Security mechanisms, attacks and security enhancements for the IEEE 802.11 WLANs

Yang Xiao*

Department of Computer Science,
University of Alabama,
Tuscaloosa, AL 35487-0290, USA
E-mail: yangxiao@cs.ua.edu
*Corresponding author

Chaitanya Bandela

Department of Computer Science,
Georgia State University,
Atlanta, GA, USA
E-mail: chaitan78@hotmail.com

Xiaojiang (James) Du

Department of Computer Science,
North Dakota State University,
Fargo, ND, USA
E-mail: Xiaojiang.Du@ndsu.edu

Yi Pan

Department of Computer Science,
Georgia State University,
Atlanta, GA, USA
E-mail: pan@cs.gsu.edu

Edilbert Kamal Dass

Department of Computer Science,
The University of Memphis,
Memphis, USA
E-mail: atchkamalbe@yahoo.com

Abstract: Wired Equivalent Privacy (WEP) protocol was adopted to protect authorised users from unauthorised access and eavesdropping in the IEEE 802.11 wireless LANs. It has been proven that the WEP protocol fails to provide data confidentiality and authentication. This paper first introduces the WEP as well as all kinds of attacks. Then, two approaches to enhance the WEP are proposed to overcome some known vulnerabilities and thus to provide better data confidentiality and authentication. Finally, simulation methodology is presented and simulation results are provided. Our studies show that the proposed enhancements provide better data confidentiality with some degree of computing cost as the trade-off.

Keywords: IEEE 802.11; security; Wired Equivalent Privacy (WEP); wireless LANs.

Reference to this paper should be made as follows: Xiao, Y., Bandela, C., Du, X., Pan, Y. and Dass, E.K. (2006) 'Security mechanisms, attacks and security enhancements for the IEEE 802.11 WLANs', *Int. J. Wireless and Mobile Computing*, Vol. 1, Nos. 3/4, pp.276–288.

Biographical notes: Yang Xiao is an IEEE Senior Member. He was a voting member of IEEE 802.11 Working Group from 2001 to 2004. He currently serves as an Editor-in-Chief for *International Journal of Security and Networks* and *International Journal of Sensor Networks*. He currently serves as an Associate Editor or on editorial boards for five other journals. He served as a guest editor for eight journal special issues. His research areas include wireless networks, mobile computing and network security. He has published more than 70 journal papers with 40 papers published in various IEEE journals. He has edited/co-edited ten books in wireless networks and security.

Chaitanya Bandela is a graduate student of Georgia State University.

Xiaojiang (James) Du is an Assistant Professor in the Department of Computer Science, North Dakota State University. He received his BE from Tsinghua University, China in 1996 and his MS and PhD from the University of Maryland, College Park in 2002 and 2003, respectively, all in Electrical Engineering. His research interests are heterogeneous wireless sensor networks, security, wireless networks and computer networks. He is an Associate Editor of Wiley Wireless Communication and Mobile Computing. He is (was) the Chair of Computer and Network Security Symposium of IEEE/ACM International Wireless Communication and Mobile Computing Conference 2007 (2006).

Yi Pan is the Chair and a Full-time Professor in the Department of Computer Science at Georgia State University. He received his BEng and MEng in Computer Engineering from Tsinghua University, China, in 1982 and 1984, respectively and his PhD in Computer Science from the University of Pittsburgh, USA, in 1991. His research interests include parallel and distributed computing, optical networks, wireless networks and bioinformatics. He has published more than 80 journal papers with 29 papers published in various IEEE journals. He has also co-edited over 20 books (including proceedings) and contributed several book chapters. He has served as an Editor-in-chief or editorial board member for eight journals including three IEEE Transactions and a guest editor for seven special issues.

Edilbert Kamal Dass is a graduate student of The University of Memphis.

1 Introduction

The IEEE 802.11 standard (1999) specifies Wired Equivalent Privacy (WEP), a wired LAN equivalent data confidentiality algorithm, to protect authorised users for security purposes. Unfortunately, the WEP protocol seriously fails to accomplish its security goals and it has been proved that prominent flaws exist (Borisov et al., 2001). Therefore, the growing popularity of the IEEE 802.11 products has been met with a growing concern for security reasons.

Wireless systems are faced with the same problems as wired systems in security aspects, that is, every system needs proper authentication, privacy of transmission and protection against attacks. Furthermore, compared with wired systems, wireless systems have limited physical security to prevent unauthorised access and security becomes more difficult. For example, wireless LANs can be used in corporate environments, where employees are presumed to have unrestricted access to the network. Guests or neighbouring offices that share the same air medium should not be allowed access network resources although they are close.

The WEP protocol employs the well-known and believed secure RC4 cipher (Rivest, 1992), a symmetrical cryptographic algorithm, with either a 40-bit or 128-bit key (Borisov et al., 2001; IEEE 802.11 WG, 1999). *Plaintext* is referred as to an originally intelligible message and *ciphertext* is referred as to the encrypted message. *Encryption* is the process of converting the plaintext into the ciphertext with a key, which is a value independent of the plaintext including a sequence of bits used throughout the encryption process. A specific key should be chosen in such a way of keeping it secret without compromising the confidentiality of their respective data. In a symmetric key cryptographic algorithm, the same key is used in the encryption process as well as the decryption process. Due to faulty implementation of the RC4 cipher in the WEP protocol, many security flaws were discovered based on known drawbacks of the RC4 cipher. The flaws give rise to a number of attacks, both passive and active, that allow eavesdropping and tampering with the wireless transmissions (Arbaugh et al., 2001; Borisov et al., 2001; Fluhrer et al., 2001; Stubblefield et al., 2001; Verton, 2001).

In this paper, we provide a comprehensive survey on vulnerabilities in the IEEE 802.11 WLANs and we propose two security enhancements on the WEP to overcome some known vulnerabilities and thus to provide better data confidentiality and authentication. In the first enhancement, a keyed-message authentication code that prevents an intruder from tampering with messages in transit is adopted, as well as a new revised authentication scheme to avoid authentication spoofing and reduce replay attacks. In the second enhancement, private IV is adopted as well as using day and session keys that counters several attacks. Simulation/experimental methodology as well as simulation/experimental results are provided.

The rest of this paper is organised as follows. Section 2 introduces association, authentication and encryption procedures in the IEEE 802.11 wireless LANs. Then we introduce the WEP protocol in Section 3. All kinds of attacks are described in Section 4. Two security enhancements are presented in Section 5. Simulation Methodology and simulation results are presented in Sections 6 and 7, respectively. Finally, we conclude this paper in Section 8.

2 Association, authentication and encryption

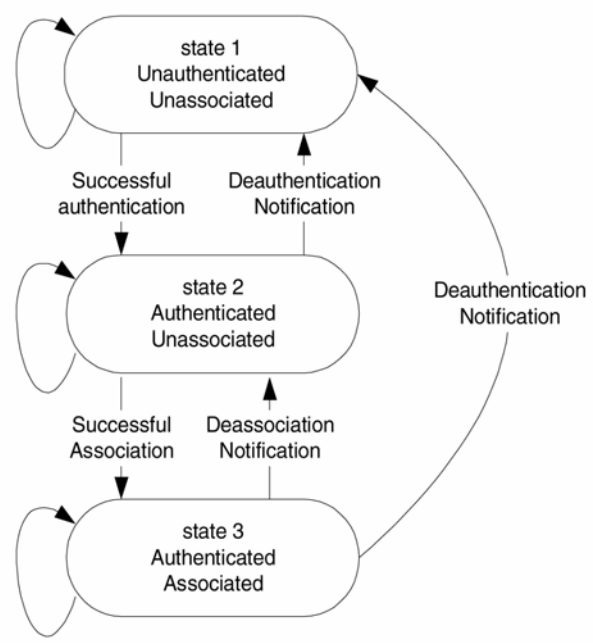
Each station in the IEEE 802.11 network may be a station or an Access Point (AP). There are two operation modes in the IEEE 802.11 standard: infrastructure mode and ad hoc mode (IEEE 802.11 WG, 1999). In an infrastructure mode network, an AP is present, whereas in an ad hoc mode network, no AP is present. A WLAN cell is called a Basic Service Set (BSS), which includes an AP and multiple associated stations. In ad hoc mode, AP is not present and the WLAN cell is referred as to Independent BSS (IBSS). Multiple BSSs connect a Distributed System (DS) and

form an Extended Service Set (ESS). The DS is the backbone of the WLAN and may be constructed over wired or wireless connection, such as Ethernet. The APs in an ESS communicate among themselves to form relay between the BSS domains, through the DS.

To become part of the BSS, a station needs to go through authentication and association procedures first. There are two kinds of services: station services and distributed services. Station services include authentication, de-authentication, privacy and data delivery. Distribution services include association, reassociation and de-association. In this section, we introduce these services.

Figure 1 shows the state diagram of a station. The station is first in state 1, that is, unauthenticated and unassociated. The station goes to state 2 after successful authentication and then goes to state 3 after successful association. On the other hand, the station in state 3 goes to state 2 after de-association and then goes to state 1 after de-authentication. A station in state 1 has nothing related to the network/BSS. A station in state 3 becomes part of the network and begins enjoying the services: transmissions and receptions of frames.

Figure 1 State diagram of a station



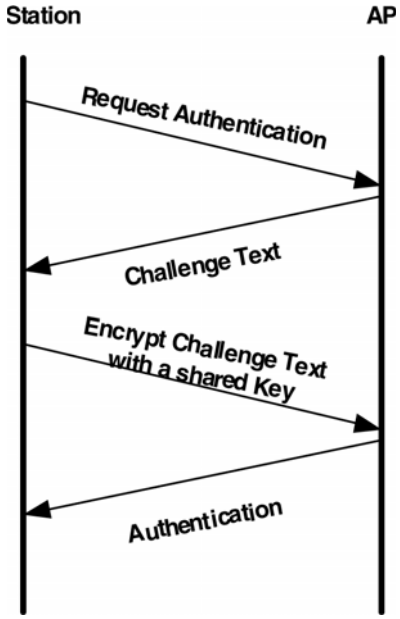
2.1 Authentication

The 802.11 standard defines the authentication service to control access to the WLAN. This service is used by all stations to establish their identities before commencing communications. This holds true for IBSS as well as ESS networks. If a mutually acceptable level of authentication is not established, communication ceases to take place. The authentication is to verify identification between a station and the AP. A station can be authenticated with many APs simultaneously, but it can only be associated with one AP. De-authentication is to notify the termination of an authentication relation by a station. Authentication is to verify a station, which has the right key to get into the network.

The authentication is a link level authentication. There are two kinds of authentication services: open system authentication and shared key authentication. Open system authentication is the default method, which is a very simple and two-step process. First, the station wanting to authenticate with another station (mostly the AP) sends an authentication management frame containing the sender station's identity. The receiving station then sends back a frame alerting whether it recognises the identity of the authenticating station. In the open system authentication, any station may become authenticated. In other words, the open system authentication is not authentication at all and therefore it is called null authentication.

Shared key authentication requires implementation of the WEP option. In this system, the identity of a station is demonstrated by the knowledge of a shared, secret WEP encryption key. This key is assumed to be delivered to each station through a secure channel, maybe manually, independent of the 802.11 network. The procedure of the shared key authentication is described in Figure 2, in which a station send a authentication request to the AP, which sends the challenge text back to the station. The challenge text is used by the AP to exam whether the station has a shared key or not. If the station has the key, it will use it to encrypt the challenge text and sends to the AP. The AP then decrypts it with the share key known by the AP. If the result of the decryption is the same as the original challenge text, the AP knows that the station has the same key and sends a successful authentication acknowledgment message back to the station.

Figure 2 Authentication procedure



The de-authentication service is used to eliminate a previously authorised station from further access to the network. Once a station is de-authenticated, it can no longer access the WLAN without performing the authentication function again. De-authentication is not a request, but a notification and can be invoked by either authenticated party (the station or the AP). De-authentication cannot be refused by either party.

2.2 Association services

Association services include association, reassociation and disassociation. The association service is used for a station to join a BSS. Each station must become associated with a BSS before it is allowed to send data in the BSS. During the association procedure, a station sends an associate message to the AP, which responses the result of the association procedure with the Association ID (AID). If a station moves from one BSS to another, within an ESS, reassociation procedure is needed. The reassociation procedure is similar to the association procedure except that it involves the information about the previous AP with which the station was associated. The disassociation service is invoked whenever an existing association is to be terminated. The service is used either to force a station to eliminate an association by the AP or for a station to inform the AP that it no longer requires the services. Disassociation is a notification, but not a request. Therefore, it cannot be refused by either party. A station shall attempt to disassociate whenever it leaves a network.

2.3 Privacy

In a wired LAN, only those stations physically connected to the wire may access LAN traffic. But with a wireless access medium, any compliant station may access all like-PHY 802.11 traffic that is within the range. The privacy service of IEEE 802.11 is designed to provide an equivalent level of the protection for data on the WLAN as that provided by a wired network. This service protects that data only as it traverses the wireless medium. It is not designed to provide complete protection of data between applications running over a mixed network. This seriously impacts the security level of a wireless link to a wired network. IEEE 802.11 counters this problem by offering a privacy service option that raises the security of the 802.11 network to that of a wired network. The privacy service, applying to all data frames and some authentication management frames, is an encryption algorithm based on the 802.11 WEP algorithm. The algorithm is not designed for ultimate security, but rather, to be at least as secure as a wired system. Only the data is encrypted and the Medium Access Control (MAC) layer is not changed after the encryption. The WEP does not protect from traffic analysis. RC4 – symmetric stream cipher algorithm with variable key length is used and the encryption and decryption algorithms are the same as well as with the same key. The WEP is introduced in more detail in the next section.

2.4 Security considerations in deploying 802.11 WLAN

For integration with existing LANs, security mechanisms in the IEEE 802.11 WLANs should be equivalent to existing mechanisms in wire-based networks in order to avoid risking security lapses. Wired networks are intrinsically secure to some extent since network jacks that serve as an access to the network are located in buildings already secured from unauthorised access. Therefore, an intruder must gain access to the building before attempting intrusion into the wired LAN. In contrast, a wireless AP with open system authentication may be accessed from off the premises, from an adjacent neighbouring area. Hence, as a precaution, several methods are used to secure access to the AP and isolate the AP from the internal private network prior to user authentication into the network domain.

The IEEE 802.11 networks secure access to an AP through three basic methods. One or all of these methods may be implemented as a security solution. Network access control can be implemented using a *Service Set Identifier* (SSID) associated with an AP or a group of APs. The SSID provides a mechanism to separate a wireless network into multiple networks serviced by one or more APs. To access the WLAN, client stations must be configured with the correct SSID. Without the knowledge of the AP's SSID, a mobile station cannot associate with it. This could be a simple way of securing an AP by not revealing the SSID to unauthorised stations.

This minimal security is compromised if the AP is configured to broadcast its SSID, which might be a requirement where it is cumbersome or restrictive to configure the client stations accessing the AP. When this broadcast feature is enabled, any station is allowed to scan for the SSID and access the AP. In addition, since users typically have access to the configuration of client stations with the appropriate SSIDs, they are widely known and easily shared.

While an AP can be identified by an SSID, a client station can be identified by the unique MAC address of its 802.11 network card. This could serve as a security measure, when each AP is programmed to filter stations requesting association based on their MAC addresses. If a station's MAC address is not known to the AP, it is not allowed to associate with the AP. This method provides good security, but is best suited to small networks. Each AP must be manually programmed with a list of MAC addresses and the list must be kept up to date. This administrative overhead limits the scalability of this approach. Furthermore, this is available software to change a station's MAC address in the viewpoint of an outsider.

Since it is easier to intercept wireless transmissions than transmissions over a wired network, to minimise the risk of security breach, the IEEE 802.11 standard specifies WEP for encryption and authentication. The WEP provides encrypted communication using an encryption key between the client station and an AP. All client stations and APs on a BSS use the same key to encrypt and decrypt data. The key resides in the client station and in the AP.

3 Wired equivalent privacy

The WEP algorithm provides the 802.11 WLANs functionality of authentication and privacy services. The IEEE 802.11 claims the WEP algorithm to be reasonably strong to withstand brute-force attack to find the secret key. It is self-synchronising, meaning that once the WEP

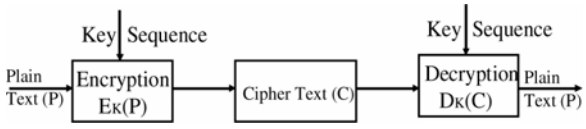
option is turned on, it automatically encrypts each message frame travelling through the medium. The WEP is efficient by making it suitable to be implemented in either hardware or software on wireless devices, which typically have limited computational power when compared to its counterparts in a wired LAN.

The WEP is used in both authentication and data privacy. In authentication, encryption on the message (challenge text) is done by the station to prove that it has the right key to get into the network, shown in Figure 2, in data privacy, encryption on message is done by the station to prevent eavesdropping by un-authorised stations.

3.1 WEP, a symmetric key algorithm

The WEP is a symmetric key algorithm, shown in Figure 3. A symmetric key algorithm is one where the same key is used in both encryption and decryption. When the plain text (P) is encrypted with an encryption algorithm using the key K, cipher text (C) is obtained, that is, $C = E_K(P)$, where $E_K(\cdot)$ denotes the encryption algorithm/function. When the cipher text (C) is decrypted using the same key K, the original plaintext is obtained, that is, $D_K(C) = P$, where $D_K(\cdot)$ denotes the decryption algorithm/function. Obviously, we have the relationship $D_K[E_K(P)] = P$. The key K is shared among the AP and all member stations of a BSS.

Figure 3 Symmetric key encryption/decryption process



3.2 Encryption

Figure 4 shows the WEP encryption block diagram and how the WEP encryption algorithm is applied to the plaintext (P). We explain Figure 4 as follows.

- An *integrity checksum value* [ICV(P)] is calculated on the P using the cyclic redundancy check 32 [CRC (32)]. The P concatenates ICV(P) to form the M. In other words, we have $M = \{P, \text{ICV}(P)\}$.
- An *initialisation vector* (IV) is chosen as a 24 bit random number by the sending station and concatenates with the secret key (K), 40-bit in length, to form a *seed*.
- The seed is input to the RC4 *pseudorandom number generator* (PRNG), which outputs a *key sequence* of pseudorandom bytes equal in length to the M. The RC4 will be introduced in a later subsection.
- The key sequence and the M are XORed ($\oplus$) to obtain the cipher text (C). The length of the key sequence and the M are the same so that they can be XORed to get the result with the same length.
- In other words, we have $C = \{P, \text{ICV}(P)\} \oplus \text{RC4(IV, } K)$.

The {IV, P, ICV(P)} triplet forms the actual data to be sent in the data frame securely by the sender station to the receiving station. The IV is sent as it is since it is assumed that an intruder can gain no useful information from its knowledge and since the recipient station must know its value to perform the decryption process.

Figure 4 WEP encryption block diagram

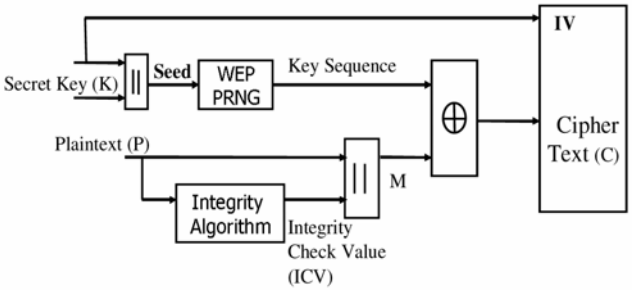
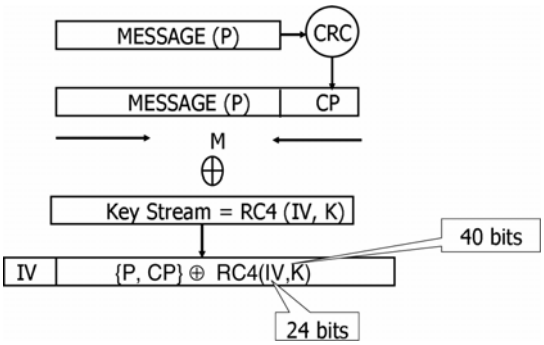


Figure 5 shows the actual encrypted packet which will be sent to the AP. The length of the plaintext and the length of the key stream are the same. It is noted that when the encrypted message is sent, the IV is also sent along with it. The reason behind this is that during the decryption process, the receiving station requires it to generate the key stream in order to carry out the decryption process.

Figure 5 WEP encrypted packet



3.3 Decryption

Decryption is the exactly reverse process of encryption, where the ciphertext is converted into the plaintext.

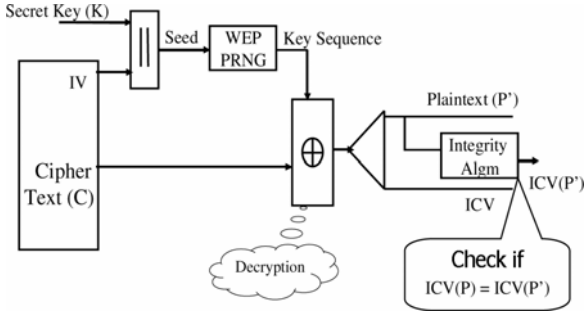
Figure 6 shows the WEP decryption block diagram. The IV of the incoming message is used to generate the key sequence necessary to decipher the incoming encrypted message. The RC4 PRNG plays a critical role since it regenerates the same key sequence as when used in encryption. This is because the same pair of the IV and the secret key (K) is passed to it as the input, as in the encryption process. The key sequence is XORed with the cipher text to extract the plaintext (P') and ICV(P). Correct decryption is verified by performing an ICV (P') and comparing it with the transmitted ICV (P). If ICV (P') is not equal to ICV(P), the received frame is rejected.

3.4 RC4

RC4 is a variable key-size stream cipher developed in 1987 by Ron Rivest for RSA Data Security Inc. (RSADSI). For seven years it was proprietary and details of the algorithm were only available after signing a non-disclosure agreement. In September 1994 someone

posted source code to the Cypherpunks mailing list – anonymously. It quickly spread to the Usenet newsgroup sci.crypt and via the internet to ftp sites around the world. Readers with legal copies of RC4 confirmed compatibility, but RSADSI has not yet made any related documents public, keeping its status as a proprietary trade secret as well as the name, a trademark. Since then, the algorithm has gained fame as *alleged* RC4 in most discussions and conferences.

Figure 6 WEP decryption block diagram



Computers are deterministic finite-state machines. At any instant of time, it can only be in a finite number of states. This finite number may be very large, but nonetheless finite. Any random number generating function, however complicated or elaborate it might be, must be a deterministic function of the computer's current state. That means that any random-number generator on a computer is, by definition, periodic. Anything that is periodic is, by definition, not random. A true random-number generator requires some random input that a computer cannot provide by itself. The best a computer can produce is a *pseudo-random sequence generator*. The generated sequence of random numbers appears random in the sense that, the sequence's *period* is long enough that a finite sequence of reasonable length is not periodic and cannot be subject to statistical analysis. Also, for a cryptographically secure pseudo-random sequence generator, it must be computationally infeasible to predict successive sequences, given the knowledge of the algorithm and all preceding sequences. This is generally achieved by using a secret key as the seed of the pseudo-random sequence algorithm, which sets the initial state of the generator.

RC4 is a pseudo-random sequence generator that takes a variable-length key as its seed. The generated sequence (keystream) is used to encrypt the plaintext and as such the keystream is independent of the plaintext. It has an 8 × 8 S-box: S_0, S_1,…, S_{255}. The entries are a permutation of the numbers 0 through 255 and the permutation is a function of the variable-length key. It has two counters, i and j, initialised to zero. To generate a random byte, following operations are performed.

$$i = (i+1) \bmod 256$$
$$j = \left(j + S_i\right) \bmod 256$$
swap S_i and S_j
$$t = \left(S_i + S_j\right) \bmod 256$$
$$K = S_t$$

The byte K is exclusive-ored (XOR) with the plaintext to produce the ciphertext or XORed with the ciphertext to produce the plaintext. Encryption is about 10 times faster than the Data Encryption Standard (DES), where the DES, known as the Data Encryption Algorithm (DEA) by ANSI and the DEA-1 by ISO, has been a worldwide standard for over 20 years. To initialise the S-box, fill it linearly: S_0 = 0, S_1 = 1,…, S_{255} = 255. Fill another 256-byte array with the key, repeating the key as necessary to fill the entire array: K_0, K_1,…, K_{255}. Set index j to zero. Then:

For $i = 0$ to 255
$$j = \left(j + S_i + K_i\right) \bmod 256$$
Swap S_i and S_j

RSADSI claims that the algorithm is immune to differential and linear cryptanalysis and has a large *period* because RC4 can be in about 2^{1700} ($256! \times 256^2$) possible states. Thus RC4 makes a good pseudo-random keystream generator and is used in dozens of commercial cryptography products, including Lotus Notes, Oracle Secure SQL, Microsoft Windows and the SSL. Although RC4 is no longer a secret it is still proprietary and any commercial product intending to use it must obtain a license from RSA Data Security, Inc.

4 Attacks

Unfortunately, the WEP has not well achieved confidentiality, access control and data integrity. Although the WEP protocol provides data privacy equivalent to that of a wired LAN, several vulnerabilities have been discovered in recent years. Attacks based on these vulnerabilities not only reveal the confidential data being transmitted, but also derive the secret key shared by the participating stations. Some of the common attacks are described as follows.

4.1 Key sequence reuse and known plaintext attack

The WEP provides data confidentiality using a stream cipher called RC4. A well-known pitfall of stream ciphers is that encrypting two messages with the same key sequence can reveal information about both messages without any knowledge of the secret key. This could lead to a number of attacks (such as cryptanalysis of XOR plaintext strings, frequency analysis) revealing the contents of each message individually (Borisov et al., 2001).

To prevent key sequence reuse, the WEP recommends varying key sequences for MPDUs so that the WEP uses a 24-bit IV (IEEE 802.11 WG, 1999), nearly guaranteeing that the same key sequence (caused from reuse of limited IVs and generally constant secret key) is being reused for multiple messages. Since IVs are public, key sequence reuse is easily detected through reuse of the IV (assuming the secret key may not have changed) exposing the system to key sequence reuse attacks. Thus, a popular pitfall of stream ciphers serves to compromise the WEP.

Key sequence reuse attack and known plaintext attack are introduced by Borisov et al. (2001). Some observations are listed as follows.

- Intruders can obtain all ciphertexts. For example, the intruders know about the ciphertext by getting the third frame during the authentication process shown in Figure 2.
- Intruders can find some duplication of the IVs: the intruders know the IVs when they constantly listen to the network and easily identify duplication of the IVs.
- Intruders can have partial knowledge of plaintexts. The intruders can get the second frame in the authentication process shown in Figure 2.

The secret key K always remains the same, but the change in the key sequence is due to the change in the IV every time. There is a chance for the IV to get reused since the length of the IV is 24 bits. The key sequence generated by the WEP algorithm is the same if the IVs are the same. If the same key sequence is used for two plaintexts (P1 and P2), the cipher texts C1 and C2, respectively, are defined as follow.

$$C1 = \{P1, \mathrm{ICV}(P1)\} \oplus \mathrm{RC4}(\mathrm{IV}, K) \tag{1}$$

$$C2 = \{P2, \mathrm{ICV}(P2)\} \oplus \mathrm{RC4}(\mathrm{IV}, K) \tag{2}$$

In the above example, RC4 (IV, K) are reused. When the same IV is used for encrypting two different plaintexts, it is called as a collision. Note that this collision concept is not that in channel access. From (1) and (2), we can obtain

$$C1 \oplus C2 = P1 \oplus P2 \tag{3}$$

By the knowledge of $C1$, $C2$ and $P1$, $P2$ can be obtained as follows.

$$P2 = (C1 \oplus C2) \oplus P1 \tag{4}$$

To find the key sequence reuse is easy and described as follows. The IVs are public and when they are sent with the ciphertexts, the intruders can obtain these IVs. Therefore, when the IVs are reused, the duplication of IVs can be easily spotted out.

The main reason behind this attack is the length of the IV, which is 24 bits and the maximum possible combinations of IVs can go up to 2^{24}. Implementation shows that the 1st collision occurs after transmitting 5000 packets which is few minutes after the data transmission. The fact reveals how easy for the attackers can get the duplicated IVs. However, the intruders can only obtain the messages using the same IV, under the condition that the triplet ($P1$, IV, $C1$) are known.

4.2 Decryption dictionary

Once the plaintext of an intercepted message is obtained, the key sequence used to encrypt the message can be easily derived. This key sequence can be used to decrypt other messages that use the same IV. In this manner, an intruder may compile a table of known key sequences and their respective IVs. Once such a table is compiled for each IV, the intruder can use it as a decryption dictionary to decrypt any message (Borisov et al., 2001).

The decryption dictionary is based on the key sequence reuse attack. If the intruders know IVs and the corresponding key sequences, a decryption dictionary can be built mapping key sequences with IVs.

For example, if the intruder finds that IV1 is used, key sequence1 can be used to obtain the plaintext by the formula:

$$P1 = C1 \oplus \text{Key sequencel} \tag{5}$$

The dictionary built requires some space which is roughly 24 GB which comes well below the limit (Borisov et al., 2001).

4.3 Key management

The 802.11 standard does not specify how the secret key is distributed to all the stations. It relies on an external system to do this. This practice seriously affects the security of the system that depends on a single key for its entire protocol to remain effective. Because a single key is used by the AP and all stations in a BSS, the administrators might find it inconvenient to change it, since that would require all stations to update their secret key as well. Thus, a constant secret key would increase chances of IV reuse and thereby key sequence reuse and the system is subject to the attacks described above. Furthermore, compromise of one station would reveal the secret key, which would thwart the security of the entire BSS (Arbaugh et al., 2001; Borisov et al., 2001).

4.4 Message tampering

The WEP protocol adopts CRC-32 to calculate a checksum integrity field and is encrypted along with the payload. This field is used after decryption to check the integrity of the message in transit. CRC-32 used in IEEE 802.11 is to ensure the data integrity by detecting random errors in messages, but not to ensure data security. Since the CRC checksum function is linear and stream ciphers such as RC4 are also linear, it is possible to tamper with the message in transit without detection through simple XOR methods (Borisov et al., 2001). The abuse of CRC, has led WEP encrypted messages to be subject to some modification without detection. Since the attacker knows the cipher text C, the message is modified without the knowledge of the key stream and even without the knowledge of the message.

Let's show the message modification process (Borisov et al., 2001). Let P be the message to be modified and C = RC4 (IV, K) $\oplus$ (P, CRC(P)) be the corresponding ciphertext. Let P' be the modified message and $\Delta = P \oplus P'$ be the modification made on P. Let C' denote the modified cipher text which is given to the AP, which will not find this message modification because CRC is a linear function, that is, CRC(P) $\oplus$ CRC(Δ) = CRC($P \oplus \Delta$). We have

$$\begin{aligned} C' &= \mathrm{RC4}(\mathrm{IV}, K) \oplus (P', \mathrm{CRC}(P')) \\ &= \mathrm{RC4}(\mathrm{IV}, K) \oplus (P \oplus \Delta, \mathrm{CRC}(P \oplus \Delta)) \\ &= \mathrm{RC4}(\mathrm{IV}, K) \oplus (P, \mathrm{CRC}(P)) \oplus (\Delta, \mathrm{CRC}(\Delta)) \\ &= C \oplus (\Delta, \mathrm{CRC}(\Delta)) \end{aligned}$$

When the ciphertext is passed to the AP, the intruder hacks the cipher and modifies the message by XOR cipher (C) with $\Delta + C$ (Δ) to the cipher. The modified text is sent to the AP, which decrypts the message and finds that the message is not modified. The main reason behind this successful modification of text without the WEP's knowledge is that during the encryption process, the secret key is not applied on the plaintext. The CRC that is applied on the text is for data integrity and it cannot handle the message modification (Borisov et al., 2001).

4.5 Message injection

Based on known key sequence attack, it is possible to introduce an arbitrary number of messages into the WEP protected WLAN, thus circumventing access control (Borisov et al., 2001) since the same IV can be reused any number of times and as long as the key sequence corresponding to a particular IV is correct, the AP cannot tell the difference between a message originating from an authenticated station or an intruder. An intruder needs only to encrypt random messages with the discovered key sequence, supply the IV along with it and transmit the message to an accepting AP.

When the intruder gets hold of the challenge text, the intruder can cause traffic to the network by simply injecting the message to the challenge text. If the intruder knows the challenge text and the cipher text, the intruder will get the key sequence according to (RC4 (IV, K) = $C \oplus P$). With the knowledge of the key sequence, the intruder will now use the key sequence to inject a message to the traffic and therefore cause increasing the traffic load, that is, $C' = (P', \text{CRC}(P')) \oplus \text{RC4 (IV, } K)$.

4.6 Authentication spoofing

A simple extension of plaintext attack leads to an authentication spoofing attack (Arbaugh et al., 2001; Borisov et al., 2001). During the authentication exchange, shown in Figure 2, an intruder can eavesdrop and obtain a plaintext and ciphertext pair. Using the pair, it is easy to obtain the key sequence. This exploit may be used to authenticate with an AP and open grounds for further attacks. An intruder may authenticate with an AP without knowledge of the secret key assuming that the AP use the same pair of IV and the challenge text.

4.7 Man-in-the-middle attack

Man-in-the-middle attack is a special case of authentication spoofing where an intruder thwarts the communication between an AP and a station and configures the communication to travel via itself. Messages from the either the AP or the station are sent to the other by manipulating them as originating from itself. Both the AP and the station cannot detect whether they are communicating with each other or an intruder as long as the authentication frames are in accordance with the protocol. Once the station sends the last authentication frame that would authenticate it with the AP, the AP authenticates the intruder instead and the station is subject to denial-of-service. This type of attack is usually countered with digital signatures. Digital signatures allow two entities engaged in communication to be assured of the identity of the messages from the other. This type of attack typically needs an elaborate manipulation in network settings and is generally considered difficult to set up, let alone execute it.

4.8 Related work in attacks

A summary of related work on vulnerabilities of the IEEE 802.11 WEP protocol: authentication spoofing (Arbaugh et al., 2001), Ciphertext-only attack (Arbaugh et al., 2001; Stubblefield et al., 2001), Dictionary attack (Borisov et al., 2001), Message Injection (Borisov et al., 2001), Message Tampering (Borisov et al., 2001), Plaintext-Ciphertext attack (Borisov et al., 2001) and Replay attack (Arbaugh et al., 2001; Borisov et al., 2001).

5 Security enhancements

The proposed enhancements attempt to rectify the vulnerabilities and make the attacks futile. We propose to enhance the WEP with Keyed Message Authentication Code and Enhanced Authentication (WEP-KMAC-EA) in Subsection A, and to enhance the WEP with Private IV and Session/Day Keys (WEP-PIV-SDK) in Subsection B. The proposed enhancements partially bases on attacks described by various researchers and their recommendations (Arbaugh et al., 2001; Borisov et al., 2001; Fluhrer et al., 2001; Stubblefield et al., 2001; Verton, 2001; Vines, 2002).

5.1 WEP-KMAC-EA

The proposed WEP-KMAC-EA adopts two enhancements of the WEP: Keyed Message Authentication Code and Enhanced Authentication.

5.1.1 Keyed message authentication code

A WEP encrypted message can be subject to message tampering and other attacks as described in above. This is due to an un-keyed linear function (CRC32). CRC (32) is linear and is used to check the data integrity, but cannot prevent the message from being tampered by the intruder. In other words, the generated message integrity check field depends only on the message and does not depend on the secret key.

Borisov et al. (2001) recommend using of a keyed message authentication code to provide considerable strength. An intruder cannot tamper with the ICV of a message since he would not have access to the secret key used to generate it. Specifically, the WEP's ciphertext C is (IV, $P \oplus$ RC4 (IV, K)), whereas the WEP with Keyed Message Authentication Code uses $C = (K \oplus \text{IV}, P \oplus K \oplus \text{RC4 (IV, } K))$.

5.1.2 Enhanced authentication

The authentication method shown in Figure 2 involves transmitting an unencrypted challenge text and an encrypted response of the same challenge text. This gives out a known plaintext-ciphertext pair to an intruder eavesdropping on the channel. Through known plaintext attacks, the intruder may spoof authentication and gain unauthorised access to the WEP. This can be avoided by prohibiting transmissions of any of plaintext-ciphertext pairs. The authentication can be enhanced in a way shown in Figure 7.

Figure 7 Enhanced authentication

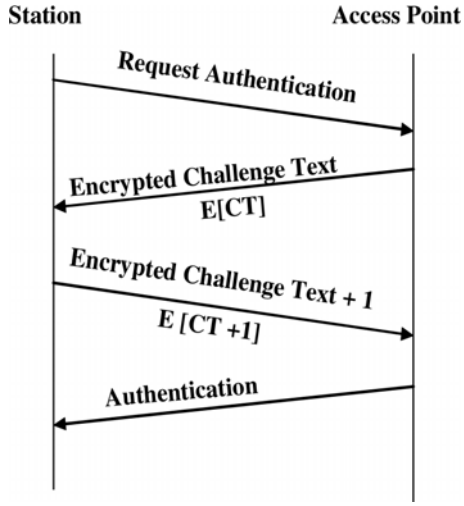


Figure 7 illustrates the enhanced authentication mechanism, in which, on request for authentication by a station, the AP can send a challenge nonce encrypted using the WEP with the shared key to the station, where a nonce is a random number guaranteed not to be repeated during the lifetime of the server generating it. The station decrypts it using the shared secret key, increments the nonce by one, encrypts it with the WEP and sends it back to the AP. The nonce needs to incremented as a way of proving to the AP that the station was in fact able to understand the challenge text through successful decryption. This authentication could be followed by transmissions of the session keys for subsequent communications.

5.2 WEP-PIV-SDK

The proposed WEP-PIV-SDK adopts two enhancements of the WEP: Private IV and Session Keys.

5.2.1 Private IV

The reason why the IV is transmitted in the clear is because the 802.11 standard assumes that an intruder gains no useful information from its knowledge. It is clearly not true as shown in the various types of attacks described in the above. The reason of using the IV is to produce randomness for the key and the reason for transmitting the IV is to help the AP decrypt the information sent from the station. To strengthen the security, the IV can be encrypted by the WEP or Day/Session key described in the next section. This will disable an intruder's ability to easily map IVs to known key sequences. Specifically, the WEP's ciphertext C is (IV, $P \oplus$ RC4 (IV, K)), whereas the WEP with Private IV uses C = (($K \oplus$ IV, $P \oplus$ RC4 (IV, K)) or C = (($K_1 \oplus$ IV, $P \oplus$ RC4 (IV, K)), where K_1 is the day key or session key introduced in the next section.

5.2.2 Session/Day Keys

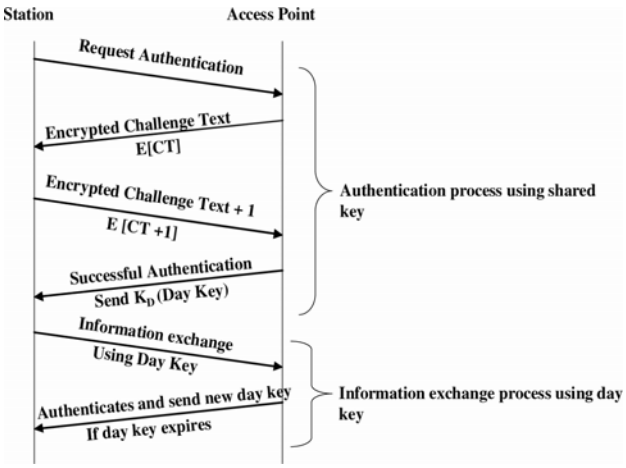
The 802.11 standard does not specify the key management and relies on an external media to distribute the secret key to all stations. Therefore, frequent changes of the secret key make a hassle for administrators. Subsequently, a constant secret key is used and it leads to reuse of key sequences that is another cause for several vulnerabilities. Instead of using the secret key for generating the key sequence used to encrypt the payload, a day key can be used and it is the output of a randomised function on the secret key. The day key could be used to encrypt all the traffic on one day. Several day keys may be used. Specifically, the WEP's ciphertext C is (IV, $P \oplus$ RC4 (IV, K)), whereas the WEP with a day key uses C = ($K_D \oplus$ IV, $P \oplus$ RC4 (IV, K_D)). The AP generates the day key everyday for the BSS. The AP can generate a day key per station and the traffic communicated between the station and the AP would be encrypted with a unique day key allocated for the station. Transmission of the day key to the station can take place as soon as a station is authenticated via the original WEP encryption. Then, the day key management is not an issue. Session keys further enhance security. The AP generates temporary session key(s) for encryption of a particular session and transmits them in encrypted form using the day key. In a BSS under high traffic, this method helps to prevent discovery of the day key. Also the key sequence reuse vulnerability is further heightened. Since the IV space is limited (24 bits in length), the above mechanism helps to change the key to achieve the requirement of supplying unique pairs of key and IV to the RC4 algorithm and therefore, the problem of key sequence re-use can be avoided largely.

Using session keys alone defends a lot of attacks, but may cause the disconnected problem, in which when the station associated with the network gets temporarily disconnected and wants to the join the network after some time, it cannot join the network since it does not have the key to get inside. The disconnected user problem can be solved by issuing a separate key for authentication process and a dedicated key for the information exchange between stations. The day key can be used to generate the key sequence which in turn is used to encrypt the payload.

As shown in Figure 8, once the station enters into the network using the shared key, it requests authentication procedure. The AP sends the challenge text encrypted with the shared key. The station decrypts and increases the text by one and encrypts it back and sends it to the AP. When the text is sent, the IV is also encrypted and sent to the AP. This is making the IV private so that the hackers would not gain any useful information. The AP gets the text, decrypts it and checks for the correctness of the text. Once the process is successful, a day key is generated for the station

by the AP and it is sent back to the station. The station uses the day key to process the information to other stations through the AP. This day key is only valid for a day. Once the day key expires, a new day key is given back to the station by the AP.

Figure 8 Modified authentication and message transfer using day key



5.2.3 Attacks

Table 1 shows a summary of vulnerabilities of the IEEE 802.11 WEP, WEP-KMAC-EA and WEP-PIV-SDK. The improvements made over the WEP are apparent from WEP-PIV-SDK and also verified in the simulations.

Table 1 Summary of vulnerabilities

	WEP	*WEP-KMAC-EA*	*WEP-PIV-SDK*
Authentication spoofing (Borisov et al., 2001)	Yes	Yes	No
Casual eavesdropping	No	No	No
Ciphertext-only attack (Arbaugh et al., 2001; Stubblefield et al., 2001)	Yes	Yes	No
Dictionary attack (Borisov et al., 2001)	Yes	Yes	No
Man-in-the-middle attack	Yes	Yes	Yes
Message injection (Borisov et al., 2001)	Yes	No	No
Message tampering (Borisov et al., 2001)	Yes	No	No
Plaintext-ciphertext attack (Borisov et al., 2001)	Yes	Yes	No
Replay attack (Arbaugh et al., 2001; Borisov et al., 2001)	Yes	Yes	May be

6 Simulation methodology

A simulation programme is developed to compare the WEP algorithm with the proposed security enhancements (Bandela, 2002). This section describes our simulation methodology.

6.1 Simulation methodology and platform

In the simulation programme, there are two entities: a sending station module and a receiving station module. The sending/receiving station modules simulate only the MAC and PHY layers. The sending/receiving station modules are assumed to communicate data with the LLC layer in a byte stream format. The data is sent/received in raw (without enclosing any header information) for simplicity since the WEP algorithm deals with the frame body of a data frame and is not concerned with how the frames arrive or the frame structure.

The medium of transmission is the local hard disk. The sending station module encrypts packets of data and outputs to a local file on the disk. The receiving station module reads this file and decrypts packets of data. Encryption takes place per packet. A packet may contain up to 2312 bytes of data. For simplicity, all packets of data have the same packet size in the simulation. Effectively, a local file that acts as the transmission medium between two stations, is simply a file which is read from or written to at the rate of 2312 bytes of data at a time.

The simulation is programmed in Java runtime environment on a Win32 platform with a Pentium III processor. Figures 9 and 10 illustrate the flow diagrams of the proposed encryption and decryption modules, respectively.

Figure 9 The encryption process

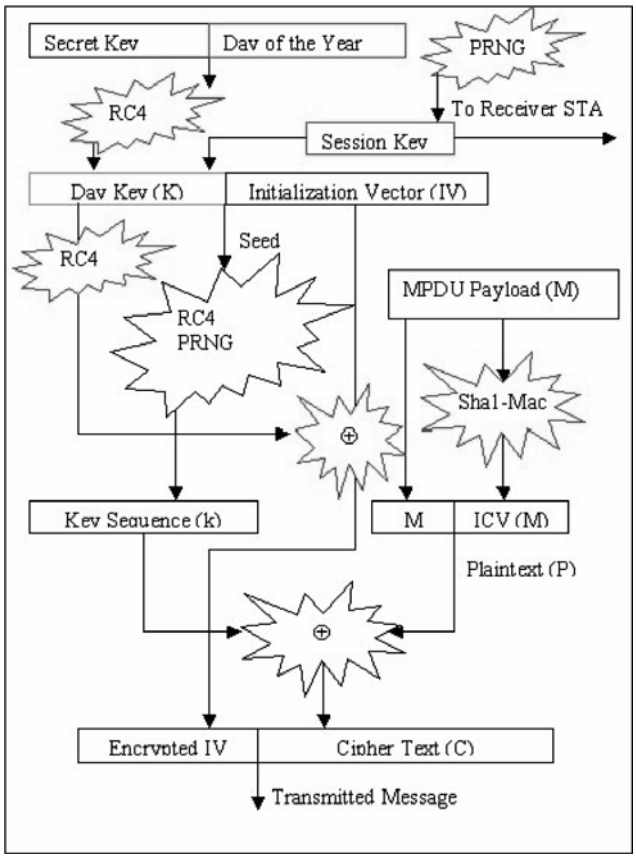
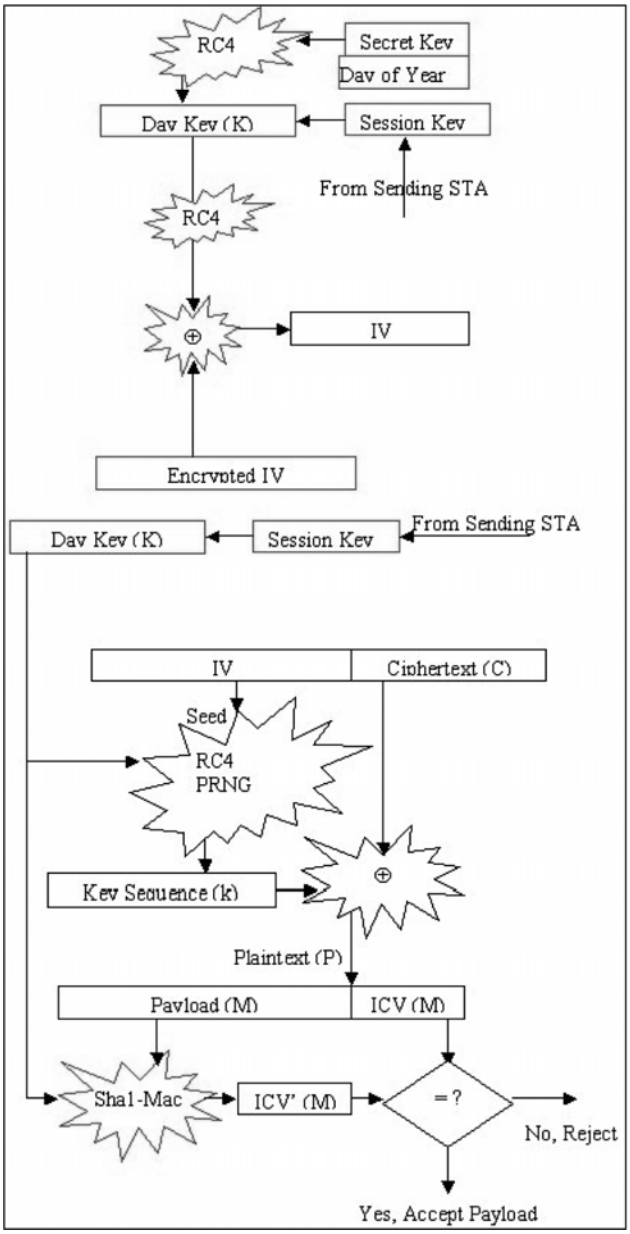


Figure 10 The decryption process



6.2 *Java object description*

Major Java objects in the simulation program are described as follows.

- *RC4* is the pseudo-random number generator used in the WEP that generates key streams of arbitrary length. The Java class *RC4* is instantiated with a secret key, which initialises the S-box. A call to the function *nextByte* returns a random *byte*.
- *WEP* contains functions to generate keystreams and integrity check values in the WEP algorithm. *WEP* defines WEP standard data fields for PDU, Intilialisation Vector, Secret Key and Integrity Check Value sizes as the 802.11 specification. It has two functions: the function *getKeyStream* takes an initialisation vector of 3 *byte* and a secret key of 5 *byte*, concatenates them to form a seed that initialises RC4 keystream generator. It calculates a random keystream and returns *byte* array whose length is equal to the plaintext data. The function *getICV* takes a plaintext *byte* array and length of plaintext (due to variations in the incoming plaintext transmission despite a standard PDU size) as arguments and calculates a CRC32 integrity check value. It returns the 4 *byte* value as an array.
- *IWEP* contains functions to generate keystreams and integrity check values as the proposed improved WEP algorithms. *IWEP* defines data fields for PDU, Initialisation Vector, Secret Key and Integrity Check Value sizes. It has two functions similar to *WEP*: the function *getKeyStream* is overloaded thrice. In addition to the function similar to that in *WEP*, it is possible to return a keystream of desired length and also keystream with only the secret key (without any IV) used as the RC4 seed. Unlink in *WEP*, the *getICV* function takes an additional parameter, the secret key, to generate a keyed-message authentication code. As described in the proposal, the integrity check value depends not only on the plaintext, but also on a secret key. HMAC-SHA1 is the method employed here to generate the ICV.
- *WepSender/WepReceiver*. *WepSender* simulates a sending station module. It reads in a packet of data that is to be transmitted, encrypts it, and outputs it to the transmission medium. It generates random IVs, calls the methods from *WEP* to generate keystreams and integrity check values and forms the encrypted packet. *WepReceiver* simulates a sending station module. It reads in a packet of encrypted data from the transmission medium, calls the methods from *WEP* to regenerate the same keystream based on the IV transmitted, does an integrity check on the data packet and thus decrypts it. The *encrypt* and *decrypt* functions read an array of *byte* equal in size to the PDU, calculate ICV of the PDU, XOR it with a random keystream generated through *WEP* and form the resulting MPDU.
- *IWepSender/IWepReceiver*. *IWepSender* simulates sending station modules with the improved WEP schemes. *IWepReceiver* simulates receiving station modules in the improved WEP schemes. *IWepSender2* improves over the *WepSender* and includes the functionality of *IWepSender1*. During initialisation of *IWepSender2*, the day key is calculated as a random keystream generated from the RC4 PRNG using the ‘day of the year’ as the IV and the secret key. Further, a session key is generated as a random number. Here instead of using RC4, Java’s *Random* is used for optimisation, since their session key cannot be subject to any analysis. The implementation of *IWepSender2* is slightly deviated from the proposal. Ideally, the session key is transmitted in a ‘management frame’ to the receiving station. But since there is no framework implementation made, a data frame is ‘tagged’ as a management frame for convenience. The receiving station examines this tag before it processes it as a data or management frame. Encryption proceeds very similar to that of *WepSender*. Data frames are encrypted using the session key. The session key is

changed after a fixed number of frames are transmitted. The new session key is encrypted using the day key and transmitted to the receiving station. *IWepReceiver2* improves over *WepReceiver* and includes the functionality of *IWepReceiver1*. During initialisation of *IWepReceiver2*, the day key is calculated as a random keystream generated from the RC4 PRNG using the 'day of the year' as the IV and the secret key. The session key is received from the sending station encrypted using the day key in a 'Management frame'. The implementation of *IWepReceiver2* is slightly deviated from the proposal. Ideally, the session key is received in a 'Management frame' to the receiving station. But since there is no framework implementation made, a data frame is 'tagged' as a management frame for convenience. The receiving station examines this tag before it processes it as a data or management frame. Decryption proceeds very similar to that of *WepReceiver*. Data frames are decrypted using the session key received. The session is updated whenever a management frame is detected. The management frame is decrypted using the day key.

- *WepReceiverKSDetector*/*IWepReceiver2KSDetector* are extensions of the receiving station modules for the WEP and the improved WEP algorithm, respectively. In addition, they also detect keystream reuse and report it. *WepReceiverKSDectector* and *IWepReceiver2KSDetector* are simple extensions of their parent versions. These receiving stations record each (IV, session key) combination which can be externally analysed for keystream reuse. A unique (IV, session key) pair maps to a unique keystream and this helps analyse the security of the proposed algorithm against the WEP.

7 Simulation results

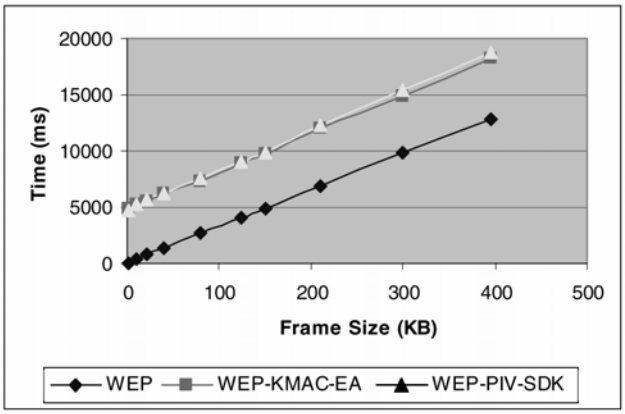
Simulation results aim to quantify the previous enhancements and enable comparisons between the original WEP protocol and the proposed schemes.

This experiment compares the rate of encryption of three pairs of station modules. WepSender (WS) and WepReceiver (WR) are station modules that implement the standard 802.11 WEP algorithm. IWepSender1 (IWS1) and IWepReceiver1 (IWR1) are station modules that implement the WEP-KMAC-EA scheme. This implementation improves WEP by changing the ICV from CRC-32 to HMAC-SHA1 which is a keyed ICV. IWepSender2 (IWS2) and IWepReceiver2 (IWR2) are station modules that implement additional WEP-PIV-SDK scheme. This implementation improves IWep1 by using day keys and session keys. Day keys are derived as an RC4 key stream using the secret key, concatenated with the day of the year, as the seed. Session keys are random bytes that are decided by the sending station and conveyed to the receiving station in encrypted form along with the data. Session keys are changed after a certain number of packets are processed.

7.1 Performance evaluation

The time required to process varying amounts of data are computed for the three pairs of station modules and tabulated. File sizes are in KB and time taken is in milliseconds. Each data was derived from an average of 10 trial tests. Figure 11 shows the results for this experiment and it shows the total times for encryption and decryption for each pair of station modules. We observe that the proposed schemes increase process time in some degree.

Figure 11 Performance evaluation: rate of encryption and decryption



7.2 Key stream reuse detection

As described earlier, key stream reuse is a major vulnerability of the WEP and if an intruder is able to detect it, the intruder can collect the respective 'collision' packets for analysis. The more the amount of the 'collision' bytes, the better is the chance of an intruder to compromise the security of the system. Therefore, key stream reuse detection is a good measure of the security of an algorithm.

Key stream reuse analysis is conducted against two pairs of station modules – the WEP stations and the WEP-PIV-SDK stations and compared. For the purpose of this experiment, instead of creating an intruder programme module that would attempt to detect key stream reuse, the receiving station simply counts key streams that repeat. Since the receiving station knows all the key streams, the total number of reused key streams are reported exactly which would serve as a worst-case scenario.

From the observations in Table 2, it can be concluded that WEP-PIV-SDK does a better job at generating random key streams without reuse than the standard WEP and hence causes fewer collisions. As expected, the WEP causes a number of collisions because of key stream reuse. Intuitively reasoning, this is because in the WEP, only the IV varies and the secret key is constant and therefore the key stream can be only in one of 2^{24} states for a given IV. But with the use of day and session keys, the key stream is generated with a much better varying seed and it can be in one of 2^{64} states. This is why no key stream reuse was detected in the above experiments.

Table 2 Key stream reuse detection

# of packets	*WEP collisions*	*WEP-PIV-SDK collisions*
22716	34	0
34908	82	0
58540	218	0
10967	534	0

8 Conclusions

In this paper, we introduced the security issues in the IEEE 802.11 WLANs and proposed two enhancements for the WEP. Furthermore, we conducted simulations/experiments on comparisons of these schemes with the original WEP scheme. The proposed enhancements provide better data confidentiality with some degree of computing cost as the trade-off. The improved schemes overcome the weaknesses resulting from Key Sequence Reuse. They make use of not only the varying IV states, but also varying key states in order to supply a higher seed space resulting in lesser key stream reuse. It is not easy to mount decryption dictionary attacks, since the total number of key streams to be discovered increases largely relative to the WEP and the key streams used change from day to day for the same IV. Key Management is partially solved since the system is not easily compromised despite the secret key remaining unchanged for a long time. Message Tampering is completely avoided from the use of Keyed Message Authentication mechanism. Security against Message Injection is heightened since discovery of a key stream is useful to the intruder only until the next session key change. If session key is refreshed frequently enough, depending on the network traffic, the vulnerability can be kept under check. Authentication spoofing is made difficult by using Kerberos based authentication.

As realised in the experiments the efficiency of the proposed algorithm is not perfect. It is apparent that the keyed message authentication is a little computationally costly. More research needs to be done to determine a satisfactory trade-off to find an easily computable integrity check value that cannot be tampered with. Other schemes may be explored that would improve the randomisation factor of key streams. Authentication remains an area to be improved since the proposed authentication mechanism is still vulnerable to replay and man-in-the-middle attacks.

Acknowledgement

Yi Pan's research was supported in part by the National Natural Science Foundation of China (NSFC) under Grant No. 60440420451 ('two base' project).

References

Arbaugh, W., Shankar, N. and Justin Wan, Y.C.W. (2001) *Your 802.11 Wireless Network has No Clothes*, Department of Computer Science, UMCP, March.

Bandela, C. (2002) 'Improving WEP security in IEEE 802.11 wireless networks', Georgia State University, Master Thesis.

Borisov, N., Goldberg, I. and Wagner, D. (2001) 'Intercepting mobile communications: the insecurity of 802.11', *The Seventh Annual International Conference on Mobile Computing and Networking (MOBICOM 2001)*.

Fluhrer, S., Mantin, L. and Shamir, A. (2001) 'Weaknesses in the key scheduling algorithm of RC4', *The Eighth Annual Workshop on Selected Areas in Cryptography*, August.

IEEE 802.11 WG (1999) *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, Standard, IEEE*, August.

Rivest, R.L. (1992) *The RC4 Encryption Algorithm*, RSA Data Security, Inc., March (Proprietary).

Stubblefield, A., Ioannidis, J. and Rubin, A. (2001) 'Using the Fluhrer, Mantin, and Shamir attack to break WEP', *AT&T Labs Technical Report*, August.

Verton, D. (2001) *Your Wireless LAN Can Be Hacked – Flaws in 802.11 can Leave Data Vulnerable*, PCWorld.com.

Vines, R.D. (2002) *Wireless Security Essentials: Defending Mobile Systems from Data Piracy*, 1st edition, 1 February 2001, John Wiley and Sons, ISBN: 0471209368.