Wireless Netw (2009) 15:279–294
DOI 10.1007/s11276-007-0039-4

SPREAD: Improving network security by multipath routing in mobile ad hoc networks

Wenjing Lou · Wei Liu · Yanchao Zhang · Yuguang Fang

Published online: 14 April 2007

Abstract We propose and investigate the SPREAD scheme as a complementary mechanism to enhance secure data delivery in a mobile ad hoc network. The basic idea is to transform a secret message into multiple shares, and then deliver the shares via multiple paths to the destination so that even if a certain number of message shares are compromised, the secret message as a whole is not compromised. We present the overall system architecture and discuss three major design issues: the mathematical model for the generation and reconstruction of the secret message shares, the optimal allocation of the message shares onto multiple paths in terms of security, and the multipath discovery techniques in a mobile ad hoc network. Our extensive simulation results justify the feasibility and the effectiveness of the SPREAD approach.

Keywords Multipath routing · Network security · Ad hoc networks

This work was supported in part by the US Office of Naval Research Young Investigator Award under grant N000140210464 and the US National Science Foundation under grants CNS-0626881, CNS-0626601 and ANI-0093241 (CAREER Award).

W. Lou
Department of Electrical and Computer Engineering,
Worcester Polytechnic Institute,
Worcester, MA 01609, USA
e-mail: wjlou@ece.wpi.edu

W. Liu
Scalable Network Technologies, Los Angeles,
CA 90045
e-mail: wliu@scalable-networks.com

Y. Zhang
Department of Electrical and Computer Engineering,
New Jersey Institute of Technology,
Newark, NJ 07102, USA
e-mail: yczhang@njit.edu

Y. Fang (✉)
Wireless Networks Laboratory (WINET), Department of Electrical and Computer Engineering, University of Florida,
Gainesville, FL 32611, USA
e-mail: fang@ece.ufl.edu

1 Introduction

Mobile ad hoc networks (MANETs) have received tremendous attention in the past few years. On the one hand, its rapid deployability and self-organizing configurability have made a MANET very attractive in tactical and military applications, such as the tactical communications in a battlefield, where the environment is hostile and fixed infrastructures are not available or reliable, but fast network establishment, self-reconfiguration and security-sensitive operations are necessary. On the other hand, the salient features of a MANET, such as the broadcast nature of the wireless channel, the infrastructureless architecture, the highly dynamic network topology, and the limited resources of mobile devices, have posed many new challenges in the design and implementation of such a network [1].

Secure data delivery from one node to another is a fundamental service in a MANET as well as in any network. Sensitive information, such as tactical military information, transmitted across a hostile MANET should be protected from passive attacks, such as eavesdropping. The wireless channel in a hostile environment is vulnerable particularly to eavesdropping due to its broadcast nature. Conventionally, data confidentiality is achieved by cryptography. However, the security of cryptographic methods highly depends on the secure and reliable key management system. In particular, many computationally efficient cryptographic algorithms, such as the stream cipher RC4 which is suitable in

the resource constrained MANET, are highly sensitive to the keying materials and susceptible to the known plaintext attacks. Many efforts have been made in developing more secure and more reliable key management systems [2–9]. However, in a highly dynamic MANET environment, end-to-end encryption is usually impractical as the end-to-end authentication and dynamic session key negotiation become less reliable, particularly when the number of nodes becomes large. So far no absolute secure and reliable key management system is available. The gap between theoretic design and practical implementation would further diminish such a possibility. Another potential threat in a MANET comes from the compromised nodes. Compromised nodes may passively collect information. They may also launch active attacks, such as altering the content of forwarded packets, disrupting routing functions, or simply selectively discarding important packets. Secure routing protocols have been proposed to ensure the correct exchange of the routing information among legitimate participating nodes [10–13]. However, they do not exclude the possibility of selecting a compromised node on a crucial communication path, nor do they prevent a compromised node from collecting information from forwarded messages or maliciously dropping important packets. As a second line of defense, some intrusion detection mechanisms [14] or misbehavior detection schemes such as the watchdog [15] have been proposed to detect such attacks, but with limited success. Before any effective prevention/reaction/recovery mechanism takes effect, the end-to-end data delivery service might have been intercepted or significantly disrupted. Therefore, developing a resilient security protocol becomes particularly important, that is, the protocol should be able to function well in adversarial environments when a certain number of nodes are compromised.

In this paper, we propose a multipath data delivery scheme, SPREAD, to provide more secure end-to-end data delivery service in a MANET. The fundamental idea of SPREAD is based on two techniques: multipath routing and secret sharing. Suppose a source node has a secret message for a destination node that is multiple hops away. If the source node sends the whole message through a single path, an adversary can intercept it at any one of the intermediate nodes along the path, or it can disrupt the delivery by dropping packets at any one of the nodes along the path. However, if the source node divides the message into multiple pieces and sends them via multiple independent paths, the adversary must intercept multiple pieces from multiple paths in order to capture the whole message, or he must disable multiple nodes on multiple paths in order to disrupt the delivery service. By this means, the secret message is less likely to be intercepted by the adversaries and more likely to reach the destination.

Our focus in this paper is on how to exploit this SPREAD idea and to develop a security enhancement protocol to strengthen the data delivery service in MANETs. We address three major design issues—how to divide the message into multiple pieces; how those pieces are allocated onto each path; and how to select the multiple paths. The contribution of this paper is threefold. First, we put together techniques from multiple disciplines (cryptography, optimization, network routing) and propose a novel scheme which effectively improves the network security. Secondly, we identify three major design issues, investigate in depth into each of them, and proposed a feasible solution to each of them—we apply the secret sharing schemes to divide the message into pieces (message shares); we propose the optimal share allocation schemes which are able to provide a certain degree of reliability without sacrificing security; we study the multipath routing algorithms and propose a secure routing cost metric which converts a non-additive cost function of security level into an additive one so that security is introduced as one dimension of Quality of Service (QoS) routing. Lastly, most of the simulation studies for multipath routing and/or security design ignore the physical layer channel dynamics and MAC layer contentions. We conduct extensive simulation with a complete setting of physical layer and MAC layer models. The significant performance impact of the shared physical channel on concurrent multipath routing is studied as well.

The rest of the paper is structured as follows. We review the related work in Section 2 and describe the overall system architecture in Section 3. Then we elaborate the three major design issues in three subsequent sections, respectively. We present the extensive simulation results in Section 7 and summarize this paper in Section 8.

2 Related work

The combination of secret sharing and multipath routing was first proposed by Zhou and Haas in [2] where the role of a certificate authority (CA) in a public key infrastructure (PKI) is distributed to multiple servers by the means of secret sharing and multipath routing. This idea was further developed by Kong et al. in [3] where CAs are further localized by distributing the servers more evenly in the network such that operations such as signing a certificate can be done locally by neighbors of the requesting node. The multipath routing in their work indicates multiple paths from one node to multiple nodes while SPREAD considers multiple paths between any two nodes. A more recent key management approach based on multipath routing is a probabilistic approach for the establishing of pairwise secret keys [6, 7]. The multipath in their schemes are logical (i.e., encrypted by different keys) rather than physically independent (node-disjoint) paths required in our SPREAD scheme. Recently, Papadimitratos and Haas proposed a Secure Message Transmission (SMT) protocol which combines Rabin's algorithm and multipath routing to safeguard the data transmission against arbitrary

malicious behavior of other nodes [16]. While the idea is similar to SPREAD, the focus of their paper is to defend against malicious packet dropping by adversaries (compromised nodes) who are on the forwarding paths, which is in fact complemental to our work here.

Multipath routing has been shown to be effective in coping with the frequent topological changes and improving resilience to node/link failures in a MANET [17–19]. Much work has been focused on the alternate multipath routing, in which a node maintains multiple paths to a certain destination but uses one path at a time. A second path is used as the alternate only when the primary one fails. The concurrent multipath routing, namely, using multiple paths simultaneously, has not been well studied. Tsirigos and Haas applied concurrent multipath routing together with diversity coding to mitigate the effect of frequent topological changes [20] and to improve the packet delivery ratio [21]. They provided an analytical framework for performance analysis but no specific routing algorithms were studied. Papadimitratos and Haas [22] studied a disjoint path set selection protocol (DPSP) which is similar to our multipath finding algorithm. Their objective was to find multiple edge-disjoint paths for reliability purpose while we are interested in node-disjoint paths for security objective. In our previous work [23], the SPREAD idea was first proposed but was studied in the wired Internet context. In [24] and [25], preliminary and partial results of this paper were presented.

3 SPREAD overview

3.1 System model

The fundamental idea of the SPREAD scheme comes from the following observation: a messenger who carries the full message from one place to another place across a hostile ground may reveal the message easily if he/she is captured, while the message will not be fully recovered by adversaries if multiple messengers are deployed, each only carrying partial information and taking different routes across the hostile ground. The SPREAD scheme works in the similar fashion: when a source node wants to send a message to a destination node securely in a MANET, the source can use a multipath routing algorithm to find multiple paths from the source to the destination with certain properties (e.g., disjoint paths); then the source determines a secret sharing scheme, depending on the message security level and the availability of multiple paths, to transform the message into multiple shares; then the message shares are routed to the destination by the multipath routing protocol and the destination reconstructs the original message upon receiving a certain number of shares.

We address the improved security by dealing with the compromised nodes and eavesdropping problem. We assume hop-by-hop link encryption, each link with different key which is negotiated between neighboring nodes. We also assume that if one node is compromised (either physically captured or remotely broken into), all the shares traveling through that node are compromised. A compromised share means the adversary has a means to decrypt it and it could be used to recover the original message. Therefore, we define that a message is compromised when the adversary has compromised enough message shares for that message. Since nodes in ad hoc networks use wireless channels to communicate with each other, we also investigate the message eavesdropping problem. We assume that anyone sitting within the transmission range of a transmitting node is able to eavesdrop the transmission of that node. However, it should be pointed out that an eavesdropped message share does not divulge any useful information before it is decrypted.

We evaluate the performance for both individual attacks and colluding attacks. For the former, we assume that each adversary is working independently to recover the message, while for the latter, we assume all the compromised nodes, by some means, can combine their compromised message shares to recover the original message.

We assume that the adversaries, after compromising the nodes, will attempt to remain in the network by launching only passive attacks in order to acquire more secure information. If the compromised nodes launch active attacks, such as stopping forwarding packets for other nodes or altering the information when forwarding, some intrusion detection mechanism [14] or the misbehavior detection schemes such as a watchdog proposed in [15] can be used to identify the compromised nodes quickly so that they can be excluded from the network. Schemes which improve reliability by re-transmission such as SMT proposed in [22] can also be combined with SPREAD to defend against such attacks.

There are several major design issues in this scheme: first, how to transform the message into multiple shares; secondly, how to allocate the shares onto each path; and thirdly, how to discover the desired multiple paths. We will briefly discuss these three design issues in this section and will elaborate each of them in the following sections.

3.2 Threshold secret sharing

The first issue is how to divide the message into multiple pieces (shares)? Simply chopping the message into multiple segments involves the least processing overhead. However, it does not provide satisfactory security protection, since each segment contains explicitly partial content of the message, which could be used to infer the content of the whole message. It also needs extra protection for the integrity of the message. In the SPREAD scheme, we use the threshold secret sharing algorithm to divide the message into multiple pieces. With a (T, N) secret sharing algorithm, the secret

message can be divided into N pieces (called *message shares*) such that in order to compromise the message, the adversary must compromise at least T shares. With fewer than T shares, the enemy cannot learn anything about the message and has no better chance to recover the secret than an outsider who knows nothing at all about the message. This gives us the desirable security properties. Another reason that we use secret sharing is that the generation of the message shares and the reconstruction of the message are all linear operations over a finite field (e.g., the Shamir's Lagrange interpolating polynomial scheme [26]). In addition, the secret sharing scheme can be designed with cheating detection and cheater identification [27] capabilities. It is possible that after compromising a node, the adversary attempts to cheat our system by sending us faked or altered message shares. By embedding the cheater detection and identification, we can deterministically detect cheating actions and identify the cheaters, no matter how many cheating shares are involved in the secret reconstruction. This is a desirable detection mechanism in an unreliable ad hoc network environment and it also helps to protect the integrity of the message transmitted.

3.3 Share allocation

The second issue is how to allocate the shares onto each selected path so that the adversary has least possibility to compromise the message. We consider the case that a message is compromised due to compromised nodes. We assume that if a node is compromised, all the credentials of that node are compromised. So the message shares traveling through that node are all intercepted and compromised. Given the available independent paths and their corresponding security characteristics, the fundamental objective is to maximize the security by allocating the shares in such a way that the adversary must compromise maximal number of paths to recover the message. The simplest and most intuitive share allocation scheme is to choose N as the number of available paths, apply (N, N) secret sharing, and allocate one share onto each path. This will achieve the desired maximum security with least processing cost. However, in an ad hoc network, wireless links are instable and the topology changes frequently. Packets might be dropped during the transmission. In case that packet loss does occur, this type of non-redundant share allocation will disable the reconstruction of the message even at the intended destination. To deal with this problem, it is usually necessary to introduce some redundancy (i.e., $T < N$) in the SPREAD scheme to improve the reliability, i.e., the destination would have better chance to receive enough shares for reconstructing the message. Generally speaking, the security and the reliability are two contradictive design goals—more redundancy implies better reliability but worse security. However, due to the salient features of the threshold secret sharing, we develop a redundant SPREAD share allocation scheme, which could tolerate certain packet loss while at the same time maintain the maximum security. We formulate the share allocation into a constrained optimization problem, with the objective to minimize the message compromising probability. Our investigation reveals that, by choosing an appropriate (T, N) value and allocating the shares onto each selected path wisely, we could tolerate certain packet loss without sacrificing the security. The maximum redundancy we can add to the SPREAD scheme without sacrificing the security is identified.

3.4 Multipath routing

The third issue is the multipath routing—how to find the desired multiple paths in a mobile ad hoc network and how to deliver the shares to the destination using these paths? Routing in a MANET presents great challenge because the nodes are capable of moving and the network topology can change continuously, dramatically, and unpredictably. A great effort has been made in designing ad hoc routing protocols in response to the frequent topological changes, among which multipath routing technique is a promising choice. One advantage is that the use of multiple paths in a MANET could diminish the effect of unreliable wireless links and the frequent topological changes. Another advantage is that, nodes in an ad hoc network is usually battery powered, by carefully distributing traffic load onto multiple paths, the energy consumption in each node can be made more evenly, hence the overall system lifetime (i.e., the time before the first node dies because of running out of battery) be prolonged.

For our SPREAD scheme, we need independent paths, more specifically, node-disjoint paths, as many as possible, because we are dealing with node compromise problem. Several multipath routing protocols have been proposed in MANETs with the design goal of finding node-disjoint paths, e.g., the split multipath routing [17], the diversity injection technique [18], and the on-demand multipath routing [19], etc. Those protocols are all on-demand, due to the network bandwidth limitation, and of the source routing type, as the source routing provides the source with the maximal capability of controlling the path disjointness. The multipath discovery technique proposed for SPREAD also takes a similar on-demand and source routing approach. We propose a security related cost function by which we define the security as one dimension in Quality of Service (QoS) routing and make the security a measurable routing cost metric. Therefore, satisfactory paths could be found according to their security levels (i.e., the probability that path might be compromised). Besides, we proposed in [28] a "link cache" organization for on-demand routing protocols, where each path returned to the source is decomposed into individual links and represented in a unified graph data structure. The

"link cache" organization allows us to take advantage of any underlying routing protocols and provide us with a partial view of network topology. Then we utilize a maximal node-disjoint path algorithm to find multiple node-disjoint paths so that the overall path set provides the maximum security for the message delivery.

4 Message share generation

To help better understand our scheme, we give a brief introduction to the threshold secret sharing system, which is used to generate the shares from a message (messages). Suppose that we have a system secret K and we divide it into N pieces, $S_1, S_2, \ldots, S_N$, called *shares or shadows*. Each of N participants of the system, $P_1, P_2, \ldots, P_N$, holds one share of the secret, respectively. The generation of the secret shares guarantees that fewer than T participants cannot learn anything about the system secret K, while with an effective algorithm, any T out of N participants can reconstruct the system secret K. This is called a (T, N) *threshold secret sharing scheme* [26]. A secret sharing scheme consists of two algorithms. The first is called the *dealer*, which generates and distributes the shares among the participants. The second is called the *combiner*, which collects shares from the participants and re-computes the secret, i.e., it re-produces the secret K from any T correct shares. A combiner fails to re-compute the secret if the number of the correct shares is less than T. Naturally, in our SPREAD, the dealer is implemented at the source while the combiner at the destination.

Several threshold secret sharing schemes have been developed in the literature. For illustration purpose, we take the Shamir's Lagrange interpolating polynomial scheme as an example. The dealer obtains the ith share by evaluating a polynomial of degree $(T-1)$

$$f(x) = (a_0 + a_1 x + \cdots + a_{T-1} x^{T-1}) \bmod p$$

at $x = i$:

$$S_i = f(i)$$

where p is a large prime number greater than any of the coefficients and is made available to both the dealer and the combiner, and the coefficient a_0 $(= K)$ is the secret while other coefficients $a_1, a_2, \ldots, a_{T-1}$ are all randomly chosen. Then, at a combiner, once T shares have been obtained, the combiner can reconstruct the original polynomial by solving a set of linear equations over a finite field GF(p). For example, assume that the received T shares are $S_{i1}, S_{i2}, \ldots, S_{iT}$, the original polynomial $f(x)$ can be recovered by Lagrange interpolation.

$$f(x) = \sum_{j=1}^{T} S_{i_j} \cdot l_{i_j}(x) \bmod p$$

where

$$l_{i_j}(x) = \prod_{k=1, k\neq j}^{T} \frac{x - i_k}{i_j - i_k}$$

Particually, the original secret K can be recovered by calculating $f(0)$.

Efficient $(O(T \log^2 T))$ algorithms for polynomial evaluation and interpolation have been discussed in [29]. Moreover, depending on the number of paths in a MANET, the (T, N) value in our SPREAD will not be large. Even the straightforward quadratic algorithms are fast enough for practical implementation.

Figure 1 is the illustration of applying the secret sharing algorithm onto the secret message at the source node. Limited by the size of the chosen prime number p, the secret sharing is applied on a block-by-block basis, which is similar to any block cipher used to encrypt a large message. Some cipher chaining mode such as the Cipher Block Chaining (CBC) might be used when concatenating the blocks together to further protect the message shares. In addition, depending on the number of paths used, the SPREAD seems to waste a lot of bandwidth. To save the network bandwidth, in SPREAD, more coefficients from $a_0, a_1, a_2, \ldots, a_{T-1}$ can be assigned message blocks, as shown in the figure.

5 Optimal share allocation

How to choose the appropriate values of (T, N) and allocate the N shares onto each selected path is another important issue in the SPREAD design. In this section, we discuss the share allocation with the objective of maximizing the message security.

5.1 Problem formulation and notations

Assume that (T, N) secret sharing algorithm is applied to the message to be protected at the source node. At the network layer, we assume that there are totally M node-disjoint paths, path 1, path 2, ..., path M, available from the source to the destination (comparable to M participants who hold the shares). We use vector $\underline{p} = [p_1, p_2, \ldots, p_M]$ to denote the security characteristics of the paths, where $p_i (i = 1, 2, \ldots, M)$ is the probability that path i is compromised.[1] Without loss of generality, we further assume $p_1 \leq p_2 \leq \cdots \leq p_M$. Then, we use vector $\underline{n} = [n_1, n_2, \ldots, n_M]$

[1] Note that such path security information, p_i, if made available, as we will discuss in the following section, is helpful in optimizing the secure path finding procedure. However, the unavailability or inaccuracy of such information does not diminish the effectiveness of SPREAD. In general, security can be improved by even blindly (e.g., assuming all the paths equally secure) spreading the traffic.

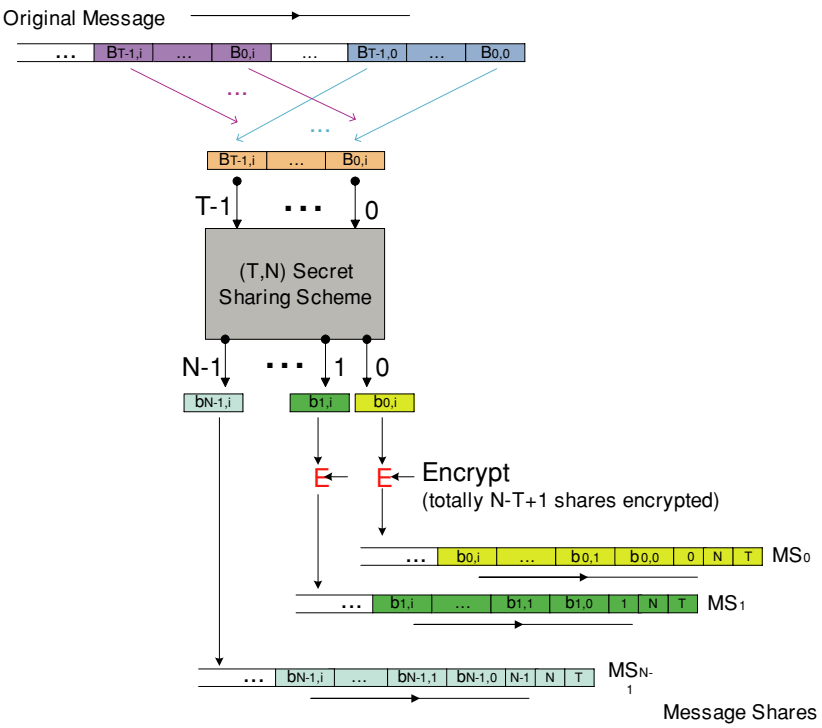


Fig. 1 (T, N) secret sharing system

to denote a share allocation that allocates the N shares onto the M available paths, where n_i is the integer number of shares allocated to path i satisfying $n_i \geq 0$ and $\sum_{i=1}^{M} n_i = N$. We assume that if one node is compromised, all the shares traveling through that node are compromised. Therefore, we define that a path is compromised when any one or more of the nodes along the path is compromised. For each path, we consider that if it is compromised, all the shares allocated to it are compromised. Otherwise, if the path is not compromised, all shares on that path are safe. As those paths are node-disjoint, we further assume that the probability that one path is compromised is independent of others. As we pointed out in previous section, SPREAD scheme only enhances the data confidentiality statistically when the data are transmitted across the network. Thus the probability p_i does not include the probability that the source or the destination node is compromised, i.e., we assume source and destination are trustworthy. The protection of a node from being compromised is another issue and is out of the scope of this paper.

According to the secret sharing algorithm, the probability that the message is compromised equals the probability that T or more shares are compromised. We denote the probability that the message is compromised in terms of the share allocation $\underline{n}$ as $P_{msg}(\underline{n})$. Then, the share allocation can be formulated as a constrained optimization problem

$$\text{minimize } P_{msg}(\underline{n})$$

$$\text{subject to } \sum_{i=1}^{M} n_i = N, n_i \text{ is an integer, } n_i \geq 0$$

5.2 Maximum security without redundancy

Let us define $r = 1 - \frac{T}{N}$ as the redundancy factor of the (T, N) secret sharing scheme. We first study the optimal allocation scheme when non-redundant SPREAD scheme where $r = 0$, e.g., $N = T$ is used. It is easy to derive that given the number of available paths, M, and the corresponding path security characteristics $\underline{p} = [p_1, p_2, \ldots, p_M]$, the non-redundant (N, N) $(N \geq M)$ secret sharing scheme would give the maximum security, i.e., the minimum message compromise probability, when at least one share and at most $T-1$ shares are allocated to each of the available paths, i.e.,

$$\begin{cases} 1 \leq n_i \leq T - M + 1, & i = 1, \ldots, M \\ \sum_{i=1}^{M} n_i = N \end{cases}$$

This share allocation forces the adversary to compromise all the paths to recover the message. This probability equals the probability that all the paths are compromised:

$$P_{msg}(\underline{n}) = \prod_{i=1}^{M} p_i$$

It is noted that, given the available multiple paths, the maximum security achievable only depends on the paths chosen. As p_i is a probability satisfying $0 \leq p_i \leq 1$, the more paths the source node uses to distribute the shares, the lower the probability is, and the more secure the message delivered. Thus, given a required security level (in terms of message

compromising probability) γ_{P_n}, the SPREAD scheme only needs to choose the first m paths, path 1, path 2, ..., path m, satisfying $P_{msg}(\underline{n}) = \prod_{i=1}^{m} p_i \leq \gamma_{P_n}$, to deliver the message. To simplify the notation and without loss of generality, in what follows, we still use M to denote the number of paths chosen to deliver the message.

5.3 Maximum security with redundancy

It is intuitive that the non-redundant secret sharing scheme provides the maximum security to the message. However, in the face of error-prone wireless channel it requires the successful reception of all the shares in order to reconstruct the original message. In an ad hoc network, wireless links are not stable. Packets might be dropped due to various reasons including broken routes, MAC layer collisions, or wireless channel fading, and so on. In order to mitigate the effect of such packet drops, it is usually necessary to add some redundancy for reliability purpose.

Again we assume that M most secure paths are selected to send the message. It is intuitive to show that, in order to achieve the maximum security, the total number of shares allocated to any $M-1$ or fewer paths should be less than T. This forces the adversary to compromise all the M paths to compromise the message. This is also a necessary and sufficient condition to achieve the maximum security. This condition can be represented as

$$\begin{cases} n_{i_1} \leq T-1, \quad \forall i_1 \in (1,2,\ldots,M) \\ n_{i_1} + n_{i_2} \leq T-1, \quad \forall i_1, i_2 \in (1,2,\ldots,M), i_1 \neq i_2 \\ \vdots \\ n_{i_1} + n_{i_2} + \cdots + n_{i_{M-1}} \leq T-1, \quad \forall i_1, i_2, \ldots, i_{M-1} \in (1,2,\ldots,M), i_1 \neq i_2, \ldots, i_{M-2} \neq i_{M-1} \\ n_1 + n_2 + \cdots + n_M = N \end{cases}$$

and simplified as

$$\begin{cases} N - n_i \leq T-1, \quad \forall i \in (1,2,\ldots,M) \\ n_1 + n_2 + \cdots + n_M = N \end{cases} \tag{1}$$

Recall $r = 1 - \frac{T}{N}$ is the redundancy factor of the secret sharing scheme. Then, we could derive a necessary and sufficient condition for achieving the maximum security, i.e.,

$$r \leq \frac{1}{M} - \frac{1}{N} \qquad (M \geq 2) \tag{2}$$

This is an important result as it defines the maximum redundancy we can add to the SPREAD scheme without sacrificing the security. It indicates that to maintain the maximum security achievable from the chosen path set, the maximum redundancy we can add to the secret sharing algorithm is bounded by $r \leq \frac{1}{M} - \frac{1}{N}$, where M is the number of chosen paths ($M \geq 2$) and N is the number of total shares generated. In other words, we could claim that for a r-redundancy SPREAD scheme, the maximum security can be achieved if and only if the redundancy factor r satisfies $r \leq \frac{1}{M} - \frac{1}{N}$.

In fact, with the (T, N) value satisfy the constraint (2), or in other words, with $T \geq \lceil N\left(1 - \frac{1}{M}\right) \rceil + 1$, any allocation that conforms to the constraints

$$\begin{cases} N-T+1 \leq n_i \leq N-(N-T+1)(M-1), \quad i=1,\ldots,M \\ \sum_{i=1}^{M} n_i = N \end{cases}$$

is an optimal share allocation in terms of security. Notice that the optimal share allocation in terms of security is not unique. Other optimization objectives, such as the minimal delivery cost, balanced bandwidth usage, or maximum reliability, might be set to further optimize the share allocation for other purposes.

6 Multipath routing and path set optimization

6.1 Multipath routing

As we mentioned earlier in the paper, routing in ad hoc networks is a great challenge. The challenge comes mainly from two aspects: constant node mobility causes frequent topological changes while limited network bandwidth restricts the timely topological updates. On-demand routing has been widely adopted in mobile ad hoc networks in response to the bandwidth constraints because of its bandwidth efficiency and effectiveness. The multipath routing technique is another promising technique to combat the frequent topological change and link instability problem in a MANET environment since the use of multiple paths could diminish the effect of possible link failures. Several multipath routing protocols have been proposed in the literature to find node-disjoint paths in an ad hoc network [17–19]. Most of the proposed protocols are on-demand, and use source routing technique to control the disjointness of the paths at the source node. For an on-demand routing protocol, whenever it needs a path to a certain destination but does not know one, it starts a route discovery process by broadcasting the route discovery messages throughout the network, the destination (or

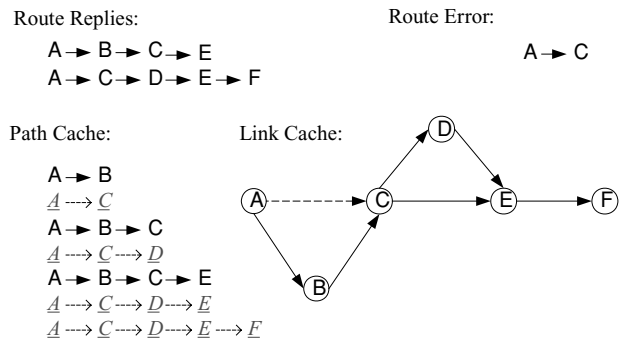


Fig. 2 Path cache and link cache

intermediate nodes that have a valid route to the destination) will reply by sending back the route. Some type of cache is necessary to store the routes previously found so that the node does not have to perform the costly route discovery for each individual packet. In many routing protocols (e.g., DSR, AODV), the routes replied back to the source contain the complete node list from the source to the destination. By caching each of these routes separately, a "path cache" organization can be formed. This type of cache organization has been widely used (e.g., DSR and multipath extension of DSR). However, the paths found by this means might not serve our purpose best because they are usually selected based on hop count or propagation delay, not necessarily the security. In [28], we designed an alternative cache organization, called a "link cache", in which routes are decomposed into individual links and represented in a unified graph data structure, as illustrated in Fig. 2. Given the same amount of route reply information, the routes existing in a path cache can always be found in a link cache, while a link cache has the potential to use the route information more efficiently by connecting individual links to form new paths which do not exist in the path cache. We also developed an adaptive stale link removal scheme to work together with the link cache. The proposed link cache scheme could be incorporated into any underlying multipath routing protocols, such as [17–19], to provide us with a partial view of network topology. Then, the optimization of the path set used to deliver the message shares can be carried independent of the routing protocols used, and is solely based on the discovered partial network topology.

6.2 Security related link cost function

We consider the security as the link cost function and discuss how to select a path based on nodes' security property. Assume in the mobile ad hoc network, each node n_i is associated with a security related parameter q_i. For simplicity and consistency, we interpret this security related parameter q_i as the probability that node n_i is compromised. In reality, q_i indicates the security level of node n_i and could be estimated by some security monitoring software and/or hardware such as firewalls and intrusion detection devices. It could also be evaluated using some theoretic trust evaluation technique [30]. It is important that these security levels are immutable, e.g., nodes should not be able to change their security levels arbitrarily in an unauthorized way. To ensure this, some form of authentication or tamper-resistant devices are needed. Here, we simply assume such a mechanism is already in place.

Based on the above argument, the probability that a (s, t) path consisting of node $s, n_1, n_2, \ldots, n_l, t$ is compromised is given by

$$p = 1 - (1 - q_1)(1 - q_2)\cdots(1 - q_l)$$

We define the following symmetric link cost function to convert the security characteristics into an additive link cost function so that the shortest path algorithm can be readily used to find the most secure path. The cost of the link between node n_i and n_j is defined as

$$c_{ij} = -\log\sqrt{(1 - q_i)(1 - q_j)}$$

Then the cost of the (s, t) path consisting of node $s, n_1, n_2, \ldots, n_l, t$ is

$$\begin{aligned}\cos t(s,t) &= c_{s1} + c_{12} + \cdots + c_{l-1,l} + c_{lt}\\ &= -\frac{1}{2}\log(1 - q_s) - \log(1 - q_1) - \log(1 - q_2)\\ &\quad - \cdots - \log(1 - q_l) - \frac{1}{2}\log(1 - q_t)\\ &= -\log\{(1 - q_1)(1 - q_2)\cdots(1 - q_l)\}\\ &\quad -\frac{1}{2}\log\{(1 - q_s)(1 - q_t)\}\end{aligned}$$

With the shortest path algorithm, s, t are fixed,

$\cos t(s, t)$ is minimized iff
$-\log\{(1 - q_1)(1 - q_2)\cdots(1 - q_l)\}$ is minimized iff
$(1 - q_1)(1 - q_2)\cdots(1 - q_l)$ is maximized iff
$p = 1 - (1 - q_1)(1 - q_2)\cdots(1 - q_l)$ is minimized.

With this definition, the non-additive security metric can be transformed into an additive one. The path found by the conventional shortest path algorithm will be the most secure path. If we treat security as a dimension of quality of service (QoS) routing, many other QoS routing protocols [31] that are developed for other additive performance metrics, such as end-to-end delay, can also be applied easily with this new metric.

6.3 Path set optimization

Ideally, given a network, we wish to find an optimal path set, such that the probability P_{msg} is minimized. Recall that

Fig. 3 Maximal node disjoint paths finding algorithm

Step 1. Find the first most secure path by modified Dijkstra algorithm, select the path
Step 2. Perform a graph transformation as follows
  a. For each selected path:
  b. Replace the links used in the path with directed arcs – for the arc that is directed towards the source, make its cost the negative of the original link cost; make the cost of the arc directed towards the destination infinite (i.e., remove it)
  c. Split each node on the selected paths (except the source and destination) into two co-located subnodes; Connect the two subnodes by an arc of cost 0 and directed towards the source node.
  d. Replace each external link that is connected to a node in the selected paths by its two component arcs of cost equal to the link cost – let one arc terminate on one subnode and the other one emanate from the other subnode such that along with the zero-cost arc, a cycle does not result.
Step 3. Run the modified Dijkstra algorithm, find the most secure path in the transformed graph
Step 4. Transform back to the original graph; erase any interlacing edges; group the remaining edges to form the new path set.
Step 5. Go to step 2, until no more path can be found or the security of the path set does not increase.

$P_{msg}(\underline{n}) = \prod_{i=1}^{M} p_i$. If given the available M paths, intuitively, since p_i is a probability which is always less than 1, the more terms of p_i, the lower the probability P_{msg}, and the better the security we can achieve. So the primary goal of our path finding algorithm is to find as many paths as possible while at the same time as secure as possible.

The maximal path finding algorithm proposed for our SPREAD scheme is modified from the node-disjoint shortest pair algorithm [32]. The basic idea of the algorithm is not simply removing the nodes on the selected paths; instead, a graph transform is applied so that the selected nodes and links can be temporarily reused; then after the interlacing removal, when the graph is transformed back to the original format, the maximal number of paths can be found by regrouping the selected links. In [22], Papadimitratos et al. proposed a similar disjoint path set selection protocol (DPSP) which aims to select multiple disjoint paths in a mobile ad hoc network. The objective of their multipath routing is to combat the path failures due to topological changes. The disjoint paths they found are edge-disjoint while in our case we find node-disjoint paths.

A modified Dijkstra algorithm is used so that negative links are allowed (but no negative loop) in the graph [32]. The modified Dijkstra algorithm modifies the standard Dijkstra algorithm by allowing the permanent labeled node to change back to a tentative label if a smaller cost to that node has been found. Based on this observation, we develop a new maximal path finding algorithm. The maximal path finding algorithm we propose here is an iterative procedure. The most secure path is found first and added to the path set. In one iteration, the number of paths in the set is augmented by one. Figure 3 summarizes the steps taken to find the maximal number of paths. Each time a new path is added to the selected pathset, a graph transformation is performed, which involves a vertex splitting of the nodes on the selected paths (except the source and destination node). Thus, the modified Dijsktra algorithm is executed to find the most secure path in the transformed graph. Then, the split nodes are transformed back to the original one, any interlacing edges are erased, and the remaining edges are grouped to form the new path set. Figure 4 shows an example of the path finding algorithm. After finding the first two node-disjoint paths, the third one temporarily makes use of the selected nodes but using the link in the reverse direction. After the interlacing removal and regrouping, a path set consisting of three paths is found instead of two.

The intuition we used in the pervious section—the more paths used, the better security achieved, is based on the assumption that candidate paths are fixed. When an additional path is added to the path set, the paths previously selected are not affected. However, with our iterative multipath selection algorithm, because of the regrouping of edges, the paths in the path set in each iteration might change. It is unnecessary that the addition of one path will decrease the overall probability. In order to cope with this dynamics, we recalculate P_{msg} after each iteration. If P_{msg} is not getting smaller in the iteration, the path set found in the previous iteration will be accepted and the path finding algorithm terminates.

This algorithm is selected because of its simplicity and its capability to find the maximum number of node-disjoint paths. Alternatively, other disjoint path finding algorithms that are capable of finding maximum number of node-disjoint paths with minimum total cost may be used, such as the algorithms proposed in [33, 34].

7 Performance evaluation

7.1 Simulation configuration

We simulate an ad hoc network with 100 nodes randomly distributed in a 1000 m by 1000 m area. Two sets of simulations

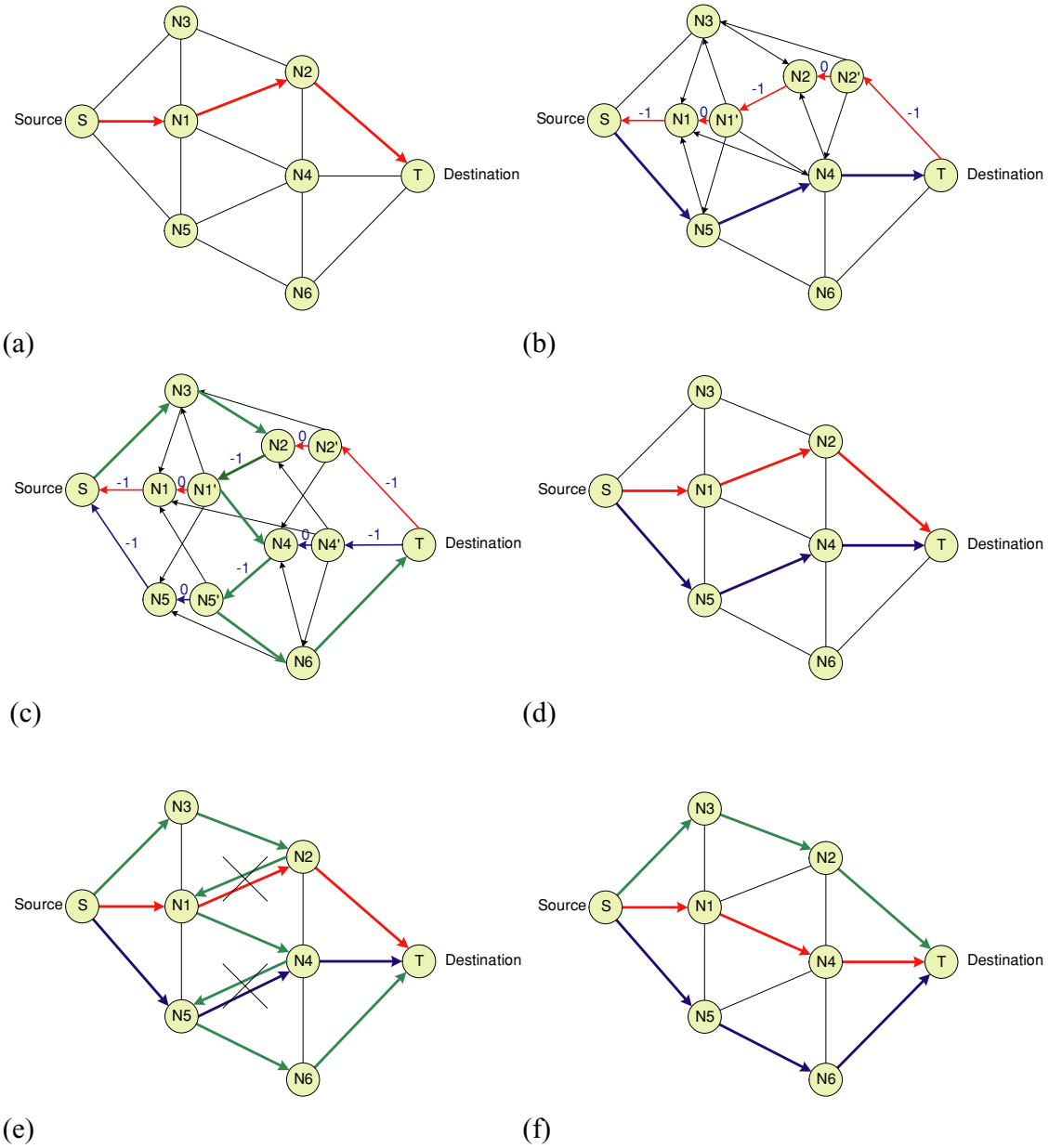


Fig. 4 An Illustration of the maximal node disjoint paths algorithm. (a) Iteration 1—modified Dijkstra algorithm; (b) Iteration 2—graph transformation and modified Dijkstra algorithm; (c) Iteration 2—resulting 2 paths; (d) Iteration 3—graph transformation and modified Dijkstra algorithm; (e) Iteration 3—edge regrouping; (f) Iteration 3–resulting 3 paths

are conducted. The first set of simulations focuses on the feasibility and effectiveness of the SPREAD. The simulation scenario is relatively ideal. Nodes are not mobile, and multiple independent logical channels are assumed among nodes so that multiple paths can be deployed independently at the network layer. This set of simulations is implemented by C/C++. The second set of simulations aims to examine more performance metrics under more realistic and dynamic scenarios with node mobility model (random waypoint model with [vmin, vmax] = [0,20 m/sec]), contention-based MAC protocol (IEEE 802.11 Wireless LAN standard), and radio model (frequency hopping spread spectrum technology with 2 Mbps capacity). The second set of simulations is implemented in OPNET [35].

To factor out the effect of routing protocols, in the simulation we assume each node knows the network topology. However, to keep the fundamental features of the on-demand routing protocols, a node only refreshes its network topology data structure when it has a message to be transmitted and there is no known path to that destination (to mimic the route discovery procedure). The multipath finding algorithm is then executed to find the desired number of node-disjoint

paths. In simulation set 1, we always assume the optimal share allocation. However, in simulation set 2, we use the following simple share allocation. Each message is divided into 10 shares and sent to the destination via the M paths. For $M=1$, $\underline{n}=[10]$; $M=2$, $\underline{n}=[5\ 5]$; $M=3$, $\underline{n}=[4\ 3\ 3]$; $M=4$, $\underline{n}=[3\ 3\ 2\ 2]$; $M=5$, $\underline{n}=[2\ 2\ 2\ 2\ 2]$. The routing is achieved by the source routing technique. A route cache is kept in each node to save the paths used. Once the paths to a certain destination are calculated, they are used till a link error occurs (to mimic the route maintenance mechanism).

Two types of security settings are simulated in each simulation set. In type 1 security setting, each node is assumed equally likely to be compromised with probability $q_i=0.14$. In the second type, each node is assigned a probability randomly: 10% of nodes being compromised with probability $q_i=0.50$, 40% of nodes with $q_i=0.20$, and 50% of nodes with $q_i=0.02$. We will denote these two types of security property settings as "equal Qi" and "different Qi" in the figures.

7.2 Feasibility

Table 1 gives some basic parameters of the network topology of the simulated ad hoc networks. It is known that in the highly dynamic ad hoc networks, in order to maintain the connectivity, ad hoc networks typically have dense connectivity that allows the exploitation of multipath routing techniques.

Figure 5 shows the probability that multiple paths are found in the simulated network. It is observed that the probability that multiple node-disjoint paths exist in an ad hoc network is pretty high. Since our SPREAD scheme depends on the availability of multiple node-disjoint paths, the existence of such multiple paths justifies the feasibility of our scheme.

In fact, if we run the maximal node-disjoint path finding algorithm purely for finding the maximum number of paths without considering the security property of the path set, the number of paths found in both sets would be equal. This implies that the maximum number of paths the algorithm is able to find is independent of the link costs; it solely depends on the network topology although the actual paths found might be different for different link costs. In our simulation, we stop augmenting the path set when the security property of the found path set does not improve. Table 2 gives the probability that the path finding algorithm stops for this reason. It indicates that when the nodes are of equal security level, the number of paths plays the most significant role. Basically, the more the paths, the better the security. However, if nodes are of different security levels (probabilities), the security of each path will have more impact on the overall security of the path set. This also explains that in Fig. 5 the number of paths selected in type 2 simulations (different Qi) is fewer than that in type 1 simulations (equal Qi).

Table 1 Network parameters

TR(m)	200	250
Node degree	10.3	15.4
Diameter	9	6.8

Table 2 The path finding algorithm stops before finding the maximum number of paths

TR(m)	200	250
Same Qi	0.45%	0.33%
Different Qi	22.7%	38.8%

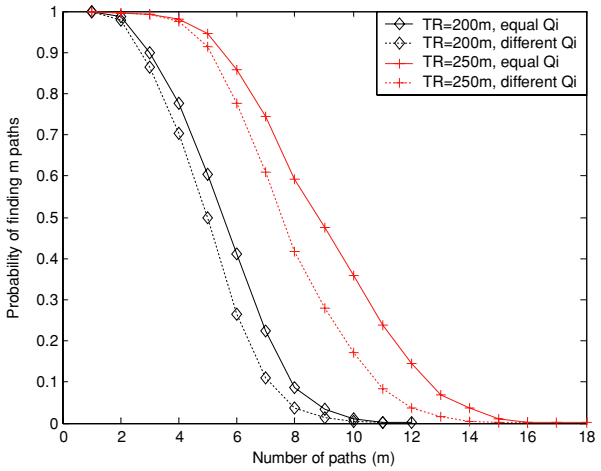


Fig. 5 Capability of path finding

7.3 Message compromise probability

We examine the security performance of our SPREAD in terms of the probability that a message is compromised. Since we assume link encryption is used, if one share is relayed by a compromised node, the share is compromised. If T out of N shares are compromised, we assume that the message is compromised. Obviously, the individual node attack on a message does not work when multiple ($M \geq 2$) paths are used because no single node is able to relay all the necessary shares. Here, we study the collusion attack where some kinds of collaborations among compromised nodes are assumed so that they could add up together their compromised shares to recover the original message. Figure 6 shows this probability when multipath paths are used and different secret sharing schemes are used. Figure 6(a), where the logarithmic scale of the Y axis is used, is from the first set of simulations and assumes the optimal share allocation. We observe that the message compromise probability drops quickly (actually exponentially fast) with the increase of the number of paths used. This result verifies the effectiveness of our SPREAD idea. We also notice that

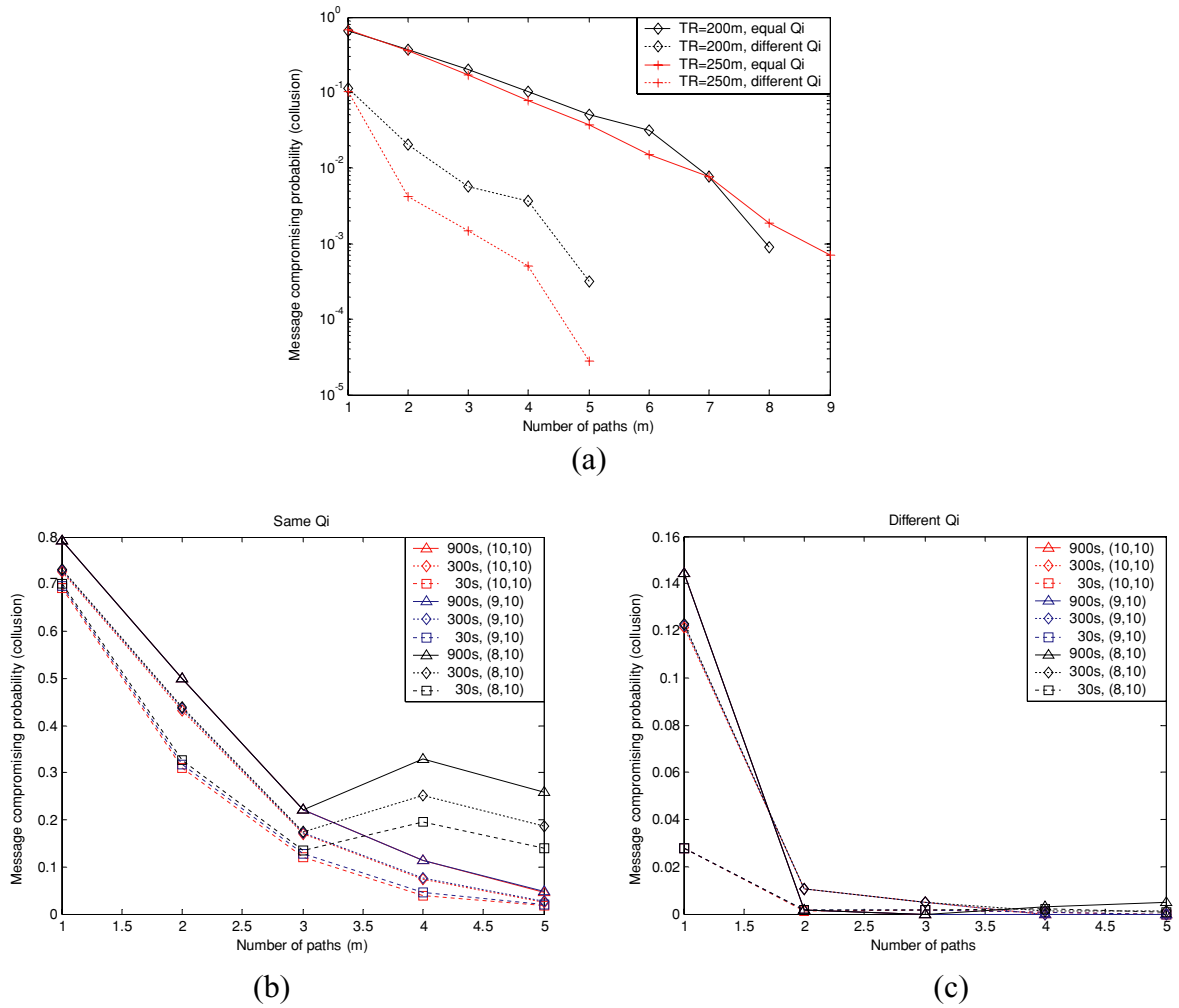


Fig. 6 Message compromise probability

when nodes are with different security levels, our algorithm tends to select more secure paths that further decrease this probability significantly. Figure 6(b) and (c) are from the second set of simulations. Notation {300s, (9, 10)} means that the curve is obtained when the pause time in the mobility model is set to 300s, (T, N) is set to (9, 10). The results verify the above observation for the message compromise probability. In addition, we observe that the node mobility seems to have a favorable impact on the message security. This could be explained as, when nodes are moving, the chosen paths are changing over time. The introduced path dynamics reduce the chance for the adversary to collect enough shares. We also notice that for $m = 4$ and 5, the security is sacrificed for (8, 10) SPREAD scheme because the non-optimal share allocation scheme is used.

7.4 Message eavesdropping probability

We also examine the message eavesdropping probability. As we use a single shared channel, when one node transmits a packet, all its neighbors would be able to overhear that packet. If a compromised node overhears T or more shares for a particular message, this message is considered eavesdropped. Figure 7 plots the message eavesdropping probability for individual node attack, which means that each node works on its own to collect the T shares. It is observed that, with the increase of the number of paths, this probability decreases. However, the decrease becomes less significant when more paths are used. In fact, there is a lower bound of this probability because anyone sits within the transmission range of the source node would be able to overhear all the shares. The message eavesdropping probability for collusion attack is pretty high (close to 1) because in our simulation, we have about 14 compromised nodes among the totally 100 nodes. The receiving range of all the compromised nodes has almost covered the whole simulation area. The simulation results with equal Qi is very similar to the ones with different Qi as shown in Fig. 7(b) and (c), which indicate that the physical security of each node has little impact on the eavesdropping of the wireless broadcast channels. The

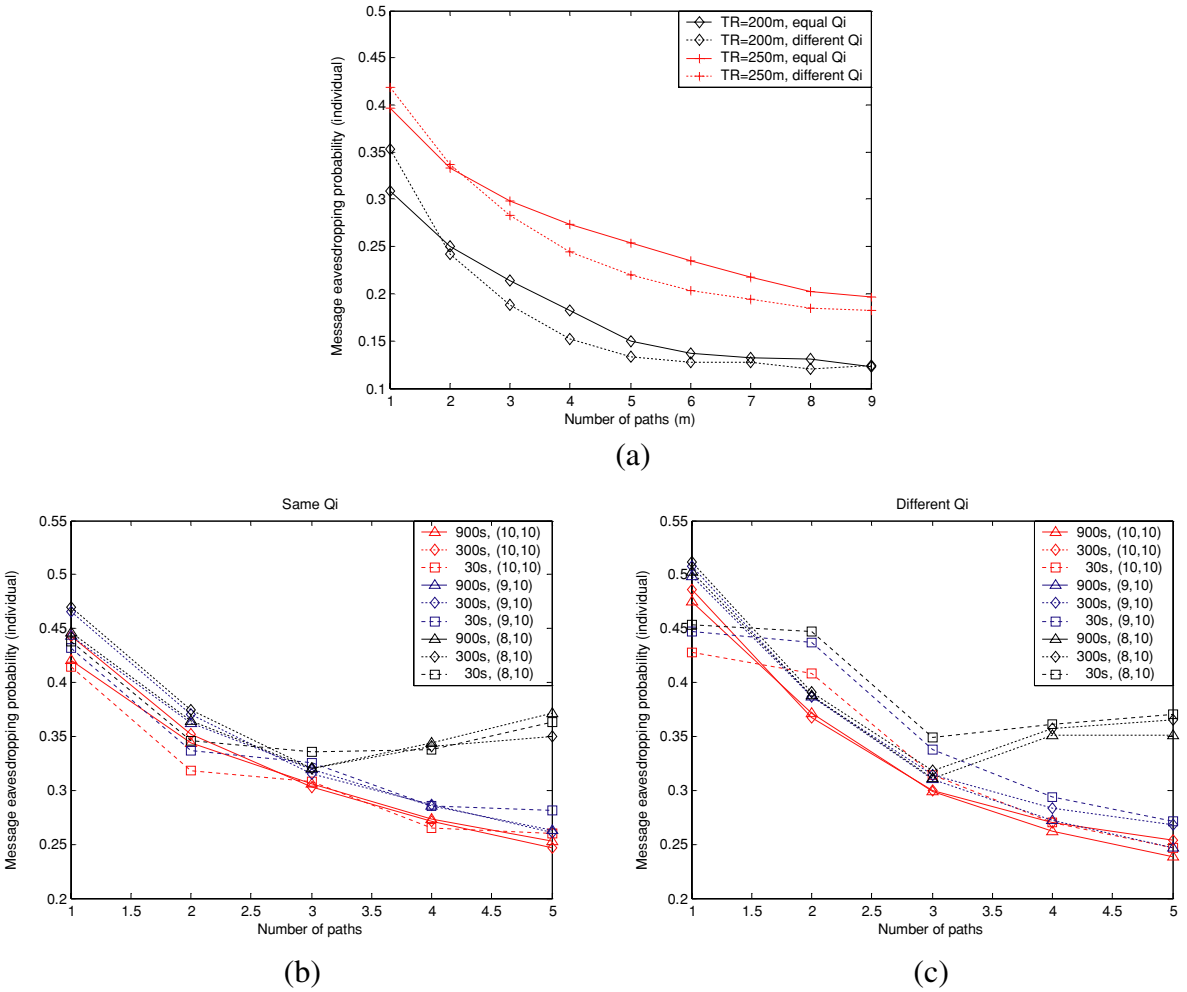


Fig. 7 Message eavesdropping probability

node mobility also has little impact on the message eavesdropping probability. However, we could observe that the optimal share allocation does have noticeable impact on the scheme. When non-optimal share allocation is used, adversaries are more likely to eavesdrop enough shares. Although SPREAD does lower the message eavesdropping probability a little bit, this result implies that more efficient ways to defend against eavesdropping in a wireless network probably should be sought at the physical layer.

7.5 Bandwidth overhead

Figure 8 shows the bandwidth overhead calculated on a per-hop basis compared with the single minimum-hop path case. We can see that using multipath does consume more network bandwidth because of longer paths used. However, this is the tradeoff. We argue that for security critical applications, the network efficiency might not be as critical a concern as security.

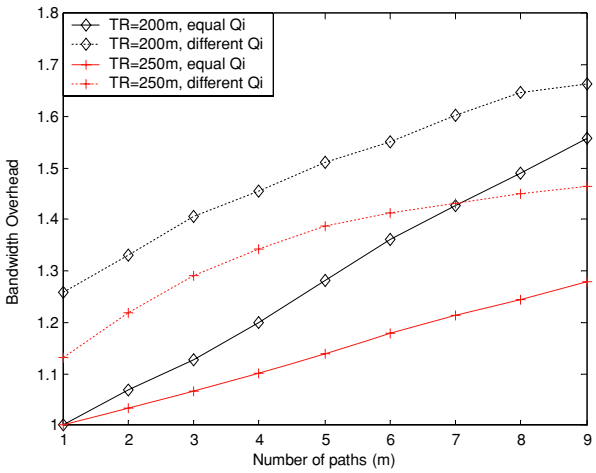


Fig. 8 Bandwidth overhead

7.6 More discussions

The security enhancement of the SPREAD scheme comes from the concept of secret sharing and is achieved in the network by spreading the message shares onto multiple paths. The key to the performance improvement relies on the independence of the multiple paths used to deliver the secret, i.e., the compromise of one path is independent of the other. This implies that in reality, fully diversified paths are desirable to achieve the security enhancement by SPREAD. For example, in a heterogeneous tactical communication scenario, paths involving various types of ground communication facilities, paths using satellite or aircrafts in the air, and paths involving ships or submarines in the sea, will provide desired path independence. The assumption used in our evaluation, namely that all the nodes are sharing the same broadcast channel and are using short range wireless transmissions, actually represents the worst case scenario. Even in a homogeneous MANET where all the nodes having identical configuration, the desired path independence can be achieved by various means. For example, when each node is equipped with a directional antenna so it can transmit to the direction of its intended receiver, or when each link is assigned a locally unique channel that is distinct from those channels used by its two-hop neighbors to avoid collision. Independent paths can also be established at the higher layer. With the multiple-channel model, each link's communication activity is independent of those of its neighbors. The concurrent data transmission in overlapped neighborhoods is possible. The message eavesdropping probability will also drop significantly to be similar to that of the message compromise probability, given the multiple-channel model.

Moreover, the SPREAD can be made adaptive by adjusting the number of multiple paths according to the security levels for a message. We admit that the SPREAD may consume more network bandwidth comparing to conventional security schemes, however, this may be acceptable for some applications, and probably desirable when reliability is a consideration.

The proposed SPREAD scheme is an enhancement to the data confidentiality service. It statistically enhances such service but it alone cannot completely guarantee data confidentiality without incorporating any underlying encryption scheme and/or LPI/LPD (low probability of interception/low probability of detection) schemes at the physical layer. Compared with the traditional end-to-end encryption and single path routing approach, SPREAD has the following advantages. First, SPREAD makes it more difficult for adversaries to intercept sufficient information to recover the secure message. As we know that the adversaries must possess both the ciphertext and the key in order to break an encrypted message. The SPREAD scheme enhances the security by protecting the former, i.e., the ciphertext, from being intercepted by the adversaries in a highly dynamic MANET environment where the key management is assumed not fully secure and reliable. Also, keys are sort of "wear out". The more encrypted data with the same key, the better chances that the adversaries break the encryption system by cryptanalysis. Protection of the ciphertext also provides "perfect forward secrecy" in the sense that a previously transmitted message can not be recovered by the adversaries even if the key used to encrypt the message is thereafter compromised. In addition, SPREAD potentially provides better reliability and failure resiliency [20, 21]. With single path routing, one compromised node might disrupt the delivery service between two end nodes, while with SPREAD, if one path fails/is disrupted, a retransmission scheme such as one described in [16] can be implemented within SPREAD framework to deliver lost data packets using other paths.

Recently, the Sybil attack [36] has been identified as one type of attacks to which multipath routing is vulnerable. In a Sybil attack, a single node presents multiple identities to other nodes in the network. Several techniques to defend against the Sybil attack have been proposed for a sensor network [37, 38]. Like many other distributed systems, our SPREAD scheme is not completely resistant to Sybil attack in the sense that our SPREAD scheme fails when a compromised node presents itself as multiple identities and our path finding algorithm happens to select at least one Sybil node on each of the paths selected. Although the defense of Sybil attacks is out of the scope of this paper, our SPREAD approach actually makes the Sybil attack harder as the Sybil nodes now need to present a greater number of identities and they need to pretend to be geographically separated during the on-demand route discovery process in order to be selected in multiple paths.

8 Conclusions

In this paper, we present a novel security enhancement scheme, namely, *Secure Protocol for Reliable Data Delivery (SPREAD)*. The basic idea of SPREAD is to distribute the secret, first by secret sharing algorithm at the source node to generate message shares and then by multipath routing to deliver message shares across the network, so that in the event that a small number of shares are compromised, the secret message as a whole will not be compromised. We investigate the major design issues of the SPREAD. Extensive simulation results show that the SPREAD can provide more secure data delivery when messages are transmitted across the insecure network. In particular, it is more resilient to compromised nodes problem. We also show that a redundant SPREAD scheme can be designed in such a way that a certain degree of reliability can be provided without sacrificing the security. Therefore, the SPREAD idea is a suitable

and promising approach to improve network security in the highly dynamic MANET environment.

Reference

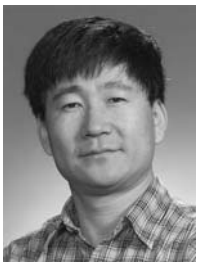
1. W. Lou and Y. Fang, A survey on wireless security in mobile ad hoc networks: Challenges and available solutions, book chapter in: *Ad Hoc Wireless Networking* (Kluwer, May 2003).
2. L. Zhou and Z.J. Haas, Securing ad hoc networks, IEEE Network Magazine 13(6) (November/December 1999).
3. J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, Providing robust and ubiquitous security support for manet, in: *ICNP* (2001)
4. L. Eschenauer and V. Gligor, A key-management scheme for distributed sensor networks, in: *ACM CCS 2002* (Washington, DC, 2002).
5. W. Du, J. Deng, Y. Han and P. Varshney, A pairwise key predistribution scheme for wireless sensor networks, in: *ACM CCS 03* (2003).
6. H. Chan, A. Perrig and D. Song, Randon key predistribution schemes for sensor networks, in: *IEEE Symposium on Security and Privacy (SP'03)* (Oakland, CA, May 2003).
7. S. Zhu, S. Xu, S. Setia and S. Jajodia, Establishing pairwise keys for secure communication in ad hoc networks: A probabilistic approach, in: *11th IEEE Internationaal Conference on Network Protocols (ICNP'03)* (Atlanta, GA, November 2003)
8. Y. Zhang, W. Liu, W. Lou and Y. Fang, Location-based compromise-tolerant security mechanisms for wireless sensor networks, IEEE Journal on Selected Areas in Communications (Special Issue on Security in Wireless Ad Hoc Networks) 24(2), (February 2006) 247–260.
9. K. Ren, W. Lou and Y. Zhang, LEDS: Providing location-aware end-to-end data security in wireless sensor networks, in: *IEEE INFOCOM 2006* (Barcelona, Spain, April 2006).
10. Y.-C. Hu, D.B. Johnson and A. Perrig, SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks, in: *WMCSA'02*
11. Y.-C. Hu, A. Perrig and D.B. Johnson, Ariadne: A secure on-demand routing protocol for ad hoc networks, in: *MobiCom 2002* (September 2002).
12. P. Papadimitratos and Z.J. Haas, Secure routing for mobile ad hoc networks, in: *CNDS 2002* (San Antonio, TX, January 2002)
13. H. Yang, X. Meng and S. Lu, Self-organized network-layer security in mobile ad hoc networks, in: *ACM WiSe'02* (September 2002).
14. Y. Zhang, W. Lee and Y. Huang, Intrusion detection techniques for mobile wireless networks, ACM Wireless Networks Journal 9(5) (Sep. 2003).
15. S. Marti, T. Giuli, K. Lai and M. Baker, Mitigating routing misbehavior in mobile ad hoc networks, in: *MobiCom'00* (Boston, MA, USA, August 2000).
16. P. Papadimitratos and Z. Haas, Secure data transmission in mobile ad hoc networks, in: *WiSe'03* (San Diego, CA, September 2003).
17. S.-J. Lee and M. Gerla, Split multipath routing with maximally disjoint paths in ad hoc networks, in: *ICC*'01.
18. M.R. Pearlman, Z.J. Haas, P. Sholander and S.S. Tabrizi, On the impact of alternate path routing for load balancing in mobile ad hoc networks, in: *MobiHOC* (2000)
19. K. Wu and J. Harms, Performance study of a multipath routing method for wireless mobile ad hoc networks, in: *9th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication System* (2001)
20. A. Tsirigos and Z.J. Haas, Analysis of multipath routing, part 1: The effect on the packet delivery ratio, IEEE Transactions on Wireless Communications 3(Issue 1) (Jan 2004) 138–146
21. A. Tsirigos and Z.J. Haas, Analysis of multipath routing, part 2: Mitigation of the effects of frequently changing network topologies, IEEE Transactions on Wireless Communications 3(Issue 2) (March 2004) 500–511.
22. P. Papadimitratos, Z.J. Haas and E.G. Sirer, Path set selection in mobile ad hoc networks, The ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'2002), EPFL Lausanne, Switzerland (June 2002).
23. W. Lou and Y. Fang, A multipath routing approach for secure data delivery, in: *IEEE Military Communications Conference (MILCOM 2001)* (Mclean, VA, USA, Oct. 2001).
24. W. Lou, W. Liu and Y. Fang, SPREAD: Improving network security by multipath routing, in: *IEEE Military Communications Conference (MILCOM 2003)* (Boston, M, Oct. 2003).
25. W. Lou, W. Liu and Y. Fang, SPREAD: Enhancing data confidentiality in mobile ad hoc networks, in: *IEEE INFOCOM 2004* (Hong Kong, China, Mar 2004).
26. A. Shamir, How to Share a Secret, Communications of the ACM 22(11) (Nov 1979) 612–613.
27. T.-C. Wu and T.-S. Wu, Cheating detection and cheater identification in secret sharing schemes, IEE Proc. Comput. Digit. Tech. 142(5) (September 1995).
28. W. Lou and Y. Fang, Predictive caching strategy for on-demand routing protocols in ad hoc networks, Wireless Networks 8(6) (Nov 2002).
29. T. Cormen, C. Leiserson and R. Rivest, *Introduction to Algorithms* (MIT Press, 1990).
30. Y. Sun, W. Yu, Z. Han and K.J.R. Liu, Information theoretic framework of trust modelling and evaluation for ad hoc networks, IEEE Journal on Selected Areas in Communications (Special Issue on Security in Wireless Ad Hoc Networks) 24(2) (February 2006) 305–317.
31. S. Chen and K. Nahrstedt, An overview of quality of service routing for next-generation high-speed networks: problems and solutions, IEEE Networks 12(6) (November/December 1998) 64–79.
32. R. Bhandari, *Survivable Networks—Algorithms for Diverse Routing* (Kluwer Academic Publisher, 1999).
33. J.W. Suurballe, Disjoint paths in a network, Networks 4 (1974) 125–145.
34. J.W. Suurballe and R.E. Tarjan, A quick method for finding shortest pairs of disjoint paths, Networks 14 (1984) 325–336.
35. http://www.opnet.com
36. J.R. Douceur, The Sybil attack, in: *1st International Workshop on Peer-to-Peer Systems (IPTPS'02)* (March 2002).
37. C. Karlof and D. Wagner, Secure routing in wilress sensor networks: Attacks and countermeasures, in: *1st IEEE Interntaional Conference on Sensor Network Protocols and Applications* (May 2003).
38. J. Newsome, E. Shi, D. Song and A. Perrig, The Sybill attack in sensor networks: Analysis & defenses, in: *IPSN'04* (Berkelwy, CA, April 2004).

Wenjing Lou is an assistant professor in the Electrical and Computer Engineering department at Worcester Polytechnic Institute. She received her Ph.D. degree in Electrical and Computer Engineering from University of Florida in 2003. She received a M.A.Sc degree from

Nanyang Technological University, Singapore, in 1998, a M.E. degree and a B.E. degree in Computer Science and Engineering from Xi'an Jiaotong University, China, in 1996 and 1993 respectively. From December 1997 to July 1999, she worked as a Research Engineer in Network Technology Research Center, Nanyang Technological University. Her current research interests are in the areas of wireless ad hoc and sensor networks, with emphases on network security and routing issues.

Wei Liu received his B.E. and M.E. in Electrical and Information Engineering from Huazhong University of Science and Technology, Wuhan, China, in 1998 and 2001. In August 2005, he received his Ph.D. in Electrical and Computer Engineering from University of Florida. Currently, he is a senior technical member with Scalable Network Technologies. His research interest includes cross-layer design, and communication protocols for mobile ad hoc networks, wireless sensor networks and cellular networks.

Yanchao Zhang received the B.E. degree in computer communications from Nanjing University of Posts and Telecommunications, Nanjing, China, in July 1999, the M.E. degree in computer applications from Beijing University of Posts and Telecommunications, Beijing, China, in April 2002, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in August 2006. Since September 2006, he has been an Assistant Professor in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark. His research interest include wireless and Internet security, wireless networking, and mobile computing. He is a member of the IEEE and ACM.



Yuguang Fang received a Ph.D. degree in Systems Engineering from Case Western Reserve University in January 1994 and a Ph.D. degree in Electrical Engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at University of Florida in May 2000 as an assistant professor, got an early promotion to an associate professor with tenure in August 2003 and to a full professor in August 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009. He has published over 200 papers in refereed professional journals and conferences. He received the National Science Foundation Faculty Early Career Award in 2001 and the Office of Naval Research Young Investigator Award in 2002. He has served on several editorial boards of technical journals including IEEE Transactions on Communications, IEEE Transactions on Wireless Communications, IEEE Transactions on Mobile Computing and ACM Wireless Networks. He have also been activitely participating in professional conference organizations such as serving as The Steering Committee Co-Chair for QShine, the Technical Program Vice-Chair for IEEE INFOCOM'2005, Technical Program Symposium Co-Chair for IEEE Globecom'2004, and a member of Technical Program Committee for IEEE INFOCOM (1998, 2000, 2003–2007). He is a senior member of the IEEE.