Provably Secure Routing Protocol for Wireless Mesh Networks

Rakesh Matam and Somanath Tripathy
(Corresponding author: Somanath Tripathy)

Department of Computer Science and Engineering, Indian Institute of Technology Patna
Patna, Bihar-800013, India
Email: {m.rakesh, som}@iitp.ac.in

(Received May 5, 2012; revised and accepted Dec. 30, 2012)

Abstract

Ensuring security of an underlying routing protocol in wireless mesh network (WMN) is a crucial issue because of the multi-hop communication environment and wireless media. This has been realized by various researchers, and several secure routing protocols have been proposed to address the security vulnerabilities. In this work, we point out the vulnerabilities of the important existing secure routing protocols proposed for WMN. Also, we present a secure routing protocol tailored to WMN. We adapt the simulation paradigm approach to prove the security of the proposed routing protocol.

Keywords: Attacker model, secure routing, simulation paradigm, wireless mesh network

1 Introduction

Wireless mesh networks have emerged as a promising technology to provide low-cost, high-bandwidth wireless access services to its clients in a variety of application scenarios [4]. A typical WMN is comprised of a set of stationary mesh routers (MRs) that form the mesh backbone and a set of mesh clients that communicate via MR's. The major functionality of a mesh backbone is to provide secure network services to its clients. Ensuring security of network services is a challenging task as the open nature of the wireless medium and multi-hop cooperative communication environment subject these services to variety of security threats.

Routing is one such network service that is vulnerable to number of threats from adversaries both internal and external to the network. A detailed survey of such routing attacks has been made in [28]. Also, Several mechanisms like [17, 26], have been proposed to mitigate the selfish packet dropping attacks in MANET.

A variety of secure routing protocols include SAODV [27], ARAN [10] Ariadne [12], SRP [21] and SAR [14] have been proposed to address the security vulnerabilities in ad hoc networks. It is observed that these protocols are still vulnerable to a number of security threats in an *Active-n-m* adversary [7] model, in which an adversary can control m compromised nodes with n compromised identifiers in the network.

The contribution of this work is in two folds. First we analyse the security of the existing on-demand hop-by-hop secure routing protocols (SAODV, ARAN and SAR). Later, we propose a secure routing protocol SWMP (secure wireless mesh protocol) to address all the known vulnerabilities. To prove the security properties of SWMP, we employ the simulation paradigm approach that has been previously used by Acs et al. [3]. The simulation paradigm approach has been shown to guarantee the security of routing protocols for ad-hoc networks in [1, 6, 12, 23].

The rest of the paper is organized as follows. Section 2 discusses the related work. In Section 3, we present the network, adversarial model and basic definitions of security used to evaluate the proposed protocol. In Section 4, we present various identified attack scenarios on existing secure routing protocols. Section 5 presents the proposed secure routing protocol for WMN (SWMP). The security proof of SWMP is presented in Section 6. In Section 7, we present a brief discussion on robustness of SWMP. Finally, Section 8 concludes the paper.

2 Related Work

Routing is one of the core networking functionalities of a network. An attacker can easily target the underlying routing protocol to compromise the security of a network. Ensuring security involves addressing of diverse security issues to preserve integrity, confidentiality, privacy and anonymity of routing messages in a routing protocol. A numerous works [10, 12, 14, 15, 16, 18, 19, 21, 27] have been carried out to address different security issues of routing protocols. Even though a lot of work has been done on resolving the routing security issues, still the existing routing protocols are not free from the present vulnerabilities. The three major factors to be considered in evaluating a secure routing protocol are: the adversary model, actual design of the protocol and its security analysis. In this section, we evaluate few of the existing secure routing protocols based on these features.

Ariadne [12] is a secure version of dynamic source routing protocol (DSR) [13] that operates in presence of an *active-n-m* adversarial model. It relies on pre-deployed pairwise symmetric keys or pre-deployed asymmetric cryptography for authentication. Even though the former is more efficient, it requires shared secrets between communicating nodes, which is not always feasible to establish. A third option for Ariadne is the TESLA authentication scheme [3], which is also based on asymmetric encryption, thus requiring a certification authority or pre-deployed keys. TESLA requires packets to be delayed by the longest RTT in the network before they are sent (thus route creation incurs this delay in both request and response phases). The security of Ariadne has been evaluated in [3, 9] and is shown to be insecure.

SAODV [27] is a secure extension to AODV [22] designed to protect route discovery mechanism and provide security features like integrity, authentication and non-repudiation in presence of an *active-n-m* adversarial model. SAODV relies on a two fold mechanism to secure routing messages. Firstly, a routing message is divided into mutable and non-mutable information. Later, digital signatures are used to authenticate non-mutable fields of these messages, and hash chains to secure the hop count information (the only mutable information in the messages). A hash chain is formed by repeatedly applying a one way hash function. Whenever a node has to send a RREQ or a RREP it generates a random number (seed), sets the Max-Hop-Count to the time-to-live (TTL) value in the IP header, and Hash to the seed value. Later, it calculates Top-Hash by hashing the seed Max-Hop-Count times. Every time a node receives a RREQ or a RREP it verifies the hop count of the message. Before rebroadcasting a RREQ or forwarding a RREP, a node hashes the Hash value one time in the Signature Extension. Several security threats against SAODV have been presented by various researchers. The formal security analysis of SAODV has been presented in [2, 11] and shown to insecure in presence of an *active-n-m* adversarial model.

ARAN [10] is an authenticated routing protocol for ad-hoc networks (ARAN) that uses cryptographic certificates to meet security goals such as authentication and non-repudiation, in presence of an *active-n-m* adversary. Route discovery in ARAN is accomplished by broadcasting a route discovery message that is authenticated end-to-end and integrity of the message is verified hop-by-hop. Similarly, the route reply is propagated in unicast fashion by the destination that is also authenticated end-to-end and verified hop-by-hop. ARAN does not consider any routing metric and therefore is not suitable for a high-performance network like WMN. The security of ARAN has been evaluated in [12, 20]. It is shown to be insecure in [20]. In addition to the identified vulnerabilities, we present additional security flaws of ARAN that have been overlooked.

Security-aware routing protocol (SAR) [14] is an on demand routing protocol that uses "level of security" as routing metric instead of hop-count. The route discovery process is similar to AODV with an exception that a source broadcasts a RREQ that contains a field specifying the desired security level called QoP (quality of protection). Only nodes whose security levels are equal to or higher than that required in the packet can forward RREQ and RREP packets. In this way, nodes having lower security level or nodes that are compromised are circumvented.

Secure routing protocol (SRP) [21] is another on-demand secure source routing protocol that secures route discovery process against non-colluding Byzantine adversaries. SRP assumes security association (SA) between source S and destination D. SRP therefore relies on end-to-end security association and does not require intermediate nodes to validate a routing message. It uses an additional header called *SRP* header that contains the query sequence number Q_{SEC}, query identifier number Q_{ID}, and a 96 bit MAC field. Intermediate nodes discard a RREQ, if the *SRP* header absent. The destination D responds to RREQ with an appropriate Q_{SEC} and Q_{ID} to the source. The source matches the Q_{SEC} and Q_{ID} in RREP with the Q_{SEC} and Q_{ID} of currently pending query, for accepting a RREQ. The formal security analysis of *SRP* is presented in [5]. Vulnerabilities of *SRP* are identified in [9, 24] employing formal verification approach.

Security of a routing protocol can be evaluated only by employing formal verification processes due to to the subtle nature of the flaws [6]. Many vulnerabilities have been identified in the existing secure routing protocols by evaluating their security using formal verification approach [3, 8, 9, 11, 12, 20]. The security of these existing protocols is formally verified employing the same adversary model in which they were proposed, and are shown to be insecure.

In this paper, we first identify the vulnerabilities of existing secure routing protocols in presence of an *active-n-m* adversary model. Later, we present a routing protocol that operates in the presence of similar *active-n-m* adversarial model and overcomes all the identified flaws/ vulnerabilities. We further evaluate the security of the proposed protocol in presence of a stronger adversary model (class VIII adversary model as proposed in [6], where attacker nodes have increased reception capability) and is shown to be secure.

3 System Model

In this section, we present the network and adversarial model employed in the design of SWMP along with the system model, its states and correctness to evaluate security of the proposed protocol. The definitions of system correctness and the security are similar to those defined in [3].

3.1 Network Model

We consider a typical wireless mesh network (WMN) architecture, where a set of MR's form the backbone of the WMN. Few of these mesh routers (MR's) designated as gateways are connected to the Internet. Mesh clients (MC's) are typical wireless clients connected to specific MR's with access point functionality. We model the backbone of the WMN as an undirected labelled graph *G(V,E)*, where V is the set of vertices and E is the set of edges. Each MR represents a node, and there is an edge between two vertices if and only if there is a communication link between the corresponding nodes. The communication links between nodes are assumed to be bi-directional. Each link is assigned a cost by a node connected to it with the help of a cost function C_{Link}: $E \rightarrow \mathbb{R}$, to represent the routing metric.

3.2 Security Model

3.2.1 Security Assumptions

We assume that each node in the network obtains a valid certificate signed by a certification authority (CA), later, to be used for authentication. Nodes use authenticated identifiers for secure peer link establishment and during path-selection process. The set of node-identifiers are denoted by L and we label each vertex v of V in G with the identifiers used by the node corresponding to v. Each non-malicious node in the network uses a single identifier that is unique in the network, whereas malicious nodes may use multiple identifiers of the compromised ones. The set of malicious nodes and their identifiers is represented by V^* and L^* respectively. The assignment of identifiers to the nodes is represented by a labelling function Φ: $V \rightarrow 2^L$, which returns the set of labels assigned to each vertex v in G. If v corresponds to a non-compromised node, then $\Phi(v)$ is a singleton and $\Phi(v) \nsubseteq \Phi(v^*)$ holds for any other vertex v^*. A configuration *conf* can be defined as follows.

Configuration: Configuration *conf* is a four tuple (*G(V,E)*, V^*, Φ, C_{Link}) that consists of a network graph, the set of compromised nodes, the labelling function and cost function.

To prevent nodes from establishing neighborhood relations with nodes that are not within each other's radio range, a secure neighbor discovery protocol like [25] is used. As part of secure neighbor discovery, a node authenticates itself with all the nodes in its two-hop range by presenting a valid certificate and thereby validating its neighborhood.

3.2.2 Adversarial Model

We consider an *active-n-m* attacker model, where n signifies the number of compromised *MR's* that hold keying material, and m is the total number of attacker nodes in the network. Usually, attacker nodes in the *active-n-m* attacker model have similar capabilities as non-compromised nodes, but, to strengthen the active-n-m attacker model, we increase the reception capability of an attacker allowing it to receive all messages in the network. This kind of an adversary with increased reception capability is considered in class VIII attacker model of [6]. Practically, higher reception capability signifies an attacker with high-range antennas operating at low power to avoid being detected.

We also assume that a compromised nodes is capable of sharing its identifier with other malicious nodes in the network. The *active-n-m* attacker model combines all neighboring attacker nodes (that can share information from captured messages during network operation) into a single node. The combined single attacker is therefore limited in its transmission capability and is represented as a single entity. Such an attacker is capable enough of launching various kinds of modification, metric manipulation, fabrication and packet dropping attacks such as grey hole and black hole. The major motivation of an attacker in our adversary model is to corrupt the routing protocol to launch various kinds of denial of service (DoS) attacks.

3.3 System State

State of the system Q can be represented by the set of routing tables of all non-compromised nodes. A routing entry in *v's* routing table can be represented as a five tuple field *(v, l_{tar}, l^1_{nxt}, l^2_{nxt}, C)* in Q with identifier l_{tar} as target and l^1_{nxt} as the one-hop and l^2_{nxt} as the two-hop identifier with routing metric C. Thus, the system state $Q \subset$ *(V $\setminus$ V^*)* x L x L x L x $\mathbb{R}$ is a collection of such tuples. A system is said to be in a correct state if all the routing-entries of non-compromised nodes are correct, i.e., if v has a routing entry for target l_{tar} with one-hop l^1_{nxt} and two-hop l^2_{nxt} with cost C, then actually there exists a route that starts at node v, ends at node l_{tar} and the path through l^1_{nxt}, l^2_{nxt} with the metric value C.

Definition 1. ***(Correct State):*** *The state Q of a system is said to be correct if for every (v, l_{tar}, l^1_{nxt}, l^2_{nxt}, C) $\in$ Q, there exists a sequence v_1, v_2,....,v_p of vertices in V such that (v_i, v_{i+1}) $\in$ E for all 1 $\leq$ i $<$ p, and*

- $v_1 = v$,
- $l_{tar} \in \ell(v_p)$
- $l^1_{nxt} \in \ell(v_2)$
- $l^2_{nxt} \in \ell(v_3)$ *and*

- $\sum C_{link} \leq C$.

3.4 Dynamic Representation of the System

3.4.1 Simulation Paradigm

The proposed secure wireless mesh protocol is evaluated by simulation-paradigm approach. The main idea of simulation-based approach is to construct two models, a real-world model, and an ideal world model to evaluate a protocol under investigation. A real-world model describes the operation of the protocol with all its details in a particular computational model whereas an ideal-world model describes the protocol in an abstract way mainly focusing on the services that the protocol should provide. Once constructed, the security of a real-world protocol is compared to that of a ideal-world implementation of the same task.

Both the models contain adversaries and their behavior is not constrained apart from the requirement that it has to run in polynomial time. The presence of an adversary in an ideal-world model essentially has no affect on the system due to the nature of its design. In other words, the ideal-world system is secure by its construction. The real-world model implements the actual protocol under consideration. Once both the models are implemented, the goal is to prove that for any real-world adversary there exists an ideal-world adversary that can achieve the same effects in the ideal-world model as those achieved by the real-world adversary in the real-world model. The security of a protocol interpreted from an adversary's generated view states that, if the view generated by an adversary after executing a protocol in the real world model, can be solely generated from the information it legitimately possesses, then the protocol is termed to be secure. This implies that an adversary cannot gain extra information from the execution of a protocol, and everything that an adversary gathers can be generated by the adversary itself.

The real-world and ideal-world models are constructed as interacting Turing machines. The real-world model consists of honest and adversarial nodes represented by M_i and A_i respectively. The nodes run the desired actual protocol in order to complete a specific task. Similarly, The ideal world model too consists of honest and adversarial nodes, but the honest nodes interact with an ideal functionality F, running the ideal protocol ϕ. The functionality F is synonymous to the protocol to be evaluated, but it is provided with all the initial conditions that allows it to detect when the system goes into an incorrect state. An adversary in the ideal-world cannot gain any extra knowledge except for the information that F chooses to provide.

Real-world Model: The real-world model is comprised of a set of interacting Turing machines those interact via common tapes. The basic rules governing the construction of a real world model are similar to the one presented in [12]. The configuration of the system, denoted by $sys^{real}_{conf,A}$, corresponds to a *conf* = ($G(V,E)$, V^*, L, C_{Link}) and adversary A. To simulate the operation of a routing protocol, the complete model is viewed as a collection of Turing machines $\{M_1,....,M_n,\ A_1,....,A_m,\ H,\ C\}$ where M_i represents a non-compromised node that corresponds to a vertex V $\setminus$ V^* and machine A_j corresponds to compromised vertex in V^* as shown in Figure 1. Machine H models an higher-layer protocol that can initiate a route discovery process to any machine M_i. Machine C models the wireless broadcast medium, represented by the edges in E.

The set of all machines in the model are initialized with certain input data to represent their initial state. Machines are connected to each other via tapes. For example, an interaction between a machine M_i and machine C is facilitated through input and output tapes in_i and out_i. Execution of the protocol begins once the machines are initialized. The machines operate in a reactive manner, that is they need to be activated in particular order for them to perform a defined computation. An activated machine, reads the content of its input tapes, processes the received data, updates its internal state, writes some output on its output tapes, and goes back to sleep (i.e., it waits for the next activation). The machines are activated in rounds by a scheduler.

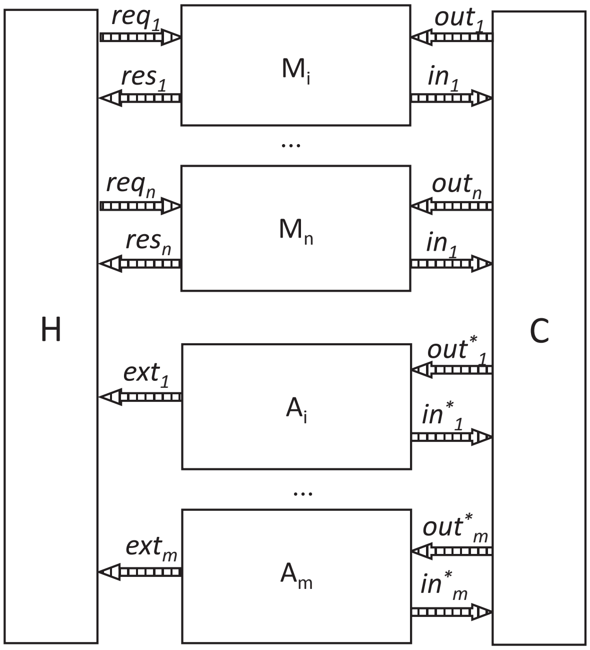

Figure 1: Interconnection of machines in real-world model

Machine C is intended to model the broadcast nature of radio communications. Its task is to read the content of the output tape of each machine M_i and A_j and copy it on to the input tapes of all the neighboring machines, where the neighbor relationship is determined by the configuration *conf*. Machine H models higher-layer protocols (i.e., protocols above the routing layer) of the end-users of the non-compromised devices. H can initiate the route discovery process at any machine M_i by placing a request on tape req_i. A response to this request is eventually returned via tape res_i. Machines M_i (1 ≤ i ≤ n) represent the non-compromised nodes, which belong to the vertices in $V \setminus V^*$. M_i communicates with machines C and H via its tapes in_i, out_i and req_i, res_i. The operation of M_i is essentially defined by the considered protocol.

The computation ends when H reaches one of its final states. The final state can either be a response to the request placed on req_i (1 ≤ i ≤ n) or a time-out due to lack of routes. The output of $sys^{real}_{conf,A}$, is an ensemble of routing tables of the non-compromised nodes. We denote the output by $Out^{real}_{conf,A}$(r), where r is the random input of the model.

Ideal-world model: The ideal-world model that corresponds to a configuration $conf = (G(V,E), V^*, L, C_{link})$ and adversary A is denoted by $sys^{ideal}_{conf,A}$. The main difference between ideal and real-world model is that the set of machines M_i (1 ≤ i ≤ n) is replaced with a new machine called T. The ideal world model is depicted in Figure 2. The operation of the ideal-world model is very similar to the real-world model except for the operation of new machine T. In effect, machine T emulates the behavior of the machines M_i (1 ≤ i ≤ n), with the difference that T is initialized with *conf*, and record a special symbol '\$' when the system gets into an incorrect state.

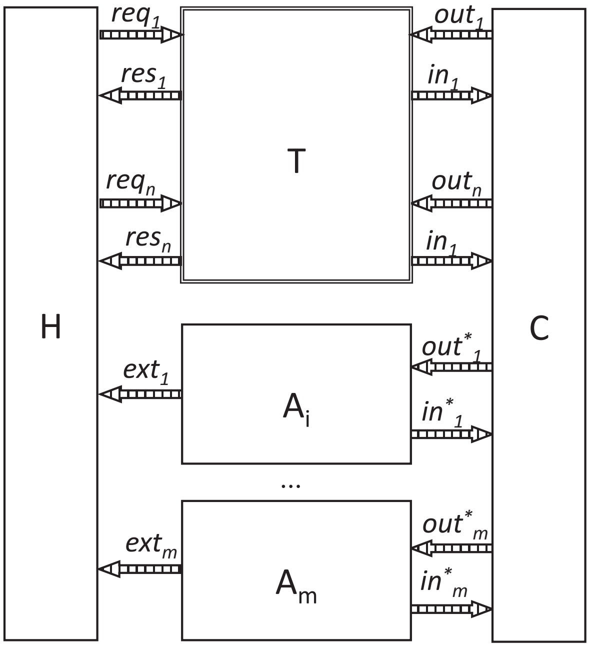

Figure 2: Interconnection of machines in ideal-world model

Similar to the real-world model, the computation ends, when H reaches one of its terminal states. The output of the ideal-world model is either an ensemble of the routing tables (for T has not recorded an incorrect state during the computation), or a special symbol '\$' that indicates that an incorrect state has been encountered. The output is denoted by $Out^{ideal}_{conf,A}$(r) when r is chosen uniformly at random.

Definition 2. ***(Statistical Security):*** *A routing protocol is said to be statistically secure if, for any configuration conf and any real-world adversary A, there exists an ideal-world adversary* $A^{'}$*, such that* $Out^{real}_{conf,A} = Out^{ideal}_{conf,A'}$*, where '=' means "statistically indistinguishable".*

Intuitively, a routing protocol is said to statistically secure if the effect of a real-world adversary on a real-world model can be "almost perfectly" simulated by an ideal-world adversary in the ideal-world model. That is, no ideal-world adversary exists that can cause the ideal-world system to go into an incorrect state, it follows that no real-world adversary can exist that can cause the real-world model to move into an incorrect state with non-negligible probability because if such an adversary exists, then no ideal-world adversary can simulate it "almost perfectly".

4 Vulnerabilities of Existing Secure Routing Protocols

n this section, we present the identified vulnerabilities in some of the existing secure routing protocols. We adapt the same definition of a correct state from [12] to evaluate the security of the considered routing protocols, since these protocols are earlier shown to be secure in their model. It states that, a routing protocol is secure, if it ensures that incorrect entries in the routing tables of non-compromised nodes are generated only with negligible probability. In other words, a routing protocol is secure only when the ideal-world model goes into an incorrect state with negligible probability. We classify the identified attacks into one of the following three types.

- **Metric Manipulation Attack:** Existing secure routing protocols are designed to prevent an attacker from decreasing the routing metric to influence path selection decisions. But, an attacker can still cause inconsistencies in the selected routes without acting on the metric field or by artificially increasing the routing metric. By increasing the routing metric, an attacker can prevent some of the routes from being selected, thus disrupting the network services due to increased congestion on few routes.

- **Route Corruption Attack:** An attacker can launch a route corruption attack by manipulating routing information to disrupt network operation. One way an attacker can achieve this is by colluding with another malicious node and misrepresenting its own identity. The main aim of route corruption attack is to cause inconsistencies in the selected routes.

- **Routing loop attack:** A routing loop attack is one of the more serious forms of active attacks where an attacker can directly influence route selection decisions. An attacker can launch a routing loop attack by impersonating the identity of another node. The main aim of the attacker in launching this attack is to prevent route discovery process from establishing routes.

4.1 SAODV

SAODV is a secure extension to AODV that protects the route discovery mechanism and provides security features like integrity, authentication and non-repudiation. SAODV assumes that each node has a signature key pair from a suitable asymmetric crypto system. Further, each node is capable of securely verifying the association between the address of a given node and the public key of that node. A routing message in SAODV contains mutable and non-mutable information. Therefore, SAODV employs two mechanisms to secure these messages. Digital signatures are used to authenticate the non-mutable fields of a message, and hash chains are used to secure the hop count information (the only mutable information in the messages). Digital signatures ensure that an adversary cannot modify the non-mutable information without the change being detected by non-compromised nodes and hash-chains prevents a node from decreasing the hop-count, and thereby preventing malicious nodes from projecting longer routes as short.

Metric Manipulation Attack: SAODV prevents nodes from reducing the hop-count of a route during route discovery process. But, an attacker can still cause inconsistencies in the selected routes by not acting on the routing metric. For example consider the scenario presented in Figure 3, where a source S tries to discover a route to target T. A malicious node M that receives a RREQ from S can simply forward a RREQ without incrementing the hop-count. A target node T that receives a RREQ from I_1 selects it over other routes as the hop-count through this route is *2*. This results in a routing inconsistency as the route selected {S, I_1, T, 2} is non-existent in the network. The ideal-world model initialized with a configuration $conf = (G(V,E), V^*, L, C_{link})$ moves into an incorrect state thus deeming it to be insecure.

SAODV is also susceptible to metric manipulation attacks where an attacker increases the routing metric to prevent some of the paths from being selected. This results in congestion due to over utilizing of some of the paths and few paths being under utilized. This is a serious form of an attack, as it can result in all kinds of DoS and is rather difficult to detect.

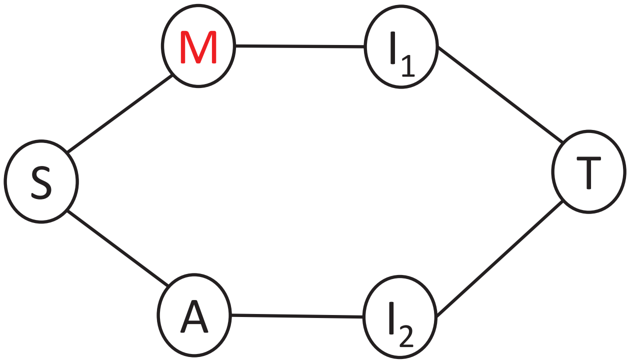

Figure 3: Metric manipulation attack

Route Corruption Attack: Consider the attack scenario as shown in Figure 4 where a source S tries to discover a route to a target node T by broadcasting a RREQ. The target node T that receives this RREQ, responds to it by transmitting an RREP. A malicious node M that receives this RREP, acts on it and forwards to S but using the node identifier of A. On receiving the RREP, node S successfully validates and accepts a non-existent route in the network. Thus, an ideal-world model moves ends up in an incorrect state.

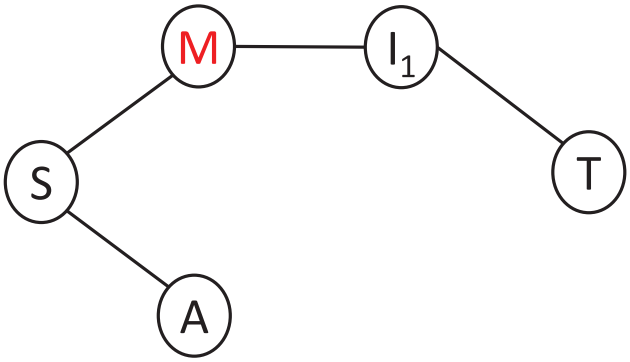

Figure 4: Route corruption attack on SAODV

Routing Loop Attack: SAODV is also susceptible to a routing loop attack when a malicious node M propagates a RREQ by impersonating an honest node. For example, consider a potential path between source S and target T. When node M receives a RREQ, it propagates the RREQ by impersonating the identity of an honest node C. Node I accepts and processes the message and creates a routing entry for S through C. Node C that receives a RREQ from I creates a routing entry for S through I, hence forming a routing loop. An adversary can exploit this flaw to force the ideal-world model to end up in an incorrect state.

$$\mathbf{S} \longrightarrow \mathbf{A} \longrightarrow \boldsymbol{B} \longrightarrow \boldsymbol{M} \longrightarrow \boldsymbol{I} \longrightarrow \boldsymbol{C} \longrightarrow \boldsymbol{D} \longrightarrow \boldsymbol{T}$$

4.2 ARAN

Authenticated Routing for Ad hoc Networks (ARAN), is an authenticated routing framework for secure routing in ad-hoc networks. ARAN utilizes cryptographic certificates to provide authentication and non-repudiation. Routing messages are authenticated end-to-end and verified for integrity at each hop. When a source node A needs to find a route, it generates, signs and broadcasts a RREQ. Upon receiving the message, node B(the one-hop neighbor of A) uses the pubic key of A to verify the signature in received RREQ. If the signature is verified, node B updates its routing table accordingly, signs the RREQ and appends its own certificate before rebroadcasting it to its one-hop neighbors. Otherwise, the received message is considered to be unauthentic and is discarded. This process is repeated by every intermediate node that processes this RREQ. The destination node that receives this RREQ, verifies the signature of source, creates a RREP and unicasts it back along the reverse path to the source.

Route Corruption Attack: Let S be the source node, T be the target node, and X and Y be the adversarial nodes as shown in Figure 5. Initially, a source node S begins a route discovery process by broadcasting a route request message:

$$(RREQ, T, cert_S, N_S, t, Sig_S)$$

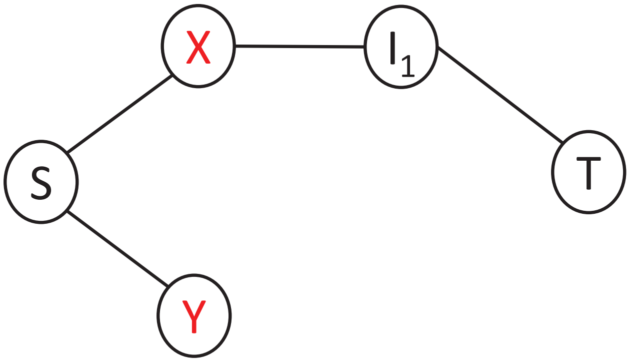

Figure 5: Route corruption attack on ARAN

X receives the request message and rebroadcasts the following request message:

$$(RREQ, T, cert_S, N_S, t, Sig_S, Sig_X, cert_X)$$

When the target T receives the request message and sends a route reply as follows:

$$(RREP, S, cert_T, N_S, t, Sig_T)$$

Then X receives the following reply:

$$(RREP, S, cert_T, N_S, t, Sig_T, Sig_A, cert_A)$$

If X forwards the reply to S, S will set a correct entry in its routing table, in which the target is T, the next hop is X. However, X has another identifier Y, and Y is a neighbor of S, so X uses the identifier Y to send the following reply to S:

$$(RREP, S, cert_T, N_S, t, Sig_T, Sig_Y, cert_Y)$$

S receives the reply, and all the verifications will be successful. So S set an entry in its routing table, in which the target is T and the next hop is Y, and this route does not exist in the actual network. Therefore, this will lead to an incorrect entry in the routing table of a normal node.

Routing Loop Attack: An adversary X can launch a routing loop attack on ARAN with the help of compromised identity information of another node Y. Consider a path in the network as shown below between source S and target T. A source S broadcasts a message after signing the RREQ.

$$S \longrightarrow A \longrightarrow \boldsymbol{B} \longrightarrow \boldsymbol{X} \longrightarrow \boldsymbol{I} \longrightarrow \boldsymbol{Y} \longrightarrow \boldsymbol{C} \longrightarrow \boldsymbol{T}$$

Node X that receives the RREQ message, broadcasts it after appending node Y's information instead of its own information. I accepts and process the message and creates a routing entry for S through Y. Node Y that receives a RREQ from I creates a routing entry for S through I, hence forming a routing loop. ARAN also suffers from delayed transmission attacks due to absence of routing metric and selects routes based on the processing delay. In such a network, a node can delay processing of a RREQ message and prevent few of the paths from being selected. Few numbers of adversarial nodes can significantly affect the performance of the network by simply delaying the processing of messages.

The above presented attacks are modelled on an *active-n-m* adversary model. That is, an adversary is assumed to operate in a constrained environment described by the adversary model. With a little enhancement of the adversarial capability (i.e., the adversary with reception capability), leads to a number of extra vulnerabilities.

5 The Proposed Secure Routing Protocol for WMN

The proposed secure wireless mesh protocol (SWMP) makes use of digital signatures to prevent all the vulnerabilities including those presented in the earlier section. SWMP requires every node to maintain secure neighborhood relations with all the nodes in its two-hop range. To prevent malicious nodes from manipulating the accumulated metric, the RREQ is protected with the help of signatures. SWMP mainly prevents nodes from modifying the metric field appended by previous nodes in the network. Route discovery in SWMP is accomplished by a broadcast route discovery message from a source node that is replied back by destination node.

SWMP uses signatures to provide authentication and message-integrity to the route discovery process. Therefore it requires the use of a trusted certificate server, whose public key is known to all valid nodes. Nodes use these certificates to authenticate themselves to other nodes during the peer link establishment process. The proposed protocol relies on neighborhood relations (both one-hop and two-hop) established during neighbor discovery process. Each node maintains a list of public keys of all the nodes in its one-hop and two hop neighborhood. The public and private keys of a node A are represented by K_A^u and K_A^v respectively. Digital signatures provide integrity to the routing messages. The route selection process of SWMP is detailed as follows.

5.1 Route Discovery Process

The goal of the route discovery process is to allow nodes to select optimal path between source and destination. Route selection process assures that the selected path is in fact an optimal of available paths.

The source node, *S*, as shown in Figure 6, begins path selection process to destination T by broadcasting a route request (RREQ):

$$S \longrightarrow * : \{RREQ, PrvHopAdr\}K_S^v\{M_S, S\_SQ\}K_S^v$$

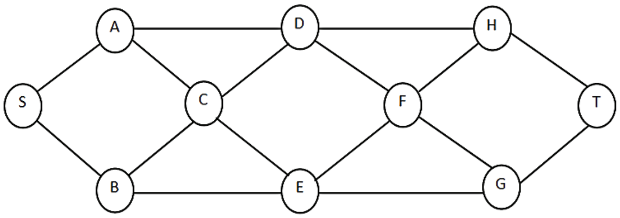

Figure 6: A simple network

The contents of the RREQ are similar to that of HWMP RREQ element. The *PrvHopAdr* element in the RREQ element is the address of the previous hop node from which the current transmitting node received the RREQ. It is not applicable to the source node transmitting the RREQ. The RREQ element is extended to include an authenticated metric field.

Nodes A and B that receive the RREQ, verify the authenticity of the message and metric field by verifying the signature. Before re-broadcasting the RREQ, nodes A and B set the *PrvHopAdr* field in the RREQ with address of *S*. Nodes A and B also compute the cumulative metric M_A and M_B(from $A \rightarrow S$ and $B \rightarrow S$) respectively. The metric field along with the source sequence number (S_SQ) is independently signed and appended to the RREQ. The source sequence number in the appended metric field ensures the freshness of the RREQ. The metric originally appended by S has kept intact.

$$A \longrightarrow * : \{RREQ, S\}K_A^v\{M_S, S\_SQ\}K_S^v\{M_A, S\_SQ\}K_A^v$$

$$B \longrightarrow * : \{RREQ, S\}K_B^v\{M_s, S\_SQ\}K_S^v\{M_B, S\_SQ\}K_B^v$$

When nodes *C*, D and E receive the broadcasted RREQ, they verify the authenticity of the RREQ message and the metric fields appended by S and A (or B). No RREQ is processed if the *PrvHopAdr* in the RREQ does not correspond to a registered two-hop neighbor (a node that is not discovered during the neighbor discovery process). They act on the RREQ accordingly and mark the path entry in their forwarding table. Let a node C receives a RREQ from A, acts on the RREQ by setting the *PrvHopAdr* in the RREQ to A. It is further authenticated by *C*. The metric field appended by S is discarded and the current cumulative metric M_C (till $C \rightarrow A$) is authenticated and appended. This process is repeated by each intermediate node until the RREQ reaches its destination. Each intermediate node repeats this process to establish an authenticated route towards the source. The RREQ broadcasted by C is given by:

$$C \longrightarrow * : \{RREQ, A\}K_C^v\{M_A, S\_SQ\}K_A^v\{M_C, S\_SQ\}K_C^v$$

5.2 Route Reply Process

The destination T responds to a RREQ that offers better air-time metric. The metric comparison is restricted to two hops similar to other intermediate nodes. On successfully validating the RREQ, the destination node responds to it by unicasting a signed route reply. The RREP element is modified to include the *PrvHopAdr* field. The *PrvHopAdr* allows a node to keep track of both the forward and reverse routes. The format of the RREP that is individually addressed to H is shown below:

$$T \longrightarrow H : \{RREP, PrvHopAdr\}K_T^v\{M_T, T\_SQ\}K_S^v$$

Even though the RREP message is generated and addressed individually to confirm the reverse path to the source, signing the metric field (along with target sequence number) independently is necessary to preserve the freshness of the RREP. Each node updates the address fields along with *PrvHopAdr* field, signs and propagates the route reply. The route reply process confirms the path selected on both sides of the route (forward and reverse directions).

6 Security Proof

Theorem 1. *The proposed SWMP is statistically secure if the signature scheme is secure against chosen message attacks.*

Proof. In order to prove the security of SWMP let us construct an appropriate ideal-world adversary $A^{'}$ as discussed in Section 3 for any real-world adversary A. Next, initialize both the systems with the same configuration *conf* and same random input *r*. The proposed protocol is said to be secure if the output ($Out^{ideal}_{conf,A'}$) of the ideal-world model for ideal-world adversary $A^{'}$ is statistically indistinguishable from the output of a real-world model ($Out^{real}_{conf,A}$) for any adversary A. In this case, $sys^{real}_{conf,A}$ and $sys^{ideal}_{conf,A'}$ are identical implies that, in each step the state of the corresponding machines and the content of the corresponding tapes are same. On the other hand, if an incorrect state is encountered, the output configurations of both the models do not match as the ideal-world model outputs a special symbol($).

A system moves into an incorrect state when a routing-entry of a non-compromised node v is incorrect. Node v sets a routing entry (l_{tar}, l^1_{nxt}, l^2_{nxt}, C) for a target l_{tar} only if it has received a signed RREQ message which has been traversed through l^2_{nxt} and l^1_{nxt} with metric equal to C, that is cumulative of $M_{l^2_{nxt}}$ and $M_{l^1_{nxt}}$.

Now, the node v shares one-hop and two-hop neighborhood relations with l^1_{nxt} and l^2_{nxt}. The routing entry can be made incorrect forcing one of the following cases to occur.

- Case 1: There is no route from v to a node that uses the label l_{tar}, but attacker forces l_{tar}, to accept the corresponding message.

- Case 2: There are routes from v to a node that uses the label l_{tar}, but attacker fabricates RREQ/RREP messages in such a way that none of the routes goes through the one-hop and two-hop neighbors (l^1_{nxt} and l^2_{nxt} of v).

- Case 3: There are routes from v to a node that uses the label l_{tar} going through l^1_{nxt} and l^2_{nxt} of v. But, attacker changes the metric values with cost lower/ higher than the actual cost C.

To succeed in Case 1, an attacker needs to generate a fabricated RREQ/RREP message that contains the RREQ element and the signed metric fields of the *PrvHopNode* (through which the RREQ has traversed) and own signed metric field.

In Case 2, to force a genuine node v for accepting an incorrect routing entry, the attacker must fabricate a RREQ/RREP message such that it has traversed through l^2_{nxt} and l^1_{nxt}. To perform this the signature mechanism needs to be forged. Alternatively, if the attacker uses the label (l^*) of other malicious nodes to fabricate a RREQ/RREP message, node v does not accept such a message since l^* is not a registered label.

In Case 3, an attacker needs to modify the metric field appended by a *PrvHopNode* in RREQ/RREP message to force the genuine node v in accepting a route whose metric value is lower/ higher than the actual cost C. Let $C^{'}$ be the minimum of the costs of routes in R. If the signatures of l^2_{nxt} and l^1_{nxt} have not been forged, the RREQ must have taken one of the routes in R. However, since the metric C is independently signed, the value signed by l^2_{nxt} cannot be acted upon by l^1_{nxt} without forging the signature of l^2_{nxt}. Therefore, for $C^{'}$ to be selected over C, either $C^{'}$ has be forged or $C^{'}$ is in fact the best of all the available metric values, which is not the case as C has been selected over $C^{'}$.

Thus the ideal-world model enters into an incorrect state, only if the signature mechanism is forgeable. Fortunately, the probability of forging a cryptographic signature mechanism is computationally infeasible. So, the output configuration of both the real-world and the ideal-world models are identical, *i.e.,* $Out^{ideal}_{conf,A'}$=$Out^{real}_{conf,A}$. □

7 Efficiency of SWMP

In this section, we present the robustness of SWMP against various known attacks. We also compare the security property achieved by SWMP with some of the important existing protocols.

7.1 Robustness of SWMP

- *Metric Manipulation Attack:* The design of SWMP prevents malicious nodes from manipulating the routing metric, using digital signature. This is because SWMP use two metric fields signed independently by the preceding two hops. Each next node verifies the validity of its neighbors (both one and two-hop nodes) through which the RREQ/RREP has traversed through signature verification. Therefore, manipulation of RREQ/RREP message can be detected. Besides this, if a node does not act on the metric and send, it would be detected by the subsequent node.

- *Route Corruption Attack:* The verification process adapted by each node for accepting a routing message allows SWMP to prevent such an attack. This attack is feasible, only when an attacker can forge signature mechanism employed by underlying neighbor discovery protocol and able to compromise two consecutive nodes (previous and second-hop). Since the probability of forging a signature scheme is negligible, SWMP is resistant to route corruption attacks.

- *Routing Loop Attack:* An attacker can cause loops in a routing protocol by exploiting the flaws in its design. SWMP is carefully designed to overcome this flaw in existing protocols and thereby preventing an attacker from causing a routing loop. Since, routing loop attack is a special case of a route corruption attack, the same design features that prevent an attacker from corrupting a route in turn prevent the attacker from forcing routing loops during route selection process. Moreover, for an attacker to successfully launch a routing loop attack, it needs to compromise both one and two-hop nodes on either side of the route.

- *Relay Attack:* To launch a relay attack on SWMP, an attacker needs to collude with a far away node to relay messages and later needs to convince the respective neighbors of relay nodes that a node located far-away node is indeed it's two-hop neighbor. This is a primary requirement that needs to be met to perform a Relay attack, as nodes accept messages that successfully pass the verification process on both one- and two-hop previous node. Therefore, an attacker

Table 1: Robustness of existing protocols

Protocol	**Metric Manipulation Attack**	**Route Corruption Attack**	**Routing Loop Attack**
SAODV [27]	Not Resistant	Not Resistant	Not Resistant
SRP [21]	Not Resistant	Not Resistant	Resistant
ARAN [10]	Resistant	Not Resistant	Not Resistant
Ariadne [12]	Resistant	Not Resistant	Resistant
endairA [3]	Resistant	Resistant	Resistant
SWMP	Resistant	Resistant	Resistant

first needs to forge the underlying signature mechanism, to successfully launch a relay attack. Since, the probability of the forging signature mechanism is negligible, SWMP is considered to be resistant against relay attack.

7.2 Discussion

The main features that enhance the security of SWMP are the two-step verification process adapted by nodes during route selection process and restricting intermediary nodes from acting on the metric field. Since, each node verifies the validity of the two-hop path traversed by a routing message, it is infeasible for an attacker to launch modification attack without being detected by a processing node. None of the existing protocols like SAODV, ARAN, SRP, SAR, Ariadne are resistant against such an *active-n-m* adversary model. The secure routing protocol *endairA* claimed to be secure against an *active-n-m* adversary model. But, *endairA* is not secure against the class VIII attacker model (*active n-m* attacker with increased reception capability). Malicious nodes in class VIII allows the attacker nodes to freely receive all the messages transmitted in the network, which enables them to launch much stronger route corruption attack like a relay attack. None of the existing protocols like SAODV, ARAN, SRP, SAR, Ariadne and endairA can withstand relay attacks launched by a class VIII adversary. Tables 1 and 2 summarizes respectively the robustness of various existing protocols including SWMP against different known attacks and against the different adversary model.

8 Conclusion

Security of a routing protocol is essential to ensure desired performance of a WMN. In this paper, we have pointed out the security pitfalls of important secure routing protocols which could be suitable for WMNs. Further, we have designed a secure routing protocol for WMN. Security of the proposed protocol is evaluated by employing the simulation paradigm approach. The proposed mechanism is found to be secure against enhanced reception capability of *active n-m* adversary model.

Table 2: Security of existing protocols against the attacker model

Protocol	**Active-n-m Attacker Model (Class VII)**	**Class VIII Attacker Model**
SAODV [27]	Not Secure	Not Secure
SRP [21]	Not Secure	Not Secure
SAR [14]	Not Secure	Not Secure
ARAN [10]	Not Secure	Not Secure
Ariadne [12]	Not Secure	Not Secure
endairA [3]	Secure	Not Secure
SWMP	Secure	Secure

References

[1] G. Acs, L. Buttyan, and I. Vajda, "Modelling adversaries and security objectives for routing protocols in wireless sensor networks," in *Proceedings of the ACM SASN*, pp. 49–58, 2005.

[2] G. Acs, L. Buttyan, and I. Vajda, "Provable security of on-demand distance vector routing in wireless ad hoc networks," in *Proceedings of the Second European Workshop on Security and Privacy in Ad Hoc and Sensor Networks (ESAS 2005)*, pp. 113–127, 2005.

[3] G. Acs, L. Buttyan, and I. Vajda, "Provably secure on-demand source routing in mobile ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 11, pp. 1533–1546, 2006.

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

[5] T. R. Andel and A. Yasinsac, "Automated security analysis of ad hoc routing protocols," in *Proceedings of the Joint Workshop on Foundations of Computer Security and Automated Reasoning for Security Protocol Analysis*, pp. 9–26, July 2007.

[6] T. R. Andel and A. Yasinsac, "Surveying security analysis in manet routing protocols," *IEEE Communication Surveys and Tutorials*, vol. 9, no. 4, pp. 70–84, 2007.

[7] T. R. Andel and A. Yasinsac, "Adaptive threat modeling for secure ad hoc routing protocols," *Electronic*

Notes in Theoretical Computer Science, vol. 197, no. 2, pp. 3–14, 2008.

[8] M. Burmester and B. D. Medeiros, "Towards provable security for route discovery protocols in mobile ad hoc networks," *IACR Cryptology ePrint Archive 2007*, p. 324, 2007.

[9] L. Buttyan and T. V. Thong, "Formal verification of secure ad-hoc network routing protocols using deductive model-checking," in *Proceedings of the Wireless and Mobile Networking Conference (WMNC 2010)*, pp. 1–6, Oct. 2010.

[10] B. Dahill, B. N. Levine, E. Royer, and C. Shields, "A secure routing protocol for ad hoc networks," in *Proceedings of the International Conference on Network Protocols (ICNP 2002)*, pp. 78–87, Nov. 2002.

[11] A. B. Gurdag and M. U. Caglayan, "A formal security analysis of secure aodv (saodv) using model checking," in *Proceedings of the 8th International Symposium on Computer Networks ISCN (2007)*, pp. 38–44, 2007.

[12] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demonad routing protocol for ad hoc networks," in *Proceedings of the ACM Conference on Mobile Computing and Networking (Mobicom 2002)*, pp. 21–38, Atlanta, Georgia, Sep. 2002.

[13] D. B. Johnson, Y. Hu, and D. A. Maltz. "The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4,". tech. rep., Feb. 2007.

[14] R. Kravets, S. Yi, and P. Naldurg, "A security-aware routing protocol for wireless ad hoc networks," in *Proceedings of ACM Symposium on Mobile Ad-Hoc Networking and Computing*, pp. 286–292, 2001.

[15] C. T. Li, , and M. S. Hwang, "A lightweight anonymous routing protocol without public key en/decryptions for wireless ad hoc networks," *Information Sciences*, vol. 181, no. 23, pp. 5333–5347, 2011.

[16] C. T. Li, C. C. Yang, and M. S. Hwang, "Improving the security of a secure anonymous routing protocol with authenticated key exchange for ad hoc networks," *International Journal of Computer Systems Science and Engineering*, vol. 23, no. 3, pp. 227–234, 2008.

[17] C. T. Li, C. C. Yang, and M. S. Hwang, "A secure routing protocol with node selfishness resistance in manets," *International Journal of Mobile Communications*, vol. 10, no. 1, pp. 103–118, 2012.

[18] C.a Li, Z. Wang, and C. Yang, "Source routing for wireless mesh networks," *International Journal of Network Security*, vol. 13, no. 2, pp. 109–120, 2011.

[19] H. Li and A. P. Dhawan, "Mosar: A secure on-demand routing protocol for mobile multilevel ad hoc networks," *International Journal of Network Security*, vol. 10, no. 2, pp. 121–134, 2010.

[20] Q. Niu, "Formal analysis of secure routing protocol for ad hoc networks," in *Proceedings of the International Conference on Wireless Communications and Signal Processing (WCSP 2009)*, pp. 1–4, Nov. 2009.

[21] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *Proceedings of the SCS Communication Networks and Distributed Systems Modelling and Simulation Conference (CNDS 2002)*, Jan. 2002.

[22] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proceedings of Second IEEE Workshop on Mobile Computing Systems and Applications(WMCSA 1999)*, pp. 90–100, Atlanta, GA, Feb. 1999.

[23] A. Perrig, R. Canetti, D. Song, and J. D. Tygar, "Efficient and secure source authentication for multicast," in *Proceedings of Network and Distributed System Security Symposium, (NDSS 2001)*, San Diego, CA, Feb. 2001.

[24] M. O. Pervaiz, M. Cardei, and J. Wu, *Routing Security in Ad Hoc Wireless Networks.* New York: Springer, 2005.

[25] B. Swathi, S. Tripathy, and R. Matam, "Secure peer-link establishment in wireless mesh networks," in *Proceedings of Advances in Intelligent Systems and Computing*, pp. 189–198, Chennai, India, July 2012.

[26] Y. Yu, L. Guo, X. Wang, and C. Liu, "Routing security scheme based on reputation evaluation in hierarchical ad hoc networks," *Computer Networks*, vol. 54, no. 10, pp. 1460–1469, 2010.

[27] M. G. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proceedings of the ACM Workshop on Wireless Security (WiSe)*, Atlanta, GA, Sep. 2002.

[28] W. Zhang, Z. Wang, S. K. Das, and M. Hassan, *Security Issues in Wireless Mesh Networks.* New York: Springer, 2008.

Rakesh Matam currently is a Ph.D. student in department of computer science and engineering, Indian Institute of Technology Patna. He obtained his M.Tech degree from Kakatiya University, Warangal. His research interests include network security, wireless routing protocols and peer-to-peer networking.

Somanath Tripathy received his Ph.D degree in Computer Science and Engineering in 2007 from Indian Institute of Technology Guwahati. At present he is an Asst. Professor in Indian Institute of Technology Patna. His research Interest includes lightweight cryptography, Network security, security issues in resource constrained devices (RFID, Sensor networks). He has published about 30 research papers in journals and conferences.