Secure Routing for Wireless Mesh Networks

Celia Li[1], Zhuang Wang[2], and Cungang Yang[2]
(Corresponding author: Cungang Yang)

Department of Computer Science and Engineering, York University[1]
M3J 1P3, Toronto, Canada (Email: cungang@ee.ryerson.ca)
Department of Electrical and Computer Engineering, Ryerson University[2]

(Received Apr. 4, 2010; revised and accepted May 11 & 25, 2010)

Abstract

This paper describes a Security Enhanced AODV routing protocol for wireless mesh networks (SEAODV). SEAODV employs Blom's key pre-distribution scheme to compute the pairwise transient key (PTK) through the flooding of enhanced HELLO message and subsequently uses the established PTK to distribute the group transient key (GTK). PTK and GTK are used for authenticating unicast and broadcast routing messages respectively. In wireless mesh networks, a unique PTK is shared by each pair of nodes, while GTK is shared secretly between the node and all its one-hop neighbors. A message authentication code (MAC) is attached as the extension to the original AODV routing message to guarantee the message's authenticity and integrity in a hop-by-hop fashion. Security analysis and performance evaluation show that SEAODV is more effective in preventing identified routing attacks and outperforms ARAN and SAODV in terms of computation cost and route acquisition latency.

Keywords: AODV, hop-by-hop authentication, MAC, wireless mesh networks

1 Introduction

Wireless Mesh Networks (WMN) [1, 6, 21] are a wireless multi-hop technology that has much in common with the mobile ad hoc networks (MANETs). WMN can be considered as a superset of the ad hoc networks. An infrastructure WMN is one that is entirely comprised of mesh routers, whereas a client WMN is a network purely made up of mobile client devices.

In most cases, a typical WMN is a hybrid network where both mesh routers and mesh clients exist simultaneously. Although a profusion of routing protocols has been proposed for other wireless networks such as Ad hoc network, the unique characteristics of WMN indicate that it demands its own solution. Hybrid routing seems to be one of the promising answers to the question of what is the trend in WMN's routing.

In hybrid routing, proactive routing is specifically used for traffics flow to the mesh portal, where a connection with Internet was created. While for intra-mesh traffic, traffics will not bypass the mesh portal and on-demand routing is preferred. In hybrid routing of WMN, such as HWMP [2, 3, 13], the security issues are not addressed. Our proposed scheme can be seen as a secure version of the on-demand part in HWMP and used to securely discover a route between any pair of mesh routers in the network.

In this paper, we present SEAODV, a security enhanced version of AODV. We utilize PTK and GTK keys to protect the unicast and broadcast routing messages respectively to ensure that the route discovery process between any two nodes in WMN is secure. We apply BLOM's key pre-distribution scheme in conjunction with the enhanced HELLO message to establish the PTK and use the established PTK to distribute GTK to the node's one-hop neighbors throughout the entire network.

We also identify various attacking scenarios specifically happened in AODV and present security analysis to prove that our proposed SEAODV is able to effectively defend against most of those identified attacks. Our Scheme is lightweight and computationally efficient due to the symmetric cryptographic operations (e.g., MAC). In addition, SEAODV supports a hop-by-hop authentication as well.

The rest of the paper is organized as follows. Section 2 discusses related work. In Section 3, we provide background knowledge of Blom's key pre-distribution scheme. Section 4 gives a brief overview of standard AODV and two well known secure routing protocols in MANETs, named SAODV and ARAN. Details of our SEAODV protocol will be presented in Section 5. Section 6 identifies various potential attacking scenarios in AODV and presents the security analysis. The performance evaluation is explained in Section 7. Finally, Section 8 concludes the paper.

2 Related Work

So far, there has been tremendous research on layer 3 secure routing for wireless networks. Each secure routing protocol is tailored to a specific type of wireless networks, such as ad hoc networks or wireless sensor networks. All of them have similar properties, and thus some routing protocols of ad hoc networks can be applied to wireless mesh networks. However, they may not provide specific security features (such as hop-by-hop authentication) for mesh networks and still vulnerable to various types of routing attacks such as flooding, route re-direction, spoofing etc.

Depending on when routes are required to be calculated, routing protocols can be divided into two categories: proactive routing and on-demand routing. In proactive routing, every node maintains one or more tables containing routing information to every other node in the network. All nodes in the network update their tables to maintain a consistent and up-to-date view of the network whenever the network topology changes or a node's routing table is updated. Example of proactive routing is Link Quality Source Routing (LQSR) [7], which giving minimum burden on relaying nodes as the source node calculates the route for a flow and stores the complete path of the flow in its packet headers. Intermediates nodes only need to forward the packets according to the path stored in the packet headers. Ad hoc On-demand Distance Vector routing (AODV) [18], Dynamic Source Routing (DSR) [15] and Load Balancing Aware Routing (LBAR) [10] are on-demand routing protocols in which a route is created only when the source node needs to send packets to the destination.

SAODV [22] is a secure variant of AODV. SAODV uses hash chain and digital signature to secure the mutable field (hop count) and non-mutable field (the rest of the routing message except hop count field) and successfully defends against impersonation attacks, modification of hop count and sequence number attacks. However, it does not provide hop-by-hop authentication. Intermediate nodes on the path cannot verify the authenticity of the messages from their predecessors. Although SAODV can prevent the hop count field in AODV routing message from decreasing, the adversary still can increase hop count and hence affect the routing decision of which node is going to be selected during route discovery process and increase the likelihood of nodes not being chosen on the established route. SAODV secures the routing messages; it does not guarantee either authentication or integrity of the subsequent data packets after route has been established between source and destination.

ARAN (Authenticate Routing for Ad hoc Networks) [20] adopts digital signature to ensure hop-by-hop authentication and routing message integrity; however, it suffers from experiencing significant computation overheads that dramatically affect its routing performance, such as route acquisition latency. Each node in the network has to verify signatures whenever it receives the signed messages. The node has to remove the certificate and signature of its predecessor, use its own private key to sign the message originally broadcast by the source and appends its own certificate before rebroadcasting to its one-hop neighbors.

Ariande is a secure on demand source routing protocol [11]. TESLA, digital signatures and standard MAC are involved to ensure the source authentication [19], however, the route request message is not authenticated before it reaches the destination. The adversary may take this advantage to initiate route request flooding attack. EndairA [9], a variant of Ariande, experiences less cryptographic computation but is still vulnerable to malicious route request flooding attack.

In HWMP [2, 3, 13], the on-demand mode allows two MPs (Mesh Point) to communicate using peer-to-peer paths. This mode is primarily used if nodes experience a changing environment and no root MP is configured. While the proactive tree building mode is an efficient choice for nodes in a fixed network topology, HWMP does not address the security issues and suffers from attacks which will be described in Section 6.

LHAP [23] is a lightweight transparent authentication protocol for ad hoc networks. It uses TESLA to maintain the trust relationship among nodes, which is not realistic due to TESLA's delayed key disclosure period. In LHAP, simply attaching the TRAFFIC key right after the raw message is not secure since the traffic key has no relationship with the message being transmitted.

3 Blom'S Key Pre-distribution Scheme

Blom's key pre-distribution scheme in Figure 1 is applied for implementing key exchange process [4, 8]. The following briefly explains how the Blom's h secure key predistribution scheme works.

During the pre-deployment phase, $a(h+1) \times N$ matrix G over a finite field $GF(q)$ is constructed, where N is the network size and M is public information that is known by any node of a network. A random $(h+1) \times (h+1)$ symmetric matrix D over $GF(q)$ and a matrix $A = (D \cdot M)T$ are then created, where $(D \cdot M)^T$ is the transpose of $D \cdot M$ and A is a $N \times (h+1)$ matrix. Matrix D cannot be disclosed to any node in the network and must be kept secret. Each node k of the network needs to store the k^{th} row of matrix A and the k^{th} column of matrix M where $k = 1, \ldots, N$. When nodes i and j need to find the pairwise key between them, they first exchange their columns of M, and then they can compute K_{ij} and K_{ji} respectively with their private rows of A because $K_{ij} = K_{ji}$. Since M is public information, its columns can be transmitted in plaintext.

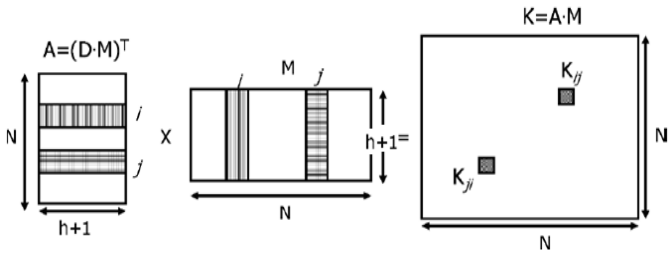

Figure 1: Blom's scheme

4 Overview of AODV, SAODV and ARAN

SEAODV is based on the standard AODV. ARAN (Authenticated Routing for Ad hoc Networks) and SAODV (Secure Ad hoc On-demand Distance Vector) are two well known secure routing protocols for Ad hoc networks. Both of them are also based on AODV, although ARAN presents different message format in route discovery process.

4.1 Standard AODV

The Ad Hoc On-Demand Distance Vector (AODV) algorithm is an on-demand reactive routing protocol, which means it seeks for routes only when required. AODV makes use of sequence numbers to avoid forming routing loops. The standard operations of AODV are described as follows.

When a source node wants to communicate with a destination node, it broadcasts a RREQ (Route Request) message to all its one-hop neighbors if it cannot find an active route in its routing table. Upon receiving the broadcasted messages, its neighbors check their routing tables and see whether there exists a route to the destination node. If not, they will forward the RREQ message to their neighbors until the RREQ reaches the designated destination or an intermediate node know the route to the destination. In that case, the destination node or the intermediate node updates its reverse route to the source node from which they received the RREQ, generates a RREP (Route Reply) message and unicasts it back to the source node. When the source node or the intermediate node receives a RREP message, they update their forward route to the destination, use the neighbors from which they receive their RREP and update their routing tables accordingly.

To maintain connectivity information, each node detects possible link breakage to its immediate neighbors with periodical HELLO message. In the case a broken link is detected for the next hop of an active route, a RERR (Route Error message) is sent to all its neighbors who are using that specific route.

4.2 SAODV

SAODV is a secure version of AODV. SAODV uses hash chains to secure hop count field and digital signatures to protect the non-mutable fields in both RREQ and RREP messages. The following explains how SAODV works in detail.

During route discovery process, a random *seed* number is generated by the source node and the maximum hop count (MHC) value is set to be the Time-To-Live (TTL) value from the IP header. Source node then computes the hash value $Hash = h(seed)$ and TOP_{Hash} as $h^{MHC}(seed)$. Upon receiving an RREQ message, an intermediate node verifies whether TOP_{Hash} equals to $h^{MHC-Hop_{count}}(Hash)$. If the two values are completely identical, the hop count is presumed to be unaltered. Moreover, before rebroadcasting the RREQ to its one-hop neighbors, the intermediate node increases the hop count by one and computes the new hash value $h(Hash)$. Except the hop count field, all other fields in the RREQ are considered to be non-mutable and secured by using digital signature. When RREQ meets the destination, destination generates a RREP in the same way towards to the source.

4.3 ARAN

The entire routing message in ARAN [20] is signed by the source and the intermediate node who forwards the RREQ or RREP message. When a source node A needs to seek a route, it generates and signs a RREQ. Upon receiving the message, node B, a one-hop neighbor of A, uses the pubic key of the certificate server to extract A's public key and applies it to verify the received signed message. If the signature is verified, the received message is considered to be authentic and unaltered. Node B then updates the routing table accordingly, signs it and appends its own certificate to the message before rebroadcasting it to its one-hop neighbors. Otherwise, the received message is considered to be unauthentic and will be discarded. Assuming node C is one of B's one-hop neighbors and C receives the RREQ message from B. Similarly, C validates the corresponding signature with the given certificate. Upon successfully verifying the signature, C then removes B's certificate and signature, updates routing table, signs the entire message originally broadcast by node A, appends its own certificate and rebroadcast it. During the route discovery, each intermediate node repeats these steps. When the destination node receives the RREQ, it creates an RREP and unicasts it back along the reverse path to the source in the same way.

5 Security Enhanced AODV (SEAODV)

This section presents our proposed security enhanced AODV routing protocol in detail. SEAODV employs

Blom's key pre-distribution scheme and enhanced HELLO message to compute the pairwise transient key (PTK), which subsequently being used to distribute the group transient key (GTK). PTK and GTK are used to secure the unicast and broadcast routing messages respectively.

5.1 Use of Keys

SEAODV is built on the existing AODV. Choosing AODV as our protocol's foot stone is due to its simplicity, maturity, popularity and availability in the research over the past few years. SEAODV requires each node in the network to maintain two key hierarchies. One is the broadcast key hierarchy, which includes all the broadcast keys from its active one hop neighbors. The other hierarchy is called unicast hierarchy and it stores all the secret pairwise keys that this node shares with its one hop neighbors. Every node uses keys in its broadcast hierarchy to authenticate the incoming broadcast routing messages (e.g., RREQ) from its one hop neighbors and applies secret pairwise keys in the unicast hierarchy to verify the incoming unicast messages such as RREP.

5.2 Enhanced HELLO Message (HELLO RREQ, HELLO RREP)

In AODV, HELLO message [5] is broadcast to its one-hop neighbors in order to maintain updated local connections. In SEAODV, we define two Enhanced HELLO messages using the idea inspired from [14]. Each node embeds its column of the public G matrix into its enhanced HELLO RREQ message. Since each column of the public known matrix G can be regenerated by applying the seed (a primitive element of $GF(q)$) from each node, every node only needs to store the seed in order to exchange the public information of matrix G. To guarantee bi-directional links, the neighboring nodes who receive HELLO RREQ message will reply with an enhanced HELLO RREP message.

5.3 Exchange Public $Seed\_G$ and GTK by Using Enhanced HELLO Message

During the key pre-distribution phase, every legitimate node in the wireless mesh network knows and stores its public known $Seed\_G$ (seed of the column of public G matrix) and the corresponding private row of the generated A matrix. The entire exchange process can be depicted in the following three major steps.

Step 1: Exchange of $Seed\_G$ of public G matrix. When node A wants to exchange its $Seed\_G$ with its one-hop neighbors, it looks for its public $Seed\_G$ from its key pool, and broadcasts the enhanced HELLO RREQ message. Node B will do the same as A if B is A's one-hop neighbor. Upon finishing step 1, every node in the network possesses the public $Seed\_G$ of all its one-hop neighbors.

Step 2: Derivation of PTK (Pairwise Transient Key). Each node uses the $Seed\_G$ it received from its neighbors and the node's corresponding private row of matrix A to compute PTK. Upon finishing Step 2, every node has stored the public known $Seed\_G$ of its neighbors and derived the PTK it shares with each of its one-hop neighbors.

Step 3: Exchange of GTK (Group Transient Key) through HELLO RREP. Upon receiving HELLO RREQ from node A, node B encrypts GTK_B with its private PTK_B and unicasts the corresponding HELLO RREP message back to A. The encrypted GTK_B is also attached in the unicast HELLO RREP message. Once A receives HELLO RREP from B, A applies its private PTK_A to decrypt the GTK_B and stores it in the database. The same process applies to node B as well. Eventually, every node possesses the GTK keys from all its one-hop neighbors and the group of secret pairwise PTK keys that it share with each of its one-hop neighbor.

5.4 Securing Route Discovery

In order to implement a hop-by-hop authentication, each node must verify the incoming message from its one-hop neighbors before re-broadcasting or unicasting the message. The trust relationship between each pair of nodes relies on their shared GTK and PTK keys, which have been obtained during the key exchange process. Route discovery process of SEAODV is similar to that of standard AODV, but a MAC extension is appended to the end of the AODV routing message. The new format of the RREQ in SEAODV is given in Figure 2.

The MAC is computed over message M using the key GTK of the node who needs to broadcast a RREQ to its one-hop neighbors. Message M refers to all the elements before the MAC field in the RREQ message. When a node wants to discover a route to a designated destination, it broadcasts the modified RREQ message to its neighbors. The receiving node computes the corresponding MAC value of the entire received message if the node possesses the GTK key of the sender. The receiving node then compares the computed MAC with the one it received. If there is a match, the received RREQ is considered to be authentic and unaltered. The receiving node will then update the mutable field (hop-count in RREQ) and its routing table, and subsequently set up the reverse path back to the source by recording the neighbor from which it received the RREQ. Finally, the node computes a MAC of the updated RREQ with its GTK key and attaches the MAC value to the end of the RREQ before the message is re-broadcast to its neighbors.

5.5 Securing Route Setup

Eventually, the RREQ message reach the destination or an intermediate node which has a fresh route to the destination. The destination node or an intermediate node

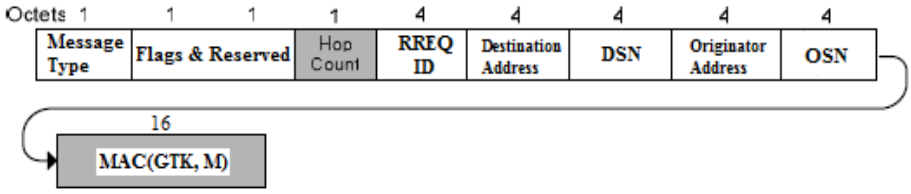

Figure 2: Modified RREQ

can generate a modified RREP and unicast it back to the next hop from which it received the RREQ towards to the originator of the RREQ (the source). Since RREP message is authenticated at each hop due to the use of PTK keys, adversary has no opportunity to re-direct the traffic.

Before unicasting the modified RREP back to the originator of the RREQ, the node first needs to check its routing table to identify the next hop from which it received the broadcast RREQ; then the node applies the PTK key it privately shares with the identified next hop to compute the $MAC(PTK, M)$ and affixes this MAC to the end of RREP as shown in Figure 3.

Upon receiving RREP from node B, A checks whether PTK_BA is in its group of PTK. If the answer is yes, node A computes $MAC'(PTK\_AB, M)$ and compares it against the $MAC(PTK\_BA, M)$ it received from node B. If $MAC'(PTK\_AB, M)$ matches $MAC(PTK\_BA, M)$, the received RREP is considered authentic. Node A then updates the hop-count field in the RREP and its own routing table, sets up the forwarding path towards to the destination. A also searches the appropriate PTK key that it shares with its next hop to which the new RREP is going to be forwarded to the source. Node A then uses the PTK key to construct the new MAC and attach it at the end of the new RREP message. Otherwise, the received RREP is deemed to be unauthentic and is dropped.

5.6 Securing Route Maintenance

In SEAODV, a node generates a RERR message if it receives data packet destined to another node for which it does not have an active route in its routing table or the node detects a broken link for the next hop of an active route or a node receives a RERR from a neighbor for one or more active routes. The format of a modified RERR message is shown in Figure 4 where the MAC field in the modified RERR is created by applying the node's GTK on the entire RERR message. Upon receiving the broadcast RERR message from node B, A first checks whether it has the GTK_B. If the answer is yes, A then computes $MAC'(GTK\_B, M')$ and compares it with the received MAC value. If the two MACs equal, node A searches its routing table and try to identify the affected routes (a new group of unreachable destinations) that use node B as its next-hop based on the unreachable destination list received from B. If no routes in node A's routing table is affected, A simply drops the RERR and starts listening to the channel again. A also discards the RERR if it fails to find the GTK_B or the $MAC'(GTK\_B, M')$ is not equals to the one received from B.

6 Attacking Scenarios in AODV and Security Analysis

This section presents possible attacks launched in AODV during a route discovery process and compare the security analysis results of our SEAODV with ARAN, SAODV and LHAP.

6.1 Attacking Scenarios in AODV

RREQ Flooding Attack

Flooding is one of the simplest attacks that a malicious node could have launched. An attacker tries to flood the entire network with RREQ message destines to a known or an unknown address. As a consequence, this causes a mass of unnecessary broadcasts and force the neighbors to process these flooding route requests, the aim is to consume the energy of the nodes in the network and the network bandwidth. Therefore, the whole network communication may be breakdown and the throughput is dropped dramatically.

RREP Routing Loop Attack

A routing loop is a path that goes through the same nodes over and over again. As a consequence, this kind of attack will deplete the resources of every node in the loop and cause the isolation of the destination and few packets can eventually reach the destination. Both RREQ flooding and RREP routing loop attacks are also called Denial-of-Service (DoS) attacks. DoS attacks do not intend to destroy the data message, but try to consume and compromise the scarce resource that available to nodes in the network. They can even disrupt the usability of the network.

Route Re-direction Attack

Figure 5 [12] explains two cases where a route re-direction attack could have been launched by malicious node M. In case A, malicious node M tries to initiate this attack by modifying the mutable fields in the routing message.

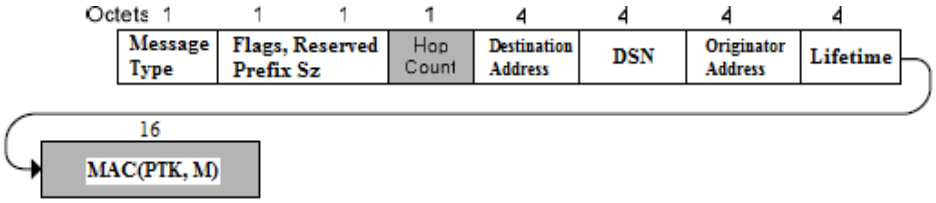

Figure 3: Modified RREP compute

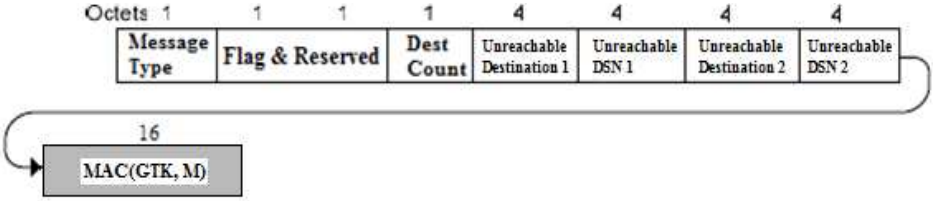

Figure 4: Modified RERR

These mutable fields include hop count, sequence numbers and other metric related fields. A malicious node M could divert the traffic through itself by advertising a route to the destination with a larger destination sequence number (DSN) than the one it received from the destination. In case B, route re-direction attack maybe launched by modifying the metric field in the AODV routing message, which is the hop count field in this case. A malicious node M simply modifies the hop count field to zero in order to claim that it has a better path to the destination. Both Case A and Case B belong to the category of modification attacks in AODV.

Formation of Routing Loops Attack

A malicious node may use other legitimate nodes' IP address (Impersonation Attack) and modify the value of the metric field to achieve the goal of creating routing loop. Figure 6 shows how this type of attack could be launched through Step a to Step c.

1) Malicious node M impersonates node A's IP address and moves closer to node B where node A cannot hear from M.

2) M sends a falsified RREP to node B indicating a better metric (in terms of hop-count in AODV) than that of the value received from node C. As a result, now node B will re-direct all the traffics destined to X towards to node A.

3) Again similar actions have been taken by malicious node M. Now M gets closer to node C and sends a RREP to C on behalf of node B indicating a better metric than the one node C receives from node E. Therefore, now C will choose B as its next hop for the route to destination X, all traffics destined to X will be routed to B. As a consequence, a routing loop now has been created and destination X is no longer reachable to node A, B, C and D.

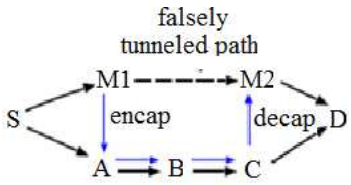

Figure 7: Tunnelling attack

Fabrication Attack

Adversary may fabricate the routing messages to disorder the routing decisions. For instance, a malicious node could simply fabricate a route error message in AODV protocol, this will put all the upstream nodes in the network into a very embarrassment situation since all of them now believe that a certain number of destination are unable to reach. This may result in these upstream nodes to re-initiate a route request to those unreachable destinations so as to discover and build another possible route to them. This brings the energy consuming issue on the table again and significantly degrades the performance of the routing protocol.

Tunnelling Attack

In Ad hoc network, a node can be located adjacent to other nodes. A tunnelling attack is referred to two or more malicious nodes in the network may collude and cooperate with each other to encapsulate and exchange routing messages between them by either using the existing data routes or potentially high power transceiver [20]. The purpose is to prevent the honest intermediate nodes from correctly incrementing the metric field that will be

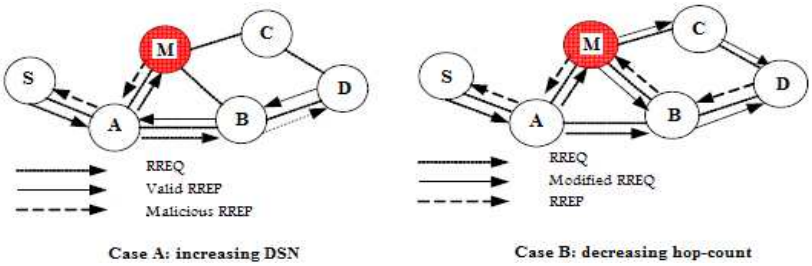

Figure 5: Route re-direction attack

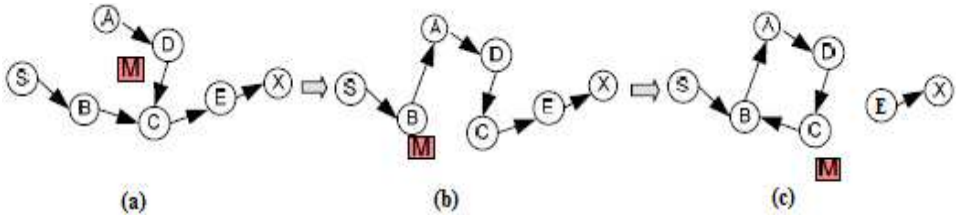

Figure 6: Formation of routing loops attack

used to measure the actual path length. In AODV, tunnelling attack may lead to the misrepresentation of the actual path length; therefore, attackers raise the chances of being included on the final established route between the source and destination and lured subsequent data traffics through them. Figure 7 illustrates such an attack where $M1$ and $M2$ are malicious nodes that are cooperating with each other in order to mis-present the actual path length by tunnelling the route request (RREQ) in AODV. The actual path is represented by the solid line in black color. The blue solid line denotes the tunnel and the dotted line in black color refers to the path that has been falsely claimed by $M1$ and $M2$.

6.2 Security Analysis

We analyze our proposed SEAODV in terms of defending against those attacking scenarios presented in Section 6.1 and compare the security analysis results against other three secure routing protocols: ARAN, SAODV and LHAP.

RREQ Flooding
ARAN suffers badly from continuously verifying digital signatures, while SAODV also incurs massive overhead in signature verification process. Contrarily, LHAP offers better immunity due to its light-weight nature by using one-way hash chain and only authenticates RREQ from its one-hop neighbors. The number of hash operations required to verify the authenticity of a message is from single hash operation up to maximum number of tolerance in terms of packet loss. SEAODV only authenticates RREQ from nodes that are in the list of its active one-hop neighbors. Hash operations are required in SEAODV and re-creation of MAC is simple, fast and one time only.

RREP Routing Loop
In ARAN, every transmission of signed routing message makes impersonation and modification of sequence numbers impossible. SAODV does not support hop-by-hop authentication. It is based on source-destination authentication and any intermediate nodes could be impersonated by any chance during the flying of RREP. LHAP uses one-way hash chain to protect the message by simply appending traffic key right after the raw message. Malicious node can simply block the wireless transmission between two neighboring nodes, modifies the messages, put the corresponding intercepted traffic keys right after the messages and send them back to the wireless channel. SEAODV is a hop-by-hop authentication. GTK and PTK keys are used to secure the broadcast and unicast routing messages respectively. The entire routing message is MACed, therefore, possibilities of impersonation and modification are eliminated.

Route Re-direction
Both ARAN and SEAODV can defeat this type of attack. ARAN employs digital signature to sign every single routing message in a hop-by-hop fashion, while in SEAODV, GTK and PTK keys are used to compute the MAC, which secures all the fields in the entire routing message. SAODV cannot effectively prevent the metric field (in this case hop count) from being increased by malicious nodes. This increases the chances of the route being de-selected from the potential candidate routes, which is another form of route re-direction attack. In LHAP, again malicious nodes can use the exact technique de-

Table 1: Vulnerabilities of various routing protocols

Attack	AODV	ARAN	SAODV	LHAP	SEAODV
RREQ Flooding	Yes	Yes	Yes	Yes	Yes
RREP Routing Loop	Yes	No	Yes	Yes	No
Route Re-direction	Yes	No	Yes	Yes	No
Formation of Routing Loops	Yes	No	No	Yes	No

scribed in the last paragraph to create this type of attack.

Formation of Routing Loops
Two conditions need to be satisfied in order to launch this attack. The malicious node has to impersonate a legitimate node in the network and is able to modify the metric such as hop count to be a better value in terms of less hop count in this case. SAODV is able to prevent the hop-count from decreasing, hence avoiding this attack. ARAN and SEAODV can also defeat this type of attack due to its hop-by-hop authentication. However, in LHAP, as long as the malicious node gets a chance to intercept the effective traffic keys and re-use them in a timely manner, there is a possibility to launch this type of attack.

RERR Fabrication
In ARAN, messages can only be fabricated by nodes with valid certificates and ARAN offers non-repudiation. Nodes keep sending fabricated routing messages might get excluded from the future route computation. While in SAODV, malicious node may simply impersonate nodes other than the one initiates the original RERR and forward the signed RERR to other nodes in the network. By doing do, malicious nodes can not only deplete the energy of the nodes, but also successfully defeat the routing protocol. LHAP also suffers from this type of attack; malicious node could use the captured traffic key to be attached after the modified RERR as long as the captured traffic keys are still "fresh" enough to be authenticated by the receivers. SEAODV experiences least negative impact against this attack since a receiving node only authenticates the RERR that comes from its active one-hop neighbors. This forces malicious node can only forward the replayed RERRs come from the receiving nodes' one-hop neighbors in order to launch this type of attack.

Tunnelling
ARAN uses the total time consumed in seeking a route as physical metric. It does not guarantee the shortest path in terms of hop count, but does offer the quickest path. This is still not enough to defeat the tunnelling attack because malicious nodes can simply adopt high-power gain transceiver to tunnel the routing messages such as RREP in order to make the source believe that the "tunnelled path" is the quickest one. As a consequence, malicious nodes would have been included on the final route towards destination and gained all the subsequent data packets passed through them. Similar methodology would be taken by malicious nodes to launch this attack on SAODV and SEAODV with the difference that now the actual routing metric is misrepresented in terms of hop counts. LHAP only authenticates messages from its one-hop neighbors, it makes tunnelling attack become more tougher to be launched since malicious nodes now have to intercept the "fresh enough" traffic keys at both ends of the tunnel.

Summaries for each routing protocol in terms of defending against those identified attacks are presented in Table 1.

7 Performance Evaluation

Performance evaluation is presented to prove that SEAODV is superior against ARAN and SAODV in terms of computation cost and route acquisition latency.

7.1 Computation Cost

Computational cost is measured and computed at every node in the network. Since every node in the wireless mesh network can be looked upon as both the sender and the receiver, the total computation cost incurred at each node is going to be the cost of this node being as a sender plus the cost of the node being as a receiver. This methodology is applied to the evaluation of the computation cost for the three secure routing protocols: ARAN, SAODV and SEAODV. Variables and notations used for computing computation and communication cost are shown in Table 2.

1) **ARAN**
ARAN is a hop-by-by secure routing protocol for Ad hoc networks; it uses public key cryptography to guarantee integrity of routing message. However, a major drawback of ARAN is that it requires extensive signature generation and verification during the routing discovery process. In ARAN, all computation cost experienced at each hop comes from the extensive signature generation and verification. To be more detailed, during the routing discovery process, each sender generates its own signature and uses it to sign the entire routing message before sending it back to the wireless channel. Once the message is received on its fly to the destination, the receiver

Table 2: Vulnerabilities of various routing protocols

$Signature_{Gen}$	Signature generation cost
$Signature_{Ver}$	Signature verification cost
H	Hash operation cost
MAC	Cost for computing a MAC
$Max_{Hop_{Count}}$	Maximum hop count
Hop_{Count}	Number of hop count
N	Total number of nodes on the established route
Broadcast	Broadcast routing message
Unicast	Unicast routing message

has to verify the signature(s) first before updating its routing table. According to the operation of ARAN, receivers are required to be classified into two different categories. Receivers that are only one-hop away from the originator of the RREQ or RREP fall into the first category and those are more than one-hop away are referred to the second category. The reason is that receivers in different category incur different computational cost in terms of number of signature verifications. Being a receiver in the first category, which means node is only one-hop away from the originator of RREQ or RREP, node is required to do two times of signature verifications if the routing message comes from the originator of the RREQ or RREP. The first signature verification is used for verifying the certificate of the originator of RREQ or RREP and obtaining the public key of the originator. The second one is required to verify the signature of the originator by using the public key of the originator. However, the node still needs to perform four times of signature verifications should the routing message come from node other than the originator of RREQ or RREP. In contrast to the node being a receiver in the first category, node in the second category experiences four times of signature verifications when receives a RREQ or RREP from its one-hop neighbor. In addition to two times of signature verifications described in the last paragraph, another extra two times of signature verifications are a must due to the verification of both certificate and signature of the node from which it receives the RREQ or RREP.

Now the computation cost in terms of number of signature generation and verification can be derived and given below.

$Signature_{Gen}$	(Sender)
$2 \times Signature$	(Receivers that are ONLY one-hop away from the originator of RREQ or RREP)
$4 \times Signature$	(Receivers that are more than one-hop away from the originator of RREQ or RREP)

Therefore, the computation cost for an established route of N nodes between source S and destination D is expressed as

$$\begin{aligned} &2 \times (N-4) \times (Signature_{Gen} + 4 \times Signature_{Ver}) \\ &+2 \times [(Signature_{Gen} + 2 \times Signature_{Ver}) \\ &+(Signature_{Gen} + 4 \times Signature_{Ver})] \\ &+2 \times (Signature_{Gen} + 4 \times Signature_{Ver}). \end{aligned}$$

This equation indicates that as the number of nodes on the established final route increases, the number of intermediate nodes who are at least two hops away from the originator of RREQ or RREP also rises, hence the total computational cost of all the nodes on the final route are going to boost up.

2) **SAODV**

SAODV is a secure variant of AODV. Routing operations are similar to that of AODV; however, it applies cryptographic extensions to provide authenticity and integrity of routing message. It uses hash chains to prevent manipulation of hop count field and digital signature to secure the rest of the routing message. However, an adversary can still increase the hop count. SAODV offers two types of signature extensions, named single signature and double signature extensions. In the evaluation of computation cost for SAODV, we only consider the single signature extension due to its simplicity. By using single signature extension, intermediate nodes cannot reply to a RREQ message simply because it cannot properly sign its RREP message. Alternatively, it just behaves as if it did not have the route and forwards the RREQ message. The only node that can reply to a RREQ is the destination itself. Before rebroadcasting a RREQ or forwarding a RREP, a node needs to apply the hash function to the Hash value in the signature extension in order to account for the new hop. If the node itself is the originator of RREQ or RREP, then it is not required to perform the hash operation, but the cost of generating digital signature should be included.

Upon receiving the RREQ or RREP, the receiver is required to apply the hash function h ($Max_{Hop_{count}} - Hop_{Count}$) times to the value in the hash field in order to secure the hop count. Apart from that, the receiving node also needs to verify the signature generated by the originator of the RREQ or RREP.

The computation cost of the node being as a sender or a receiver in SAODV is given below.

$Signature_{Gen}$	(Sender, Originator of RREQ or RREP)
H	(Sender, Intermediate Node
$H \times (Max_{Hop_{count}}$	(Receiver)

$$-Hop_{Count})$$
$$+2 \times Signatue_{Ver}$$

The equation indicates that as the total number of nodes on the finalized route increase, more hash operations and signature verifications are required to be performed during the route set up process.

3) **SEAODV**
The computation cost of SEAODV is simple and straightforward in contrast to that of ARAN and SAODV. In SEAODV, the computation cost involved to every node on the route is exactly the same whenever the node acts to be a sender or a receiver. The computation cost for a finalized route with N nodes can be deduced $2\times(N-2)\times 2\times MAC+2\times 2\times MAC$. In this equation, our scheme only involves the operation of MAC and with no signature at all.

In this part, three secure routing protocols (ARAN, SAODV and SEAODV) are evaluated in terms of computation cost. The computation cost for computing signature, hash operation and MAC are listed in Table 3 in which different cryptographic primitives have been implemented based on the open source Crypto++ library in the PDA platform with an Intel Xscale 400 MHz CPU, 64 MB SDRAM, and 32 MB Flash ROM [17]. These implemented primitives include RSA, SHA-1 and HMAC. The RSA in Table 3 is 1024 Bits with exponent being 17.

Table 3: Computational cost of cryptographic operations

Abbreviation	Definition	Computational time
$Signature_{Gen}$	RSA Signature generation	33.3
$Signature_{Ver}$	RSA Signature verification	1.42
H	SHA-1	0.009
MAC	HMAC	0.015

By applying the cryptographic costs from Table 3 and setting the number of nodes N to be 10, 30, 50, 70 and 100 respectively, the computation cost for secure routing protocols ARAN, SAODV and SEAODV can be calculated in millisecond and shown below. Figure 8 shows that the computation cost almost can be negligible for our SEAODV in contrast to SAODV and ARAN. Ours is much faster (1000 folds on average) than ARAN and almost 100 folds quicker on average than SAODV.

The SEAODV is more functional in wireless mesh networks due to its following superiorities.

- Extreme low computation cost as the number of nodes on the selected route increases;

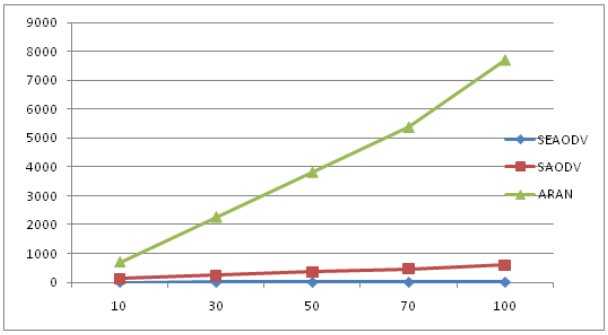

Figure 8: Computation cost for ARAN, SAODV and SEAODV

- Better immunity against to DoS attack in terms of energy consumption;
- Extends entire lifetime of the selected route under the condition that a certain number of nodes on the selected route are classified as mesh client, which tends to be power constrained;
- As being a mesh client, low computation cost simply means longer lifetime the node can enjoy and share valuable information with other nodes in the network;
- Computation cost is computed in terms of timing expense in millisecond, indicates that SEAODV performs excellent in authentication latency due to its efficient cryptographic operation.

7.2 Communication Cost

ARAN, SAODV, and our SEAODV are similar in the way of discovering routes. RREQ is broadcast by the originator of the on-demand node towards the destination. Upon receiving the RREQ, destination unicasts the RREP back to the source from which the RREQ is generated by using the reverse path which has been set up during the flooding of the RREQ. All of these routing protocols apply the same methodology in their routing mechanisms. Therefore, their communication costs (the number of routing messages) are the same. However, ARAN, SAODV, and SEAODV experience various number of control bytes within RREQ and RREP. The more number of control bytes incurred in a single routing message, the larger the entire routing message. Therefore, routing message with bigger size in terms of bytes tends to have a lower probability of successful reception at the destination and suffer longer delay.

Before computing the latency produced by the communication overhead for each of the routing protocols mentioned above, the following assumptions are made:

1) Network throughput is 400 Kbps for a single flow [16];

2) Signature used in ARAN and SAODV are based on 1024 bit RSA algorithm with exponent being 17 and the signed message is 1024 bits;

3) In ARAN, the size of RREQ or RREP generated by the source or destination is smaller than those forwarded by intermediate nodes, which include two signatures and two certificates. While the RREQ or RREP originates by either the source or destination is only comprised of one signature and one certificate. Presume that the route discovery packet (RDP) in ARAN is the same size as that of used in AODV, which is 24 bytes. Therefore, for RREQ and RREP with single signature and certificate, the total size is 312 bytes which is the same as that of in SAODV given below. For RREQ or RREP with double signatures and double certificates, the total size is extended to 568 bytes;

4) In SAODV, 312 bytes in total for both RREQ and RREP, which include original AODV message (24 bytes), signature (128 bytes), top hash (16 bytes), hash (16 bytes) and certificate (128 bytes);

5) In SEAODV, there are totally 40 bytes for either RREQ or RREP. The AODV message costs 24 bytes and the HMAC is 16 bytes.

There are two routing messages in ARAN with single signature, others are double signatures. The total number of bytes required to be transmitted in order to ensure a secure route set up is bytes. In SAODV, all routing messages are the same size, hence the total number of bytes required $568\times(N-2)\times2+312\times2$ to be transmitted is $312\times(N-2)\times2+312\times2$ bytes. Similarly, the total number of bytes of SEAODV incurred during the route set up process is $40\times(N-2)\times2+40\times2$ bytes. Therefore, the number of RREQ and RREP incurred during the entire route setup process can be derived for ARAN, SAODV and SEAODV. For an established route with N nodes on it, the total number of RREQ and RREP is $(N-2)\times2+2$. The number of bytes that are required to be transmitted in order to safely setup a route for ARAN, SAODV and SEAODV can be computed. The communication cost of transmitting those required bits can be calculated below:

$$\text{Communication Cost} = \frac{\text{Total No. of bits need to be Transmitted (bits)}}{\text{Network Throughput}}$$

Now the average route acquisition latency can be derived by using the following equation:

$$\text{Average Route Acquisition Latency} = \text{Computation Cost} + \text{Communication Cost}$$

The following Tables 4, 5, and 6 summarize the total cost required to safely setup a route between a source and a destination for the three routing protocols in terms of route acquisition latency in millisecond.

Table 4: Average route acquisition latency for ARAN

N	Computation cost (ms)	Communication Cost (ms)	Total Latency (ms)
10	695.96	24.28	720.24
30	2255.16	81.08	2336.24
50	3814.36	137.88	3952.24
70	5373.56	194.68	5568.24
100	7712.36	279.88	7992.24

Table 5: Average route acquisition latency for SAODV

N	Computation cost (ms)	Communication Cost (ms)	Total Latency (ms)
10	122.976	14.04	137.016
30	241.796	45.24	287.036
50	353.416	76.44	429.856
70	457.836	107.64	565.476
100	600.966	154.44	755.406

Table 6: Average route acquisition latency for SEAODV

N	Computation cost (ms)	Communication Cost (ms)	Total Latency (ms)
10	0.54	1.8	2.34
30	1.74	5.8	7.54
50	2.94	9.8	12.74
70	4.14	13.8	17.94
100	5.94	19.8	25.74

The average route acquisition latency (the total computation and communication costs) is defined as the average delay between sending a RREQ packet by a source for discovering a route to a destination and the receipt of the first corresponding RREP. Figure 9 shows that the average route latency of our SEAODV is much less in contrast to SAODV and ARAN due to the use of MACs and the smaller size of routing messages.

8 Conclusions

In this paper, we have presented a security enhanced AODV routing protocol, SEAODV. In SEAODV, Blom's key pre-distribution scheme is used to establish keys to ensure that every two nodes in the network uniquely share the pairwise keys. Each node in the network possesses two types of keys: PTK and GTK. PTK is used to accomplish the distribution of GTK while GTK is used to secure the broadcast routing messages between the node and its one-hop neighbors. Security analysis and performance evaluation show that SEAODV is more effective in preventing identified routing attacks and outperforms

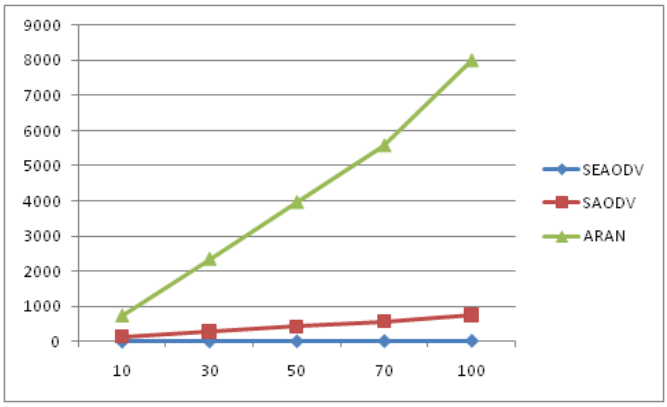

Figure 9: Average route acquisition latency for ARAN, SAODV and SEAODV

ARAN and SEAODV in terms of computation cost and route acquisition latency.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: A survey," *Computer Networks*, vol. 47, no. 4, pp. 445-487, 2005.

[2] M. Bahr, "Proposed routing for IEEE 802.11s WLAN mesh networks," *2nd Annual International Wireless Internet Conference (WICON)*, pp. 133-144, Boston, MA, USA, Aug. 2006.

[3] M. Bahr, "Update on the hybrid wireless mesh protocol of 802.11s," *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*, pp. 1-6, MASS, 2007.

[4] R. Blom, "An optimal class of symmetric key generation systems," *Proceedings of Eurocrypt'84*, LNCS 209, pp. 335-338, 1985.

[5] I. D. Chakeres and E. M. B. Royer, "The utility of hello messages for determining link connectivity," *Proceedings of the 5th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 504-508, 2002.

[6] R. Bruno, M. Conti, and E. Gregori, "Mesh networks: Commodity multihop ad hoc networks," *IEEE Communications Magazine*, vol. 43, pp. 123-131, 2005.

[7] R. Draves, J. Padhye, and B. Zill, "Routing in multi-radio, multi-hop wireless mesh networks," *Proceedings of ACM Mobicom*, pp. 114-128, 2004.

[8] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," *ACM Transactions on Information and System Security*, vol. 8, no. 2, pp. 228-258, Oct. 2003.

[9] L. B. Gergely and I. Vajda, "Provably secure on-demand routing in mobile adhoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1533-1546, 2006.

[10] H. Hassanein and A. Zhou, "Routing with load balancing in wireless Ad hoc networks," *Proceedings of ACM MSWiM*, pp. 89-96, 2001.

[11] Y C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," *Proceedings of MobiCom*, pp. 21-38, Atlanta, GA, Sep. 2002.

[12] M. S. Islam, Y. J. Yoon, M. A. Hamid, and C. S. Hong, "A secure hybrid wireless mesh protocol for 802.11s mesh network," *Proceedings of ICCSA 2008*, LNCS 5072, pp. 972V985, 2008.

[13] IEEE 802.11s Task Group, *Draft Amendment to Standard for Information Technology Telecommunications and Information Exchange Between Systems V LAN/MAN Specific Requirements V Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Amendment: ESS Mesh Networking*, IEEE P802.11s/D1.06, July 2007.

[14] X. Jing and M. J. Lee, "Energy-aware algorithms for AODV in Ad hoc networks," *Proceedings of Mobile Computing and Ubiquitous Networking*, pp. 466-468, Yokosuka, Japan, Jan. 2004.

[15] D. B. Johnson and D. A Maltz, "Dynamic source routing in Ad hoc wireless networks," *Mobile Computing*, vol. 353, pp. 153-181, 1996.

[16] Y. Lin, A. H. M. Rad, V. W. S. Wong, and J. H. Song, "Experimental comparisons between SAODV and AODV routing protocols," *Proceedings of WMuNeP*, pp. 113-122, Oct. 2005.

[17] M. Long and J. Wu, "Energy-efficient and intrusion-resilient authentication for ubiquitous access to factory floor information," *IEEE Transactions on Industrial Informatics*, vol. 2, pp. 40-47, Feb. 2006.

[18] C. E. Perkins and E. B. Royer, and S. R. Das, *Ad Hoc on-demand Distance Vector Routing*, IETF RFC 3561, July 2003.

[19] A. Perrig, R. Canetti, J. D. Tygar, and D. Song,"Efficient authentication and signing of multicast streams over lossy channels," *Proceedings of IEEE Symposium on Security and Privacy*, pp. 56-73, 2000.

[20] K. Sangiri and B. Dahil, "A secure routing protocol for ad hoc networks," *Proceedings of 10th IEEE International Conference on Network Protocols*, pp. 78-89, 2002.

[21] M. L. Sichitiu, "Wireless mesh networks: Opportunities and challenges," *proceedings of the Wireless World Congress*, pp. 5-10, 2005.

[22] M. Zapata and N. Asokan, "Securing ad-hoc routing protocols," *Proceedings of ACM Workshop on Wireless Security*, pp. 1-10, Sep. 2002.

[23] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "LHAP: A Lightweight Hop-by-Hop Authentication Protocol for Ad-Hoc Networks," *Proceedings of ICDCS International Workshop on Mobile and Wireless Network*, pp. 749-755, Providence, Rodhe Island, May 2003.