lt.htm
[5] S. Madden, M. J. Franklin, J. Hellerstein, and W. Hong, "TAG: A tiny aggregation service for ad hoc sensor networks," in Proc. 5th USENIX Symp. Operating Syst. Des. Implement., 2002, pp. 1–3.
[6] J. Zhao, R. Govindan, and D. Estrin, "Computing aggregates for monitoring sensor networks," in Proc. 2nd Int. Workshop Sensor Netw. Protocols Appl., 2003, pp. 139–158.
[7] J. Considine, F. Li, G. Kollios, and J. Byers, "Approximate aggregation techniques for sensor databases," in Proc. IEEE 20th Int. Conf. Data Eng. (ICDE), 2004, pp. 449–460.
[8] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis diffusion for robust aggregation in sensor networks," in Proc. 2nd Int. Conf. Embedded Netw. Sensor Syst. (SenSys), 2004, pp. 250–262.
[9] M. Garofalakis, J. M. Hellerstein, and P. Maniatis, "Proof sketches: Verifiable in-network aggregation," in Proc. 23rd Int. Conf. Data Eng. (ICDE), 2007, pp. 996–1005.
[10] Y. Yang, X. Wang, S. Zhu, and G. Cao, "SDAP: A secure hop-by- hop data aggregation protocol for sensor networks," in Proc. ACM MOBIHOC, 2006, pp. 356–367.
[11] H. Yu, "Secure and highly-available aggregation queries in large-scale sensor networks via set sampling," in Proc. Int. Conf. Inf. Process. Sensor Netw., 2009, pp. 1–12.
[12] S. Roy, M. Conti, S. Setia, and S. Jajodia, "Secure data aggregation in wireless sensor networks," IEEE Trans. Inf. Forensics Security, vol. 7, no. 3, pp. 1040–1052, Jun. 2012.
[13] S. Roy, S. Setia, and S. Jajodia, "Attack-resilient hierarchical data aggregation in sensor networks," in Proc. ACM Workshop Security Sensor Adhoc Netw. (SASN), 2006, pp. 71–82.
[14] M. B. Greenwald and S. Khanna, "Power-conservative computation of order-statistics over sensor networks," in Proc. 23th SIGMOD Principles Database Syst. (PODS), 2004, pp. 1–11.
[15] P. Flajolet and G. N. Martin, "Probabilistic counting algorithms for data base applications," J. Comput. Syst. Sci., vol. 31, no. 2, pp. 182–209, 1985.
[16] L. Hu and D. Evans, "Secure aggregation for wireless networks," in Proc. Workshop Security Assurance Ad Hoc Netw., 2003, pp. 384–391.
[17] H. Chan, A. Perrig, and D. Song, "Secure hierarchical in-network aggregation in sensor networks," in Proc. ACM Conf. Comput. Commun. Security (CCS), 2006, pp. 278–287.
[18] B. Chen and H. Yu, "Secure aggregation with malicious node revocation in sensor networks," in Proc. 31st Int. Conf. Distrib. Comput. Syst. (ICDCS), 2011, pp. 581–592.
[19] D. Wagner, "Resilient aggregation in sensor networks," in Proc. ACM Workshop Security Sensor Adhoc Netw. (SASN), 2004, pp. 68–79.
[20] L. Buttyan, P. Schaffer, and I. Vajda, "Resilient aggregation with attack detection in sensor networks," in Proc. 2nd IEEE Workshop Sensor Netw. Syst. Pervasive Comput., Mar. 2006, pp. 331–336.
[21] B. Przydatek, D. Song, and A. Perrig, "SIA: Secure information aggrega- tion in sensor networks," in Proc. 1st Int. Conf. Embedded Netw. Sensor Syst. (SenSys), 2003, pp. 255–265.
[22] K. Frikken and J. A. Dougherty, "An efficient integrity-preserving scheme for hierarchical sensor aggregation," in Proc. 1st ACM Conf. Wireless Netw. Security (WiSec), 2008, pp. 68–76.
[23] S. Nath, H. Yu, and H. Chan, "Secure outsourced aggregation via one-way chains," in Proc. 35th SIGMOD Int. Conf. Manag. Data, 2009, pp. 31–44.
[24] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in Proc. Int. Conf. Mobile Comput. Netw. (MobiCOM), 2001, pp. 189–199.
[25] W. He, X. Liu, H. Nguyen, K. Nahrstedt, and T. F. Abdelzaher, "PDA: Privacy-preserving data aggregation in wireless sensor networks," in Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM), May 2007, pp. 2045–2053.
[26] S. Roy, M. Conti, S. Setia, and S. Jajodia. (2013). Secure Data Aggregation in Wireless Sensor Networks: Filtering out the Attacker's Impact. An Extended and Online Version [Online]. Available: http://people.cis.ksu.edu/~sroy/attackResilientAgg.pdf

Surabhi Singhal is the author of this paper. She has done her B.tech in Computer Science from Galgotias College of Engineering and Technology. This is her first paper.