An Automated System Interoperability Test Bed for WPA and WPA2

Sebastian Bohn*, Stephan Groß[†], René Nüßgen*, and Paul Schwann*
*Philips Semiconductors Dresden AG, Am Waldschlößchen 1, D-01199 Dresden, Germany,
Email: {sebastian.bohn, rene.nuessgen, paul.schwann}@philips.com
[†]Technische Universität Dresden, Department of Computer Science, D-01062 Dresden, Germany,
Email: st.gross@inf.tu-dresden.de

***Abstract*—The discovery of several attacks on WEP during the past few years has rendered the first WLAN security standard useless. Thus, new mechanisms had to be defined to protect current and future wireless infrastructures. However, some parts of the new standards WPA and WPA2/IEEE802.11i respectively require changes in the used hardware. To ensure interoperability between different vendor's products the Wi-Fi alliance provides a certificate that can be obtained by passing several fixed tests. Unfortunately, there exists no standard solution so far to get your products ready for the certification process. Each vendor has to do his homework by hand. To overcome this manual and error-prone process we have developed a test environment for conducting automated system interoperability tests. In this paper we outline the Wi-Fi certification process and categorize necessary test requirements to be fulfilled. We further discuss our solution, i.e., the setup of our test environment and selected implementation details of the associated control software.**

***Index Terms*—Automatic test software, Certification, Wireless LAN**

I. Introduction

In the end of the twentieth century Wireless LANs became more and more popular. People use it widely both in corporate locations and at home. During the first years of production, WLAN devices were shipped with a protection mechanism called Wired Equivalent Privacy (WEP) [1]. It was known to be a convenient suite providing message integrity and privacy protection. As the success of WLANs increased, the underlaying security algorithms were analysed by security experts. It came about that more and more vulnerabilities of WEP could be disclosed. This led to the need for development of new security mechanisms. Wi-Fi Protected Access (WPA) [2] and later IEEE802.11i [3] and WPA2 [4] respectively have been proposed as solutions to overcome this issue. However, new security could not be achieved without new hardware in all cases, i.e., to benefit from the whole range of new security mechanisms one has to invest in new WLAN network equipment.

The WLAN boom has also resulted in a multitude of manufacturers offering appropriate equipment. In order to assure interoperability among the variety of different vendor's WLAN devices, the Wi-Fi alliance[1] has developed a certification process. Amongst others, obtaining that certificate requires supporting the new encryption types as well as passing several specified tests. Unfortunately, the Wi-Fi alliance does not provide a standardized solution to prepare for such a certification process. Thus, each vendor has to build up his own customized test lab in compliance with the Wi-Fi regulations. This is obviously an error-prone and expensive task. To reduce both costs and failures we have developed an universal test software for Wi-Fi certification tests. It encapsulates all requirements defined in the Wi-Fi specifications into several software modules and supports the setup and operation of a standard conform automated environment for system interoperability tests.

The remainder of this paper is structured as follows. In section II we give an introduction into common solutions for secure Wireless LANs. Following that, we elucidate the system interoperability test procedures provided by the Wi-Fi alliance (section III) before we introduce our solution for the automation of these tasks (section IV). Finally, we close with a conclusion of our findings and suggested further work.

II. A Brief History of WLAN Security

Wired Equivalent Privacy or short WEP has been the only known method for wireless networks security during the first five years of IEEE 802.11. With increase of Wi-Fi LAN's popularity in 2000, security experts turned their attention to the cryptographic methods used, namely the RC4 algorithm. This led to disclosure of significant vulnerabilities within short time. Towards end of 2001 WEP was classified as completely broken, and tools for various attacks were available freely in the Internet. However, many of those attacks required a large amount of captured traffic. Therefore, WEP remained quite a good security solution for home users with a comparatively small quantity of traffic. But since August 2004 WEP must be considered dead again and everybody using it is at high risk.

[1] The Wi-Fi alliance is an international nonprofit association founded by major manufacturers of WLAN hardware in 1999. For more informations please refer to http://www.wi-fi.org/.

0-7803-9412-7/06/$20.00 ©2006 IEEE

TABLE I
Comparison of WEP, WPA, and WPA2/IEEE802.11i

	WEP	WPA	WPA2
Cipher Type	stream	stream	block
Cipher	RC4	RC4	AES
Key Size	40 or 104 bits	128 bits	128 bits
IV Size	24 bits	48 bits	48 bits
IV Reuse Protection	no	yes	yes
Security Protocol	WEP	TKIP	CCMP
Message Integrity	ICV	Michael	MIC
Key Management	direct key	multiple key derivation	
Weak Keys	yes	padded out	no
Overall Security	broken	secure	state-of-the-art

The so-called "KoreK attack" [5] is based on statistical crypt analysis and allows breaking a 104 bit WEP key with little effort in only a few minutes. Today, we can state that WEP failed in almost any security requirement. Methods for key management were poor and did not scale in large networks. Key length was too short and many tiny logical errors in the basic concept allowed successful attacks on disclosure of the secret key by only monitoring traffic. The willing reader might have a look into [6]–[9] for full details. WEP is a good example for problems that can arise if a security protocol is developed without proper review by security experts. Understanding why WEP fails will help to understand why next-generation security methods are so much stronger.

To overcome the drawbacks of WEP the IEEE 802.11 Working Group originated an own task group to define a new and more competitive security standard for wireless networks. This standard, IEEE 802.11i, defines a secure WLAN called Robust Security Network (RSN). In some respects, this is the same as a traditional or WEP based network. But a wireless device has to show a number of new features in order to be able to join a RSN. However, since legacy Wi-Fi hardware is not compatible with RSN and customers definitely do not want to replace all their existing equipment, Task Group i has developed a solution compatible to legacy hardware. Inter alia, this led to the definition of the Temporal Key Integrity Protocol (TKIP). TKIP is permitted as an optional mode under RSN. To save the industry from waiting for finishing the lengthy process of standards ratification, the Wi-Fi alliance has adapted a new security approach based on the RSN draft specifying TKIP only. This subset of RSN is called Wi-Fi Protected Access (WPA) [2]. RSN and WPA share a common architecture and approach. After the completion of IEEE 802.11i in 2004 the complete RSN definition has been adapted by the Wi-Fi alliance as WPA2.

Table I shall oppose all three standards and give a brief overview of their most important differences. For further details we refer to [6].

III. Wi-Fi System Interoperability Test Plans

The Wi-Fi Alliance was formed in 1999 as an international nonprofit association to certify interoperability of wireless local area network devices based on the IEEE 802.11 specifications. Its objective is to ensure interoperability among IEEE 802.11g and 802.11b products from multiple vendors, and to promote this technology across all market segments. The Wi-Fi Alliance has instituted a test suite that defines how member products are tested to certify that they are interoperable with each other. The test plan is divided into two parts: implementation requirements and interoperability test procedures for access points and stations respectively. The latter include tests for BSS and IBSS scenarios. However, IBSS tests only utilise WEP or no security configurations. Thus, they will be omitted in this paper. Due to the lack of space we will also neglect access point and concentrate on implementation requirements and station, i.e., WLAN adapter, test procedures. For further details please refer to [10], [11].

A. Implementation Requirements

A product generally shall comply with IEEE 802.11g to pass interoperability testing. According to the test plan it must support and correctly handle several features, such as SSID element, beacon interval, TIM element, RTS/CTS, CTS-to-self and protection mechanisms, fragmentation, etc. Stations must be capable of operating at each specified data rate from 1 to 54 Mbit/s.

To pass the WPA portion of interoperability testing, which will be confined to the BSS mode only, a device must either support WPA-Enterprise (WPA 802.1X/EAP plus PSK) or WPA-Personal (PSK only). To ensure the correct operation of TKIP, fragmentation is also subject to verification with WPA turned on. Along with correct handling of the WPA Information Element (WPA IE) and proper key management, a station is required to be capable of supporting countermeasures as defined by the WPA for 802.11 Specification. The test plan requires that countermeasures to detect and handle MIC failures implemented in the two supplicants to be used by the station under test (STAUT). The vendor may provide a supplicant along with the product. The STAUT has to interoperate using the Microsoft Windows XP operating system, and if it ships with a Windows 2000 driver, both operating systems are tested. For the WPA2 interoperability testing, analogous requirements apply. Additionally, TLS client certificates must conform with RFC 3280 [12]. WPA2 certified products shall be backwards compatible with WPA certified products.

B. Station Testing

Station testing is divided into two categories, station configurability and interoperability tests. When a STAUT only supports WPA-Personal, all tests that specify WPA-TLS shall be replaced with WPA-PSK, and use a pre-shared key of "12345678".

The Wi-Fi alliance has specified a certain network setup for properly running all stated test procedures. Basically a STAUT shall be connected to the test LAN via different access points. The LAN contains at least one authentication server (RADIUS) and a test server running special software to create traffic or receive frames from the STAUT. Figure 1 depicts a

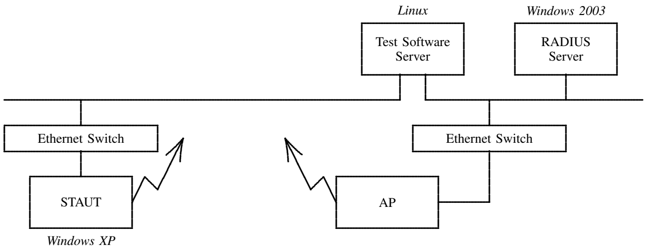

Fig. 1. Sample test scenario setup

sample setup for a conform test scenario used in our test lab. It must be pointed out that the STAUT as well as the AP can be reached via Ethernet from the test server and, thus, can easily be configured.

The test suite starts with out-of-box (OOB), configurability and initial ping tests. Therefore, the STAUT is set to its factory default configuration to simulate first-time use. Three access points participate in the OOB test. They are configured to have no encryption, WEP encryption, respectively WPA-PSK encryption enabled. The pass criteria is that the STAUT will allow association and pings only with the first AP, having security disabled. The initial ping test is done utilising access points configured to use WPA-TLS security. It shall verify that the STAUT can authenticate, associate and support pings to a wired authentication server on a subnet connected to the test configuration. The STAUT is to ping through each of the access points in the test bed to a server within 90 seconds to pass this test.

Having passed the first group of tests, the STAUT is verified to be able to work with 802.11g and 802.11b access points from different vendors, with a variety of ESSIDs, beacon intervals, with and without RTS and fragmentation, on several channels. The tests are conduced on four sets of security configurations: no security at all, WEP, WPA-TLS, and WPA-PSK. Each configuration includes three scenarios of combinations of 802.11b/g pure and mixed BSS styles: pure-G, pure-B, and mixed-G/B. Proper association is tested first. Subsequent throughput tests comprising mutual data transfer of diverse data types finalise the second group of tests.

Furthermore, a station's ability to roam across access points from different manufacturers is tested. These tests will be performed involving the two security options WPA-PSK and WPA-TLS. For a roaming test the STAUT shall first associate and authenticate with an access point. A repeating ping command demonstrates application level connectivity. When this is working, a second access point is started and when beaconing the first AP is stopped. The STAUT passes this test if it will be able to associate and authenticate with the second AP and can resume the ping application within 90 seconds. The cycle shall be repeated with different access points.

The STAUT's ability to receive and correctly handle broadcast and multicast packets is verified next. Therefore, UDP packets will be sent from a machine inside the test network via broadcast or multicast stream to at least two other machines, including the STAUT. Successful data verification is required for passing this test.

Moreover, the station under test is connected to access points in a mismatched configuration to check and ensure that neither association nor any data transfer occurs. The mismatches include wrong ESSID, wrong case in ESSID characters, ESSID substrings, a WEP key mismatch, as well as the STAUT configured for WEP but the AP for non-WEP, and vice versa.

WPA respectively WPA2 specific tests conclude the interoperability tests for BSS scenario. WPA specific countermeasures are examined first. Its purpose is to ensure that the STAUT can recover from a message integrity check (MIC) failure as described in [2]. The station must de-authenticate or disassociate if receiving bad frames from the attacker access point. The aim of the test here is to ensure that re-establishment can only occur after 60 seconds. WPA and WPA2 negative tests are performed to validate that the STAUT configured for WPA or WPA2 does not interoperate with any configuration that will compromise the security, e.g. WEP or different authentication modes.

WPA2 certification requires passing additional tests. A pre-authentication test is optional and shall only be performed if the STAUT supports pre-authentication completion. However, testing correct pairwise master key (PMK) caching is mandatory. Its aim is to verify that when re-associating to a certain access point PMK caching is used by the STAUT instead of a full EAP authentication.

Looking at these test plans from a more developer's point of view we can distinguish the following test categories:

Connectivity Tests: All tests probing for association and simulating application layer connectivity in a correctly configured network setup.

Non-Connectivity Tests: This includes all "negative" tests, i.e., the participating devices are configured for a mismatching network scenario, and succeeding tests fail if connectivity can be measured.

Performance Tests: All tests where link throughput is verified by measuring TCP and UDP performance with variously sized file transfers using FTP, TFTP, or a raw packet generator.

Feature Tests: This includes WPA or WPA2 specific tests that verify the existence of a certain (mandatory or optional) hardware behaviour, countermeasure tests, pre-authentication tests, or PMK caching tests.

Configurability Tests: Enumerates testing of configurability, such as user interface masks. These test are hard to automate. Hence, our solution does not care about this group of tests.

Having analysed the Wi-Fi test plan we will now come to the basic design of our automated test suite.

IV. Automating System Interoperability Tests

Our implementation of an automated test software server as depicted in figure 1 was developed in Perl. It basically operates as follows: all devices involved in a test setup are set to factory default, are subsequently configured as needed, and eventually test macros are run. The test result will be logged as XML output. To control the STAUT we use a SSH channel to alter the machine's registry. Thus, we have direct access to the wireless adapter's driver as well as the supplicant's configuration. The same applies to the RADIUS server. Adjusting the AP's settings is more difficult due to the variety of different vendor's access points. It is quite certain that each device offers other ways and different user interfaces for configuration. A web interface can be considered as the least common denominator. However, each is varying in look and feel. Thus, the only acceptable way of automatic configuration is to emulate web interface usage by sending appropriate HTTP requests to the desired AP.

In order to make tests portable, we have designed a generic script language to define abstract test procedures. Parameters of each command in such a procedure will be evaluated and wrapped into the appropriate device commands. This allows to use on procedure file to configure different vendor's hardware. The testing functionality is also described in so-called test atoms making use of our language. Again the test atoms are transformed into commands according to the target operating system. To further enhance the modularity of our software for the sake of flexibility and extensibility we added constructs to realize nested test definitions. It is also possible to define a specific threshold to be excessed in order to pass a certain test. For more details on our scripting language we refer to [13].

We will conclude this section describing the most important modules of our implementation (see figure 2). The class *TestDescription* is responsible for reading out all config files, including test description, test setup, and macro description files. *DeviceConfiguration* acts as gateway for all types of devices. It offers interfaces for wrapping generic configuration directives into operating system specific commands. *TestAtom* provides an interface for all test atoms that are defined in inherited classes. *TestResult* is used for generating XML test logs. One cycle of the program looks as follows: At first, given command line parameters are checked for validity. If everything is OK, a new *TestDescription* object will be created. Initially, it contains the paths to all relevant configuration files. Thereafter, the test description will be opened utilising the procedure *GetProcedureFile()*. If this succeeds, the function *main()* will be called. It reads the description by calling *readProcedure()*. Subsequently, macros are read and tested for validity, the hardware database is examined, and all participating devices are initialised by creating a new *DeviceConfiguration* object. Finally, each defined macro will be executed by repeated calling of *runTests()*. The latter function tests device association if necessary and calls the macro's test atoms with given parameters. These calls will be wrapped by the respective concrete *TestAtom* classes into appropriate system calls. The return code and results of each execution are examined and written in readable form into the log file using *TestResult*.

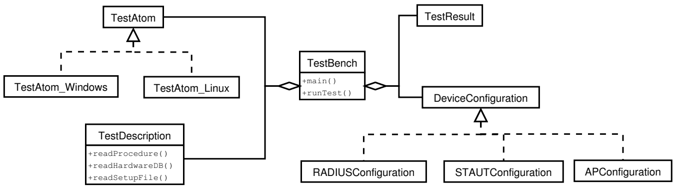

Fig. 2. Basic design of our test bench

V. Conclusion

We have presented a solution for automated system interoperability tests according to the Wi-Fi alliance's regulations. After introducing common wireless security standards we analyzed the interoperability test plans provided by the Wi-Fi alliance. We identified several distinct test categories and derived common requirements for an automated software solution. Our implementation is based on the Perl language and follows an object-oriented approach, thus, offering a maximum of flexibility and extensibility. It was developed for a competitor on the WLAN market and is used on a regular basis to prepare new devices for the Wi-Fi certification process. Our approach has been proven useful for everyday work during several trial tests. However, as some tests have not been implemented yet, there is still some work to be done for the future. We are also planning to enhance support for access points and WLAN devices running on embedded systems.

References

[1] B. O'Hara and A. Petrick, *IEEE 802.11 Handbook, A Designer's Companion*. IEEE Press, 1999.

[2] Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks. Retrieved June, 28 2005. [Online]. Available: http://www.wifialliance.com/OpenSection/pdf/Whitepaper_Wi-Fi_Security4-29-03.pdf

[3] IEEE 802.11 Working Group, Task Group I, "802.11i. IEEE Standard for Information technology. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. Amandment 6: Medium Access Control (MAC) Security Enhancements," July 2004, retrieved June, 28 2005. [Online]. Available: http://standards.ieee.org/getieee802/download/802.11i-2004.pdf

[4] Wi-Fi Protected Access 2. Retrieved June, 28 2005. [Online]. Available: http://www.wi-fi.org/OpenSection/protected_access.asp

[5] KoreK attack. Retrieved June, 28 2005. [Online]. Available: http://www.netstumbler.org/showthread.php?t=11869

[6] J. Edney and W. A. Arbaugh, *Real 802.11 Security: Wi-Fi Protected Access and 802.11i*. Addison-Wesley, 2004.

[7] S. Fluhrer, I. Mantin, and A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," in *Proceedings of the 8th Annual International Workshop on Selected Areas in Cryptography (SAC 2001)*, ser. LNCS, vol. 2259. Springer, 2001, pp. 1–24.

[8] M. Schmidt, "Der WEP-Wall bricht," *c't Magazin für Computertechnik*, no. 10, 2005.

[9] M. Ossmann. WEP: Dead again. Retrieved June, 28 2005. [Online]. Available: http://www.securityfocus.org/infocus/1814

[10] "Wi-Fi 802.11g with WPA System Interoperability Test Plan," Wi-Fi Alliance, Tech. Rep. Version 2.2.

[11] "Wi-Fi 802.11 with WPA2 System Interoperability Test Plan for IEEE 802.11a, b & g Devices," Wi-Fi Alliance, Tech. Rep. Version 2.3.

[12] R. Housley, W. Polk, W.Ford, and D. Solo. (2002, Apr.) RFC 3280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. IETF Request for Comment. Retrieved June, 28 2005. [Online]. Available: http://www.ietf.org/rfc/rfc3280.txt

[13] Sebastian Bohn, "An Automated Interoperability Test Bed for WPA and WPA2," Master's thesis, Technische Universität Dresden, 2005.