# SECURE DATA AGGREGATION IN WIRELESS SENSOR NETWORKS USING RANDOMIZED DISPERSIVE ROUTES

**Vijay Kumar S**

**ABSTRACT**

In a Wireless sensor network, specifically data aggregation reduces the amount of communication and energy utilization. Research community has proposed a strong aggregation framework called synopsis diffusion which combines multipath routing schemes with duplicate-insensitive algorithms to perfectly compute aggregates (e.g., predicate Count, Sum) unkindness of message losing results from node and communication failures. But this aggregation framework does not solve the problems which are appearing at base station side. These problems may occur due to the irrespective of the network size, the per node communication over-head. In this paper, we make the synopsis diffusion approach secure against attacks in which compromised nodes put in false sub aggregate values. In particular, we present a novel lightweight verification algorithm by which the base station can determine if the computed aggregate (predicate Count or Sum) includes any false input.

In this paper, we study the compromised node and denial-of-service is the two key attacks in wireless sensor networks. These attacks are disagreeing that multipath routing approaches are highly helpless. So, for we develop the mechanisms that generate randomized multipath routes. In this designing, the routes are taken by the shares of dissimilar packets change over time. So, we analytically examine the security and energy performance of proposed schemes.

**Keywords** Sensor Networks, Aggregation, Security, Base Station, Randomized Multipath Routing.

## I. INTRODUCTION

In a WIRELESS sensor network (WSN) different types of security problems are encountered. In this paper, we are exclusively warfare with two types of attacks: compromised node (CN) and denial of service (DOS).In the CN attack, a follower actually compromises a subset of nodes to eavesdrop information, whereas in the DOS attack, the adversary interferes with the normal operation of the network by actively disrupting, changing, or even paralyzing the functionality of a subset of nodes. These two attacks are similar in the sense that they both generate black holes: areas within which the adversary can either passively intercept or actively block information delivery. CN and DOS attacks can disturb normal data delivery between sensor nodes and the sink, or even partition the topology. Likewise, an adversary can always perform DOS attacks (e.g., jamming) even if it does not have any knowledge of the underlying cryptosystem.

One remedial solution to these attacks is to exploit the in-network's routing functionality. It should be locating the black holes are as priori, if the data can be delivered over paths that bypass these holes, whenever possible. The above idea is implemented in a probabilistic manner, typically through a two-step process. First, the packet is divided into M shares (i.e., components of a packet that carry partial information) using a Shamir's algorithm [13]. The original aggregation can be recovered from a combination of at least T shares, but no aggregation can be guessed from less than T shares. Second, multiple routes from the source to the destination are computed according to some multipath routing algorithms such as distance routing algorithm, optimum routing algorithm. These routes are node-disjoint or maximally node-disjoint subject to certain constraints (e.g., min-hop routes). The M shares are then distributed over these routes and delivered to the destination. As longer as at least T shares bypass the compromised (or jammed) nodes, the adversary cannot acquire (or deny the delivery of) the original packet.

We argue that three security problems exist as following: approaches. First, this approach is no longer valid if the supporter can selectively compromise or jam nodes. It is the route computation in the above multipath routing algorithms is deterministic in the sense of given topology and given source and destination nodes are always computed by the routing algorithm. As a result, once the routing algorithm becomes known to the rival (this can be done, e.g., through memory cross-examination of the compromised node), the rival can compute the set of

routes for any given source and destination. Second, actually very few node-disjoint (min-hop) routes can be found when the node density is moderate and the source and destination nodes are several hops apart. For example, for a node degree of 8, on average only two node-disjoint routes can be found between a source and a destination that are at least 7 hops apart. The lack of sufficient routes much undermines the security performance of this multipath approach. Last, because the routes are computed under certain constraints, the routes may not be spatially dispersive enough to circumvent a moderate-size black hole.

In a WSNs are increasingly used in several applications, such as wild habitat monitoring, forest fire detection, and military surveillance. After being deployed in the field of interest, sensor nodes organize themselves into a multi-hop network with the base station as the central point of control. Typically, an aggregate (or summarized) value is computed at the data sink by applying the corresponding aggregate function, e.g., predicate count and sum to the collected data. A straightforward method to collect the sensed information from the network is to allow each sensor node's reading to be forwarded to the base station, possibly via other intermediate nodes, before the base station processes the received data. However, this method is prohibitively expensive in terms of communication overhead (or energy spent).

However, most of the existing in-network data aggregation algorithms have no provisions for security. A compromised node might attempt to thwart the aggregation process by launching several attacks, such as eavesdropping, jamming, message dropping, message fabrication, and so on. This paper focuses on one of the most vexing attacks: the *falsified subaggregate attack*, in which a compromised node relays a false subaggregate to the parent node with the aim of injecting error to the final value of the aggregate computed at the base station.

In this paper, we propose a randomized multipath routing algorithm that can overcome the above problems. In this algorithm, multiple paths are computed in a randomized way each time an aggregating packet needs to be sent, such that the set of routes taken by various shares of different packets remain altering over time. As a result, a large number of routes can be potentially generated for each source and destination.

However, the algorithm ensures that the randomly generated routes are as dispersive as possible, i.e., the routes are geographically separated as far as possible such that they have high probability of not concurrently passing through a black hole. A naive algorithm of generating

random routes, such as Wanderer scheme (a pure random-walk algorithm), only leads to long paths(containing many hops, and therefore, consuming lots of energy) without achieving good depressiveness.

## II. RELATED WORK

Several researchers have studied problems related to data aggregation in WSNs.

### *A. Data Aggregation Without Security*

The tiny aggregation service (TAG) to compute aggregates, such as Count and Sum, using tree-based aggregation algorithms were proposed in. Moreover, tree based aggregation algorithms to compute an order statistic have been proposed in. To address the communication loss problem in tree based algorithms the authors in designed an aggregation frame-work called *synopsis diffusion* to compute Count and Sum, which uses a ring topology and duplicate insensitive algorithms for computing aggregates based on the algorithm in for counting distinct elements in a multi-set.

### *B. Secure Aggregation Techniques*

Several secure aggregation algorithms have been proposed assuming that the base station is the only aggregator node in the network. It is not straightforward to extend these works for verifying in-network aggregation unless we direct each node to send an authentication message to the base station.

A tree-based verification algorithm was designed in by which the base station can detect if the final aggregate, Count or Sum, is falsified. We are unable to extend this idea for verifying a synopsis because the synopsis computation is duplicate insensitive. A verification algorithm for computing Count and Sum within the synopsis diffusion approach was designed in. Recently, a few novel protocols have been proposed for "secure outsourced aggregation".

Although algorithms in and our verification protocol prevent the base station from accepting a false aggregate, they do not guarantee the successful computation of the aggregate in the presence of the attack. Some researchers also designed *attack-resilient computation algorithms* to empower the base station to filter out the false contributions of the compromised nodes from the aggregate. The first attack-resilient hierarchical data aggregation protocol was designed in. However, this scheme is secure when only one malicious node is present. An attack-

resilient aggregation algorithm for computing Count and Sum has been proposed in, which is based on a sampling technique. This algorithm can produce an approximation of the target aggregate. We previously presented an attack-resilient aggregation algorithm for the synopsis diffusion framework. The verification protocol we propose in this paper has a very light overhead compared to all these attack resilient solutions.

***C. Randomized Multipath Delivery***

**1. Overview**

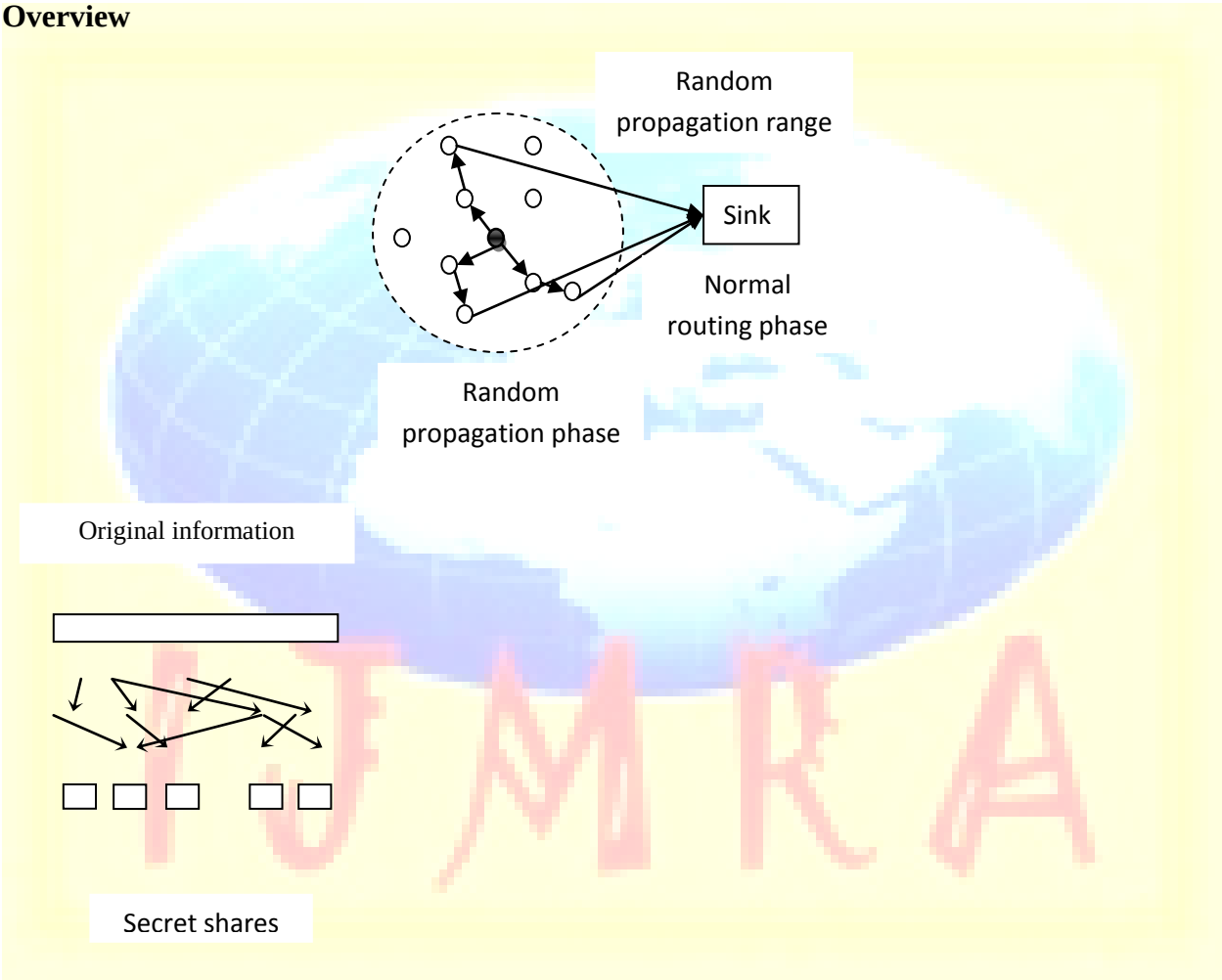
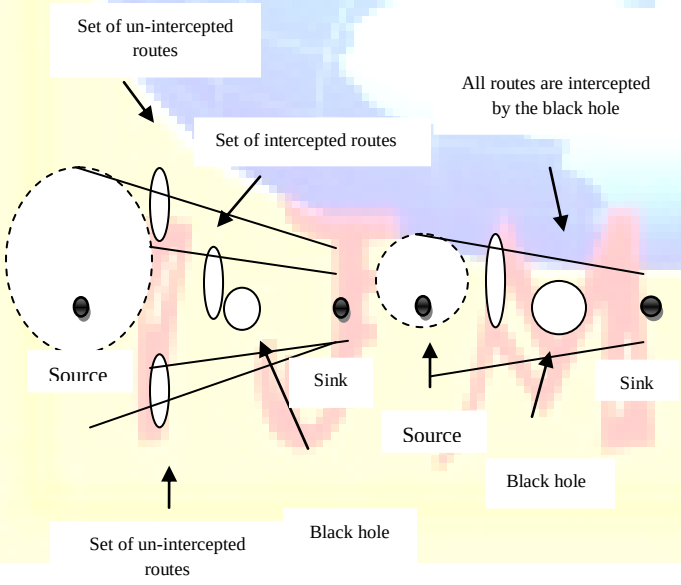



Fig.2 Randomized dispersive routing in a WSN.

As illustrated in Fig. 2, we consider a three-phase approach for secure information delivery in a WSN: secret sharing of information, randomized propagation of each information share, and normal routing (e.g., min-hop routing) toward the sink. More specifically, when a sensor node wants to send a packet to the sink, it first breaks the packet into M shares, according to a ðT; MÞ-threshold secret sharing algorithm, e.g., Shamir's algorithm. Each share is then transmitted

to some randomly selected neighbor. That neighbor will continue to relay the share it has received to other randomly selected neighbors, and so on. In each share, there is a TTL field, whose initial value is set by the source node to control the total number of random relays. After each relay, the TTL field is reduced by 1. When the TTL value reaches 0, the last node to receive this share begins to route it toward the sink using min-hop routing. Once the sink collects at least T shares, it can reconstruct the original packet. No information can be recovered from less than T shares.

The effect of route dispersiveness on bypassing black holes is illustrated in Fig. 3, where the dotted circles represent the ranges the secret shares can be propagated to in the random propagation phase. A larger dotted circle implies that the resulting routes are geographically more dispersive. Comparing the two cases in Fig. 3, it is clear that the routes of higher dispersiveness are more capable of avoiding the black hole. Clearly, the random propagation phase is the key component that dictates the security and energy performance of the entire mechanism.




(a) (b)

Fig: 3 Implication of route dispersiveness on bypassing the black hole. (a) Routes of higher dispersiveness. (b) Routes of lower dispersiveness.

## 2. Random Propagation of Information Shares

To diversify routes, an ideal random propagation algorithm would propagate shares as dispersively as possible. Typically, this means propagating the shares farther from their source. At the same time, it is highly desirable to have an energy-efficient propagation, which calls for limiting the number of randomly propagated hops. The challenge here lies in the random and distributed nature of the propagation:

### *2.1 Purely Random Propagation (Baseline Scheme)*

In Purely Random Propagation (PRP), shares are propagated based on one-hop neighborhood information. More specifically, a sensor node maintains a neighbor list, which contains the IDs of all nodes within its transmission range.

The main drawback of PRP is that its propagation efficiency can be low, because a share may be propagated back and forth multiple times between neighboring hops.

### *2.2 Non-Repetitive Random Propagation*

NRRP is based on PRP, but it improves the propagation efficiency by recording the nodes traversed so far. Specifically, NRRP adds a "node-in-route" (NIR) field to the header of each share. This non-repetitive propagation guarantees that the shares will be relayed to a different node in each step of random propagation, leading to better propagation efficiency.

### *2.2.3 Directed Random Propagation*

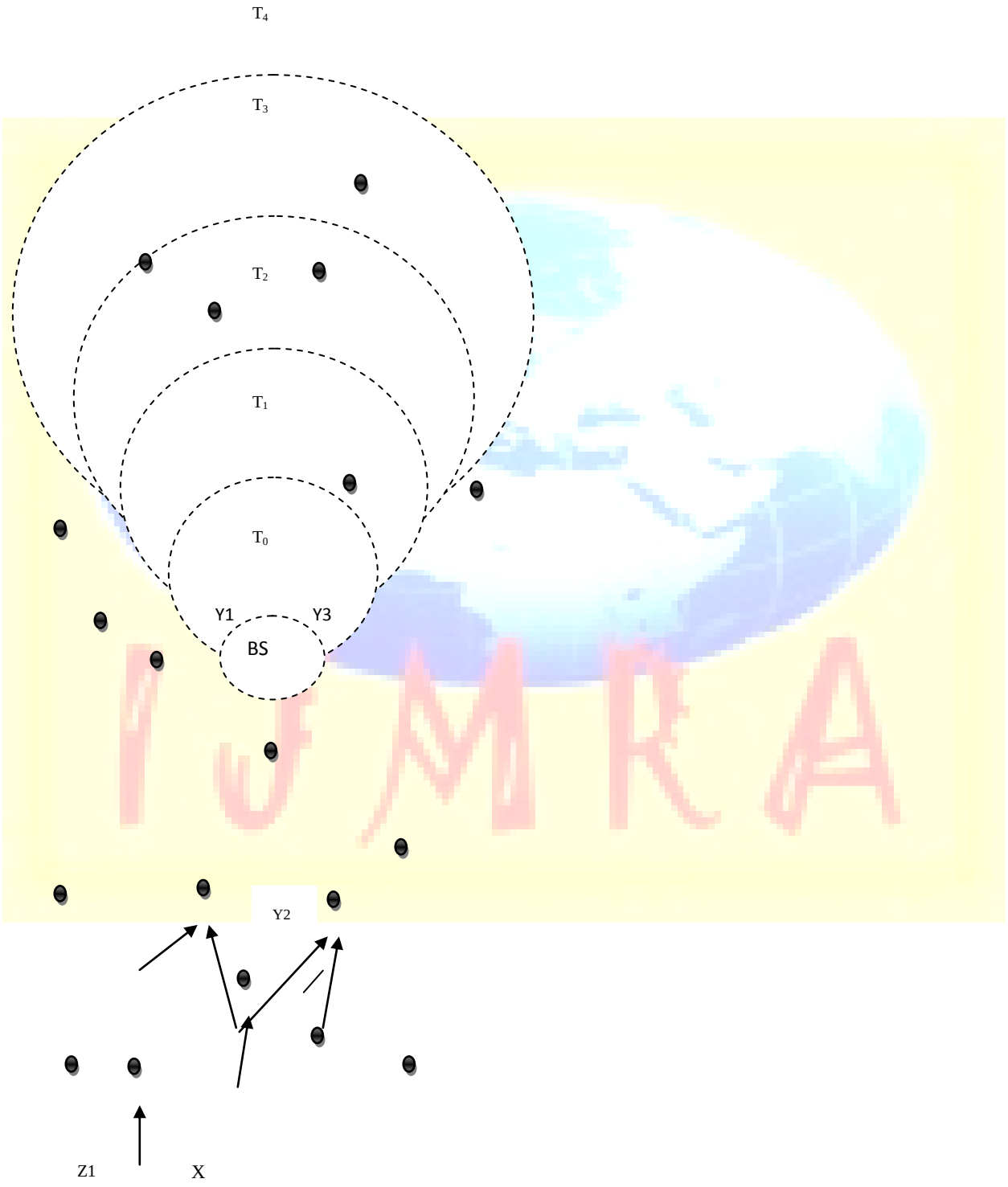
DRP improves the propagation efficiency by using two-hop neighborhood information. More specifically, DRP adds a "last-hop neighbor list" (LHNL) field to the header of each share. Before a share is propagated to the next node, there laying node first updates the LHNL field with its neighbor list.

### *2.2.4 Multicast Tree-Assisted Random Propagation*

MTRP aims at actively improving the energy efficiency of random propagation while preserving the dispersiveness of DRP. The basic idea comes from the following observation of Fig.3: Among the three different routes taken by shares, the route on the bottom right is the most energy efficient because it is the shortest end-to-end path. So, in order to improve energy efficiency, shares should be best propagated in the direction of the sink. In other words, their propagation should be restricted to the right half of the circle in Fig.3.

## III. PRELIMINARIES

### *A.SYNOPSIS DIFFUSION*



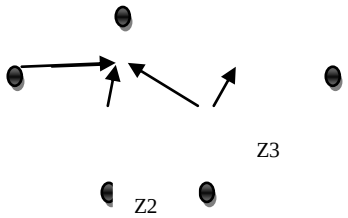



Fig.1. Synopsis diffusion over a ring topology—A node may have multiple parents, e.g., X has three parents, Y1, $Y_2$, and Y3.

An aggregation framework called *synopsis diffusion* which uses a ring topology as illustrated in Fig. 1. During the query distribution phase, nodes form a set of rings around the base station (BS) based on their distance in terms of hops from BS. By Ti we denote the ring consisting of the nodes which are hops away from BS.

We now describe the duplicate-insensitive synopsis diffusion algorithms for Count and Sum. These *algorithms* are based on a probabilistic algorithm for counting the number of distinct elements in a multiset.

***1. Count***

In this algorithm, each node X generates a local synopsis $Q^X$ which is a bit vector of length$\eta > log^{N'}$, where $N'$ is the upper bound on Count. To generate $Q^X$, node X executes the function CT(X, η) given as follows (Algorithm 1), where X is the node's identifier. Algorithm 1 can be interpreted as a coin-tossing experiment (with a cryptographic hash function *h()*, modeled as a random oracle whose output is 0 or 1, simulating a fair coin-toss), which returns the number of coin tosses, say, until the first head occurs or η+1 if ηtosses have occurred with no heads occurring. In the synopsis generation function $SG_{count}$, the $i^{th}$ bit of $Q^X$ is set to "1" while all other bits are "0". Thus, $Q^X$ is a bit vector of the form $0^{(i-1)}\,10^{(n-i)}$ with probability $2^{-i}$.

***Algorithm 1 CT(X, η)***

Begin

i = 1;

While i <η +1 AND h(X,i) = 0 do

```
        i =i+1;
    end
    return i;
end
```

***Definition:*** The *fused synopsis* of a node X, $B^X$, is recursively defined as follows. If X is a leaf node (i.e., X is in the outer most ring), $B^X$ is its local synopsis $Q^X$. If X is a non-leaf node, $B^X$ is the logical OR of X's local synopsis $Q^X$ with X's children's *fused synopses*.

If node X receives synopses $B^X_1$, $B^X_2$, - - - - -, $B^X_d$ from d child nodes $X_1$, $X_2$, - - - -, $X_d$ respectively, then X computes as follows (denotes the bitwise OR operator):

$B^X = Q^X || B^X_1 || B^X_2 ||$- - - - -$|| B^X_d$

Note that the represents the sub-aggregate of node, including its descendant nodes. We note that is same as the final synopsis.

**2. Sum**

The Count algorithm can be extended for computing Sum. The synopsis generation function SG() for Sum is a modification of that for Count, while the fusion function SF() and the evaluation function SE() for Sum are identical to those for Count. To generate the $Q^X$ local synopsis to represent its sensed value $v_x$, node X invokes CT(), used for Count synopsis generation, $v_x$ times. In the $i^{th}$ $1 \leq i \leq v_x$ invocation, node X executes the function CT ($X_i$, η) where $X_i$ is constructed by concatenating its ID and integer i (i.e., $X_i$= (X, i)), and η is the synopsis length. The value of is taken as $\log_2 S'+4$, where S' is an upper bound on the value of Sum aggregate. Unlike the local synopsis of a node for Count, more than one bit in the local synopsis of a node for Sum may be equal to "1". The pseudo code of the synopsis generation function $SG_{sum}(X, vx, \eta)$, is presented in following Algorithm.

***Algorithm 2 SGsum(X, $v_x$, η)***

```
Begin
    Q^X [j] = 0 all j 1 ≤ j ≤ η;
    i = 1;
    while i ≤ v_x do
    X_i = (X, i);
    j = CT (X_i, η);
```

$Q^X[j] = 1;$

$i = i + 1;$

end

return $Q^X$;

end

## C. VERIFICATION ALGORITHM

In the rest of the paper, by the term *false MAC* we refer to any string that does not correspond to the MAC generation scheme described previously. Note that a false MAC can be associated either to a false "1" or to a true "1" bit. In particular, a compromised node can generate a false MAC (in the context of computing the function MAC) in four ways: 1) by using a false; 2) by using a false key; 3) by doing both of 1) and 2); and 4) by simply sending a bogus string of bits.

### 1. Protocol Operation

We describe our verification protocol with respect to one single synopsis. Each synopsis can be verified independently and hence our algorithm is readily applicable for computing multiple synopses.

*a) Query Dissemination:* In this phase, BS broadcasts the name of the aggregate to compute, a random number Seed and the chosen value of "test length", k. The query that BS broadcasts is as follows ($F_{agg}$is the name of the aggregate (e.g., "Sum")):

BS→→ : ($F_{agg}$, Seed, k).

During this phase, nodes form a set of rings around BS based on their distance in hops from BS.

*b) Aggregation Phase:* Each node executes the aggregation phase of the original synopsis diffusion protocol along with sending some authentication messages. Recall that during the falsified subaggregate attack the fused synopsis, $B'^X$ computed at a node X can be different from X's true fused synopsis $B^X$.

**Example (No Attack):** Fig. 4 illustrates the protocol operation with k=5. Node P is in ring *i* and nodes X, Y, and Z are in ring *i*+1. X, Y and Z send to P their fused synopses, $B'^X$, $B'^Y$, and $B'^Z$, respectively. Node

X also forwards one MAC each for the 4th, 5th, 6th, 8th and 10th bit, which is denoted as $M_4$, $M_5$, $M_6$, $M_8$, and $M_{10}$, respectively. Similarly, P receives MACs from nodes Y, and Z. Let the local synopsis of node P, $Q^P$ be 001000000000. P fuses all of the received synopses ($B\hat{}^X$, $B\hat{}^Y$, and $B\hat{}^Z$), including its local synopsis ($Q^{P)}$, to compute its fused synopsis ($B\hat{}^P$), and sends it to the parent nodes in ring *i- 1*. In this example, $B\hat{}^P$ = 111111111100. P also forwards the MACs for the five rightmost "1" bits ($M_6$, $M_7$, $M_8$, $M_9$, and $M_{10}$) to its parent nodes.

**Example (With Attack):** If P is malicious, it may inject a false "1" in $B\hat{}^P$ at the 11th bit resulting in $B\hat{}^P$=111111111110. An example of such an attack is shown in Fig. 4. In this example, MAC is claimed to be generated by an arbitrary node selected by the adversary, and sensed value being $v_W$. Also, note that Seed set to the 11$^{th}$ bit equal to "1". For ease of exposition, we only show in this example the relevant messages and assume the forged MAC is forwarded directly to the BS (BS being the parent of node P). We see that BS does the verification and detects this attack.

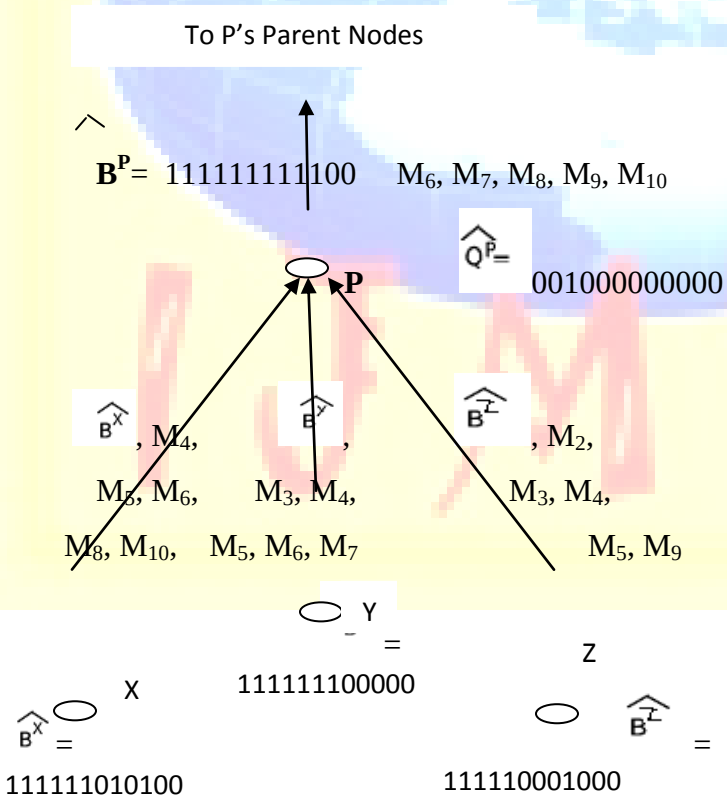



Fig. 4. Aggregation phase of verification algorithm. An example (without attack).

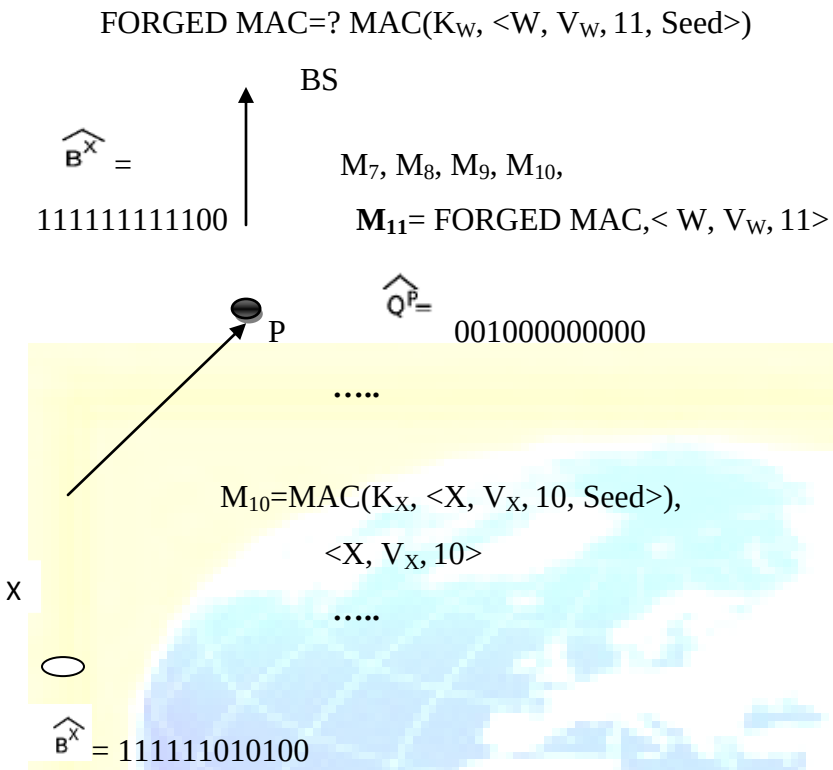



Fig.5. Example of MAC forging during aggregation phase (with attack).

## IV. PROBLEM DESCRIPTION

Our goal is to detect the falsified subaggregate attack against Count or Sum algorithm. More formally, our goal is to detect if $B\hat{}$, the synopsis received at BS is the same as the "true" final synopsis B. Without loss of generality, we present our algorithm in the context of Sum aggregate. As Count is a special case of Sum, where each node reports a unit value, this algorithm is readily applicable to Count aggregate also.

Note that a compromised node C can introduce a false "1" at bit j in $B\hat{}^C$ by launching either of the following attacks.

1) Falsified subaggregate attack: C just flips bit j in $B\hat{}^C$ from"0" to "1" — not having a local aggregate justifying that "1"inthesynopsis $B\hat{}^C$.

2) Falsified local value attack: C injects a false "1" at bit j in its local synopsis, $Q\hat{}^C$. The falsified synopsis, $Q\hat{}^C$, induces bit j in $B\hat{}^C$ to be "1". Note that true local sensed value, $v_C$, corresponds to $Q^C$.

Fig. 6 illustrates an example of the falsified subaggregate attack. Node C has three child nodes which are X, Y and Z, and C receives from them synopses $B\hat{}^X$, $B\hat{}^Y$, and $B\hat{}^Z$, respectively.

Node C is supposed to aggregate its local synopsis $Q^C$ with the received synopses using the Boolean OR operation. That means, the fused synopsis of C should be $B\hat{}^C = Q^C \| B\hat{}^X \| B\hat{}^Y \| B\hat{}^Z$. However, in this example, malicious node C increases the number of "1"s in $B\hat{}^C$ by injecting false "1"s into $B\hat{}^C$ without forging $Q^C$. The fabricated $B\hat{}^C$ represents a bogus subaggregate at C, which is higher than C's true subaggregate.

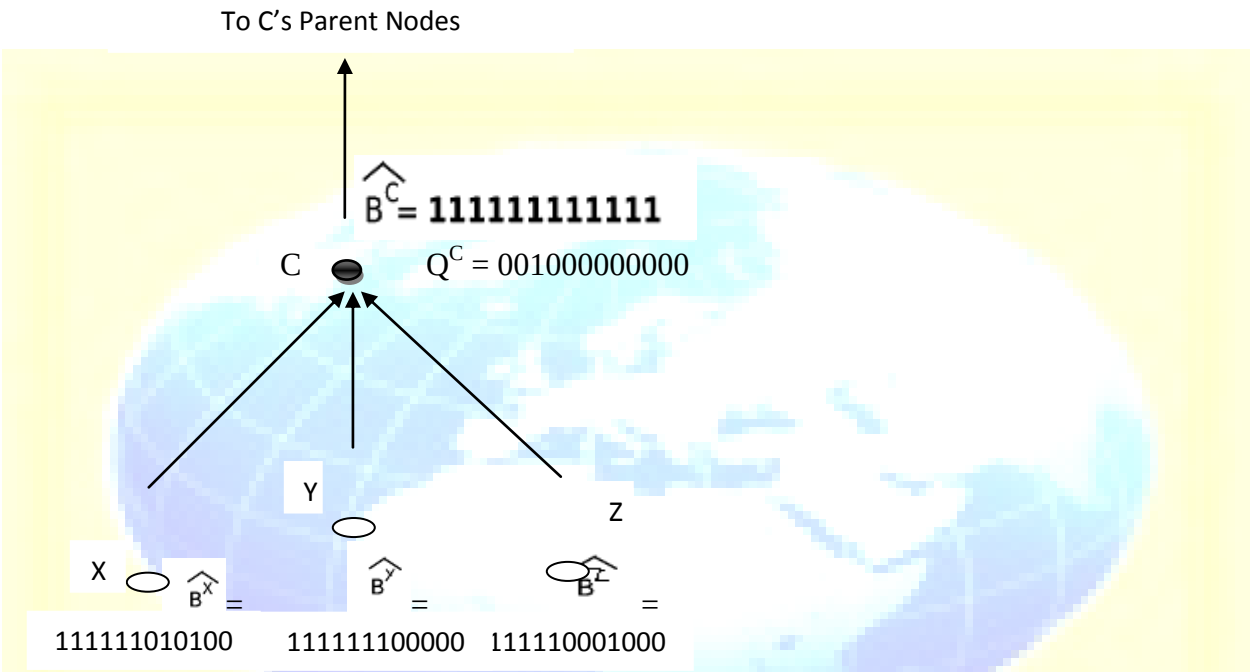



Fig.6. Example of falsified subaggregate attack: Node C is supposed to aggregate its local synopsis $Q^C$ with received synopses (from child nodes X, Y, and Z) using the Boolean OR operation. However, malicious node C injects false "1"s in its fused synopsis $B\hat{}^C$. Fabricated $B\hat{}^C$ represents a bogus subaggregate at C, which is higher than C's true subaggregate.

In the rest of this paper, we do not further discuss the deflation attack (changing "1" to "0"). We restrict our discussion to the inflation attack (changing "0" to "1"), which we call the false "1" *injection attack*. That means the goal of our attacker is only to increase the estimate of the aggregate.

In this paper, we introducing the Randomized Dispersive Routes for computing the packets in multiple paths between the networks based on accessing the signals from BS. If the packets are computing through the intermediate nodes from in-network to BS using the aggregation functions.

## V. RESULT

We now present the results of the experiments. As Count can be considered as a special case of Sum, here we discuss only the results related to Sum aggregate. We did not study the false positive rate of the verification protocol. Recall that integrity checks in node-to-node communication ensures that if no attack is launched, BS will receive at least one MAC for each of the rightmost "1"s in the final synopsis. A corrupted MAC that is a consequence of something besides an attack (e.g., communication error) can reach the BS. However, this problem is not protocol-dependent and it is out of the scope of our work. Since the verification protocol completes in one epoch irrespective of the final result (success or failure), we did not study the latency in our simulation. We present the following results for a single synopsis, which can be extended for multiple synopses.

We evaluate the average number of hops of the end-to-end route as a function of the TTL value in Fig. 19. This hop count metric can be considered as an indirect measurement of the energy efficiency of the routes generated by the routing schemes. It can be observed that the hop count under PRP, DRP, and NRRP increases linearly with N, while the hop count under MTRP only increases slowly with N. The TTL value N does not play a role in the H-SPREAD scheme. Under large N, e.g., when N ¼ 25, the randomized algorithm achieves better security performance than H-SPREAD. However, the hop count of H-SPREAD is about 1=3 of that of PRP, DRP, and NRRP, and about 1=2 of that of MTRP. The relatively large hop count in the randomized algorithms is the cost for stronger capability of bypassing black holes.

## VI. CONCLUSION

Our results have shown the effectiveness of randomized dispersive routing in combating CN and DOS attacks. By appropriately setting the secret sharing and propagation parameters, the packet interception probability can easily be reduced by the proposed algorithms to as low as 10−3, which is at least one order of magnitude smaller than approaches that use deterministic node-disjoint multi-path routing. At the same time, we have also verified that this improved security performance comes at a reasonable cost of energy. Our current work does not address this attack. Its resolution requires us to extend our mechanisms to handle multiple collaborating black holes, which will be studied in our future work.

## REFERENCES

[1] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis diffusionfor robust aggregation in sensor networks," in *Proc. 2nd Int. Conf.Embedded Networked Sensor Systems (SenSys)*, 2004.

[2] D. Wagner, "Resilient aggregation in sensor networks," in *Proc. ACMWorkshop Security of Sensor and Adhoc Networks (SASN)*, 2004.

[3] L. Hu and D. Evans, "Secure aggregation for wireless networks," in*Proc. Workshop Security and Assurance in Ad hoc Networks*, 2003.

[4] T. Claveirole, M.D. de Amorim, M. Abdalla, and Y. Viniotis,"Securing Wireless Sensor Networks Against Aggregator Compromises,"IEEE Comm. Magazine, vol. 46, no. 4, pp. 134-141, Apr.2008.

[5] S.Roy, M.Conti, S.Setia, S.Jajodia, "Secure Data Aggregation in Wireless Sensor Network", IEEE Transactions on Information Forensics and Security, vol. 7, no. 3, June 2012.

[6] T.Shu, M.Krunz, S.Liu, "Secure Data Collection in Wireless Sensor Networks Using Randomized Dispersive Routes" IEEE Transactions on Mobile Computing, vol. 9, no. 7, July 2010.

[7] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," IEEE Comm. Magazine, vol. 40, no. 8, pp. 102-114, Aug. 2002.

[8] M. Burmester and T.V. Le, "Secure Multipath Communication in Mobile Ad Hoc Networks," Proc. Int'l Conf. Information Technology: Coding and Computing, pp. 405-409, 2004.

[9] D.B. Johnson, D.A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks," Ad Hoc Networking, C.E. Perkins, ed., pp. 139-172, Addison-Wesley, 2001.

[10] S.J. Lee and M. Gerla, "Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks," Proc. IEEE Int'l Conf. Comm. (ICC), pp. 3201-3205, 2001.

[11] M.K. Marina and S.R. Das, "On-Demand Multipath Distance Vector Routing in Ad Hoc Networks," Proc. IEEE Int'l Conf. Network Protocols (ICNP), pp. 14-23, Nov. 2001.

[12] Z. Ye, V. Krishnamurthy, and S.K. Tripathi, "A Framework for Reliable Routing in Mobile Ad Hoc Networks," Proc. IEEE INFOCOM, vol. 1, pp. 270-280, Mar. 2003.

[13] D.R. Stinson, Cryptography, Theory and Practice. CRC Press, 2006.

[14] A.D. Wood and J.A. Stankovic, "Denial of Service in Sensor Networks," Computer, vol. 35, no. 10, pp. 54-62, Oct. 2002.

[15] N.F. Maxemchuk, "Dispersity Routing," Proc. IEEE Int'l Conf.Comm. (ICC), pp. 41.10-41.13, 1975.