IOSR Journal of Computer Engineering (IOSRJCE)
ISSN: 2278-0661 Volume 4, Issue 1 (Sep-Oct. 2012), PP 01-05
www.iosrjournals.org

Formal Modeling and Verification of Trusted OLSR Protocol Using I-SPIN Model Checker

[1]Harpreet Kaur, [2]Amandeep Verma
[1,2]Punjabi University Regional Centre for IT &Mgmt., Mohali, India

Abstract: *An ad hoc network is a momentary network set up by the self-managed nodes that operate and communicate randomly with or without a little support of a network infrastructure. Due to security vulnerabilities ad hoc networks are defenseless against attacks of malicious nodes as the nodes in these networks are not secured by firewalls. In order to enhance the security of conventional OLSR Protocol trust is incorporated as additional security measure in the functioning of the protocol. To validate the improved version of the protocol a technique of formal modeling and verification is used by the utilization of established Model Checker I-SPIN and PROMELA language for validation of Trusted Optimized link state Protocol (TOLSR).*

Keywords: *Ad Hoc Network, Formal modeling and verification, OLSR, Model checker I-SPIN, PROMELA, LTL*

I. Introduction

An ad hoc network is a momentary network set up by the self-managed nodes that operate and communicate randomly with or without a little support of network infrastructure. Routing the packets in the ad hoc network is a vital and crucial task as routing protocols play an important role to redirect the packets and to discover routes through the computer network. Developing new routing protocols and applications for ad hoc networks is a challenging and error prone task due to the unreliable transmission medium, highly dynamic network topology, limited amount of energy and attacks of malicious nodes. The mobility of the nodes makes the network topology dynamic with multi-hop path.

Ad hoc network is dynamic due to frequent changes in topology. The use of dynamic links makes ad hoc network susceptible to attacks. Eavesdroppers can access secret information, thus violating network confidentiality. Hackers can directly attack the network to delete messages, inject erroneous messages, or impersonate a node, which violates availability, integrity, authentication, and non-repudiation. Thus one needs to consider malicious attacks not only from outside but also from within the network from compromised nodes. The conventional cryptographic approaches do not prove an effective measure to defend against threats from malicious nodes [9].

Thus to assuage the effect of these malicious nodes and to achieve the higher level security and reliability the trust based framework is suggested [9]. Trust and reputation of a node on the other node is taken as an interesting criterion to explicit the different types of trust relations before any operation and interaction in the network. The reliability of the protocol must be validated before deployment in order to ensure that it is free from all the errors. Formal modeling and verification is referred as a mathematical-based technique which can be used for specification, development and verification of software system [1]. The use of this technique helps to increase the rigor on the design and development of systems, leading to more reliable products. This approach can help the protocol designers to decrease the development time, find design errors and validate the proposed solutions. In this paper, a error validated in the traditional OLSR Protocol using formal modeling and verification technique is presented and the security issues are investigated, especially with emphasis on providing the secure extension to OLSR, trust is incorporated in the core functioning of the protocol by modifying the model specifications so that this new Trusted OLSR protocol can overcome some of the security threats by malicious nodes.

The paper is organized as follows: The section 2 gives the review of literature. The section 3 gives a general idea of the OLSR protocol and model checker I-SPIN. The section 4 describes the modeling specifications of the protocol using PROMELA language and results of formal verification. The section 5 gives summarizes the work and provides future directions to this work.

II. Review Of Literature

Various research papers have been reviewed from initialization to completion of this work. Many Routing protocols have been proposed and a number of comparative studies have been performed on ad hoc routing protocols using various methods of simulation. A performance analysis of 4 protocols have been done [8] and concluded that OLSR performance is best among all other ad hoc network protocols namely DSR, AODV and GRP. So it is better to implement in the applications of ad hoc network.

There are number of formal approaches and their applications in diverse areas that are suggested for validation and for proving correctness of the models [7]. The formal verification techniques can be applied to all areas of ad hoc network such as authentication, access control, routing etc. A thorough appraisal of the formal verification of ad hoc network routing protocols using the variety of formal specification languages, modeling techniques and verifying tools are specified in earlier work. AVISPA, BAN Logic, Petri nets, SDL, SPIN, PROMELA and UPPAAL are some of the keywords of this study.

To reveal the methodology for formal verification of routing protocols of ad hoc network [1], a case study of OLSR protocol using PROMELA specification language and SPIN model checker is depicted. The OLSR protocol is also a focus of case study in the description of testing methodology for an ad hoc routing protocol using SDL language [4].

The Z/EVES formal verification tool is used for formal modeling and verification of OLSR protocol [9]. In this study trust based framework is proposed and verified in order to improve security of the protocol.

In our approach I-SPIN model checker and PROMELA language is used as a formal modeling and verification technique for Trusted OLSR Protocol. The formal modeling is done using PROMELA language and verification is performed to validate this model using I -SPIN Model checker which is advanced version of SPIN 6.0 and have Graphical User Interface.

III. Olsr And Model Checker Overview

A. THE OLSR PROTOCOL

Optimized Link State Routing (OLSR) is proposed by IETF's MANET Group at 2003. The original version of the protocol is used in this work which is described in protocol RFC [2].

Optimized Link State Routing (OLSR) protocol is a proactive routing protocol with an optimization version of a pure link state protocol. Flooding is done with the help of Multipoint relays. Multipoint Relays (MPRs) are optimized in OLSR where each node must select MPRs from among its neighbor nodes such that a message emitted by a node and repeated by the MPR nodes will be received by all nodes two hops away [2]. MPR selection is performed through the exchange of HELLO messages and TC messages. HELLO messages are exchanged periodically among neighbor nodes, in order to detect links to neighbors, to detect the identity of neighbors and to signal MPR selection. TC messages are periodically flooded to the entire network, in order to signal link state information to all nodes. The malicious nodes introduce the invalid HELLO and TC messages in the network due to which protocol does not function well.

B. I-SPIN MODEL CHECKER

SPIN is a popular open-source software tool [6], used by thousands of people worldwide that can be used for the formal verification of distributed software systems. The tool was developed at Bell Labs in the original UNIX group of the Computing Sciences Research Center, starting in 1980. The software has been available freely since 1991, and continues to evolve to keep pace with new developments in the field. In April 2002 the tool was awarded the prestigious System Software Award for 2001 by the ACM.

SPIN has been used to detect design errors in applications ranging from high-level descriptions of distributed algorithms to detailed code for controlling telephone exchanges [5]. I -SPIN Model checker which is advanced version 6.0 and have Graphical User Interface. I- SPIN accepts design specifications written in the verification language PROMELA (a Process Meta Language), and it accepts correctness claims specified in the syntax of standard Linear Temporal Logic (LTL) as shown the Figure 1.

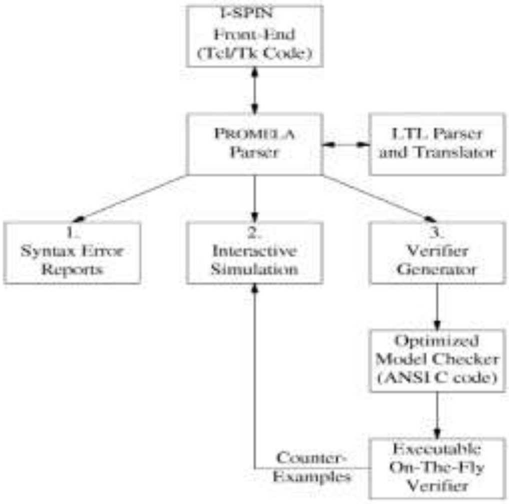

Figure 1: The I-SPIN Model Checker.

SPIN uses a *hash factor* to serve as a predictor coverage function, which is defined as:

$\boldsymbol{H\,f}{=}{-M/N'}$

N' equals to the number of states reached and M equals to the maximum number of storable states.
The corresponding coverage is:

$\boldsymbol{H\,f} > 100 ==+ C > 99.9\%$

$10 < \boldsymbol{H\,f} < 100 ==+ 98\% < C < 99.9\%$

The results of all validations performed in super trace mode are superior to any other type of validation performed within the same physical constraints of the host machine e.g., memory size and speed [6].

IV. Modeling specifications and Verification Results

SPIN is a model checker for the communication protocols by the use of the PROMELA language. The OLSR protocol is initially specified using the basic model provided in the case study [1]. Then its complexity is increased step by step to find the error in the protocol specifications and then the trust factor is added to these specifications in order to enhance the security of the protocol and verified it for different properties such as liveness, deadlock, and safety.

A. Modeling the channel

One of the key aspects of the methodology is the channel modeling. Modeling the channel in a non-deterministic way makes the results more general and it makes possible for the intermediate node to act as a cloud of nodes. This simple assumption greatly decreases the complexity of the models and allows their verification independently of any specific topology.
The channel could be modeled in PROMELA as shown in the Figure 2.

```
mtype={Origin,Intermediate,Destination};
 /*it describes whether the node is
origin,destination or intermediate*/

mtype={Hello,TopologyControl,Datapacket}/*
it gives the type of msg*/

chan medium = [nnodes] of {mtype, byte, byte,
bool, bool};
/*this is the chan declaration*/
```

Figure 2: Specifications for modeling the channel in PROMELA.

However, the most important aspect of the channel is the way nodes receive messages from it. The choice must be random and any node should be able to receive the message at any time. Thus, when two nodes want to communicate they use one predetermined channel.

B. Creating The Model

In this research work firstly modeling of the basic model or Level 1 of OLSR protocol functioning is done and then level 2 model is verified by increasing the functioning of protocol in the network. No error is validated in these two basic models. Then the complexity of model is increased to a higher level 3 by specifying the message format as shown in the Figure 3 and validating this model for an error.

```
chanmedium=[nnodes]of
{mtype,byte,byte,bool};
type={Hello,TopologyControl,DataPacket};
mtype={Origin,Intermediate,Destination};
bool packt=false; bool test=true;byte myid;
```

Figure 3: Specifications of Message format in the PROMELA.

The error is validated in the traditional protocol. Every time a message is send from origin node to destination node, the MPR entries are checked by the origin node and also the table entries are updated. In this model a claim is done that the error exists in the OLSR protocol specification. When the model is verified a design flaw regarding the security in the specification of the OLSR protocol is verified. The error claimed is that any of malicious nodes can declare itself as a MPR node and can drop messages without sending it to the destination when the recalculation of these table entries done to update these tables as at this time no route is available to the node that wants to communicate.

So, in order to make the protocol secure from malicious nodes a new TOLSR model is specified and verified in which we have added the trust factor in the specifications of the protocol which is considered as a criteria by nodes to communicate in the network. Whenever nodes have to communicate their trust variable value will be considered.

```
byte trust;*/trust variable/*
byte origin_t;*/origin trust/*
byte dest_t;*/destination trust/*
if
{
::true->trust=1;
::true->trust=2;
::true->trust=3;
.::true->trust=9;
::true->trust=10;
}
```

Figure 4: Adding Trust variable in the specifications of OLSR Model

The Trust can be any value from 0 to 10 where 0 indicates complete distrust and 10 signifies the full trust in the node. Using if loop this trust factor can be assigned a random value and consider all the cases as shown in the Figure 4.

After this trust model is validated it is also increased to a higher trust level 2 in which indirect trust relations are incorporated. Indirect trust means the source node decision to send a packet does not depends only upon the trust level of that node on destination node but also on the other trust relations among the nodes. The trust variables used for adding the additional trust framework for providing a higher level security are shown in the Figure 5.

```
byte trust;
byte origin_intr_t; //trust value of  the origin node
on the intermediate node
byte intr_dest_t;//trust value of intermediate node
on the destination node
byte origin_dest_t;//trust of origin on the
destination
```

Figure 5: Indirect trust variables added in the TOLSR model.

C. Verification Results

The verification of the model is done using I-SPIN model checker in order to verify the error which is found in the protocol specifications of the level 3 protocol and to validate the proposed solution of incorporating the trust factor in the basic trust model and level 2 TOLSR model of the protocol. These models all verified for the properties such as safety, liveness and deadlock using the bit state mode of the model checker which verifies it for all the cases without space state explosion problem.

The results for all the models specified and validated under I-SPIN model checker are tabulated in the Table 1. This table shows the Depth reached Total number of transitions, size and memory used for states for each OLSR model that is modeled and verified in this study.

Table 1: OLSR Verification Results for all Models.

Implementation	Depth reached	States Stored	Transitions	Memory usage (MB)
Basic OLSR Model	3	4	4	0.003
OLSR Level 2 Model	10	55	67	0.002
OLSR Level 3 Model	18	637	637	0.024
Trust Based OLSR Model	22	785	1025	3.442
Trust based Level 2 Model	83	364202	584017	27.786

V. CONCLUSION

This paper presented the specifications of TOLSR model which is enhancement of basic OLSR protocol by incorporating the trust factor in the functioning of the protocol. This model is validated for direct as well as indirect trust relations between the nodes of the network. As a result it is proved that with this technique of Formal Modeling and Verification the memory usage increases as protocol complexity increases but a more secure protocol can be modeled in short time. Thus use of formal modeling and verification technique leads to more reliable protocol and is a promising technique for modeling and validation of ad hoc network routing protocols. The future work of this study is extension of this trust relation to higher level and to compare it using other formal methods technique such as equivalence checking.

Refrences

[1] Camara, D., Loureiro, A. A. F., and Filali F., Methodology for Formal Verification of Routing Protocols for Ad Hoc Wireless Networks, *In Proceedings of the IEEE Conference on Global Communications,* pp. 705 – 709. 2007.

[2] Clausen, T. Ed., Jacquet, P. Ed., Optimized Link State Routing Protocol (OLSR), *IETF INTERNET DRAFT,* RFC 3626, 2003.

[3] Hafslund A., Tonnesen A., Rotwik R., Jon A., Kure O., Secure Extension to the OLSR Protocol, *OLSR Intershop and Workshop,* 2004.

[4] Maag S., Zaidi F., Testing Methodology for an Ad Hoc Routing Protocol, *IEEE Conference on Testing of Ad hoc Networks,* October 6,2006.

[5] R. de Renesse, A.H. Aghvami, Formal Verification of Ad hoc routing Protocols Using SPIN model Checker, *IEEE MELCON*, May 12-15, 2004.

[6] SPIN Manual available at: http://www.spinroot.com.

[7] Verma, A., Formal Verification of Ad Hoc Network Routing Protocols, *International Journal of Advanced Research in Computer Science*, Vol 2, No. 4, pp. 526 -530, July-August 2011.

[8] Verma, A. and Gujral, M S , Performance Analysis of Routing Protocols for Ad hoc Networks, *International Journal of Computer Science and Emerging Technologies*, , Vol 2, No. 4, pp. 484 – 487 , August 2011.

[9] Verma a. and Gujral M S, Formal Specifications of Trusted OLSR Protocol of Ad hoc Network in Z, *International Journal of Computer Applications*, Vol 37-NO.2, Janurary 2012.