A New Evaluation Model for Security Protocols

Chao YANG, Jianfeng MA, Xuewen DONG
Key Laboratory of Computer Networks and Information Security, Ministry of Education, Xidian University, Xi’an 710071, China;
School of Computer Science and Technology, Xidian University, Xi’an 710071, China

Abstract—Till today, the study of performance of security protocols of WLAN has been one of research focuses. Whereas, owing to enormous complexity and low efficiency of modeling security protocols, there has been by now no uniform method or technology that can be used generally to simulate and evaluate security protocols, and no simulation system available that can well support research on performance of security protocols. So in view of the status quo, we, in this paper, mainly propose a novel security protocol simulation architecture and a simulation extending method for simulating security protocols of WLAN, and then set up a simulation platform for modeling such protocols based on OPNET, a famous simulation software. Finally, with an instance of extending and simulating a security protocol of WLAN, the validity and the universality of the simulation architecture and the extending method as well as the feasibility and correctness of the simulation platform are demonstrated.

I. INTRODUCTION

Security in wireless network, such as WLAN, is a critical concern. More and more scholars focus on the study of WLAN security protocols. However, how to evaluate security protocols in terms of security and performance is an important and tough task. There are three common approaches which can be exploited to carry out evaluation of security protocols—testing, mathematical analysis and simulation. Due to lower costs and maneuverability, the simulation approach has been widely used. However, the simulation approach has some disadvantages in terms of efficiency and extensibility. To the best of our knowledge, there exists no unified simulation approach on how to evaluate WLAN security protocols, because of the complexity and inefficiency of security protocols modeling. Moreover, popular simulation tools, such as OPNET and NS2, do not have built-in security protocol models or universal interfaces to support the evaluation of WLAN security protocols, which leads to lots of repetitive works and low comparability of different performance researches.

To address the problem mentioned above, we propose, in this paper, a universal and extensible security protocol simulation platform for WLAN. First of all, the architecture design of the platform is presented. One of the key features of it is the ability to provide a universal interface to incorporate various security protocols into the simulation platform conveniently and exactly. Furthermore, a general method for security protocols extension is demonstrated. It specifies how to modify the protocol module and embed it into the simulation platform, which standardize the extension process. Finally, according to the proposed architecture and method, we implement the simulation platform in OPNET and validate its feasibility, correctness and universality through extending and simulating WEP and EAP-TLS security protocols. The results show that our simulation platform has characteristics of universality and extensibility.

The rest of this paper is organized as follows. Section II discusses the related work of security protocol simulation. Section III designs the architecture of the simulation platform for WLAN security protocols. Section IV provides a detailed description of the simulation interface and the core components of the architecture. Section V describes the method for security protocols extension. In section VI, we present a simulation experiment for the validation of the proposed platform. Finally, Section VII concludes this paper.

II. RELATED WORK

There are two lines of researches focusing on security protocol simulation in the research community.

The first one focus on how to utilize various kinds of methods and tools to simulate and evaluate WLAN security protocols, and improve them through the analysis of the simulation results. This kind of research is extremely popular, such as [9] [10] [11] and so on.

The second one concentrates on how to develop a simulation platform or framework to standardize the simulation and evaluation of security protocols, which is very useful in improving the efficiency and extensibility of security protocol simulation. However, only a few research works, described below, have been reported on this topic.

Zhao et al. established a simulation framework to model BGP security protocols and PKI system [12]. With this framework, they evaluated certification path building...
algorithms and improved the performance of the algorithms. However, the main consideration of this paper focuses on how to use the framework to evaluate the performance of security protocols, which is similar to that of the first kind of research focus. In fact, the proposed framework cannot serve as a universal simulation platform for certain sort of security protocols.

Yang et al. proposed a model to simulate authentication protocols at application layer of mobile ad hoc network based on OPNET simulation environment[13]. The main idea of their work is seemingly the same as ours. However, this paper does not present a universal and extensible security protocol simulation platform but a concrete process model in OPNET. The feasibility and correctness of the process model is also not validated. Furthermore, the simulation model proposed in their paper is built on the application layer, which not only falls short of reality, but also fails to provide an extensible and unified security protocol simulation interface.

Yang et al. constructs a node model to implement IEEE802.1x authentication function in OPNET[14]. However, their paper focuses on designing a special simulation model to simulate a concrete protocol, rather than designing a universal simulation platform. Because of its special interface to the simulation tools, the model proposed in their paper lacks universality and reusability. Furthermore, due to its non-standard design method, the results conducted by this model have low comparability.

III. THE ARCHITECTURE

This section presents the architecture of the simulation platform. The designing ideas of the architecture are as follows. Although all kinds of WLAN security protocols are distinctly different in terms of data field, exchange process, encryption algorithm and so on, they perform main functions, such as authenticating, port controlling and framing, all on the data link layer of the OSI architecture. To satisfy the needs of incorporating various security protocols into the simulation platform conveniently and exactly, an extensible and standard security protocol simulation interface is designed between data link layer and network layer, and divided into several function modules to manage data stream, identify and schedule protocols. It makes the simulation architecture a universal, highly configurable and practical solution to simulate WLAN security protocols.

As illustrated in Figure 1, the architecture has the following functional components.

Security Protocol Modeling

This component is responsible for modeling the simulated protocols. Due to the complexity of security protocols, we need to simplify them and abstract their exchange processes in order to incorporate them into the simulation platform. Furthermore, correlative parameters of security protocols should also be represented in appropriate forms so that they can be correctly coded. The exact parameters and abstract process of protocols are necessary in simulation. Finally, basic algorithms of security protocols, such as encryption algorithm, should also be implemented in this component.

Protocol Simulation

This component is the core of the simulation architecture.

In this component, an extensible and standard security protocol simulation interface is proposed and integrated into the data link layer of WLAN nodes. The interface consists of several sub-components, including Data Stream Management Module (DSMM), Multi-Protocol Extension Slot (MPES) and so on. As one of the most important components, the interface can be used to distinguish between normal data stream and authentication stream, guide them into right modules, load and initialize new security protocol modules, and encapsulate security protocol packets, which plays a key role in extending various security protocols. The detail of simulation interface will be described in section IV.

Simulation is based on the specific scenario with different network topologies, link characteristics and node mobility models. Based on the scenario settings, WLAN nodes, mainly supported by physical layer and data link layer, communicate with each other by the wireless channel. All nodes are independently managed and controlled by a core module. The generator module generates a set of traffic packets; the debugger module is used to test and debug the simulation program; the output module exports and displays the simulation information at any moment. All the behaviors of nodes make up of an integrated simulation environment as well as the actual WLAN.

Evaluation and Optimization

The evaluation component is responsible for evaluating security protocol performance according to the simulation results. The appropriate performance metrics, such as
complexity of encryption algorithms, packet loss rate, communication overhead, throughput, handover delay and so on, should be determined initially. After the evaluation is completed, it is required to determine which component should be modified so as to obtain optimal performance. This is the main purpose of the optimization module.

IV. SIMULATION INTERFACE

As the key component of the architecture, the security protocol simulation interface can be exploited to embed various security protocols into the simulation platform conveniently and exactly. Based on the layered OSI architecture, the simulation interface is implemented between data link layer and network layer and divided into three function modules: Data Stream Management Module (DSMM), Security Protocol Identifying and Scheduling Module (SPISM), and Multi-Protocol Extension Slot (MPES). These modules are responsible for managing data stream, identifying and scheduling protocol, and extending various security protocols, respectively. The function modules and their relationship are illustrated in Figure 2, in which the solid line with arrowhead and the dashed line with arrowhead denote the direction of authentication and normal data stream respectively.

A. Data Stream Management Module (DSMM)

The DSMM is mainly responsible for identifying, managing, encrypting and decrypting data stream. After entering the simulation interface, data streams is intercepted by the DSMM. Then, according to the interface control information, data streams will be divided into two classes—authentication data stream and normal data stream, and be guided into right modules. Furthermore, a Session Key Cache providing keys used to encrypt data stream, and an Authenticated Station List storing the authenticated station information are built in the DSMM.

The control flow chart of the DSMM is depicted in Figure 3. The solid line with arrowhead and dashed line with arrowhead denote the authentication data stream and normal data stream respectively.

B. Security Protocol Identifying and Scheduling Module (SPISM)

The SPISM is mainly responsible for identifying the type of authentication protocols and scheduling the corresponding security protocol module. When a node joins a new network, the SPISM will identify the type of the authentication protocol in the current network and select the scheduling algorithm to wake up the corresponding authentication protocol module, which will in turn establish a security association between the node and the authentication server. To realize the function above, the SPISM has a sub-module, Security Protocol Type Code Cache, in which the type of security protocols is registered during the initialization phase.

The control flow chart of the SPISM is depicted in Figure 4. The solid line with arrowhead and dashed line with arrowhead denote the authentication data stream and normal data stream respectively.
with arrowhead denote the authentication data streams and initial data streams respectively.

C. Multi-Protocol Expansion Slot (MPES)

The MPES mainly responsible for adding, registering, managing and deleting various security protocols through a built-in Security Protocol Management Interface sub-module. It also has a sub-module of Security Protocol Register Data Base to store correlative information of all kinds of security protocols and the mapping relationship between these security protocols and their type codes. Furthermore, the “plug and play” function of security protocol modules is realized with the help of the sub-modules.

The control flow chart of the MPES is depicted in Figure 5. The solid line with arrowhead and dashed line with arrowhead denote the authentication data stream and register data stream respectively.

V. Extension Method of Security Protocol

This section proposes a general method of how to embed various security protocols into the simulation platform, which standardize the extension process. The main designing idea is that based on the functions provided by the simulation interface, the extension method modifies the security protocol module in terms of frame format, management interface and protocol address, and must be simple and universal enough to embed new protocols into the simulation platform. In detail, the extension method includes three manipulations:

**Designing Unified Frame Format (UFF):** A unified frame format to encapsulate all sorts of security protocols is defined. Through the standard frame format, the simulation platform could support different types of security protocols without amending the way function modules deal with different security protocol frames.

**Designing Interface Control Unit (ICU):** A standard ICU between the security protocol modules and other modules is definitely defined. It provides a unified interface for function modules to exchange management information such as loading, deleting and configuration information.

**Designing Service Access Point (SAP):** Security protocol modules are marked with protocol address called SAP and registered in a database, which provides a convenient method for other modules to search for and communicate with them.

The extended security protocol modules are shown in Figure 6.

![Figure 6 Extended Security Protocol Modules.](imageURL)

The workflow to add a security protocol to the simulation platform is illustrated in Figure 7.

**Step 1:** Design the unified frame format to encapsulate all kinds of security protocols, as shown in Figure 8.

![Figure 7 workflow of Extending Security Protocol.](imageURL)

The type of protocol denotes the sort of security protocols; the address of module denotes the SAP of security protocols; the frame number field includes the serial number of frames; the control field is used to implement certain control functions; raw data field carries the original security protocol packets.

![Figure 8 Unified Frame Format.](imageURL)
Step 2: As illustrated in Figure 9, design the standard ICU between security protocol modules and other modules for exchange of management information.

Figure 9 Interface Control Unit.

The management interface denotes the address through which the management frame could be sent and received; the status code denotes the current state of management; the operation code is used to identify different management operation, such as adding or deleting operation; the registration info denotes the protocol name, type and address; the result field is used to inform the results of operations.

Step 3: Mark security protocol modules with designed ASP and register them in a database.

Step 4: Code the security protocol module, configure its data stream port and connect it with the Security Protocol Simulation Interface.

VI. PLATFORM VALIDATION

In this section, we implement the simulation platform for WLAN security protocols in OPNET and demonstrate its correctness, feasibility and universality through extending and simulating two concrete security protocols—WEP and EAP-TLS.

1) Platform implementation

First, we modify the OPNET built-in modules and implement the new protocol modules in the process layer of OPNET. These modifications and implementations mainly include three parts denoted by 1, 2 and 3 in Figure 10. Part 1: modify the data stream management function in Wireless_Lan_Mac module to control the security protocol data stream; Part 2: code the Security Protocol Simulation Interface and their sub-modules to fulfill the function of identifying, scheduling and managing security protocols; Part 3: write programs to implement the Protocol module to be tested. These new added and modified modules correspond to the DSMM, SPISM and MPES modules respectively.

Second, we add the Security Protocol Simulation Interface module and the Protocol module to the WLAN nodes in OPNET and define the logical relationship between these modules, as shown in Figure 10.

2) Evaluation criteria

To exactly evaluate the simulation platform, three evaluation criteria are established:

Feasibility: If a simulation platform could accept input by rule, successfully perform simulation process, output right results and fulfill functions of modules, it can be considered to be feasible.

Correctness: we can compare the results of a simulation platform with actual test results. If the difference between the two results is slight and in an acceptable rang, the simulation platform can be considered to be correct.

Universality: If various kinds of protocol modules could be added by a unified extension interface to a simulation platform without modifying other communication modules, the simulation platform can be considered to be universal.

3) Validation of Feasibility

A. Protocol extension

First, we code the protocol module of WEP in OPNET according to the standard IEEE 802.11[15]. Second, we extend and embed it into the simulation platform. Finally, we debug and install it.

B. Simulation scenarios

A typical IEEE 802.11 network, illustrated in Figure 15, is set up. This network consists of two Basic Service Sets(BSSs). BSS1 is composed of Station A, Station B and AP_A; BSS2 is composed of Station C and AP_B. Two BSSs are connected by Bridge forming an ESS. In detail, Station B keeps on sending data packets to Station A and Station C, and hands off periodically between the two BSSs during first 20 minutes of simulation. In this network, all wireless nodes adopt the standard 802.11a[15] and four scenarios are developed.

Scenario 1 is named authentication_1, in which each node has an authentication function. The pre-shared key between Station B and AP_A is identical; and another pair of keys between Station B and AP_B is the same too.
Scenario 2 is named authentication_2, in which each node has an authentication function. The pre-shared key between Station B and AP_A is identical; however, another pair of keys between Station B and AP_B is not the same.

Scenario 3 is named authentication_3, in which each node has an authentication function. The pre-shared key between Station B and AP_A differs; and another pair of keys between Station B and AP_B is not the same either.

Scenario 4 is named non_authentication, in which all nodes consist of OPNET built-in modules that do not have an authentication function.

C. Simulation and collection of statistics

The task of simulation utilizes the optimized simulation kernel and continues for 30 minutes. During simulation, four kinds of statistics are collected: rate of dropping data packets of Station B, average receiving rate of data packets of Station A, throughput of MAC module of Station A, overall transmission delay of data packets from Station B to Station A, all of which are illustrated in Figure 12-15 respectively.

D. Analysis

According to the proposed simulation architecture, the simulation platform for WLAN security protocols, based on OPNET, is implemented. Meanwhile, under the direction of the proposed extension method, the WEP security protocol
is embedded into our platform and the corresponding simulation statistics and results are obtained. It is demonstrated that the architecture and method have practical significance and can be referred as a principle to set up security protocol simulation platforms. Furthermore, it is also validated that the simulation platform possesses certain characteristic of feasibility. Detailed analysis is as follows.

Figure 12 shows the packet drop rate of Station B in each Sub-Scenario. The results are arranged in descending order: Sub-Scenario3, Sub-Scenario2, Sub-Scenario1, and Sub-Scenario4 (does not have dropped packets). Because Station B, in Sub-Scenario 3, can not succeed in authenticating with any AP for different session key pre-shared between them, it discards all the data packets, and its packet drop rate is highest. In contrast, Station B in Sub-Scenario 4 does not drop any data packets, because it does not have the authentication function. In Sub-Scenario 1, Station B successfully completes the authentication with both APs, and sends and receives data packets normally. Therefore, the number of data packets it drops is the smallest. In addition, as illustrated in Figure 12, four peaks are appeared in Sub-Scenario1, 2, 3, respectively. This is because that Station B hands off four times during the first 20 minutes simulation time and then requires re-authenticating with the corresponding AP.

As seen from Figure 13 and Figure 14, the average data receiving rate and throughput of Station A in Sub-Scenario 3 are zero, which demonstrates that it does not receive any data packets under these conditions. On the contrary, Station A can receive data packets in Sub-Scenario 1 and 4. The reason is that Station B cannot send out any data packets in Sub-Scenario 3 due to the failure in the authentication. However, in Sub-Scenario 1 and 4, the data packets sent from Station B could be received by Station A. Furthermore, the average data receiving rate and throughput of Station A in Sub-Scenario 1 are nearly the same with and yet a little lower than those in Sub-Scenario 4. This is because Station B, in Sub-Scenario 1, will drop some normal data packets during the authentication procedure, whereas it will not perform authentication in Sub-Scenario 4.

Figure 15 shows that the transmission delay of data packets in Sub-Scenario 1 is far larger than that in Sub-scenario 3 during the initial phase of simulation, and that the difference between them dwindles to a relatively constant afterwards. The reason is that more time will be spent at the beginning of simulation due to the authentication procedure. Afterwards, because of the process of encryption and decryption, the overall transmission delay of data packets in Sub-Scenario 1, during the next simulation time, is larger than that in Sub-scenario 4; and their difference is almost constant.

In conclusion, with WEP embedded into the simulation platform, it can accept input by rule, successfully perform simulation process, and output reasonable results, which clearly demonstrate that the simulation platform proposed possesses the characteristics of feasibility.

4) Validation of Correctness

We plan to test the WLAN security protocol EAP-TLS in a practical test-bed we established, and obtain relative results which would be compared with the simulation results in order to validate the correctness of the proposed simulation platform.

A. Protocol extension

First, we code the protocol module of EAP-TLS in OPNET and in the practical test-bed respectively. Second, we extend and embed it into the simulation platform and the test-bed. Finally, we debug and install it.

B. Simulation scenarios

A typical IEEE 802.11 network, illustrated in Figure 16, is set up. This network consists of eight Stations, two APs, two Bridges and one AS which is connected with AP1 and AP2 by wire; all wireless nodes adopt the standard 802.11 and EAP-TLS security protocol. In detail, Station1 is supplicant, AP1 is authenticator and AS is authentication server; Station1 initiates authentication request retransmitted to AS by AP1, and perform standard authentication process with AS. The Station1’s attribute is shown in Figure 17.
C. Practical test-bed

As illustrated in Figure 18, a practical WLAN security protocol test-bed is set up in our lab, the topology of which is similar to the topology in Figure 16. More detailed parameters of the test-bed are as follows.

a) Supplicant
   Hardware: LINKSYS Wireless-G USB wireless NIC
   Software: Windows XP, WPA_supplicant 0.5.10
b) AP
   Hardware: LINKSYS Wireless-B PCI wireless NIC
   Software: Linux2.6, Hostap drivers, Hostapd 0.5.10
c) AS:
   Hardware: Broadcom NetLink(TM) Ethernet NIC
   Software: Linux2.6, FreeRadius 2.1.1

D. Simulation and test data

To contrast simulation data with practical test data, detailed information in every round of protocol, including transmission delay, propagation delay and processing overload, is precisely recorded.

In practical test scenarios, protocol packets are captured in NIC to obtain the difference between the packets’ sending time and receiving time. In one case, the difference refers to the span of time when a packet enters the NIC and another packet of next round of protocol departs from the NIC. In another case, it refers to the span of time when a packet departs from the NIC and another packet of next round of protocol is received at the same NIC. At the same time, breakpoints are inserted in the testing program and variables are set to record the packets’ processing time, i.e. MAC protocol processing time, which includes encapsulation and decapsulation, encryption and decryption and protocol FSM operations.

In simulation scenarios, the packets’ receiving time and sending time are recorded in transmit-receive module of OPNET.
between the packets’ sending time and receiving time. The time of processing refers to the protocol’s processing time. In Figure 20, the sending time denotes the time when a packet departs from the transmit module of OPNET, and the receiving time denotes the time when a packet enters the receive module of OPNET.

**E. Analysis**

It can be learned from the principles of communication and Media Access Control that there are several time-consuming states in one round of protocol exchange.

- **T0**: Delay of MAC protocol operation
- **T1**: Delay of sending
- **T2**: Delay of propagation
- **T3**: Delay of receiving
- **T4**: Delay of resolving and decapsulating
- **T5**: Delay of processing protocol packets (verifying signature and decrypting and so on)
- **T6**: Delay of forming new packets (computing signature, Hash, and encrypting and so on)
- **T7**: Delay of encapsulating
- **T8**: Delay of ACK

In practical test scenarios, because of the adoption of WLAN standard topology and protocol configuration, the practical running time of EAP-TLS includes following time-consuming states:

\[ T_{\text{run}} = T0 + T1 + T2 + T3 + T4 + T5 + T6 + T7 + T8 \]

Because the responding ACK and the processing packets are the simultaneous performance of operations, T8 is not counted in the Trun. Moreover, we also discover that during the running of protocol, the delay of processing packets accounts for the majority of the total running time which is not same as what we have commonly thought — the majority of the total running time mainly consists of the delay of sending, receiving and propagation.

In simulation scenarios, due to the lack of modeling CPU computing capacity and RAM storage capacity, the delay of resolving, processing and encapsulating protocol packets are not counted in the total simulation time which includes following time-consuming states:

\[ T_{\text{simulate}} = T0 + T1 + T2 + T3 + T8 \]

Apart from the delay of processing packets, the simulation time is almost same as the practical running time, i.e. the difference between them is slight and in an acceptable range, which clearly demonstrates that the proposed simulation platform can be considered to be correct.

5) **Validation of Universality**

In the previous sections, two types of WLAN security protocols, WEP and EAP-TLS, are added by the designed unified extension interface to the proposed simulation platform without modifying other communication modules, and reasonable and similar results are obtained from the simulation and the practical test, which definitely shows that the proposed simulation platform possesses the characteristics of universality.

**VII. CONCLUSION**

In this paper, we propose a novel simulation platform for WLAN security protocols and investigate its architecture design, simulation interface and protocol extension method. The simulation platform provides a universal, flexible and extensible simulation solution for the evaluation of WLAN security protocols. Furthermore, based on the architecture and the method, we also implement the simulation platform in OPNET and validate its feasibility, correctness and universality through simulating and testing concrete security protocols. The results show that the platform has characteristics of correctness, universality and extensibility.

**VIII. ACKNOWLEDGMENT**

This work is supported by the the Fundamental Research Funds for the Central Universities under Grant No. JY10000903006.

**REFERENCES**


Chao YANG received the PHD. degree in cryptology from Xidian University, China, in 2008; He is currently an Associate Professor in the School of Computer of Xidian University. His research interests include: wireless network, security protocol and cryptology.