Design of Improving the Usability and Security of WPA Wireless Network

Albert Chen, Zhiyong Cheng, and Henry Lam
alphax777@gmail.com, zhiyongc@cs.ubc.ca, hlundertaker@hotmail.com

***Abstract*—Nowadays, wireless networks are everywhere. Despite its popularity, wireless networks are not as secure comparing to wired networks: Wired equivalent privacy (WEP) has already been proven to be insecure and even Wi-Fi protected access (WPA) is vulnerable to offline dictionary attacks. Despite the weakness, WPA authentication protocol with a 63-character computer generated passphrase is proven to still be secure. Unfortunately, most users choose relatively weak passphrases due to convenience.**

In this report, we present a naïve approach to improve the security of WPA wireless networks without sacrificing usability. A proof-of-concept prototype is created and evaluated.

***Index Terms*— 802.11 Wireless Networks, Wi-Fi, .Net framework, WEP, WPA, Usability, Security.**

I. Introduction

WITHIN the last decade, wireless network rapidly gains popularity due to its flexibility for both network owners and users. The following facts contribute to this phenomenon: almost all the new laptops, PDAs and smart cell phones are Wi-Fi equipped; many Internet service providers such as Telus and Shaw are giving away wireless gateways/routers as a fixed term contract signing bonus; wireless hotpots are available at cafés, campus, libraries and other public places; flexibility of wireless allows user to work anywhere within the covered area. While general users enjoying the improved productivity and mobility offered by the wireless networks, they also suffer from a wide range of security problems range from theft of Internet bandwidth to lost of confidential information.

In response to these security threats, several encryption/ authorization protocols are introduced to secure wireless networks. As the name suggests, Wired Equivalent Privacy (WEP) [1] protocol which published as IEEE 802.11 standard in 1997 supposes to provide a wireless security level equivalent to wired networks. Unfortunately, critical design flaws in WEP such as key stream reuse, linear checksum and small initialization vector (IV) length were quickly identified which prevent WEP from achieving its security goals [2][3]. As a remedy, a new security standard for 802.11 wireless networks was released by Wi-Fi Appliance in 2002: Wi-Fi Protected Access (WPA)[4]. The primary goal of WPA is to fix all of the known security design flaws in WEP. WPA improves the WEP design by increasing the length of IV/key, replacing the CRC checksum in WEP with message integrity check (MIC) and new features such as key management system provided by temporal key integrity protocol (TKIP) and data encryption using AES-CCMP. WPA has been successful and today almost all new Wi-Fi devices are WPA compatible.

Although WPA and its successor 802.11i [5] (a.k.a. WPA2) are considered to be more secure than WEP[1], WPA protocol is still vulnerable to offline dictionary attacks when a weak pre shared key (PSK) is used [6]. Unfortunately, in the real world, normal users tend to choose weak passphrases because it is simply too hard to memorizing randomly generated 63-characters. The problem is, how to prevent user from using weak passphrases to improve the security of WPA-PSK wireless networks without sacrificing the system usability? Several products are available in the market to attempt to address this problem, but all of them have usability issues as we will analyze in the later section. In this paper, we will present our own solution to the problem and implement a proof-of-concept application: OneClickSetup.

In Section II, we examine the usability of two related solutions in detail. Then we state our design goals in Section III and present our idea design in Section IV. The implementation of our proof-of-concept prototype is explained and evaluated in Section V and Section VI separately. Finally, we address unsolved issues in future work (Section VII) and reach our conclusions in Section VII.

II. Related work

Major vendors which dominated the market such as Microsoft and Linksys also realize the security risks related to weak WPA-PSK passwords. As a result, both Microsoft and Linksys attempt to address the problem with their own solutions. These solutions arrive in the form either in hardware, software or hybrid solution. However, both of the solution we analyzes here have their own weaknesses in the context of usability:

A. Linksys/Broadcom: SecureEasySetup (SES)

According to Linksys website [8], SES is designed to “make setting up a wireless network and installing WPA security as simple as pushing a button.” Ideally, a user can setup a WPA wireless network following the “Two-Step SES Process”:

--First, press the SES button on the Linksys wireless router

[1] At the time this report is finished, a draft paper[7] released from the same group (aircrack-ng) claims WPA can be cracked within 15 minutes by attacking on TKIP even a strong password is used. However, no proof is shown and we assume that WPA with strong password is secure enough for normal home users.

or access point.

--Second, click on the SecureEasySetup "START" button on computer without wireless network access.

The SecureEasySetup software which comes with the wireless card driver CD must be installed on the computer prior to the setting up process. This solution does simplify the process of configuring a wireless work however it has the following weaknesses:

1) The solution only works with Linksys products which mean that a user has to purchase both access point and wireless network card from Linksys/Cisco. This increases the operating cost of wireless networks.
2) SecureEasySetup software only works on Windows operating systems. Computers with Linux or Mac OSX installed cannot run SecureEasySetup thus are unable to setup wireless network.
3) The SES process only generates 15-character long passphrase and during the synchronization process, the passphrase is transmitted in clear text for several minutes

B. Windows XP/Vista Wireless Network Setup Wizard

Microsoft provides a wireless network setup tool in the latest operating system products such as Windows XP SP2 and Vista. In order to apply the solution, a wireless router/access point which has a USB port and support Windows Smart Network Key (MSNK) is required. Developed together by Microsoft and HP, MSNK enables Windows users to configure their wireless networks quickly and securely together with wireless network setup wizard application. Like many other windows wizard applications, users must go through a series of steps to export wireless network settings to a USB flash driver and repeat the same process while importing the wireless network settings back. This solution has some fundamental flaws in terms of usability:

1) MSNK-enabled wireless devices are almost impossible to find in the market. Many wireless network hardware manufacturers do not provide device with support of MSNK. The reason is not clear but we are unable to find a MSNK-enabled wireless router/access point online using Google.
2) Similar to SecureEasySetup, wireless network setup wizard application only works on Windows operating systems. Computers with other operating systems cannot use this solution to setup wireless network.
3) The wizard user interface is excessively complicated for general users who have little or no network experiences. At each page of the wizard, more than necessary information is presented which only confuses the users. This can be proved by our usability tests performed later.
4) As a general wireless network setup tool, wireless network setup wizard does not enforce the length of WPA passphrases. Therefore, users are still able to select weak passphrases which reduce the security of the wireless networks.

C. Analysis

Applying ten principles of desiging secure systems learned from the class, we identify that several principles are violated in the two solutions mentioned above:

--First of all, both solutions breach the “Psychological Acceptability” [9] principle. The SES solution requires user to verify that both the wireless router/access point and wireless network card are Linksys products which are SES compatible. Similarly, the wireless network setup tool expects user to make sure that wireless router is MSNK enabled. These security mechanisms add difficulty to access wireless network resources and could result in security configuration failure.

--Secondly, the Windows XP SP2/Vista wireless network setup tool violates the “question assumptions” principle. The tool leaves too much information to the user and expects the user to make the correct decision. This design is based on the assumption that users have some network experiences or even have done some network administrator’s work. Considering the large user base of Windows operating systems, this assumption is apparently false.

--Last but not least, the SES solution doesn’t obey the “open design”[10] principle. The details of synchronization process between the wireless router and the computer are not publicly released. As the result, when the public learned that passphrases are transmitted in plaintext during the synchronization process, it is too late for users and vendors to take any countermeasures against the theft of passphrases during the synchronization. This is the perfect example of no “security through obscurity.”[11]

Despite these usability flaws, these two solutions also have some good designs which could greatly improve user’s wireless network configuration experience. Learning from the two solutions' mistakes we came up with a design that is similar to those mentioned above but improved on area where the two solutions fell short.

III. Design Goals

Basing on the analysis of currently available solutions and what we learned from EECE 412 course, we attempt to design an application which helps user to securely setup WPA wireless network without reducing usability. The followings are the design goals we want to achieve:

1) Simple one click solution.
2) Automatic setup WPA wireless connection profile within the operating system.
3) Cross platform support which means the application can be ported to other operating system such as Linux and Mac OSX.
4) User friendly which means the application is easy to use even for grandparents.
5) Portable to USB flash drivers which means the application should be small in size.

IV. System Design

The ideal solution we proposed includes design in both hardware and software.

A. Hardware design

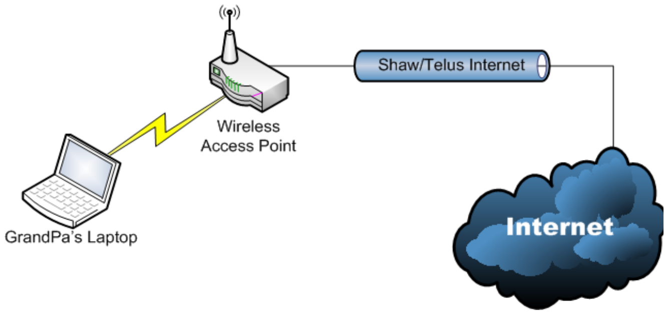


Fig. 1. A typical user scenario

In our ideal design, all the wireless router/access point manufactured would be equipped with a synchronization button, USB ports and a USB flash drive. When user press the synchronization button, the SSID, secure password would be written in an XML based profile.

B. Software design

We propose a layered architecture for the software part. The rationale are:

- operating systems are layered
- Most of Windows SDK APIs and Linux library are written in C/C++ and not object oriented.
- Our application is small and easy enough to be code in a single layer

Layers:

Hardware: wireless network cards. Wireless router/access point

Operating system: Windows SDK API, Linux libraries, Wireless card device drivers.

Application: Cross platform frameworks such as wxWidgets, QT4 and java. Wireless profile parsing (XML). Wrapper classes for lower level APIs or libraries. User Interface.

C. A typical user scenario

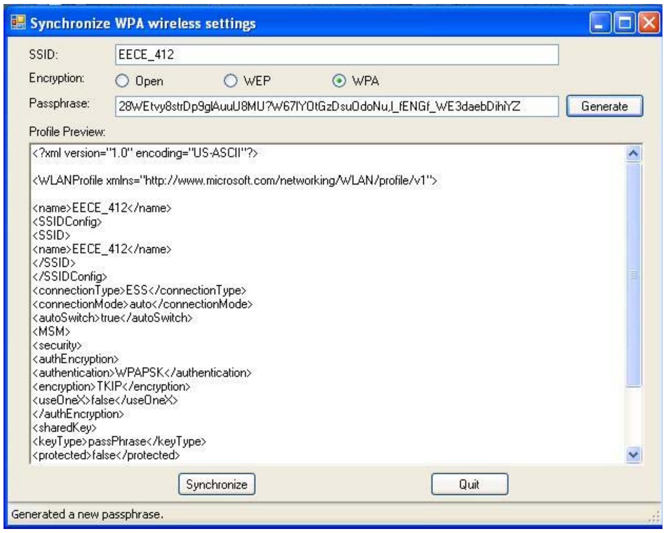
Grandpa scenario: Grandpa login using an unprivileged user account on Windows Vista. He wants to access the Internet but have little wireless network experience… Finally, with the help of OneClickSetup, grandpa is able to access the Internet with minimal learning curve and efforts.

V. Proof-of-Concept Prototype Implementation

To verify the design, we implemented proof-of-concept prototypes: hardware simulator and OneClickSetup.

A. Hardware simulator application

Due to the limited resources especially in time and supports from wireless router manufactures, we are unable to implement the hardware design proposed. Instead, we implement a hardware simulator to show how wireless access point in our design should work when the user press the synchronization button. The screenshot in figure 2 generates a 63-character WPA passphrase then synchronize the wireless profile settings to the USB flash drive.



SSID: EECE_412

Encryption: ○ Open ○ WEP ● WPA

Passphrase: 28WEtvy8strDp9glAuuU8MU?W67lYOtGzDsuOdoNu,l_fENGf_WE3daebDihiYZ

Profile Preview:

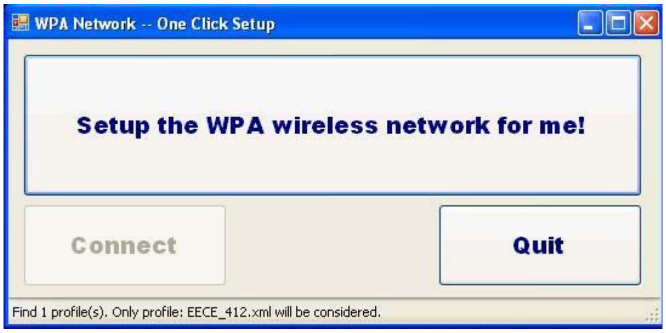
```
<?xml version="1.0" encoding="US-ASCII"?>

<WLANProfile xmlns="http://www.microsoft.com/networking/WLAN/profile/v1">

<name>EECE_412</name>
<SSIDConfig>
<SSID>
<name>EECE_412</name>
</SSID>
</SSIDConfig>
<connectionType>ESS</connectionType>
<connectionMode>auto</connectionMode>
<autoSwitch>true</autoSwitch>
<MSM>
<security>
<authEncryption>
<authentication>WPAPSK</authentication>
<encryption>TKIP</encryption>
<useOneX>false</useOneX>
</authEncryption>
<sharedKey>
<keyType>passPhrase</keyType>
<protected>false</protected>
```

Generated a new passphrase.

Fig. 2. Hardware Simulator Screenshot

B. OneClickSetup application

The OneClickSetup application is coded using Visual Studio 2008 using C# The application also included Open source project: Managed Wi-Fi API. To use the application, user simply click on the "Setup the WPA wireless network for me!" and then click on the "Connect" button to initialize the connection


WPA Network -- One Click Setup
Setup the WPA wireless network for me!
Connect
Quit

Find 1 profile(s). Only profile: EECE_412.xml will be considered.

Fig. 3. OneClickSetup Screenshot

VI. Evaluation

To evaluate our prototype application OneClickSetup, we strictly followed the procedures described in [12] and designed the usability test. We gave our participants a questionnaire and asked them to carry out the security tasks described in the questionnaire: setup a WPA wireless work using both Windows wireless network setup wizard and OneClickSetup. We select Windows wireless network setup tool as our comparison subject because of its availability and popularity.

A. Participants

How to recruit/find participants?

Due to the restricted budget we had, there were no proper funding for finding test participants. Thus our test participants are consists of friends, family members, and classmates. We have 7 participants in total, range from age 22 to 78. The table below describes their knowledge of computers and experience in setting up wireless network.

B. Test environment and process

Out test requires 1 laptop and 2 USB flash drives. One USB for OneClickSetup and one for Windows Vista Wireless setup. The laptop has Windows Vista on it with no wireless connection setup. We have created a non administrator user account called user to simulate a normal user environment.

In each session with our participants, we ask to do the following in the same order

1) Fill out the first half of the usability form, which stated their knowledge with computer and experience in setting up a network connection.

2) Login to our laptop using the user account

3) Try to create a wireless connection using Window Vista's setup. We will provide minimum assistants in the event they do not know what to do.

4) Try to create a wireless connection using OneClickSetup. We will provide minimum assistants in the event they do not know what to do.

5) Fill out the second half of the usability form, which ask about their experience in using the two software.

C. Result analysis

We decided to use weighted average to total up the responses of the participants. We add up their score from their previous experience. Then, we multiply their score for "Windows Vista Wireless Setup" by this number, to get their total score for "Windows Vista Wireless Setup". We also did the same for the OneClickSetup. This way, the participants with least knowledge in computers will be weighted the highest. As our primary audience is user with little or no experiences, thus we rated the response of this type of participants the highest. The result is shown in Figure 4.

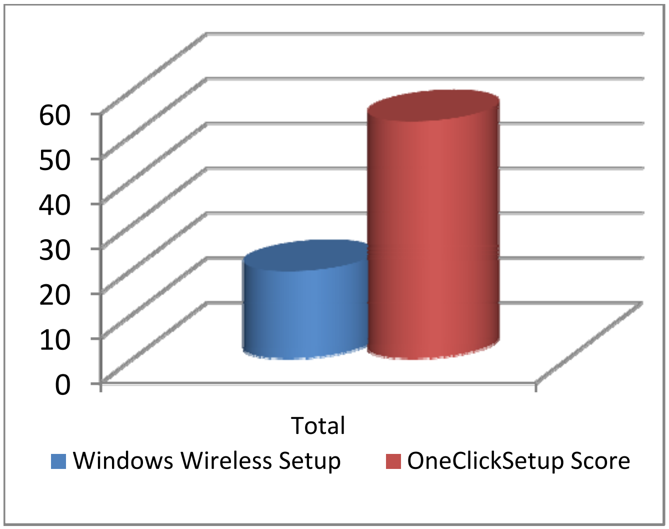


Fig. 4. Evaluation Result

VII. FUTURE WORK

Future work of the OneClickSetup include:

- Add cross platform support using WxWidgets or QT4
- Encrypt/protect the WPA key exported (currently the key is stored in plaintext).
- Finish hardware design and implementation (Wireless router with synchronization button, USB flash drive)
- Establish a wireless network profile standards for the different manufacture which all will follow.

VIII. CONCLUSION

Through the usability study, we have shown that the OneClickSetup prototype is a much easier solution to WPA key management problem. We have completed our goals which we have set at the beginning of this project. The final product will automatically create a wireless WPA connection within Windows Operating System. The product is very user friendly and extremely portable.

APPENDIX 1 USABILITY TEST QUESTIONNAIRE

Name: _______________
Subject#:______________
Date: _________________
Please ask the subjects the follow questions BEFORE you start doing the experiment
1) How much experience have you had setting up a computer
(1 = very experience, 10 = very inexperience)

2) How much experience have you had setting up network
(1 = very experience, 10 = very inexperience)

3) How much experience have you had setting up wireless network
(1 = very experience, 10 = very inexperience)

Now, ask the subject to try setting up a wireless router using Windows Vista's wireless setup. After they have completed the experiment, ask the following questions.
4) How easy was it to you to setup a wireless router in the Windows Vista version
(1 = very hard, 10 = very easy)

Now, ask the subject to try setting up a wireless router using OneClickSetup. After they have completed the experiment, ask the following questions.
5) How easy was it to you to setup a wireless router in the OneClickSetup (1 = very hard, 10 = very easy)

Total Score for Windows = (Q1 + Q2 + Q3) x Q4 = _________
Total Score for OneClickSetup = (Q1 + Q2 + Q3) x Q5 = _________

Acknowledgment

We would like to thank classmates, TA, Kosta, and all of the test participants.

References

[1] *Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Standard 802.11, 1997

[2] W. Arbaugh, N. Shankar, Y. Wan, and K. Zhang, "*Your 802.11 wireless network has no clothes*," IEEE Wireless Communications, vol. 9, no. 6, pp. 44-51, Dec. 2002.

[3] M. Borse and H. Shinde, "*Wireless Security & Privacy*," 2005 IEEE International Conference, pp. 424-428, Jan. 2005.

[4] Alliance, W.F. "*Wi-Fi Protected Access: Strong, standards-based, interoperable security for today's Wi-Fi networks,*" Apr. 2003.

[5] *Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Specifications for Enhanced Security*, IEEE Standard 802.11i, Jun. 2004.

[6] Aircrack-ng project. [Online]. Available: http://www.aircrack-ng.org/

[7] M. Beck and E. Tews, "*Practical attacks against WEP and WPA,*" Prepare to publish. Nov. 2008

[8] Linksys – What is SecureEasySetup? [Online]. Available: http://www.linksys.com/servlet/Satellite?c=L_Promotion_C2&childpagename=US%2FLayout&cid=1121874561907&pagename=Linksys%2FCommon%2FVisitorWrapper

[9] J. H. Saltzer and M. D. Schroeder, "*The Protection of Information in Computer Systems,*" In Proceedings of the IEEE, vol. 63, pp. 1278-1308, Sep. 1975.

[10] P. Baran, "*Security, secrecy, and tamper-free considerations*," On Distributed Communications, vol. 9, Aug. 1964.

[11] R. Anderson, *Security Engineering: A Guide to Building Dependable Distributed Systems.* New York: John Wiley & Sons, Inc. pp. 240. 2001

[12] A. Whitten and J. D. Tygar, "*Usability of Security: A Case Study,*" Carnegie Mellon University School of Computer Science Technical Report CMU-CS-98-155, Dec. 1998.