Poster Abstract: On the Implementation Code of the Secure Mesh Routing Protocol PASER in OMNeT++: The Big Picture

Mohamad Sbeiti
Communication Networks Institute (CNI)
Faculty of electrical Engineering and Information Technology
Dortmund University of Technology
44221, Dortmund
mohamad.sbeiti@tu-dortmund.de

Christian Wietfeld
Communication Networks Institute (CNI)
Faculty of electrical Engineering and Information Technology
Dortmund University of Technology
44221, Dortmund
christian.wietfeld@tu-dortmund.de

ABSTRACT

The Position Aware Secure and Efficient Mesh Routing Protocol (PASER) aims to efficiently establish accurate routes in terms of legitimate mesh nodes in wireless mesh networks in presence of external attackers. This poster gives an overview of the implementation code of PASER in OMNeT++, the modules it incorporates and their interaction. Striving for code validation and security verification, PASER is first evaluated in comparison to the well-known routing protocols HWMP, OLSR and BATMAN in simulation. Afterwards, the OMNeT++ code of PASER is ported to Linux and the simulation results are validated by experimental measurements. The results show that PASER achieves a reasonable trade-off between security and performance in the evaluated scenario.

Categories and Subject Descriptors

C.2.0 [**Computer-Communication Networks**]: Security and Protection; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing Protocols*

1. INTRODUCTION

Wireless mesh networks (WMNs) have recently become a promising technology to establish a high-performance and low-cost network anywhere anytime without the need for an existing infrastructure. To establish WMNs, routing protocols are necessary to discover and maintain routes on the fly between all network nodes. The latter makes WMNs prone to a new type of attacks [4], e.g., the wormhole attack. A pair of malicious nodes linked via a fast transmission path (e.g., Ethernet or directional antenna) forward route discovery messages faster than legitimate nodes. This causes victim nodes to always use the tunneled route to transmit their data packets, which are then dropped by the attacker. Even if the network is protected via conventional cryptosystems e.g., IEE802.11i-PSK, this attack still succeeds. The main reason for this is that routing messages are simply forwarded, without any changes, from one end to the other end of the tunnel.

Thus, without a satisfactory level of security, end users or organizations lack motivation to utilize the WMNs technology. Otherwise, malicious users, terrorists or benefiting organizations might easily disrupt the communication channel. To address this issue, many approaches to secure routing in WMNs have been recently proposed, see [3]. However, none of these protocols has been adopted in the practice. Besides, to the best of the authors' knowledge, none of these protocols have a running implementation in OMNeT++ [6] or other user-friendly simulation environments, which could have helped to enhance these protocols. The high overhead of the security mechanisms of these protocols or the hard assumptions taken by their design burdened their deployment in real life applications. For this end, PASER has been proposed, to achieve a reasonable trade-off between security and performance and this poster highlight the implementation design of PASER in OMNeT++.

2. OVERVIEW OF PASER: THE POSITION AWARE SECURE AND EFFICIENT MESH ROUTING PROTOCOL

PASER is a secure routing protocol tailored for wireless mesh networks. In order to secure the network against routing attacks such as replay, blackhole and wormhole attacks, it fulfills the following goals: *Node authentication*, *message freshness and integrity*, and *neighbor transmissions authentication*. The protocol is briefly described in the following. A detailed description of the protocol is provided in [2].

2.1 Routing

PASER is a hierarchical reactive routing protocol. It distinguishes between gateways and mesh routers/access points. The latter are always responsible to register themselves once at a gateway in order to join the network. Gateways are typically connected to a key distribution center that manages the network keys and certificates. During the registration process, new neighbors always establish a trust relationship between each other. Afterwards, they mainly use unicast messages to communicate over trusted routes. To maintain

OMNeT Workshop 2013, March 05-07

DOI 10.4108/icst.simutools.2013.251690

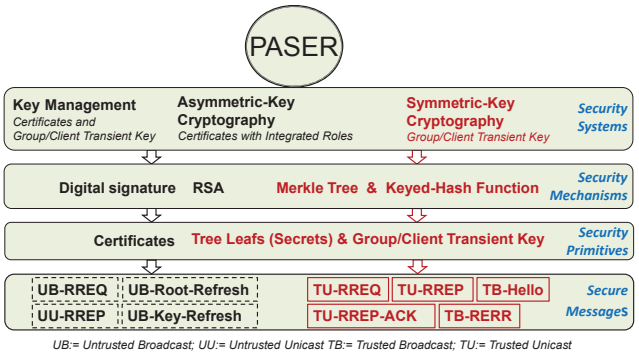

Figure 1: Overview of the security of PASER.

established routes and to detect and react on broken links, apart from specific timeouts defined for an existing route, a node deletes a route in two cases: In case of a Hello message based failure detection or a link layer based failure detection. While the link layer feedback enables the fast reaction of PASER on route breaks in case of active data transfer, Hello messages allow the detection of route changes also in case of no data transfer.

2.2 Security

PASER combines digital signature with lightweight Merkle tree and a keyed hash function to secure the routing messages. Apart from that, to address the problem of node compromise, PASER endorses a key revocation scheme to exclude those nodes in a fast and efficient way. The main building blocks of PASER with respect to security are depicted in Figure 1.

3. IMPLEMENTATION DESIGN

Bearing in mind that the final target beyond the simulation of PASER is its deployment in real life scenarios, the PASER implementation in OMNeT++ have had to fulfill the following goals in the order of priority:

1. Performance evaluation of the protocol and design optimization if necessary
2. Verification of the protocol robustness against well-known attacks and design optimization if necessary
3. Portable code

To achieve these goals, the PASER code has been divided into sub-modules as depicted in Figure 2, where each sub-module is responsible for a given task. This modular design allowed low-effort protocol optimizations. Besides, apart from small functions, OMNeT++ specific code is mainly kept in the 'Socket' sub-modules. The latter enabled a straightforward porting of the rest of the code to Linux. As Figure 2 shows, the PASER code in OMNeT++ is mainly composed of:

- PASER Logic
  - **Route Discovery:** It is a set of functions that manage a node registration at a gateway and handle a route discovery.

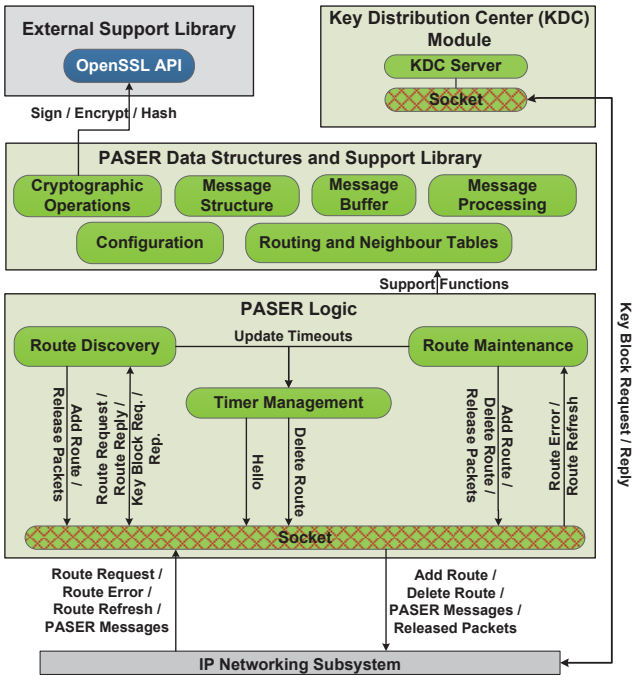

Figure 2: The big picture of the PASER implementation in OMNeT++.

  - **Route Maintenance:** It provides functions to keep routes up-to-date. It manages several PASER timers and the link layer feedback.
  - **Timer Management:** It manages all PASER timers.
  - **Socket**: it comprises the handleMessage function that is called upon receipt of in/outcoming messages, i.e., it is the interface to the outside world.
- PASER Data Structures and Support Library
  - **Cryptographic Operations:** It handles all security related operations. This sub-module depends on the external library OpenSSL [1].
  - **Message Structure:** It comprises the classes of all the PASER messages, which are illustrated in Figure 1 - bottom.
  - **Message Buffer:** It manages a buffer of all data packets that must be forwarded to an unknown destination.
  - **Message Processing:** It provides functions to manage all PASER messages.
  - **Configuration:** It comprises a parser of the PASER configuration parameters in the NED file. It also comprises additional PASER configuration parameters for developers.
  - **Routing and Neighbour Tables:** It manages the PASER tables.
- Key Distribution Center Module
  This module handles the key management in PASER. Nodes contact this module in order to get the required keys to access the network.

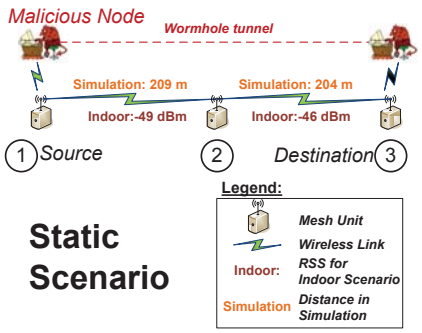

Figure 3: Network setup for simulation and experimental measurements.

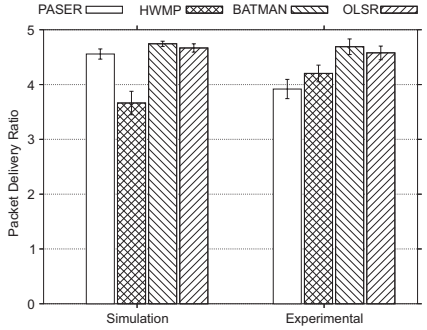

Figure 4: Packet delivery ratio in case there is no wormhole.

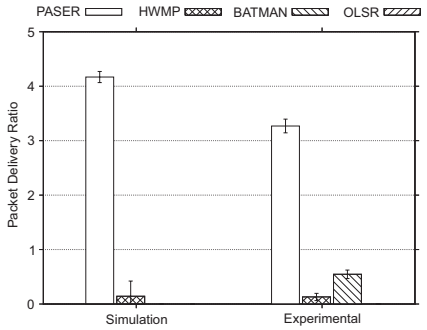

Figure 5: Packet delivery ratio in case of an activated wormhole attack.

Table 1: Simulation Parameters

Parameter	Value
Mac Layer	802.11g
Antenna Type	Omni
Transmission Range [m]	250
Channel Model	Nakagami m=9
Traffic Model	CBR-UDP
Data Rate [Mbit/s]	5
Packet Size	512 Bytes
Buffer Size [Packets]	100
Number of Runs	10
Simulation Time [s]	100

4. IMPLMENETATION VALIDATION

In this section, the evaluation of PASER and the well-known routing protocols HWMP, BATMAN and OLSR in OMNeT++ as well as in a real testbed is covered. Hereby, a small network is considered as illustrated in Figure 3. First, the performance of the protocols is analyzed in a case of no wormhole attack. Second, wormhole attack is activated and its impact and the behaviour of the protocols are investigated.

4.1 Simulation Configuration

The code of PASER in OMNeT++ and the INETMANET framework are used for the evaluation of PASER, HWMP, BATMAN and OLSR in simulation. Table I illustrates the network and traffic models used. The protocols are configured by their default parameters.

4.2 Experimental Testbed

The mesh nodes used in the experimental testbed are based on the Roboard RB110 (Vortex86 CPU running at GHz with a 256 MB DRAM). The latter is equipped with a DNMA92 miniPCI WLAN device operating according to the IEEE802.11g standard in channel 13. The WLAN device incorporates two omni-directional antennas with a gain of 5 dBi. As for the software configuration, we have used Debian Wheezy with the 3.4.5 Linux kernel. The following versions of the protocols have been installed: PASER 0.1.b, HWMP compatwireless 3.5.snpc (ath9k), BATMAN 0.3.2-12 and OLSRd 0.6.3. The protocols configuration is the default one. The measurements are run using Iperf-UDP from mesh unit 1 serving as the client to mesh unit 3, which hosts the server. Ten repetitions for each measurement are run in average. The measurements are performed indoor in an office environment. All nodes have an altitude of 85 cm and we limited the transmit power of sender and destination from 20 dBm (default) to 4 dBm in order to enforce the desired topology.

4.3 Results

The results of the network evaluation with respect to packet delivery ratio in case of no wormhole are depicted in Figure 4. The latter shows that BATMAN and OLSR perform similar in the practice and in simulation. In case of PASER, the experimental results are a bit lower than in simulation. This is explained by the higher collision probability of the large-sized PASER messages in the practice due to more interference sources. In case of HWMP, PDR in simulation is unexpectedly lower than in the practice. In contrast to simulation where a node drops a packet after 7 failed retransmissions, in the practice the node tries to retransmit the packet using CTS/RTS after 7 retransmission failures. This explains the lower PDR of HWMP in simulation since dropping the packet causes HWMP to delete the route due the its link layer feedback mechanism, which then leads to lower PDR. The results of the evaluation of the protocols in case of wormhole are depicted in Figure 5. They clearly show that PASER, in contrast to its counterparts, is robust against wormhole attack in simulation as well as in the practice. The general behaviour of the protocols in presence of this attack is described in more details in [5].

5. CONCLUSION

In this work, we elaborated the implementation code of PASER in OMNeT++. Especially, its modular design and splitting OMNeT++ specific code from the rest of code are noteworthy. To validate the implementation of PASER and to highlight the robustness of the protocol against routing attacks such as wormhole, the protocol is evaluated in simulation as well as in a real testbed in comparison to other well-known protocols. The experimental results validate the PASER code in simulation but with respect to the impact of PASER's large-sized messages. Simulation and Experimental results show that PASER protects the network against the wormhole attack.

6. ACKNOWLEDGMENT

The authors would like to thank Eugen Paul for his technical support. Our work has been conducted within the AVIGLE project, which is co-funded by the German federal state North RhineWestphalia (NRW) and the European Union (European Regional Development Fund: Investing In Your Future).

7. REFERENCES

[1] OpenSSL http://www.openssl.org/, Nov. 2012.

[2] Postion Aware Secure and Efficient Mesh Routing Protocol (PASER) http://www.paser.info/, Nov. 2012.

[3] L. Abusalah, A. Khokhar, and M. Guizani. A Survey of Secure Mobile Ad hoc Routing Protocols. *IEEE Communications Surveys and Tutorials*, 10(4):78–93, Fourth Quarter 2008.

[4] B. Kannhavong, H. Nakayama, Y. Nemoto, N. Kato, and A. Jamalipour. A survey of routing attacks in mobile ad hoc networks. *IEEE Wireless Communications*, 14(5):85–91, Oct. 2007.

[5] M. Sbeiti, J. Pojda, and C. Wietfeld. Performance Evaluation of PASER - an Efficient Secure Route Discovery Approach for Wireless Mesh Networks. In *IEEE PIMRC*, Sydney, Australia, Sep. 2012.

[6] A. Varga and R. Hornig. An overview of the OMNeT++ simulation environment. In *SIMUTools*, Marseille, France, Feb. 2008.