Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

Implementation of Protected Data Accumulation Using Randomized Dispersive Routes in Wireless Sensor Networks

A.Subramani[1], M. Praveen[2], G.Sathishkumar[3], Mr.T. Karthikeyan[4]

[1,2,3] 3rd Year Student, [4] Assistant Professor
Department of Computer Science & Engineering
Knowledge Institute of Technology, Salem

Abstract--In large amount of sensor network, specifically in case of data aggregation it should reduce the amount of communication and energy consumption. Recent research on wireless sensor networks has proposed a robust aggregation framework called synopsis diffusion which combines multipath routing schemes with duplicate-insensitive algorithms to accurately compute aggregates (e.g., predicate Count, Sum) in spite of message losses resulting from node and transmission failures. But these aggregation frameworks aggregation frameworks does not solve the problems which are appearing at base station side. Overhead of per node communication has been occurred regardless of increase in network size. In this paper, we make the synopsis diffusion approach secure against attacks in which compromised nodes contribute false sub combined values. In this regard, the alternative verification algorithm by which the base station can determine if the computed aggregate (predicate Count or Sum) includes any false contribution.

In this paper, we study the compromised-node and denial-of-service is the two key attacks in wireless sensor networks. We argue that multipath routing approaches are highly vulnerable for such kind of attacks. The mechanisms that we developed so far will generate randomized multipath routes. The routes taken by the shares of different packets changes over time in this design. So, we analytically investigate 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) various possible security threats encountered, in this paper, we are specifically interested in combating two types of attacks: compromised node (CN) and denial of service (DOS) [14]. In the CN attack and DOS attack, a supporter physically compromises a subset of nodes to information overhead and the adversary interferes with the normal operation of the network by changing the functionality of a subset of nodes actively. Those two attacks such as CN and DOS 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 disrupt normal data delivery between sensor nodes and sink, or it will even screen the topology. In such case, 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 can be resolved by finding black holes when the data deliver that can bypass the holes. The above proposed solution can implemented by two phases. First, the packet is divided into M shares (i.e., components of a packet that carry partial information) using a Shamir's algorithm [13]. Second, multiple routes from the source to the destination are computed according to some multipath routing algorithm (e.g., [9], [10], [11], [12]). 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. The 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. When the adversary chose compromise or Jam modes, the first approach will no longer useful. 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. To finalise, 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.

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

II. RELATED WORK

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

A. Data Aggregation Without any Provision for Security

B. Secure Aggregation Techniques

C. Randomized Multipath Delivery

1. Overview

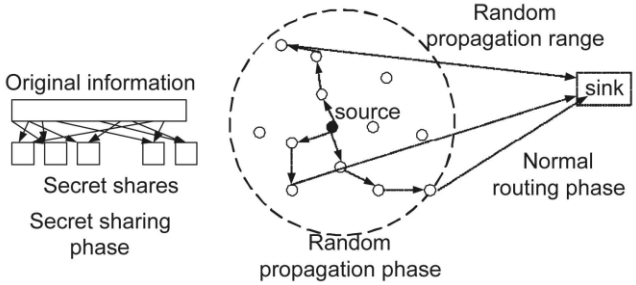

Fig. 2. Randomized dispersive routing in a WSN.

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.

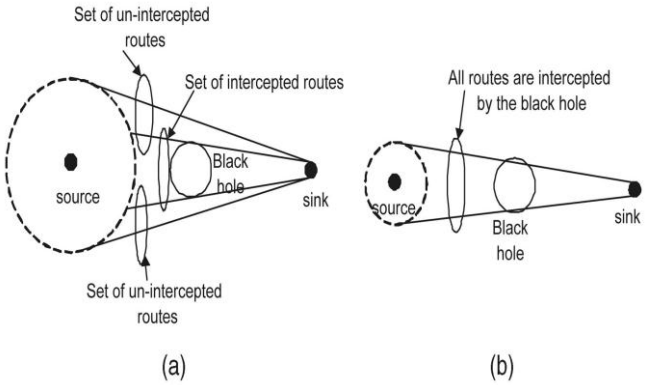

Fig. 3. Implication of route dispersiveness on bypassing the black hole. (a) Higher dispersiveness routes. (b) Lower dispersiveness routes.

The effect of route dispersiveness on bypassing black holes is demonstrated in Fig. 3, where the dotted circles represent the ranges the secret shares can be propagated to in the random propagation phase. A dotted circle infers 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. In evident, the random propagation plays a key role in both security performance as well as energy consumption in the entire mechanism.

2. Random Propagation of Information Shares

To diversify routes, an ideal random propagation algorithm would propagate shares as dispersively as possible. It means that propagating the shares beyond from their source. Simultaneously, it is highly desirable to have an energy-efficient propagation, which will limit the number of randomly propagated hops. The experimental 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 broadcasted based on one-hop neighborhood information. More explicitly, a sensor node maintains a neighbor list, which comprises the IDs of all nodes within its transmission range.

The main disadvantage 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. The share will be relayed to a different node in each step of random propagation that will lead to better propagation efficiency which will be guaranteed by the non-repetitive propagation.

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 distributed to the next node, there placing node first updates the LHNL field with its neighbor list.

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 bottom right route is the most energy efficient because it is the shortest end-to-end path. Accordingly to improve the energy efficiency, the best propagation will be shared in the direction of the sink. In different, their propagation should be limited to the right half of the circle in Fig.3.

III. PRELIMINARIES

A. SYNOPSIS DIFFUSION

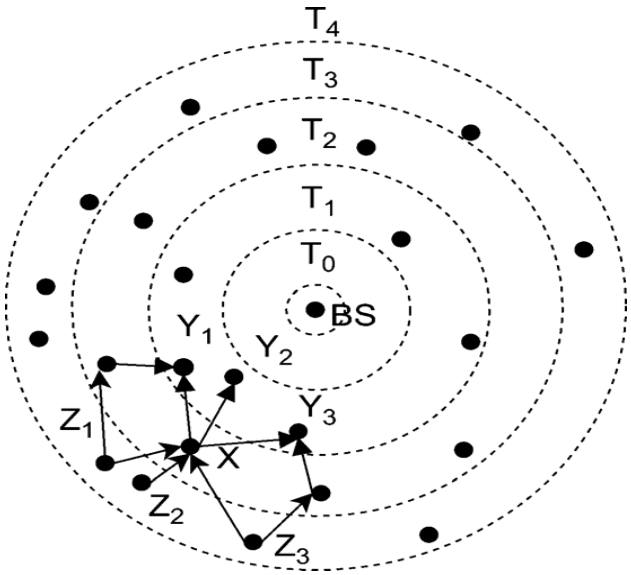

Fig. 1. Synopsis diffusion over a ring topology—A node may have multiple parents, example X which will contain three parents such as Y-1,Y-2 and Y-3.

An aggregation framework called *synopsis diffusion* which uses a ring topology as illustrated in Fig. 1. During the query distribution phase, nodes produce a set of rings around the base station (BS) based on their distance in terms of hops from BS. Ti denotes 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 diffusion 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 implemented as a coin-tossing experiment (with a cryptographic hash function *h()*, designed as a random oracle, their output is 0 or 1, pretending 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. During 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 outermost ring), B^X and its local synopsis is represented as 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 represents the subaggregate 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, although 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. While comparing the local synopsis of a node for count with local synopsis of node for sum which contains more than one bit that may be equal to one. 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. It should be noted that a false MAC can be associated either to a false "1" or to a true "1" bit. 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. Independently each synopsis can be verified and hence our algorithm is readily applicable for computing multiple synopses.

a) Query Dissemination: Aggregate name will be computed by the BS broadcasts in this phase, a random number Seed and the chosen value of "test length", k. The request that BS broadcasts mentioned as follows ($F_{agg}$is the name of the aggregate (e.g., "Sum")):

BS→ →∗∗ : (F_{agg}, Seed, k).

A set of rings will be formed around BS by the nodes based upon their distance that is in hops(BS).

b) Aggregation Phase: Each node executes the aggregation phase of the original synopsis diffusion protocol along with sending some authentication messages. It is also noted that during the falsified subaggregate attack the fused synopsis, $B\hat{}^X$ 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, $B\hat{}^X$, $B\hat{}^Y$, and $B\hat{}^Z$, respectively. Node X also forwards one MAC each for the 4,5,6 and 8th 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. Q^P will 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's sensed value being v_W. We need to set 11th bit as 1 in Seed. 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.

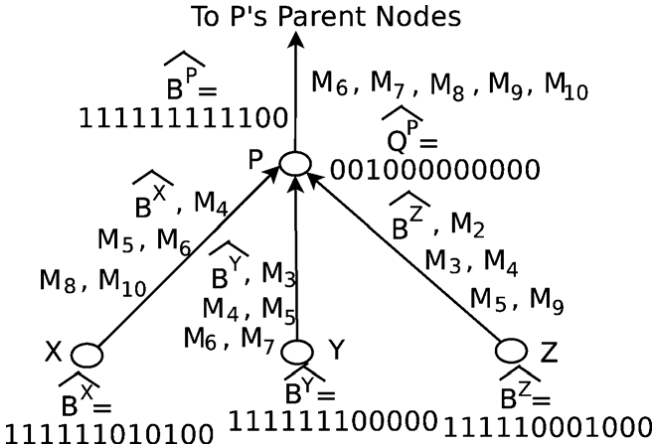

Fig. 4. Aggregation phase of verification algorithm. An example (without attack)

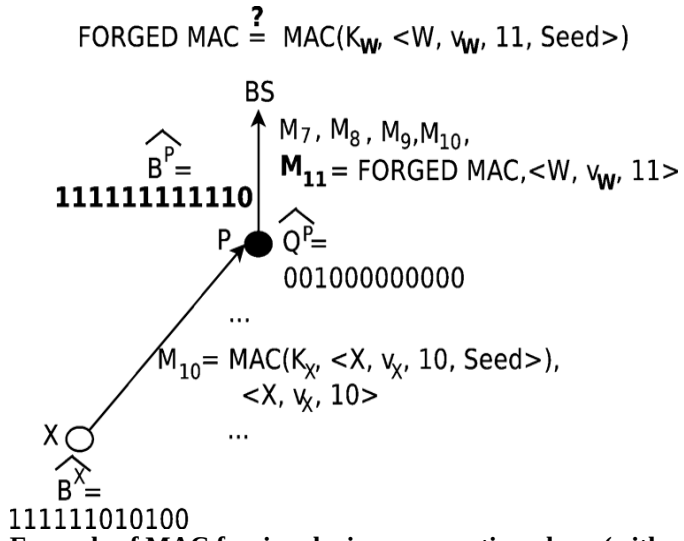

Fig.5. Example of MAC forging during aggregation phase (with attack)

IV. PROBLEM DESCRIPTION

Our goal is to detect the falsified subaggregate attack against Count or Sum algorithm. Our main goal is to find out the BS Synopsis is same as B. Without loss of generality, we present our algorithm in the context of Sum aggregate. Each node will represented a unit value , as the algorithm is easily applicable to Aggregate count also

It should be noted that 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.

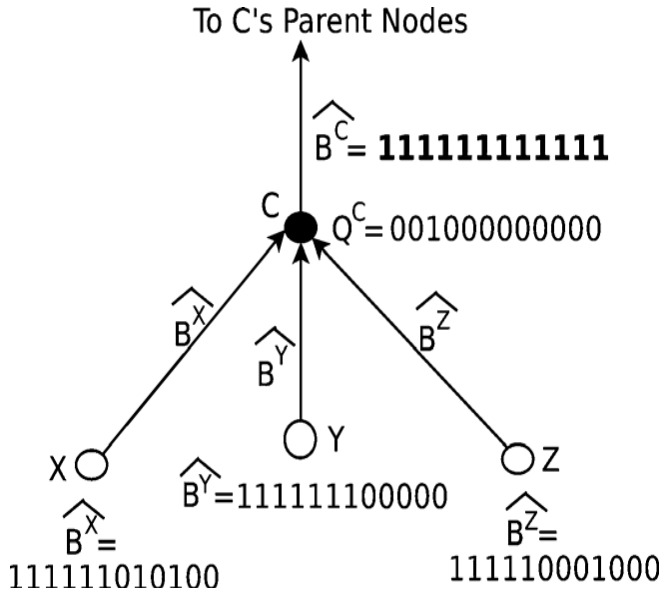

Fig.6. Example of falsified subaggregate attack: Node C is supposed to aggregate its local synopsis

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*. 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 intermediate nodes from in-network to BS using the aggregation functions.

V. RESULT

We now present the results of the experiments Even though there is an importance of SUM case, here we discuss only the issues that are related to aggregate sum. We did not study the false positive rate of the verification protocol. BS will receive at least one MAC when there is an integrity check on node to node communication. It will ensure that no attack has been launched A corrupted MAC that is a consequence of something besides an attack (e.g., communication error) can reach the BS. However, this problem is protocol independent 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), the latency is not simulated in our proposed model. We proposed 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 is detected that MTRP hop count increases slowly with N while PRP, DRP and NRRP hop count increases gradually with N. In H-SPREAD scheme, TTL value doesn't play a vital role. As a result, randomized algorithm yields a better security while compared to H-Spread. Still, H-SPREAD hop count is about 1=3(PRP, DRP, and NRRP) and 1=2(MTRP). The relatively large hop count in the randomized algorithms is the cost for stronger capability of bypassing black holes.

VI. CONCLUSION

As a result, the probability of packet interception can be reduced such as 1-3 while setting the secret sharing and their propagation parameters. The security performance also improved at a reasonable price at the same time. As current work does not propose those issues, so in future, it is planned to implement multiple collaborating black holes.

REFERENCES

[1] S. Nath, P. B. Gibbons, S. Seshan, and Z. Anderson, "Synopsis diffusion for 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.

