SHWMP: A Secure Hybrid Wireless Mesh Protocol for IEEE 802.11s Wireless Mesh Networks*

Md. Shariful Islam[1], Md. Abdul Hamid[2], and Choong Seon Hong[1,**]

[1] Department of Computer Engineering, Kyung Hee University, Republic of Korea
sharif@networking.khu.ac.kr, cshong@khu.ac.kr
[2] Department of Information and Communications Engineering,
Hankuk University of Foreign Studies, Republic of Korea
hamid@hufs.ac.kr

Abstract. In recent years, mesh networking has emerged as a key technology for the last mile Internet access and found to be an important area of research and deployment. The current draft standard of IEEE 802.11s has defined routing for Wireless Mesh Networks (WMNs) in layer-2 and is termed as Hybrid Wireless Mesh Protocol (HWMP). However, security in routing or forwarding functionality is not specified in the standard. As a consequence, HWMP in its current from is vulnerable to various types of routing attacks such as flooding, route disruption and diversion, spoofing etc. In this paper, we propose SHWMP, a secure HWMP protocol for WMN. The proposed protocol uses cryptographic extensions to provide authenticity and integrity of HWMP routing messages and prevents unauthorized manipulation of mutable fields in the routing information elements. We show via analysis that the proposed SHWMP successfully thwarts all the identified attacks. Through extensive ns-2 simulations, we show that SHWMP provides higher packet delivery ratio with little increase in end-to-end delay, path acquisition delay and control byte overhead.

Keywords: Wireless Mesh Network, Secure Hybrid Wireless Mesh Protocol, Authentication, Merkle Tree.

1 Introduction

Wireless mesh networking (WMN) has emerged as one of the most promising concept for self-organizing and auto-configurable wireless networking to provide adaptive and flexible wireless Internet access solutions for mobile users. Potential application scenarios for wireless mesh networks include backhaul support for cellular networks, home networks, enterprise networks, community networks, and intelligent transport system networks [1]. The increased interest in WMN has reflected in producing a standard named IEEE 802.11s, which is in progress and expected to be finalized by mid 2009. Our work is based on the current draft version D2.02 [2] of

* "This research was supported by the MKE, Korea, under the ITRC support program supervised by the NIPA)"(NIPA-2009-(C1090-0902-0016)).

** Corresponding author.

M.L. Gavrilova and C.J.K. Tan (Eds.): Trans. on Comput. Sci. VI, LNCS 5730, pp. 95–114, 2009.

IEEE 802.11s that introduces the concept of embedding routing in layer-2 named Hybrid Wireless Mesh Protocol (HWMP). HWMP has been developed to ensure interoperability between devices of different vendors and is the key reason for integrating routing in MAC layer.

Wireless mesh networks (WMNs) consist of mesh clients and mesh routers, where the mesh routers form a wireless infrastructure/backbone and interwork with the wired networks to provide multihop wireless Internet connectivity to the mesh clients. The network architecture of a 802.11s WMN is depicted in Fig. 1. A mesh point (MP) is an IEEE 802.11s entity that mainly acts as a relay node. A mesh access point (MAP) is an MP but can also work as an access point. Legacy wireless mobile stations (STA) are connected to an MAP through generic WLAN protocols. Thus, configuration of an MAP allows a single entity to logically provide both mesh functionalities and AP functionalities simultaneously. A mesh portal (MPP) is a logical point and has a bridging functionality and connects the mesh network to other networks such as a traditional 802.11 WLAN or a non-802.11 network and act as the gateway to the WMN infrastructure. In a WMN, traffic flows between MP to MP for intra-mesh traffic and between MP-MPP or vice-versa for traffic to / from outside mesh.

The security in routing or forwarding functionality is not specified in IEEE 802.11s [3]. Our study identifies that existing HWMP is vulnerable to various types of routing attacks such as flooding, route disruption and diversion, spoofing etc. The main reason is that the intermediate nodes need to modify mutable fields (i.e., hop count, TTL, metric etc) in the routing element before forwarding and re-broadcasting them. Since other nodes will act upon those added information, these must also be protected somehow from being forged or modified. However, only source authentication does not solve this problem, because the information are added or modified in intermediate nodes. This motivates us to devise a hop-by-hop authentication mechanism in our proposal. An earlier version of this work can be found in [4].

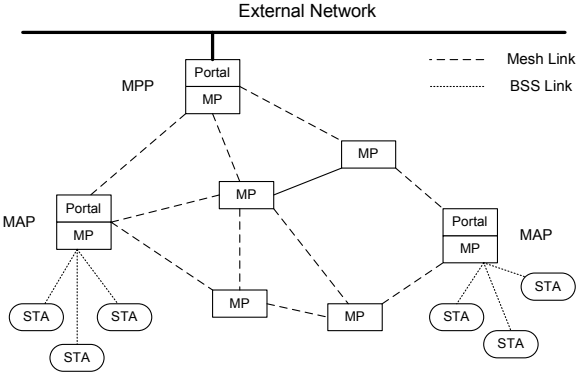


Fig. 1. Network architecture of IEEE 802.11s WMN

The contributions of this paper are as follows. We identify the security vulnerabilities of HWMP. Particularly we show that flooding, route disruption, route diversion, routing loop formation through spoofing are the major threats posed by an adversary

in HWMP. We propose a Secure Hybrid Wireless Mesh Protocol (SHWMP) for IEEE 802.11s based WMN. The fields of the routing information elements are classified as mutable and non-mutable fields. The proposed SHWMP protects the non-mutable part using symmetric key encryption and authenticates mutable information exploiting the concept of Merkle tree [5]. Results from analysis and simulation demonstrate that SHWMP is robust against the identified attacks, provides higher packet delivery ratio, requires no extra communication cost and incurs little path acquisition delay, end-to-end delay, control byte, computation and storage overhead.

The rest of the paper is organized as follows. Following section discusses some of the related works. We give a brief overview of the cryptographic primitive used in this work in Section 3. Section 4 briefly introduces HWMP in 802.11s. Possible attacks are shown in section 5. Section 6 shows the key distribution in 802.11s. We have proposed our idea in section 7 followed by security analysis in section 8. We have evaluated network performance through simulation in section 9. Finally, section 10 concludes our work.

2 Related Works

Research on layer-2 routing is still in its early age. As of now, there is no state-of-the-art solution exists in the literature for securing layer-2 routing. In [6], the authors have just summarized the proposed routing from IEEE 802.11s draft D0.01 [7]. However, the optional routing protocol RA-OLSR, that was described in [6] is no longer considered as a candidate protocol for IEEE 802.11s routing and is omitted from current draft D.2.02 [2]. In [8], the author has described just an update of layer-2 routing in the current draft. IEEE 802.11s's framework and research challenges are summarized in [3]. A hybrid centralized routing protocol is presented in [17] that incorporates tree-based routing with a root-driven routing protocol.

Apart from these, there has been some research on securing layer 3 routing. Ariadne in [9] ensures a secure on-demand source routing. Authentication is done using TESLA [10], digital signatures and standard Message Authentication Code (MAC). However, as the route request is not authenticated until it reaches the destination, an adversary can initiate route request flooding attack. A variant of Ariadne named endairA is proposed in [11] with a difference that instead of signing a route request, intermediate nodes sign the route reply. It requires less cryptographic computation, but still vulnerable to malicious route request flooding attack. SAODV [12] is a secure variant of AODV. Operations are similar to AODV, but uses cryptographic extensions to provide authenticity and integrity of routing messages. It uses hash chains in order to prevent manipulation of hop count field. However, an adversary can always increase the hop count. Another secure on-demand distant vector protocol, ARAN (Authenticate Routing for Ad hoc Networks), is presented in [13]. Just like SAODV, ARAN uses public key cryptography to ensure integrity of routing message. However, a major drawback of ARAN is that it requires extensive signature generation and verification during the route request flooding.

In our proposed scheme, we will use the existing keying hierarchy specified in current 802.11s specification. So, there is no extra burden for enforcing external keying mechanism (like PKI, KDC etc.). That is, we are not assuming that a pairwise key

exists between any two nodes in the networks as path security can not be assured in 802.11s. We have used Merkle Tree in our scheme for authenticating mutable fields in the routing information elements. Our secure routing employs symmetric cryptographic primitives only and uses Merkle tree-based hop-by-hop authentication mechanism by exploiting existing key-hierarchy of 802.11s standard.

3 Cryptographic Primitives

In this section, we briefly describe two of the important cryptographic primitives Merkle tree and Authentication Path used in our proposed scheme.

3.1 Merkle Tree

Merkle Tree is a useful technique to build secure authentication and signature schemes from hash functions. A Merkle tree [4][14][15] is a complete binary-tree that has equipped with a function *hash* and an assignment function F such that for any interior node n_{parent} and two child node n_{left} and n_{right}, the function F satisfies: $F(n_{parent}) = hash(F(n_{left}) \,||\, F(n_{right}))$.

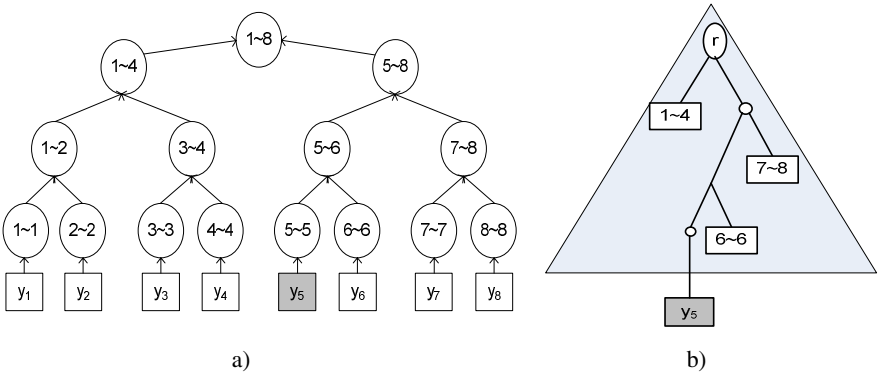


Fig. 2. a) A Merkle tree with 8 leaves and 8 leaf pre-images (y_1, y_2,…,y_8). Each leaf node is a hash of its corresponding pre-image and each internal node is the hash of the concatenation of two child values. y_5 is the value of the pre-image needs to be verified and the root of the tree is known to be public. b) The set of white rectangular nodes makes up the authentication path for the leaf pre-image y_5. Node r represents the root and value of y_5 is verified if both the publicly known root (1~8) (in Fig. 2a) and computed root *r* (in Fig. 2b) are same.

The hash function used is a candidate one-way function such as SHA-1[16]. Fig. 2a depicts a Merkle tree with eight leaf nodes, each being a hash of a leaf pre-image denoted by a box. Then, function F is used to assign the values of each internal node. The value of the root is considered public while all the values associated with a leaf pre-image are known by the owner of the tree.

3.2 Authentication Path

The authentication path of a leaf pre-image consists of values of all the nodes that are siblings of the nodes on the path between the leaf pre-image and the root. Fig. 2b shows the authentication path of the marked leaf pre-image y_5. To verify the value of a leaf pre-image a receiver needs to compute the potential values of its ancestors by iteratively using of the F function shown in previous section. Note that, computing the F function α times, where α denotes the number of leaf nodes in the authentication path, will result in getting the value of the root. A leaf pre-image is authenticated and accepted as correct if and only if the computed root value is equal to the already known root value.

4 Overview of HWMP for IEEE 802.11s WMN

The Hybrid Wireless Mesh Protocol (HWMP) has combined the flavor of reactive and proactive routing strategy by employing both on-demand path selection mode and proactive tree building mode. On-demand mode allows two MPs to communicate using peer-to-peer paths. This mode is mainly used by nodes that experience a changing environment and when there is no root MP configured. On the other hand, proactive tree building mode can be an efficient choice for nodes in a fixed network topology. The mandatory routing metric used in HWMP is the airtime cost metric [2] that measures the link quality (e.g. amount of channel resource consumed by transmitting a frame over a particular link). In HWMP, both on demand and proactive mode can be used simultaneously.

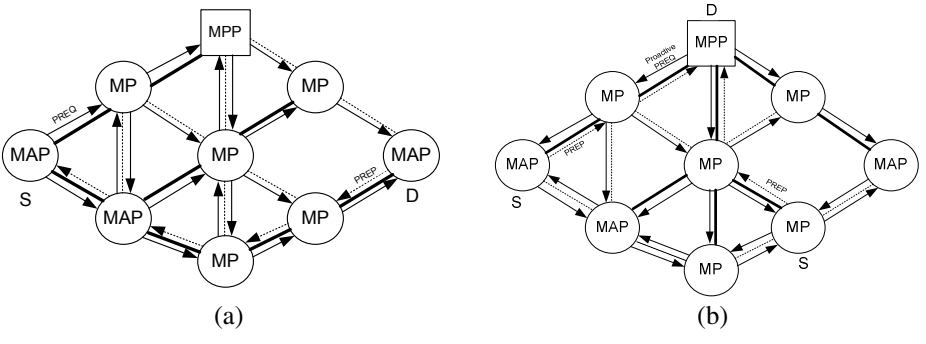


Fig. 3. a) On-demand mode. b) Proactive mode.

4.1 On-Demand Mode

In an On-demand mode a source MP broadcast *path request* (PREQ) message requesting a route to the destination. The PREQ is processed and forwarded by all intermediate MPs and sets up the reverse path from the destination to the source of route discovery. The destination MP or any intermediate MP with a path to the destination may unicast a *path reply* (PREP) to the source MP that creates the forward path

to the destination. As shown in Fig. 3a, source MAP *S* broadcasts PREQ message and intermediate nodes re-broadcasts PREQ after updating its path to the source as shown in solid lines. The destination MAP *D*, unicast a PREP message to the source MAP *S* using the reverse path shown in solid lines.

4.2 Proactive Mode

In *Proactive Tree Building* mode, the MP that configured as a root MP (i.e usually the MPP) can initiate route discovery process in two ways. *Firstly,* it announces its presence by periodically broadcasting a *root announcement* RANN message that propagates metric information across the network. Upon reception of a RANN message, an MP that has to create or refresh a path to the root MP sends a unicast PREQ to the root MP. The root MP then unicast a PREP in response to each PREQ. The unicast PREQ creates the reverse path from the root MP to the originating MP, while the PREP creates the forward path from the MP to the root MP. *Secondly*, the root MP proactively disseminates a proactive PREQ message to all the MPs in the networks with intent to establish a path as shown in Fig. 3b. An MP after receiving a proactive PREQ, creates or updates its path to the root MP by unicasting a proactive PREP, if and only if the PREQ contains a greater sequence number, or the sequence number is the same as the current path and the PREQ offers a better metric than the current path to the root MP. Thus, a routing tree is created with the MPP being the root of the tree.

4.3 Hybrid Mode

HWMP also allows both on-demand and proactive mode to work simultaneously. This hybrid mode is used in situations where a root MP is configured and a mesh point *S* wants to send data to another mesh point *D* but has no path to *D* in its routing table. Instead of initiating on-demand mode, S may send data to the root portal, which in turns delivers the data to *D* informing that both *S* and *D* are in the same mesh. This will trigger an on-demand route discovery between *S* and *D* and subsequent data will be forwarded using the new path that performs better. A more detailed description regarding existing HWMP can be found in [2][6][8].

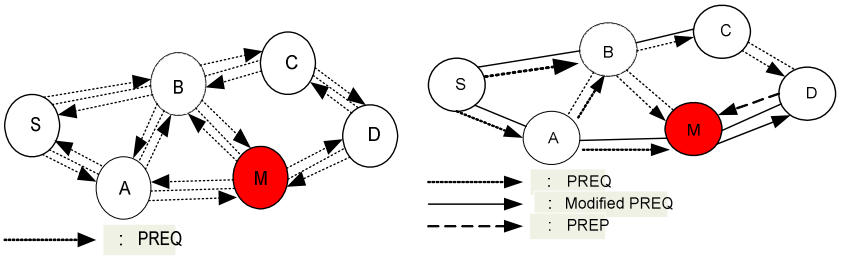
5 Security Vulnerabilities of HWMP

The existing HWMP routing mechanism relies on the fact that all participating mesh entities cooperate with each other without disrupting the operation of the protocol. Without proper protection, the routing mechanism is susceptible to various kind of attacks. In the following, we identify and describe possible attacks that can be launched in the existing HWMP routing mechanism.

5.1 Flooding

The simplest of attacks that a malicious node can launch is by flooding the network with a PREQ messages destined to an address which is not present in the network. As the destination node is not present in the network, every intermediate node will keep

forwarding the PREQ message. As a result, a large number of PREQ message in a short period will consume the network bandwidth and can degrade the overall throughput. As shown in Fig. 4a, the malicious node *M* initiates route discovery with a PREQ for a destination that is not in the network. So that intermediate nodes re-broadcasts PREQ and within a short time the network is flooded with fake requests.



(a) (b)

Fig. 4. a) Flooding. b) Route Disruption.

5.2 Route Disruption

By launching a route disruption attack, an adversary can prevent discovering a route between two legitimate nodes. In other word, if there exist a route between the two victim nodes, but due the malicious behavior of the attacker, the routing protocol can not discover it. In HWMP, route-disruption attacks can easily be launched by a malicious node as shown in Fig. 4b. The malicious node *M* can prevent the discovery of routes between nodes *S* and *D*. *M* can modify the metric field value to zero on the PREQ message it receives from *A* or *B* and re-broadcast. So, after receiving the modified PREQ, *D* will choose *M* as the next hop in the reverse path and unicast PREP to *M*. Now, *M* can prevent the route discovery by dropping the valid PREP message destined for *S*.

5.3 Route Diversion

A malicious node can launch a route diversion attack by modifying mutable fields in the routing information elements such as hop count, sequence number and metric field. A malicious node *M* can divert traffic to itself by advertising a route to a destination with a Destination Sequence Number (DSN) greater than the one it received from the destination. For example, the malicious node *M* in the Fig. 5a receives a PREQ from *A* which was originated from *S* for a route to node *D*. As HWMP allows intermediate PREP, *M* can unicast a PREP to *A* with a higher destination sequence number than the value last advertised by *D*. After getting the PREQ message *D* will also unicast PREP to the source *S*. At some point, *A* will receive both the PREPs and consider the PREP with higher destination sequence number as the valid one and discards the original PREQ as if it was stale. So, *A* will divert all subsequent traffic destined for *D* to the malicious node *M*.

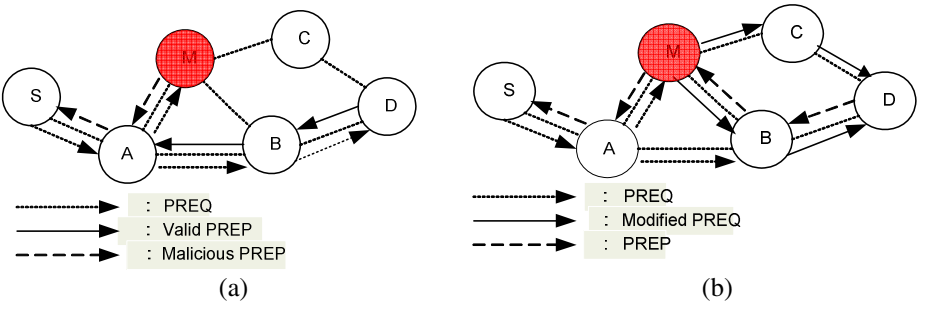


Fig. 5. Route diversion attacks. a) Increasing DSN. b) Decreasing Metric.

Route diversion attack is also possible by modifying the metric field used in the HWMP PREQ messages. Value of the metric field is used to decide the path from a source to a destination, which is a cumulative field contributed by each node in the path. A malicious node can modify the mutable metric field to zero to announce a better path to a destination. As depicted in Fig. 5b, *M* can modify the metric field in the PREQ to zero and re-broadcasts it to the network. So, the reverse path created should go through the malicious node *M*. As a result, all traffics to the destination *D* will be diverted through the attacker.

5.4 Routing Loops

A malicious node can create routing loops in a mesh network by spoofing MAC addresses and modifying the value of the metric field. Consider the following network (Fig. 6) where a path exists between the source *S* and destination *X* that goes through node *B* and *C*. Also, there is a path exists from *A* to *C* through *D*. Assume that a malicious node *M* as shown in Fig. 6a is present in the vicinity where it can listen to the PREQ/PREP messages pass through *A*, *B, C* and *D* during route discovery process. It can create a routing loop among the nodes *A, B, C* and *D* by impersonation combined with the modification of metric field in PREP message.

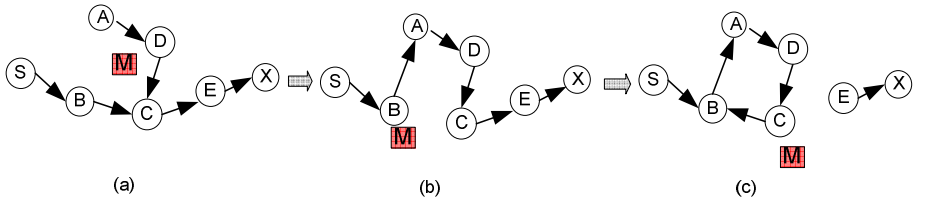


Fig. 6. Formation of Routing loops

First it impersonates node *A*'s MAC address and moved out of the reach of node *A* and closer to node *B*. And then it sends a PREP message to node *B* indicating a better metric value than that of the value received from *C*. So, node *B* now re-establishes its route to *X* that should go through *A* as shown in Fig. 6b. At this point, the malicious node impersonates node *B* and moves closer to node *C* and sends a PREP to node *C*

indicating a better metric than the one received from *E*. So, node *C* will now choose *B* as the next hop for its route to the destination *X* as shown in Fig. 6c. Thus a loop has been formed and the destination *X* is unreachable from all the four nodes.

6 Key Distribution in IEEE 802.11s

IEEE 802.11s ensures link security by using Mesh Security Association (MSA) services. 802.11s extends the security concept of 802.11 by a key hierarchy, inherits functions of 802.11i and uses 802.1X for initial authentication [2]. The operation of MSA relies on mesh key holders, which are functions that are implemented at MPs within the WMN.

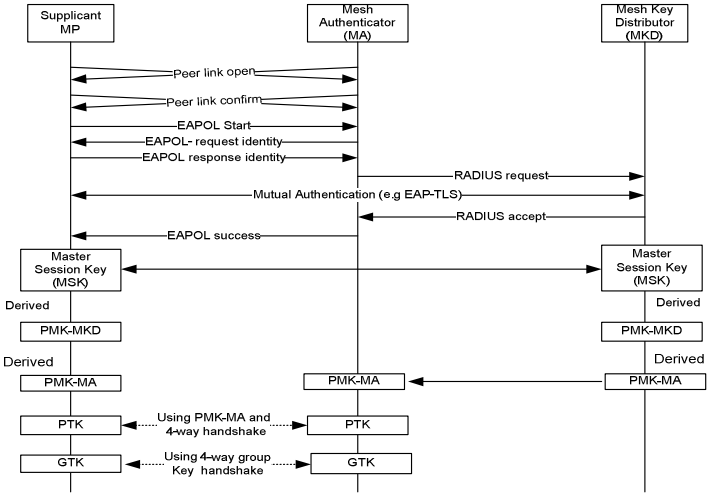


Fig. 7. Key establishment procedure in IEEE 802.11s

Two types of mesh key holders are defined: mesh authenticators (MAs) and mesh key distributors (MKDs). A single MP may implement both MKD and MA key holders, an MA alone and no key holders. Fig. 7 depicts the key establishment procedure between two MPs in IEEE 802.11s.

The first level of link security branch, PMK-MKD is mutually derived by the supplicant MP and MKD, from the Master Session Key (MSK) that is created after the initial authentication phase between supplicant MP and MKD or from a pre-shared key (PSK) between MKD and supplicant MP, if exists. The second level of link security branch PMK-MA is also derived by the supplicant MP and MKD. MKD then delivers the PMK-MA to the MA and thus permits to initiate MSA 4-way handshake which results in deriving a PTK of 512 bits between supplicant MP and MA.

During the MSA 4-way handshake, an MA receives the GTK of the supplicant MP. After the completion of MSA 4-way handshake, a group handshake is used to send the GTK of the MA to the supplicant MP. The GTK is a shared key among all supplicant MPs that are connected to the same mesh authenticator (MA). In our proposed

secure routing algorithm, PTK is used for encryption of unicast messages and GTK is used for encrypting broadcast messages.

7 Secure Hybrid Wireless Mesh Protocol (SHWMP)

The routing protocol proposed in this section is a secure version of Hybrid Wireless Mesh Protocol (SHWMP). HWMP routing information elements have a mutable and a non-mutable part. We exploit these existing mutable and non-mutable fields to design a secure layer-2 routing. More specifically, we (i) use the existing key distribution, (ii) identify the mutable and non-mutable fields, (iii) show that mutable fields can be authenticated in hop-by-hop fashion using the concept of Merkle hash tree, and (iv) use symmetric key encryption to protect non-mutable fields. We describe the proposed protocol in details in the following subsections.

7.1 Use of Keys

All the entities in the mesh infrastructure (MP, MAP and MPP) can act as supplicant MP and Mesh Authenticator (MA). Before initiating a route discovery process, all the MPs authenticate its neighboring MPs, send its GTK and establish PTK through key distribution process described in Section 6. We use this GTK for securing broadcast messages such as PREQ, RANN and PTK is used to secure unicast messages such as PREP, proactive PREQ.

7.2 Identification of Mutable / Non-mutable Fields

The information elements in the HWMP contain fields that can be modified in the intermediate routers which we termed as mutable and those that can not be modified termed as non-mutable fields.

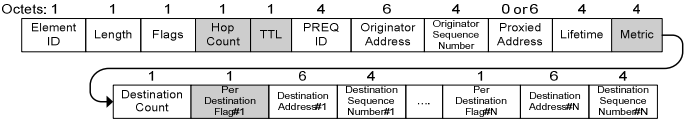


Fig. 8. Format of a PREQ element

Fig. 8 shows the format of a PREQ element where the mutable fields are:

a. Hop count field: Provides information on the number of links in the path, incremented by each intermediate node, but it is not used for routing decision.
b. TTL field: The time-to-leave field defines the scope of the PREQ in number of hops. TTL value is decremented by 1 at each intermediate node.
c. Metric field: HWMP uses an airtime link metric instead of hop count metric as in AODV, to take a decision on path selection. Whenever an intermediate node receives a PREQ that is to be forwarded, it calculates the airtime cost to the current path and adds the value to the existing metric field.

d. Per destination flag: The Destination Only (DO) and Reply and Forward Flag (RF) determine whether the route-reply message (RREP) will be sent by intermediate node or only by destination. If DO flag is not set and RF flag is set, the first intermediate node that has a path to the destination sends PREP and forwards the PREQ by setting the DO flag to avoid all intermediate MPs sending a PREP. In this case, per destination flag field is also a mutable field.

Fig. 9 and Fig. 10 show the format of a PREP and RANN information element. In both the cases, the mutable fields are hop-count, TTL and metric indicated by shadowed boxes.

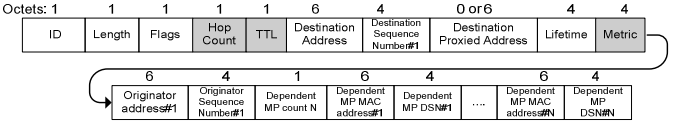


Fig. 9. Format of a PREP element

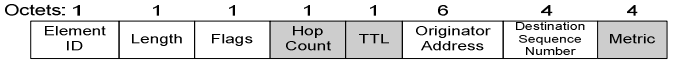


Fig. 10. Format of a RANN element

7.3 Construction of Merkle Tree

Let the mutable fields of routing information elements that need to be authenticated are v_1, v_2, v_3 and v_4. We hash each value v_i into u_i with a one-way hash function such that $u_i = h(v_i)$. Then we assign the hash values to the leaves of the binary tree as shown in Fig. 11.

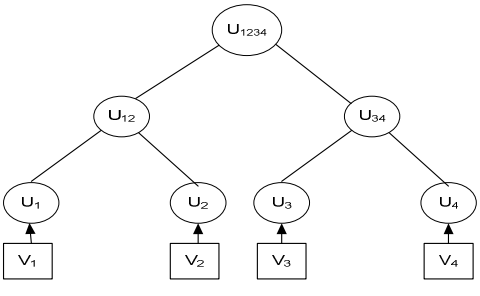


Fig. 11. Construction of Merkle tree

Moreover, to each internal vertex u of this tree, we assign a value that is computed as the hash of the values assigned to the two children of u such as $u_{12} = h(u_1 \| u_2)$. Finally, we found the value of the root and make a message authentication code on

the root by using GTK of the sender or by PTK between sender and receiver for authenticating broadcast and unicast messages, respectively. The sender can reveal a value v_i that needs to be authenticated along with the values assigned to the siblings of the vertices along the path from v_i to root that we denote as authentication path, *authpath(v_i)*. The receiver can hash the values of the authentication path in appropriate order (as described in Section 3.2) to compute the root and create a MAC on the root using the derived key. It then compares the values of the two MACs, If these two values match, then the receiver can be assured that the value v_i is authentic.

7.4 Securing on Demand Mode

Consider a source MP *S* that wants to communicate with a destination MP *X* as shown in Fig. 12.

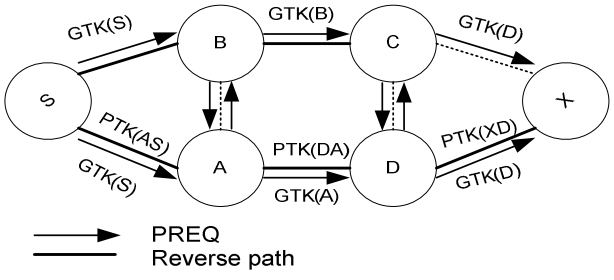


Fig. 12. Secure on-demand path selection

In order to establish a secure route, source node *S*, destination node *X* and set of intermediate nodes (F_1 that includes *A, B* and F_2 that includes *C*, *D*) executes the route discovery process in the following way:

$$S \rightarrow *: MAC_{GTK}root(S),\ \{v_i, authpath(v_i)\},\ \{PREQ-MF\}_{GTK} \quad (1)$$

$$F_1 \rightarrow *: MAC_{GTK}root(F_1),\ \{v_i, authpath(v_i)\},\ \{PREQ-MF\}_{GTK} \quad (2)$$

$$F_2 \rightarrow *: MAC_{GTK}root(F_2),\ \{v_i, authpath(v_i)\},\ \{PREQ-MF\}_{GTK} \quad (3)$$

$$X \rightarrow F_2: MAC^{X,F_2}_{PTK}root(X),\ \{v_i, authpath(v_i)\},\ \{PREP-MF\}^{X,F_2}_{PTK} \quad (4)$$

$$F_2 \rightarrow F_1: MAC^{F_2,F_1}_{PTK}root(F_2),\ \{v_i, authpath(v_i)\},\ \{PREP-MF\}^{F_2,F_1}_{PTK} \quad (5)$$

$$F_1 \rightarrow S: MAC^{F_1,S}_{PTK}root(F_1),\ \{v_i, authpath(v_i)\},\ \{PREP-MF\}^{F_1,S}_{PTK} \quad (6)$$

From the key management of 802.11s, the node *S* is equipped with one GTK that it shares with its neighbors and set of PTKs for communicating with each neighbor individually. Before broadcasting the PREQ, it first creates a Merkle tree with the leaves being the hash of mutable fields of PREQ message. *S* then creates a MAC on the root of the Merkle tree it just created. Then, *S* broadcasts message (1) which includes the MAC of the root created using the GTK, mutable fields v_i *s* that need to be

authenticated along with the values of its authentication path *authpath(v_i)*and encrypted PREQ message excluding the mutable fields.

Any of the neighboring nodes of *S*, after receiving the PREQ, tries to authenticate the mutable fields by hashing the values received in an ordered way, create a MAC on it using the shared GTK and comparing that with the received MAC value of the root. If the two values match, the intermediate MP is ascertain that the values are authentic and came from the same source that created the tree.

Let us consider for example that *B* receives a PREQ from its neighboring node *S* and wants to authenticate the value of the metric field *M*. According to our protocol, *B* and *C* should receive the value *M* along with the values (i.e. U_H and U_{TF}) of the authentication path as shown in Fig. 13. *B* and *C* can now verify the authenticity of *M* by computing $h(h(h(M)|| U_H)|| U_{TF})$ and a MAC on this value using the key GTK. It then compares the received MAC value with the new one, if it found a match, then it can assure that the value *M* is authentic and came from the same entity that has created the tree and computed the MAC on the U_{root}.

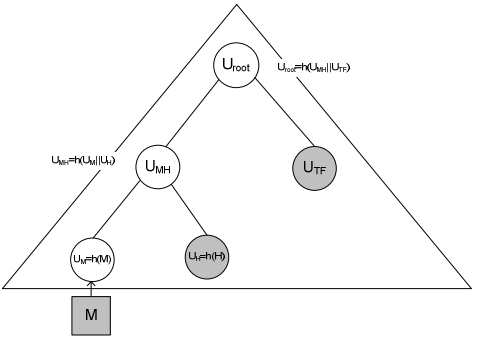


Fig. 13. Circles in grey construct the Authentication path for the metric field M

The intermediate nodes then update the values of the mutable fields like hop count, metric and TTL and create Merkle trees from the modified fields. They also decrypt the non-mutable part of the PREQ message and re-encrypt it with their own broadcast key and re-broadcast (as shown in (2) and (3)) the PREQ message following the same principle. After receiving the PREQ, the destination MP updates the mutable fields; creates its own Merkle Tree and unicasts a PREP message as (4) using the same principle but this time using PTK instead of using GTK. The PREP is propagated as (5) and (6) to the source MP in the reverse path created using PREQ and thus a secure forward path from the source to the destination is established.

7.5 Securing Proactive Mode

In the *Proactive RANN* mode, the RANN message is broadcasted using the group transient key as shown in Equation (7) to (9) to protect the non-mutable fields and authenticate the mutable fields (hop count, TTL and metric) using the Merkle tree approach. As there are only three mutable fields in the RANN message a node requires generating a random number to construct the Merkle tree. After receiving the

RANN message an MP that needs to setup a path to the root MP unicast a PREQ to the root MP as per Equation (10) to (12). On receiving each PREQ the root MP replies with a unicast PREP to that node as described in Equation (13) to (15). *Proactive PREQ* mode can also be secured by transmitting proactive PREQ and PREP in the same way discussed above.

$$R \rightarrow *: \ MAC_{GTK}root(R), \{v_i, authpath\ (v_i)\}, \{RANN\text{-}MF\}_{GTK} \quad (7)$$

$$F_1 \rightarrow *: \ MAC_{GTK}root(F_1), \{v_i, authpath\ (v_i)\}, \{RANN\text{-}MF\}_{GTK} \quad (8)$$

$$F_2 \rightarrow *: \ MAC_{GTK}root(F_2), \{v_i, authpath\ (v_i)\}, \{RANN\text{-}MF\}_{GTK} \quad (9)$$

$$D \rightarrow F_2: MAC^{D,F_2}_{PTK}root(D), \{v_i, authpath\ (v_i)\}, \{PREQ\text{-}MF\}^{D,F_2}_{PTK} \quad (10)$$

$$F_2 \rightarrow F_1: MAC^{F_2,F_1}_{PTK}root(F_2), \{v_i, authpath\ (v_i)\}, \{PREQ\text{-}MF\}^{F_2,F_1}_{PTK} \quad (11)$$

$$F_1 \rightarrow R: MAC^{F_1,R}_{PTK}root(F_1), \{v_i, authpath\ (v_i)\}, \{PREQ\text{-}MF\}^{F_1,R}_{PTK} \quad (12)$$

$$R \rightarrow F_1: MAC^{R,F_1}_{PTK}root(R), \{v_i, authpath\ (v_i)\}, \{PREP\text{-}MF\}^{R,F_1}_{PTK} \quad (13)$$

$$F_1 \rightarrow F_2: MAC^{F_1,F_2}_{PTK}root(F_1), \{v_i, authpath\ (v_i)\}, \{PREP\text{-}MF\}^{F_1,F_2}_{PTK} \quad (14)$$

$$F_2 \rightarrow D: MAC^{F_2,D}_{PTK}root(F_2), \{v_i, authpath\ (v_i)\}, \{PREP\text{-}MF\}^{F_2,D}_{PTK} \quad (15)$$

Notations used in Equation (7) to (15) are as follows: R is considered as the root MP and D is the MP that needs to setup a path to R. F_1 and F_2 are the intermediate nodes in the path. $MAC_k root(X)$, represents the MAC of the Merkle tree's root created by node X using a shared key k. {RANN/PREQ/PREP-MF} represents the routing information elements without the mutable fields. v_i and $authpath(v_i)$ denote the fields that need to be authenticated and the values assigned to the authentication path from v_i to root of the tree, respectively.

7.6 Securing Hybrid Mode

The hybrid mode is the combination of both on-demand and proactive mode. As described in Section 4.3, the hybrid mode is first initiated with the proactive mode where an MP can establish route to the destination via the proactively built tree with the MPP as the root, using our secure proactive routing explained in previous section. After getting the notification from the root MPP that the destination is within the mesh, the source MP initiates the secure on-demand node as described in Section 7.4. The path that performs better will be chosen for subsequent communication. Therefore, individually securing on-demand and proactive mode can ensure security for the hybrid mode too.

8 Analyses

In this section, we will analyze the proposed SHWMP in terms of robustness against the attacks presented in Section 4 and also the overhead required for ensuring secure routing.

8.1 Security Analysis

1) Preventing Flooding: In the proposed SHWMP, a node can participate in the route discovery process only if it has successfully establishes a GTK and PTK through key distribution mechanism of 802.11s. Thus it will not be possible for a malicious node to initiate a route discovery process with a destination address that is not in the network. Again, as the PREQ message is encrypted during transmission, a malicious node can not insert new destination address.
2) Preventing Route Disruption: This type of attack is caused by the malicious behavior of a node through modification of a mutable field and dropping routing information elements. Note that, in our proposed scheme only authenticated nodes can participate in the route discovery phase. Moreover, routing information elements are authenticated and verified per hop. So, it is not possible to launch a route disruption attack in SHWMP.
3) Preventing Route Diversion: The root cause of route diversion attack is the modification of mutable fields in routing messages. These mutable fields are authenticated in each hop. If any malicious node modifies the value of a field in transit, it will be readily detected by the next hop while comparing the new MAC with the received one. It will find a miss-match in comparing the message authentication code (MAC) and the modified packet will be discarded.
4) Avoiding Routing Loops: Formation of routing loops requires gaining information regarding network topology, spoofing and alteration of routing message. As all the routing information is encrypted between nodes, an adversary will be unable to learn network topology by overhearing routing messages. Spoofing will not benefit the adversary as it will require authentication and key establishment to transmit a message with spoofed MAC. Moreover, fabrication of routing messages is detected by integrity check. So, proposed mechanism ensures that routing loops can not be formed.

8.2 Overhead Analysis

1) Computation cost: The computation cost of a sender and receiver are defined by following equations:

$$k \times h + m + e \text{ (sender)} \tag{16}$$

$$\alpha \times h + m + d \text{ (receiver)} \tag{17}$$

Where, k is the number of hash operations required to construct a Merkle tree. Cost of computing a hash function is defined by h. m is the cost involved in computing the MAC of the root, whereas e and d are encryption and decryption cost. To authenticate a particular value, a receiver needs to compute the root by calculating α hash operations, where α defines the number of leaf nodes in the authentication path as described in Section 3.2. Note that, the computation cost to verify a single mutable field requires $O(\log_2 \alpha)$ hash evaluations.
2) Communication Overhead: It is defined by the number of routing messages required to establish a secure path and defined by (18), (19) and (20).

$$(n-1)\times broadcast + h\times unicast \quad (on\text{-}demand) \tag{18}$$

$$n\times broadcast + h\times unicast \quad (proactive\ PREQ) \tag{19}$$

$$n\times broadcast + 2h\times unicast \quad (proactive\ RANN) \tag{20}$$

Where, n is the number of nodes in the network, h is the number of hops in the shortest path. The number of messages required for establishing a path in HWMP is same as our proposed one. So, our protocol does not incur any extra communication overhead.

3) Storage Requirements: A node needs to store the number of fields that need to be authenticated, hashed values of the Merkle tree and the MAC of the root value. So, storage requirement of a node is given by (21)

$$\sum_{i=1}^{n} d_i + (k\times l) + S_M\ , \tag{21}$$

where, d_i is the size of a mutable field, k is the number of hashes in the Merkle tree, l is the size of a hashed value and S_M is the size of the MAC.

9 Performance Evaluation

We use *ns*-2 [18] to simulate our proposed secure routing (SHWMP) protocol and compare that with the existing HWMP. We have simulated 50 static mesh nodes in a 1500 x1500 m^2 area. We use 5 to 10 distinct source-destination pairs that are selected randomly. Traffic source are CBR (constant bit-rate). Each source sends data packets of 512 bytes at the rate of four packets per second during the simulation period of 900 seconds.

In order to compare HWMP with SHWMP, both protocols were run under identical traffic scenario. Both on-demand and proactive mode were simulated. We consider the following performance metrics:

1. **Packet delivery ratio:** Ratio of the number of data packets received at the destinations to the number of data packets generated by the CBR sources. It in turn determines the efficiency of the protocol to discover routes successfully.
2. **Control overhead (in bytes):** Ratio of the control overhead bytes to the delivered data bytes.
3. **Path acquisition delay:** Time required to establish a route from source to destination which actually measures the delay between sending a PREQ/proactive PREQ to a destination and the receipt of corresponding PREP.
4. **End-to-end delay:** Average delay experienced by a data packet from a source to destination. Note that, end-to-end delay includes all the delays including medium access delay, processing delays at intermediate nodes etc.
5. **Average path length:** Average length of a path (in terms of hop count) discovered by the routing protocol. We calculate this by averaging the number of hops taken by each packet to reach the destination.

As shown in Fig. 14, the packet delivery ratio is better in SHWMP for both on-demand and proactive mode than that of HWMP. We assume that 10% misbehaving nodes are present in the network. Since the misbehaving nodes participates in the route discovery process, in HWMP sometimes packets are intentionally dropped by the misbehaving nodes. But, in the proposed protocol, misbehaving nodes can not participate in the route discovery process and thus always achieve a higher packet delivery ratio.

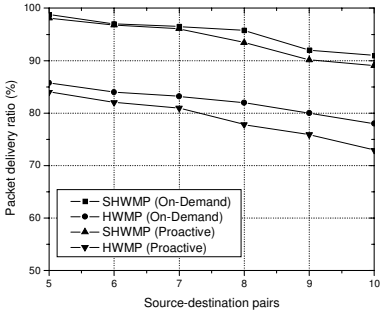


Fig. 14. Packet delivery ratio

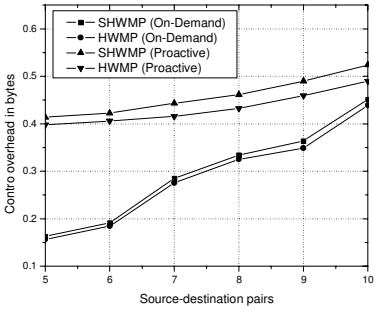


Fig. 15. Control overhead

Fig. 15 shows that the control overhead in bytes for the two protocols are almost identical both in the case of on-demand mode and proactive tree building mode. However, SHWMP has a little more overhead as it needs to include MAC values along with the routing messages that increases the size of a control message. However, the number of control packets transmitted by the two protocols is roughly equivalent.

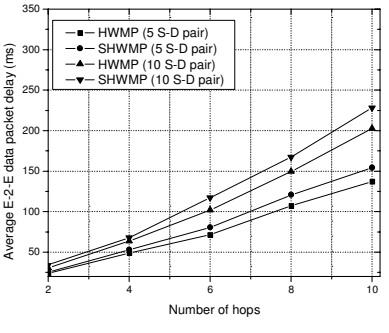


Fig. 16. End-to-end delay for data

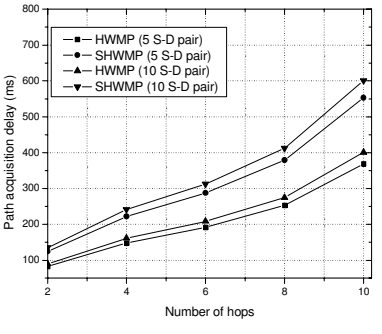
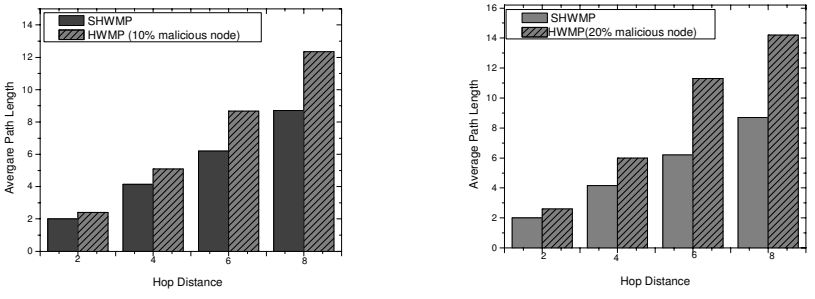


Fig. 17. Path acquisition delay

Fig. 16 depicts that the average end-to-end delay of data packets for both protocols are almost equal. We run the simulation using 5 and 10 source-destination pairs, and as the traffic load increases, end-to-end delay also increases. It is also evident that the

effect of route acquisition delay on average end-to-end delay is not significant. Average route acquisition delay for the proposed SHWMP scheme is much higher than that of the HWMP mechanism as shown in Fig. 17. Because, in addition to normal routing operation of HWMP, the proposed SHWMP scheme requires computing hash and MAC values to verify the authenticity of a received packet, which require extra processing delay.



(a) (b)

Fig. 18. Average path length, a) 10% malicious node. b) 20% malicious node.

In Fig. 18a and Fig. 18b, it is shown that the average path length is less with the proposed SHWMP protocol compared to HWMP protocol in the presence of 10% and 20% malicious nodes, respectively. Since, malicious nodes, when, participate in the route discovery process, can increase the value of the metric field while observing a small hop-count values in the PREQ message. This in turn restricts a node to select a shorter path in the HWMP protocol. With secure HWMP, a malicious node cannot participate in the route discovery process until it is authenticated by the neighbour node(s). We chose source-destination pairs those are 2, 4, 6 and 8 hops away. We derive the hop distance between each source-destination pair using Euclidean geometry. However, in simulation, the path from source to destination is created using HWMP and SHWMP routing protocol. The result shows that average path-length increases for HWMP as the distance between source-destination pairs increases. Since, more malicious nodes can participate in the route discovery process and cause increase in average path length. On the other hand, in case of SHWMP average path length is close to the hop distance measured using Euclidean geometry.

10 Conclusion

The goal of this paper is to develop a secure routing mechanism for wireless mesh networks. We have identified that several security attacks can be launched by the adversary with the existing HWMP routing protocol. Then, we have designed SHWMP, a secure extension of layer-2 routing protocol for IEEE 802.11s. Through analysis, it is shown that all the identified attacks can be defended with the proposed

SHWMP protocol. Furthermore, through extensive simulations we have compared different performance metrics for the existing HWMP and the proposed SHWMP. Our simulation results show that the proposed protocol outperforms existing one in terms of packet delivery ratio and average path length. Due to cryptographic extension (which is a must to provide secure routing), our protocol incurs little overhead in terms of control overhead in bytes and path acquisition delay. In future work, we intend to address the attack mounted by colluding adversarial nodes against the secure routing protocol.

References

1. Akyildiz, I.F., Wang, X., Wang, W.: Wireless Mesh Networks: a Survey. Computer Networks 47(4) (2005)
2. IEEE 802.11s Task Group, Draft Amendment to Standard for Information Technology Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications: Amendment: ESS Mesh Networking, IEEE P802.11s/D2.02 (September 2008)
3. Wang, X., Lim, A.O.: IEEE 802.11s Wireless Mesh Networks: Framework and Challenges. In: AdHoc Networks, pp. 1–15 (2007), doi:10.1016/j.adhoc.2007.09.003
4. Islam, M.S., Yoon, Y.J., Hamid, M.A., Hong, C.S.: A Secure Hybrid Wireless Mesh Protocol for 802.11s Mesh Network. In: Gervasi, O., Murgante, B., Laganà, A., Taniar, D., Mun, Y., Gavrilova, M.L. (eds.) ICCSA 2008, Part I. LNCS, vol. 5072, pp. 972–985. Springer, Heidelberg (2008)
5. Merkle, R.C.: A Certified Digital Signature (subtitle: That Antique Paper from 1979). In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 218–238. Springer, Heidelberg (1990)
6. Bahr, M.: Proposed Routing for IEEE 802.11s WLAN Mesh Networks. In: 2nd Annual International Wireless Internet Conference (WICON), Boston, MA, USA (2006)
7. IEEE P802.11s^{TM}/D0.01, Draft amendment to standard IEEE 802.11^{TM}: ESS Mesh Networking. IEEE (March 2006)
8. Bahr, M.: Update on the Hybrid Wireless Mesh protocol of 802.11s. In: Proc. of IEEE International Conference on Mobile Adhoc and Sensor Systems, 2007. MASS, pp. 1–6 (2007)
9. Hu, Y.-C., Perrig, A., Johnson, D.B.: Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In: Proc. MobiCom 2002, Atlanta, GA (2002)
10. Perrig, A., Canetti, R., Tygar, J.D., Song, D.: Efficient Authentication and Signing of Multicast Streams over Lossy Channels. In: Proc. of IEEE Symposium on Security and Privacy, 2000, pp. 56–73 (2002)
11. Gergely, A., Buttyan, L., Vajda, I.: Provably Secure On-demand Routing in Mobile Ad Hoc Networks. IEEE transactions on Mobile Computing 5(11), 1533–1546 (2006)
12. Zapata, M.G., Asokan, N.: Securing Adhoc Routing Protocols. In: Proc. of ACM Workshop of Wireless Security(Wise), pp. 1–10 (2002)
13. Sangiri, K., Dahil, B.: A Secure Routing Protocol for Ad Hoc Networks. In: Proc. of 10th IEEE International Conference on Network Protocols, ICNP 2002 (2002)
14. Szydlo, M.: Merkle Tree Traversal in Log Space and Time. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 541–554. Springer, Heidelberg (2004)

15. Jakobsson, M., Leighton, T., Micali, S., Szydlo, M.: Fractal Merkle Tree Representation and Traversal. In: Joye, M. (ed.) CT-RSA 2003. LNCS, vol. 2612, pp. 314–326. Springer, Heidelberg (2003)
16. FIPS PUB 180-1, Secure Hash Standard, SHA-1, http://www.itl.nist.gov/fipspubs/fip180-1.htm
17. Lim, A.O., Wang, X., Kado, Y., Zhang, B.: A Hybrid Centralized Routing Protocol for 802.11s WMNs. Journal of Mobile Networks and Applications (2008)
18. The Network Simulator – ns-2, http://www.isi.edu/nsnam/ns/index.html