AN INTERFERENCE-BASED PREVENTION MECHANISM AGAINST WEP ATTACK FOR 802.11B NETWORK

Wen-Chuan Hsieh[1], Yi-Hsien Chiu[2] and Chi-Chun Lo[3]
[1]Shu-Te University, wch@mail.stu.edu.tw; [2]National Yunlin University of Science & Technology, g9223701@yuntech.edu.tw; [3]National Chiao-Tung University, cclo@cc.nctu.edu.tw

Abstract: WEP has a potential vulnerability that stems from its adaptation of RC4 algorithm. As indicated by prior researches, given a sufficient collection of packets, speculation on shared key is possible by extracting IVs that matched a specific pattern. With the primary protection becomes void, there is a pressing need for new WLAN security measure. However, establishing new security protocol requires considerable time and financial resources. This research proposes an alternative solution to WEP hacking, without modification on present wireless settings, called Interference-Based Prevention Mechanism.

Key words: IEEE 802.11, Wireless Local Area Network (WLAN), RC4, Wired Equivalent Privacy (WEP)

1. INTRODUCTION

Wireless Local Area Network (WLAN) offers organizations and users a both convenient and flexible way of communication. It provides mobility, increases productivity, and lowers installation costs. However, WLAN is susceptible to attacks due to the use of radio frequency which incurs data exposure. WLAN does not have the same physical structure as LANs do, and therefore are more vulnerable to unauthorized access. While access points (AP) offers convenient and flexible way of communication, the fact that they are connected to internal network exacerbates security problem. Without additional protection, APs can as well be entries for potential attackers.

To enhance data security of wireless transmission to the level of a wired network, IEEE 802.11 standard defined WEP (Wired Equivalent Privacy), which encrypts traffics between clients and AP. However, WEP has a potential weakness stems from its adaptation of RC4 algorithm, which utilizes a plain IV (initial vector) as part of key stream computation. As indicated by prior researches[1,2,3], given a sufficient collection of packets, speculation on shared key is possible by extracting IVs that matched a specific pattern. That is, any anyone with WEP attack tools, such as AirSnort[4] and WEPCrack[5], can obtain the key in a matter of hours or days.

Obviously, WLAN suffers severe security problem and it offers merely limited privacy guarantee. Installing WLAN incurs tremendous risks since APs can as well be entries for potential attackers into internal network. With WEP, the primary WLAN protection, being compromised, the condition of wireless security is considered critical. With the primary protection becomes void, there is pressing need for new WLAN security measure.

However, security protocol requires considerable resources to upgrade legacy network for the supporting features. Instead of altering or replacing present WLAN, this research offers an alternative solution without modifying the present setting. This research proposes an Interference-Based Prevention Mechanism which is proven effective in preventing adversaries from deducing WEP key based on weak key detection.

2. THEORETICAL BACKGROUND

2.1 Wireless Security

The most prominent feature about WLAN is the absence of wires and its mobility. As compares to the traditional network, WLAN requires no complicate configuration on its physical topology. Its prestigious nature of mobility is made possible by transmitting data using radio frequency. However, as data travels through the air, it can easily be tapped by any one including unauthenticated personnel using sniffer.

Many attacks on traditional network also applied to wireless environment; for instance, DOS attack, session hijack and man-in-the-middle. Also, unauthorized clients may attempt to access WLAN without authorization. Since WLAN does not constraint users to physical connection ports, users are able to access the AP anywhere. Borisov et al.[6] conducted a detailed research on insecurity of 802.11.

As defined in IEEE 802.11b standard, WEP is applied to encrypt data so that it becomes unreadable to the intruder. Despite the effort, WEP is recently proved insecure since its key can be stolen or cracked using tools such

as AirSnort. In addition, most APs are deployed with WEP setting switched off by default. APs offers MAC filter as a supplementary security feature to WEP; however, keeping track of MAC addresses list is both time consuming and inconvenient.

Because of the weaknesses in WEP security, several entities are developing stronger security technology, such as TKIP (Temporal Key Integrity Protocol) [7,8] and 802.1X[9,10]. TKIP is proposed, as part of wireless standard 802.11i, to replace WEP. 802.1X is an IEEE standard for EAP encapsulation over wired or wireless Ethernet. 802.1X is also known as EAPoL (EAP over LAN). However, the fact that majority of the legacy wireless hardware are 802.11b based requires potential adopters to either upgrade firmware or even replace the incompatible devices. The cost of such hardware and software renovation and reconfiguration is just too expensive for entities with limited budget. Therefore, for the mean time, current WLAN is considered insecure.

2.2 Wired Equivalent Privacy (WEP)

The concept of WEP is to prevent eavesdroppers by encrypting data transmitted over the WLAN from one point to another. Data encryption protects the vulnerable wireless link between clients and access points; that is, WEP does not offer end-to-end security because AP decrypts the frames before passing them to destinations that are beyond WLAN.

WEP adopts RC4 algorithm, a stream cipher, developed by RSA security. "A stream cipher operates by expanding a short key into an infinite pseudo-random key stream. The sender XORs the key stream with the plaintext to produce cipher text. The receiver has a copy of the same key, and uses it to generate identical key stream. XORing the key stream with the cipher text yields the original plaintext"[11]. In other words, RC4 is a symmetric algorithm relies on a single shared key that is used at one end to encrypt plain text into cipher text, and decrypt it at the other end[12].

Current WEP implementations support key length up to 64 bits and 128 bits; technically, the key length of both version are shorten by 24bits due to the use of plaintext Initial Vector (IV). In this research context, a key (or key combination) is a series of ASCII bytes often presented in hexadecimal; whereas a key value is one byte (8 bits) out of the total combination. Figure 1 shows a WEP encrypted frame which consists of IV(24 bits), padding(6 bits), key index(2 bits), encrypted message and Integrity Checksum Value (ICV)(32 bits). Note that the frame is transferred with the first 32 bits in plaintext and the rest of the body encrypted. This is because a sender generates IV, either incrementally or randomly, as part of inputs to encryption process. That is, the receiver must know the exact IV to decrypt the frame.

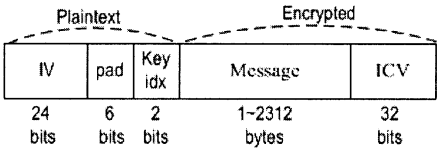

Figure 1. WEP encrypted frame format

As indicated by the length of key index in the diagram, WEP can have up to 4 (2^2) keys. However, using shared static keys can be dangerous. Therefore, the purpose of constantly changing IV is to achieve the effect as if having a greater number (2^{24}) of key combinations. This gives WEP the capability of encrypting each frame with different keys (packet key).

Figure 2 illustrates WEP encryption process which starts by generating IV and selecting a predefined key. Next, RC4 uses both IV and chosen key (k) as inputs to generate key stream. Then, plaintext message (M), along with its ICV, is combined with key stream through a bitwise XOR process, which produces ciphertext (C). Upon sending the encrypted frame, WEP appends IV in clear to the front of the frame. The encryption process can be summarized as following formula: C = (M, crc32(M)) XOR RC4(IV, k)

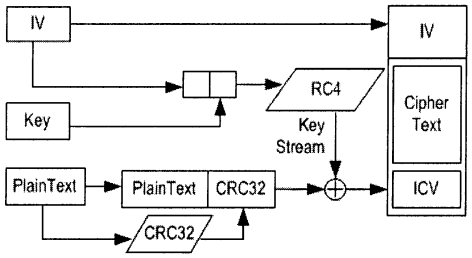

Figure 2. WEP encryption process

To decrypt, the receiving station uses the first 32bits IV and the shared key (k) as indicated by key index bits to generate the same key stream that encrypted the frame. Next, WEP XOR key stream with ciphertext (C), along with it ICV, to retrieve the plaintext (M). Note that, plaintext has ICV attached at the end. Finally, WEP computes plaintext, without ICV, CRC32 and compares the output with the ICV.

Wireless environment is prone to interference; hence, data may be lost or damaged before reaching the destination. To ensure data integrity, sender computes CRC32 against the plaintext message and inserts the output (32 bits) at the back of the message prior to encryption. The receiver ensures data integrity by matching ICV of decrypted frame with the CRC32 result done locally with the resolved message. Frames with disconfirmed check-

sum will be discarded. The decryption process can be summarized as following formula based on the encryption formula:
(M, crc32(M)) = C XOR RC4(IV, k)

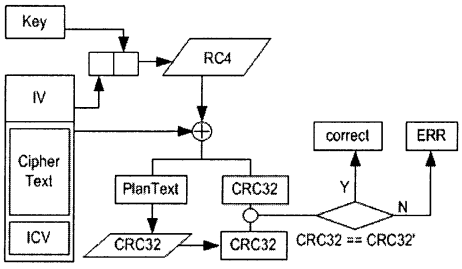

Figure 3. WEP decryption procedure

2.3 WEP Vulnerability

Though combining IV into key stream computation increases key complexity so that it appears unpredictable, the reality that IV has to be transferred in clear may divulge WEP key as first discovered in the research undertaken by Fluhrer, Martin and Shamir[1]. Specifically, frames with IV that matched (B+3, 255, X) form, where B points to the position of the key value in the combination and X can be any value between 0 and 255, may reveal key values. The probability of retrieving the right key value from the frame is 5%[13]. Given sufficient time and traffic, one is able to obtain the WEP within hours or days. For instance, an IV (4, 255, 31) may resolve the value of the K[1], where IV (7, 255, 72) may resolve the K[4] (fig 4). Often, attackers determine the key combinations by running statistic on all the potential key values computed from frames that matched such pattern.

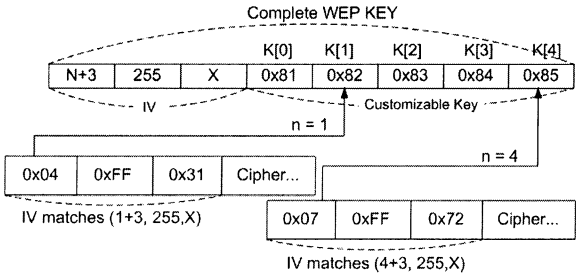

Figure 4. IV pattern resolving key combination

Part of the key value extracting concept is based on the nature of XOR. Suppose C is the result of P XOR K, then we are able to retrieve K by XOR C with P. In the case of WEP, the idea is extended and is much complicated due to RC4 algorithm; nevertheless, the fundamental idea is the same. That is, the initial step of cracking WEP key is to obtain ciphertext with its matching plaintext, which is almost readily available. As defined in 802.11 standard, any frames of type ARP or IP has to begin with 0xAA (known as SNAP). In IPX environment, 0xFF or 0xE0 is used instead. In fact, majority of the data transferred in WALN is in either format.

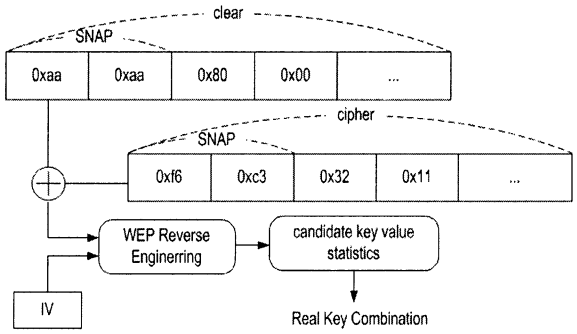

Figure 5. XOR plaintext and ciphertext to resolve key value

All in all, to crack WEP, one must first capture as much frames that matched the specified pattern as possible. Then, for each of the captured frame, XOR the first byte of the ciphertext with 0xAA to obtain the exact key stream that were used during encryption. By reverse-engineering RC4, attacker would be able to retrieve the key value (fig 5). Please refer to Fluhrer's study for detailed explanation on specific algorithms. Seth Fogie[13] has published an article which describes detailed steps of WEP cracking. Also, WEP attack implementation can be found in the research done by Stubblefield et al[2].

3. INTERFERENCE-BASED PREVENTION MECHANISM

The major WEP vulnerability is the fact that attacker is able to extract the key from gathered frames. Usually, statistics is used to assists in determining the real key values from the candidates. The real key value often has the highest occurrence among all. Therefore, it is reasonable to conclude that the resulting key is based on the amount and quality of the frames. That is, the

attacker is unlikely to get the right key combination if traffic is scarce or frames reveal more false key values than that of the right ones.

Since it is impossible and unreasonable to keep WLAN traffics from increasing, we propose that the best option to prevent attacker from getting the correct key value is by poisoning the traffic with frames that are deliberately tailored to generate false result.

Based on the understanding of the frames that the attackers are interested in and the logics of detecting the key, this research devised an innovative solution called Interference-Based Prevention Mechanism (IBPM). IBPM creates interference effect by injecting spoofed frames to delude the attacker resulting in inaccurate statistic. Since injecting frames increases traffic load, hence, an effective and space-efficient method must be applied.

IBPM utilizes the same technique similar to WEP crackers. That is, IBPM monitors the traffic and keeps computing the key values. The difference is that IBPM is implemented in a client station within a WEP protected WLAN; therefore, it is assumed that IBPM station possesses the key as a legitimate user. Having the key gives it the capability of interfering network traffic in advance. Figure 6 shows IBPM generates spoofed frames whenever the speculated key value matches the real key value (we refer such event as weak-key occurrence). Consequently, the automated statistic program at the offense side takes those frames into account and increments false key values. What actually happened is that, IBPM pollutes attacker's statistic in a way that causes false key values to increase to prevent real key value becoming distinct. Since IBPM has disrupted the statistic long before it reveals the real key value, WEP is, therefore, secured. This research proposes several schemes, which are discussed under interference schemes section, to distribute spoofed frames that generates false key across all possible key values.

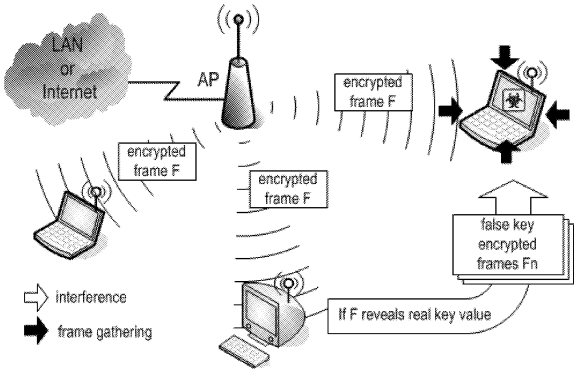

Figure 6. Interference generation

4. INTERFERENCE SCHEMES

The effect of interference is accomplished by increasing the tally of false key value whenever a weak-key is detected by IBPM. For instance, a weak-key 0xBB is detected, one may decide to increase all false-key tallies range form 0x00 to 0xFF excluding 0xBB. However, incrementing the tally arbitrarily incurs flaw that may eventually allows the attacker to discern the fixed pattern in the resulted statistic. Interference not only conceals the key, but also should prevent attackers from speculating the key based on the spoiled statistic again. Therefore, a sound interference scheme does not exhibits traceable patterns. This research proposes three schemes with each takes a different approach to poison the traffic.

4.1 Random Distribution Interference

This scheme randomly selects any amount of key values from the false set. The increment scale can also be any number. However, it is recommended to use a scale less than or equals to 3, because drastic change may ultimately cause the real key value to become the least and apparent. The scale can also be randomly assigned given a specified range.

4.2 Perfect Probability Distribution Interference

Under such scheme, every false key value is given a 50% probability of becoming candidates for increment. That is, there are roughly half of the false set members will be incremented upon each weak-key occurrence. In this case, we use a fixed increment scale of 2. This scheme is able to maintain a stable increasing rate of 1 (2 x 0.5), just like the detected weak-key, while avoiding exposing the pattern when using scale of 1. Therefore, the overall average increasing rate remains constant yet leaves no traces.

4.3 Mixed Interference

Though a properly designed scheme should avoid revealing a traceable pattern; nevertheless, it is still recommended to implement multiple schemes and have each scheduled or randomly assigned to activate upon weak key occurrence. The advantage of such mixed scheme over the others is that it prevents attackers from recognizing a fixed pattern due to adoption of a single scheme.

5. IMPLEMENTATION

5.1 System Requirements

As mentioned earlier, IBPM requires no change on the legacy network configuration and is compatible to any WEP-enabled 802.11 WLAN. The IBPM-enabled device (preferably a desktop PC) appears just like any other regular wireless clients; therefore, attackers are unlikely to realize the intension behind such deployment. As shown in IBPM system framework (fig 7), IBPM joined the WLAN as a member client station which issues bogus frames upon weak key occurrences. At the same time, the attacker, being unaware of the spoofed frames, keeps gathering the frames.

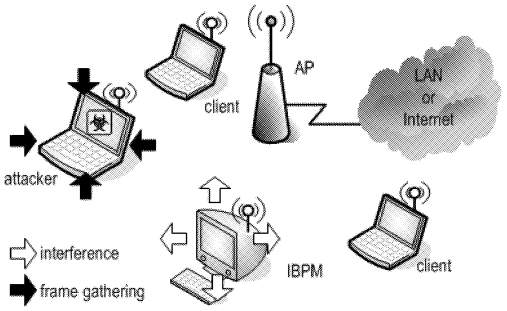

Figure 7. IBPM System Framework

IBPM involves both proactive and passive activities which include traffic sniffing and injection. Presently, this research has implemented an experimental system under Linux. Various wireless drivers are available in the open-source community[14,15,16] with each offers slightly different capabilities. This research has modified and integrated some of the drivers in achieving features to support both monitor WLAN traffics and send frames with arbitrary format, including WEP encrypted, through individual wireless network interface cards.

This research implemented IBPM using Python and C libraries under Redhat 9 Linux. The IBPM machine is equipped with two wireless network interfaces: one for sniffing frames and the other is used to inject spoofed frames whenever weak-key is detected.

5.2 Demonstration

To demonstrate the effectiveness of IBPM, two independent WEP attacks were launched against the experimental WLAN in the lab. At the end of each attack, the offender's statistical result is captured. This demonstration adopted perfect probability interference scheme. Since WEP-128 is as vulnerable as WEP-64 despite of its extended key length, therefore WEP-64 is applied just to illustrate the concept due to space limitation. AP is configured with WEP key setting as K = {76, 210, 126, 196} and 24. Figure 8 shows the result of the statistical result of the first attack without IBPM. Note that the thick line indicates the real key value.

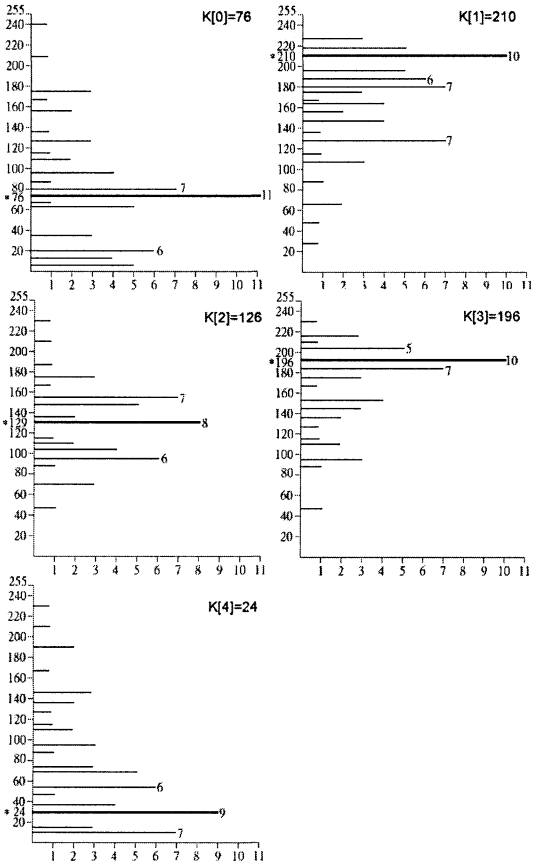

Figure 8. K[0]~K[4] statistic result without interference

Clearly, the attacker can easily points out the real key value based on the statistical result. Evidently, each of the real key value stands out prominently. In contrasting to the previous test, the result captured in the second

experiment with IBPM conceals the real key values (fig 9). In addition, the overall distribution is almost random and there is no fixed pattern to follow. As for better observation, we deliberately thicken the line of the real key. In reality, the attackers will not be able to determine the real key value from such random formed statistical result.

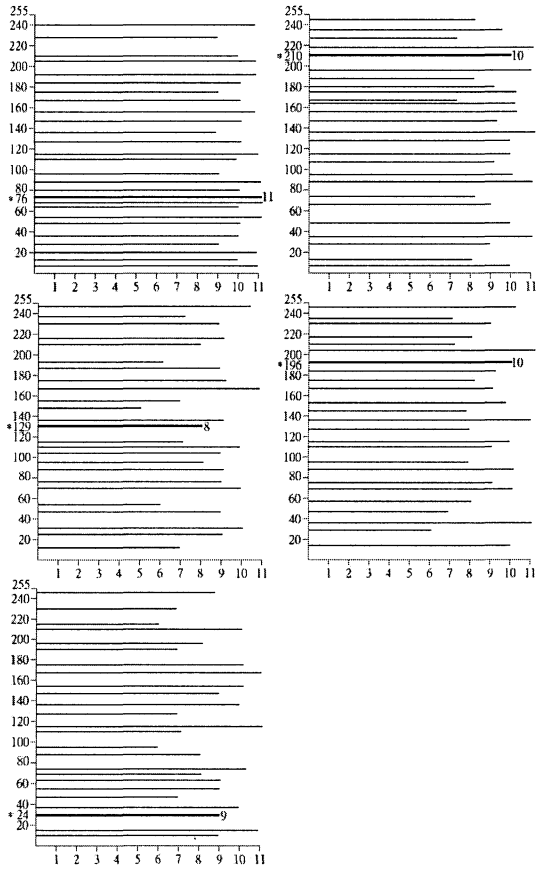

Figure 9. K[0] and K[1] statistic result with interference

6. CONCLUSION

This research covered discussion on WEP encryption, its vulnerability and basic concept on the technique applied to extract key from frames that matched the weak-key form. More importantly, we developed Interference-Based Prevention Mechanism (IBPM), which is proven to be effective in preventing attackers from speculating WEP key by means of frame gather-

ing. Presently, two interference schemes are proposed. However, for further studies, more effort should be devoted in testing and developing of new schemes.

REFERENCES

1. S. Fluhrer, I. Mantin & A. Shamir, "Weaknesses in the key scheduling algorithm of RC4". Eighth Annual Workshop on Selected Areas in Cryptography, August 2001.
2. A. Stubblefield, J. Ioannidis and A. Rubin, "Using the Fuhrer, Mantin and Shamir Attack to Break WEP", AT&T Labs Technical Report TD-4ZCPZZ (08/2001).
3. Liao Jyun-Ruei , "Analysis Data Security in Wired Equivalent Privacy Algorithm for Wireless Local Area Network", 2002
4. AirSnort, http://airsnort.shmoo.com/
5. WEPCrack, http://wepcrack.sourceforge.net/
6. Nikita Borisov, Ian Goldberg & David Wagner, "Intercepting Mobile Communications: the insecurity of 802.11", 7th Annual International Conference on Mobile Computing and Networking
7. Mark, Joseph & Edwards, "Increasing Wireless Security with TKIP", 2002, URL: http://www.winnetmag.com/Articles/Print.cfm?ArticleID=27064
8. The TECH FAQ, "What is TKIP (Temporal Key Integrity Protocol)?", URL: http://www.tech-faq.com/wireless-networks/tkip-temporal-key-integrity-protocol.shtml
9. Joel Snyder & Network World Global Test Alliance, "What is 802.1X", 2002, URL: http://www.nwfusion.com/research/2002/0506whatisit.html
10. Jim Geier, "802.1X Offers Authentication and Key Management", 2002, URL: http://www.wi-fiplanet.com/tutorials/article.php/1041171
11. Nikita Borisov, Ian Goldberg & David Wagner, "Security of the WEP algorithm", URL: http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html
12. Nuruddin Mohd. Alamgir, "Insecurities of WEP and Securing the Wireless Networks", June 2002, URL: http://www.giac.org/practical/nuruddin_alamgir_gsec.doc
13. Seth Fogie, "Cracking WEP", July 2002, URL: http://www.informit.com/articles/printerfriendly.asp?p=27666
14. SourceForge, URL: http://sourceforge.net
15. AirJack, URL: http://sourceforge.net/projects/airjack/
16. HostAP, URL: http://hostap.epitest.fi/