Formal Verification of Authenticated AODV Protocol using AVISPA

Kinchit Vishesh
Punjabi University Regional Centre
for I.T. and Mgmt., Mohali

Amandeep Verma
Punjabi University Regional Centre
for I.T. and Mgmt., Mohali

ABSTRACT
Due to the advancement in the communication technologies, people can communicate with each other anywhere on the move. The concept of ad hoc networks comes into existence in such scenarios. These networks require secure routing protocols; as such networks are vulnerable to various attacks, because of their open and dynamic infrastructure. In this paper we analyze ad hoc on-demand distance vector routing protocol (AODV) using formal verification technique. The verification has been carried out with the help of an automated tool, AVISPA. The result reveals poor authentication between the nodes of the routing protocol. To overcome the weak authentication problem found, we incorporate a secure authentication technique in AODV specification and prove it to be secure by formally verifying the results.

General Terms
Ad hoc networks, Formal Verification, Authentication.

Keywords
AVISPA, HLPSL, AODV.

1. INTRODUCTION
Ad hoc networks change the current scenario of communication. The concept of ad hoc network allows the wireless devices to communicate without any central server or access point. Ad hoc network are very much flexible and adaptable to the needs and requirement of the data traffic travel throughout the network. The network infrastructure is not fixed in such kind of networks, as the nodes involve in these networks are movable and also the participation of the nodes is ad hoc. In ad hoc networks each node can move into the network and leave it at any point of time. Mobile nodes that are within each other's range communicate directly via wireless links, while those that are far apart rely on other nodes to relay messages as routers [9]. Each node guides the routing messages according to the routing protocol designed for such kind of networks.

An ad hoc routing protocol is a standard that decides the routes and the flow of the packets in the network. The general mechanism of ad hoc routing protocols is that, they broadcast the message to discover the path between the source and the destination node. And when it is found, the data packets are then travel along that path. An ad hoc routing protocol could be proactive (table driven), reactive (on demand) or hybrid [8]. Proactive routing protocols, which is also known as table driven routing protocols maintains the table, consisting of fresh routes to the destination, at the regular interval of time, by exchanging the table information between the nodes periodically. Examples of proactive routing protocols are Destination-Sequenced Distance –Vector Routing (DSDV), Clusterhead Gateway Switch Routing (CGSR), Wireless Routing Protocol (WRP) and many others. Reactive routing protocols on the other hand initiate the route discovery only when a node requires a route to the destination node, to which the node wants to send the data. That is why they are also termed as on-demand routing protocols. Examples of reactive routing protocols are AODV, Dynamic Source Routing (DSR), Temporally Ordered Routing Algorithm (TORA), Associativity-Based Routing (ABR), Signal Stability Routing (SSR) and many others. Hybrid routing protocols are the combination of both the techniques used in proactive and reactive routing protocols. The technique applied according to the demand of the situation. Examples of hybrid routing protocols are Core Extraction Distributed ad hoc Routing (CEDAR) protocol, Zone Routing Protocol (ZRP) and Zone based Hierarchical Link State (ZHLS) routing protocol. These routing protocols must be tested and verified earlier their implementation in the scenario of real-world. To prove the system's truthfulness, various simulation and testing techniques are available. From one of them a method use to validate the system is formal verification.

We use formal verification technique for the routing protocols to verify that the behavior of the protocols must exhibit the same functionality for which they are designed. Formal verification is the process of proving the correctness, validity and consistency of the protocol model, constructed from the given specifications. Formal verification uses mathematical techniques for the verification purpose. Their main difference compared to simulation methods and tools is that instead of only examining a limited area of the operational space of the system under consideration, they are used to examine the whole state space of possible operations and conditions, under which the system may operate, hence all the possible combinations of input and actions can be taken into account and therefore, all possible outputs can be derived [13]. It is used in theorem proving, equivalence checking, correctness of cryptographic protocols and circuit design. Because the mathematics involved in the process, people tend to believe that the use of formal verification, worthies only for safety critical systems [4]. But this is not the case; formal methods can also be used in the system development purposes. Formal verification can be performed by using the numerous available formal tools. Formal tools are the automated software packages that are used to execute the formal specification of the system, written in the respective language provided along with the tool. The formal languages are required to shape the properties of the system, the algorithms and the routing protocols and act as an input to the formal verification tool. For our analysis purpose, we consider Automated Validation of Internet Security Protocols and Applications (AVISPA), a formal verification tool best suited for verifying security properties that takes an input written in High Level Protocol Specification Language (HLPSL).

This paper is organized as follows: in Section 2, we briefly describe the work done in the field of formal verification and the tools used to verify the properties of the protocols. In Section 3, the overview of the AODV protocol is given. The features and architecture of AVISPA is described in Section 4.

Next the formal verification of AODV is performed and the proposed solution for the authentication is then verified in Section 5. Section 6, consists the conclusion of the paper.

2. RELATED WORK

A large number of formal verification tools are used to verify the model, security properties and various other aspects of the ad hoc routing protocols. One of them is the SPIN model checker used for model checking. It confirms the properties of Wireless Adaptive Routing Protocol (W.A.R.P), by proving the correctness, at 98% in [11]. The case study has been carried out using DT-FPNs (Dynamic topological Fuzzy timing high level Petri Nets) in [3] to verify the correctness of the AODV and the verification results shown proves that the formal verification approach is powerful in verifying the routing protocols in ad hoc networks. To make the advantageous use of both the techniques simulation and formal verification, J-Sim network simulator has been extended to be an integrated environment for both simulation and model checking of network protocols in [1], to model check the AODV that is a widely used and fairly complex network protocol.

The formal analysis of correctness for routing protocols from IETF standards and drafts shown in [7], by proving the correctness of the RIP standard, automated proof of a sharp real-time bound on the convergence of RIP and proof of AODV loop freedom in Higher Order Logic (HOL) and SPIN, demonstrates that the formal verification technique are the effective means of assuring the correctness of the behavior of the ad hoc routing protocols.

Our work is mainly inspire by the model checking of Authenticated Routing for ad hoc Networks (ARAN) and endairA in [5], which shows the results of formal verification using the formal verification tool AVISPA. The results shows that the HLPSL specification of ARAN is found out to be unsuccessful and discovered three kinds of attacks: route disruption, route diversion and creation of an incorrect routing state. On the other hand analysis of endairA reveals no attacks. Thus the results shows the AVISPA as an easy to use tool to verify the security properties of ad hoc routing protocols and motivates us to carry out the formal verification of most widely used ad hoc routing protocol AODV.

3. AODV: AN OVERVIEW

AODV is an on-demand routing protocol which is used to find a route between the source and destination node, whenever a source node demand for a route to the destination node. The detail operation of AODV is described in the rfc [2] provided by the Internet Engineering Task Force (IETF).

AODV complete its operation by executing both route discovery and route reply phase. It consists of three message types: Route Request (RREQ), Route Reply (RREP) and Route Error (RERR).

As shown in Fig 1, in route discovery phase, the RREQ is generated by the source node propagates throughout the network until it reaches the desired destination node for which the route request has been made. The intermediate node that receives any RREQ, searches for the fresh route available to the destination node in its routing table. If the route to the destination node is available by the intermediate node, then it forwards the route request to the neighboring node, and updates the latest information contained in the RREQ message in its routing table. At last when the RREQ message received by the destination node, the destination node generates the RREP message and unicast the message along the path from which the data packets will be sent.

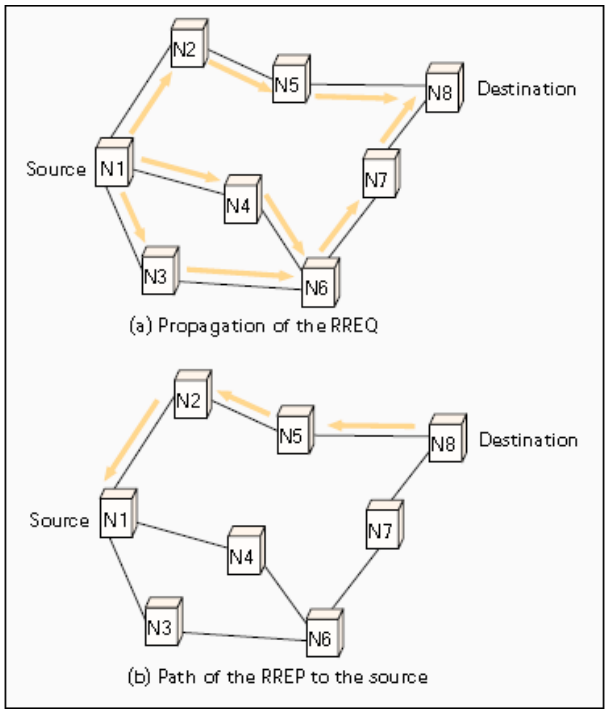

Fig 1: AODV route discovery [6].

The RERR message occurs or comes into existence at a point, when there is some link brokerage or network failure occurs in the network due to which the data packets are unable to reach the destination. This message unicast the information of non availability of route to the destination to previous nodes, and hence, nodes stop sending messages on that route.

These three messages are identified by an unique id and the format of these messages consists of the information like ip-address, destination sequence number, number of hops in between the nodes, etc. Any unauthorized changes made in such message contents may lead to the poor performance of the routing protocol. Hence there must be some security mechanism applied in the routing protocols that prevents any alteration made in the contents in the messages formats.

The work in this paper is mainly focus on securing such routing messages by making it to be sure that only authenticated nodes in the network sends the messages throughout the network and encrypt the messages, so that if any intruder comes into existence in the network, it cannot be able to modify the message contents. The proof has been provided by formally verifying the concept using formal verification tool AVISPA.

4. AVISPA/HLPSL

AVISPA is an automated formal verification tool used to verify the security properties of the routing protocols. The usage of AVISPA for analyzing the protocols can be obtained from the tutorials [15].

AVISPA requires the protocol specification to be written in HLPSL language, which is then provided as an input to the tool, AVISPA, which results in the correctness of the behavior of the protocol described in the specification. HLPSL is a modular, role based language, in which the functionality of each node is described in the respective role written in the specification. As shown in Fig 2, the HLPSL specification written is provided to the tool, which is then converted to the Intermediate Format (IF). Now this IF is then verified by the back-end tools such as On-the-Fly Model-Checker (OFMC), SAT-based Model-Checking (SATMC), Constraint-Logic-

based Attack Searcher (CL-AtSe) and Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP).

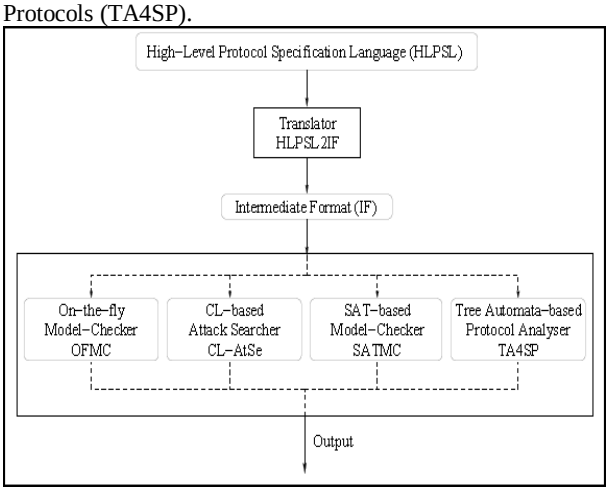

Fig 2: Architecture of AVISPA tool [14].

A special characteristic of AVISPA is that a same specification can be validated by four tools: OFMC, CL-AtSe, SATMC and TA4SP [10]. Hence the verification results are more reliable and meaningful to the user.

5. RESULTS

In this section the results of formal verification of AODV protocol are presented to check the correctness of security properties of the protocol. To formally verify the AODV routing protocol we consider three nodes, i.e. source node (S), intermediate node (IM) and destination node (D) as shown in Fig 3.

Fig 3: Nodes consider for analysis purpose.

5.1 Formal Analysis of AODV

The role of node S describes the start of the session of the routing protocol, in which the RREQ message format is generated as shown in Fig 4.

```
SND(RREQ.RREQ_ID.S.Dest_Seq_No'.Hop_Count'.D)
```

Fig 4: RREQ message format of AODV in HLPSL.

The message format is written in HLPSL specification and the contents of the message are same as the RREQ message format of AODV provided in [2]. The RREQ, in the message shows that the message kind is route request generated by the source node S, asking for the route to destination node D with the destination sequence number Dest_Seq_No, and the nodes in between sends the RREQ request forward by adding the number of hop counts, Hop_Count from the source node. Hence when the source node created the message the value of Hop_Count is set to be zero. This route request message is uniquely identified by the RREQ_ID, through which the nodes that receives the route request message identifies the RREQ request and discards the message, if they receive the same message again or already process that message.

Security Protocol Animator for AVISPA (SPAN) is used to write the AODV protocol HLPSL specification. SPAN helps in developing the specification by providing an interactive user interface. It is designed to help in building the message sequence chart from the HLPSL specification.

In our specification, the basic working of AODV routing protocol is specified in the HLPSL language. Each node in the routing sends and receives the message packets in the format specified by the IETF task group for the AODV routing protocol. Along with the RREQ message, a witness message to the authentication of the particular content of the message is sent, which means that the sender is the witness to the content that it is correct and generated by the authenticated node.

```
witness(S, IM, rreq, RREQ_ID)
```

Fig 5: Authentication message for RREQ_ID.

For example in Fig 5, witness message is used in HLPSL specification which is used for a weak authentication property of S by IM on RREQ_ID, which declares that the S is the witness for the RREQ_ID in the message and this declaration is identified by the constant rreq. The same is done with the RREP message received by any node.

The RREP message format specification written in HLPSL is shown in Fig 6. The RREP in the message format shows that the message kind is route reply originated by the node which replies with the route to the destination node. Dest_Seq_No is the destination sequence number of the destination node D for which the route discovery has been taken place. Here the hop count, Hop_Count, is the number of hops from the node, from which the route reply message is sent to the destination node D. this RREP message is uniquely identified by the RREP_ID.

```
RCV(RREP.RREP_ID.D.Dest_Seq_No'.Hop_Count'.S)
```

Fig 6: RREP message format of AODV in HLPSL.

Now along with the RREP message, a wrequest message is also sent, which requests for the authentication to the particular content of the message from the node that generates the route reply message.

```
wrequest(S, IM, rrep, RREP_ID)
```

Fig 7: Authentication message for RREP_ID

For example the message shown in Fig 7 is the weak authentication property which is used in HLPSL specification to describe that the node S request for the authentication of RREP_ID by IM, which sends a rout reply message to the source node S. This authentication request is identified by the constant unique id rrep. The properties which are to be verified for their correctness and validity are then declared in the goal section of the HLPSL specification.

The goal provided in this specification is to verify the authentication properties that are declared along with the RREQ and RREP messages in the form of witness and wrequest. Both the witness and wrequest are identified by their unique id's, rreq and rrep which are used to define the goal in the goal section of HLPSL specification as shown in Fig 8.

```
goal
        weak_authentication_on rreq
        weak_authentication_on rrep
end goal
```

Fig 8: Goal section of HLPSL specification

The specification written for the AODV protocol is then used as an input to the tool, AVISPA, to verify the defined goals for the protocol verification. The tool first converts the protocol specification into the IF, which is a machine readable form of language and hence, interpreted by the back end tools of the AVISPA. After executing the HLPSL specification of the AODV protocol, we found the specification to be unsafe as shown in Fig 9.

The result shows OFMC output of the protocol specification. Under the SUMMARY section of output, it is given that the protocol is unsafe, and DETAILS section provides the information that an attack is found in the protocol specification. The IF form of the protocol resides in the path given under the PROTOCOL section of the output, with the file name, aodv.if. The GOAL section of output describes the failure of the goal, which is written in the specification for the verification process. The backend that verifies the protocol specification is OFMC, which is given under the BACKEND. The STATISTICS gives us the time required to execute the protocol specification by the tool and the number of the visited nodes during the execution. There is an ATTACK TRACE section in the output that gives the details of

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\aodv.if
GOAL
  weak_authentication_on_rrep
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.03s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i -> (s,3): start
(s,3) -> i: dummy_nonce.dummy_pid.s.0.0.d
i -> (s,3): dummy_nonce.dummy_pid.d.x242.x243.s


% Reached State:
%
% wrequest(s,im,rrep,dummy_pid,i)
% witness(s,im,rreq,dummy_pid)
% state_destination_node(d,s,i,2,dummy_nat,dummy_nat,dummy_nonce,dummy_nonce,dummy_pid,dummy_pid,9)
% state_source_node(s,i,d,0,dummy_nat,dummy_nat,dummy_nonce,dummy_nonce,dummy_pid,dummy_pid,7)
% state_source_node(s,im,d,6,x242,x243,dummy_nonce,dummy_nonce,dummy_pid,dummy_pid,3)
%state_destination_node(d,s,im,2,dummy_nat,dummy_nat,dummy_nonce,dummy_nonce,dummy_pid,dummy_pid,3)
% state_intermediate_node(im,s,d,1,dummy_nat,dummy_nat,dummy_nonce,dummy_nonce,dummy_pid,dummy_pid,3)
```

Fig 9: Output of AODV protocol specification verified in AVISPA.

attack found in the protocol specification. Here the intruder shows itself as a valid node and sends and receives the messages with the authorized nodes in the network. Now we discuss the attack found in the specification in more detail.

i -> (s,3) : start

The i, which is an intruder, initiates the session with the source node s, with a session id number 3. A special start message is sent by the intruder to begin the execution of the protocol.

(s,3) -> i : dummy_nonce.dummy_pid.s.0.0.d.

Now when the source node receives a message to begin the session, it start sending the message in the specified format, with nonce and a protocol id values, pid in this case, that are the identity of the messages. The message shows that the source node s, sends the message intended to the destination node d. But all the messages are now passed through the intruder, even if the intruder is not the respective recipient.

i -> (s,3) : dummy_nonce.dummy_pid.d.x242.x242.s.

After receiving the message from the source node, the intruder replies back to the source node without changing the message contents, but showing itself as the valid user as if the message has been sent by the destination node d. The message contents with the value x242, is a variable that relates with the internal working of the OFMC and doesn't affects the normal protocol specification.

This attack trace in the specification shows that the intruder can easily make changes in the respective messages defined by the protocol to send the request and reply for the routes. Hence degrades the performance of the AODV routing protocol under ad hoc routing attacks.

To make the protocol work efficiently and effectively even under the attacks, there must be some security measure that works along with the functioning of the protocol. We applied a security solution against the authentication attack, found in the protocol specification, which uses the signature scheme to encrypt the message formats, such that even if the intruder receives the messages, it cannot able to decrypt and modify the contents and hence the messages receives and sent by the authorized nodes.

5.2 Formal Verification of Proposed Solution for Authentication in AODV

In the previous section we found an attack trace in the routing session of the AODV. The goal specified in the HLPSL specification for the authentication of the RREQ_ID and RREP_ID fails, after verifying the result in the AVISPA. Now we have to make sure about that the message contents cannot be modified by the intruder in the execution of the protocol session.

We incorporate a scheme known as aggregate designated verifier signature scheme (Ag_DVS), described in [12], which is prove to be an efficient authentication scheme for reactive routing protocols. We apply the Ag_DVS scheme in the AODV routing protocol and write down the combined security and AODV routing protocol in HLPSL specification and formally verifies the result in AVISPA.

In Aggregated Designated Verifier Signature scheme, there are two keys that are generated, one for the signer and other for the verifier. The signer key (Ag_DVS.S) is a combination of two keys, a secret key (sk) of the sender and public key of the verifier (pk). Now by combining the both keys, a pair of keys is formed (sk , pk), which is then used by the signer to sign the message. Similarly again the same combination of the keys is formed for the designated verifier and combined with the message, and if the output result is 1, then the verification is correct otherwise not. We use only the signature mechanism of the signer in our protocol verification to check whether the message sign by the signature of the signer prevents the modification of the message contents in the protocol specification or not.

In the formal specification by using the concept of aggregate designated verifier signature scheme for signing the message, a signature is prepared by combining the sender's secret key and the designated receiver's public key. Now this signature is used to sign the RREQ and RREP messages. The node that receives the messages, prepare their own aggregated signature and appends their signature with the message and forwards the message to the next node. The HLPSL specification of signing a message format by the sender S is shown in Fig 10.

```
Ag' := (inv(Ks).Kd)
/\ SND({RREQ.RREQ_ID.S.Dest_Seq_No'.Hope_Count'.D}_Ag')
```

Fig 10: Aggregate signature of the sender to sign the RREQ message.

Ag' is a signature prepared by combining the secret key of the sender S and the public key of the destined node D, which is used to sign the RREQ message and sent. Now when the intermediate node IM receives such message sent by the sender S, it appends its own aggregate signature to the message and forwards the request message, shown in Fig 11.

```
Ag' := inv(Kim).Kd
/\ SND({{RREQ.RREQ_ID.S.Dest_Seq_No'.Hope_Count'.D}_Ag'}_Ag')
```

Fig 11: Aggregate signature of the intermediate node appended to the RREQ message.

The value of Ag' is now changed by combining the secret key of the intermediate node IM and the public key of the designated verifier D. And the new value of Ag' is then appended to the RREQ message received from the sender S.

Again the same goal is specified in the goal section of the HLPSL specification of the AODV, to verify the authentication between the nodes in the AODV routing protocol. The HLPSL specification is then input to the AVISPA, and the verification of the specification is carried out using the same OFMC backend. The output of the given specification of the AODV routing protocol comes out to be safe. This shows that the security mechanism applied in the AODV specification secures the messages and prevent the intruder to modify or tampered the contents of the messages. The output of the verification is shown in Fig 12. Output shows that the verification result has been carried out by the OFMC backend of the AVISPA tool. The SUMMARY of the specification is that the protocol specification provided for the verification, shows that the result is SAFE. All the sessions described in the protocol specification executed completely and the protocol who's IF format is present in the file named aodv(secure).if which is placed in the path given under the PROTOCOL section of output, verifies the goal as specified in the specification of the protocol. The statistics and the backend information are given under their respective sections of the output. Hence, the proposed solution comes out to be helpful in securing the messages of the AODV routing protocol, proved by the formal verification tool AVISPA.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  C:\progra~1\SPAN\testsuite\results\aodv(secure).if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.01s
  searchTime: 0.02s
  visitedNodes: 4 nodes
  depth: 2 plies
```

Fig 12: Output of the proposed solution in AODV protocol specification verified in AVISPA

6. CONCLUSIONS AND FUTURE WORK

The ad hoc routing protocols are the backbone of the ad hoc networks. If ad hoc routing protocols does not perform as they are designed to, then the whole ad hoc network functionality degrades. There are various techniques available to check the ad hoc routing protocols performance before they are used in the real scenarios. Testing and the simulation are the common techniques to investigate the behavior and working of the protocols. One more technique for verification purpose that is parallel to the available techniques is the formal verification.

We use the formal verification technique in our paper to verify the results and the specifications of the protocols. The formal verification tool used is AVISPA that takes the input written in the form of HLPSL specification language. AVISPA is used to verify the security properties of the routing protocols. The AODV routing protocol is taken into consideration for the verification purpose. Due to its low memory consumption and ability to perform better even in the excessive load, AODV is a most widely used ad hoc routing protocol. First the simple AODV is specified in the HLPSL specification language, which is required by the tool to formally verify the working of the AODV protocol. The result carried out by OFMC, backend of the formal verification tool AVISPA.

The result found an attack trace in the protocol specification, and proves the protocol unsafe. Then we incorporate a security feature of aggregate designated verifier scheme in the specification of the AODV protocol and then again use the formal verification tool AVISPA for the verification purpose. The result found out to be safe and hence the incorporated security feature proves to be effective in securing the AODV routing messages.

Our work is a contribution to the work laid out in the field of formal verification. The formal verification is the excellent technique to formally verify the properties or behavior of the particular system and the routing protocols, as in formal verification all the possible states are comes into the verification exhaustively. This increases the reliability and the effectiveness of the verification results. As a future work, it can be extended to the verify the different properties of the protocol and the different tools can be used for this purpose.

7. REFERENCES

[1] A. Sobeih, M. Vishwanathan, D. Marinov, and J.C. Hou, "J-Sim: An integrated environment for simulation and model checking of network protocols", IEEE International conference on Parallel and Distributed Processing Symposium, pp. 1-6, March 2007.

[2] C. Perkins, E. Belding-Royer, S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing", RFC 3561, July 2003.

[3] C. Xiong, T. Murata, and J. Leigh, "An approach for verifying routing protocols in mobile ad hoc networks using Petri Nets",Proceedings of 6th IEEE Circuits and Systems Symposium on Emerging Technologies: Frontiers of Mobile and Wireless Communication, vol. 2, pp. 537-540, May-June 2004.

[4] D. Camara, A.A.F. Loureiro and F.Filali, "Methodology for formal verification of routing protocols for ad hoc wireless networks", IEEE conference on Global Communications, pp. 705-709, November 2007.

[5] D. Benetti, M. Merro and L. Vigano, "Model checking ad hoc network routing protocols: ARAN vs. endairA", 8th IEEE International Conference on Software Engineering and Formal Methods, pp. 191-202, September 2010.

[6] E.M. Royer and C.K. Toh, "A review of current routing protocols for ad hoc mobile wireless networks", IEEE Personal Communications, vol. 6, no. 2, pp. 46-55, April 1999.

[7] K. Bhargavan, D. Obradovic, and C.A. Gunter, "Formal verification of standards for Distance Vector Routing Protocols", Journal of ACM, vol. 49, no. 4, July 2002.

[8] L. Abusalah, A. Khokhar, and M. Guizani, "A survey of secure mobile ad hoc routing protocols", IEEE Communications Surveys and Tutorials, vol. 10, no. 4, pp. 78-93, 2008.

[9] L. Zhou and Z.J. Haas, "Securing ad hoc networks", IEEE Network Journal, November-December 1999, 13(6), pp. 24-30.

[10] M. L. Pura, V. V. Patriciu and I. Bica, "Formal verification of secure ad hoc routing protocols using AVISPA: ARAN case study",Proceedings of 4th European Computing Conference, 2010.

[11] R. de Renesse and A.H. Aghvami, "Formal verification of ad-hoc routing protocols using SPIN model checker", 12th IEEE Mediterranean Electrotechnical Conference, vol. 3, pp. 1177-1182, May 2004.

[12] R. Bhaskar, J. Herranz and F. Laguillaumie, "Efficient authentication for reactive routing protocols", 20th International Conference on Advanced Information Networking and Applications, vol. 2, pp. 57-61, April 2006.

[13] S. Georgoulas, K. Moessner, B. Mcaleer, and R. Tafazolli, "Using formal verification methods and tools for protocol profiling and performance assessment in mobile and wireless environments", 21st Annual IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, pp. 2471-2476, September 2010.

[14] The AVISPA team, "AVISPA v1.1 User Manual", June 2006.

[15] The AVISPA team, "HLPSL Tutorial", June 2006.