A Code-Book Cipher Methodology for Wireless Networks

Vikram Jain
Dept. of IT, Shri Ram Institute of Technology, Jabalpur, M.P., India.

Abstract
This paper illustrates a new encryption methodology that can be useful in present wireless technologies. The algorithm uses a code-book cipher as the basis. The proposed algorithm is designed with some degree of self-organizing mechanism that makes it a strong cipher to break. The present encryption protocol, WEP (Wired Equivalent Privacy) has many weaknesses and drawbacks. This made researchers to go for more sophisticated algorithms and improvements to WEP (like WPA). The algorithm proposed in this paper helps to solve the current problem.

Keywords
WEP, Code-book Cipher, WLAN, Self-organizing Networks.

I. Introduction

There have been a number of imperfections revealed in the WEP algorithm, which critically weaken the security claims of the system. In particular, the following types of attacks are found:

- Passive attacks to decrypt traffic based on statistical analysis.
- Active attack to inject new traffic from unauthorized mobile stations, based on known plaintext.
- Active attacks to decrypt traffic, based on tricking the access point.
- Dictionary-building attack that, after analysis of about a day's worth of traffic, allows real-time automated decryption of all traffic.

Analysis suggests that all of these attacks are practical to mount using only inexpensive off the shelf equipment [1]. It's recommended that anyone using an 802.11 wireless network not rely on WEP for security, and employ other security measures to protect their wireless network.

A. WEP Setup

The 802.11 standard describes the communication that occurs in wireless local area networks (LANs). The Wired Equivalent Privacy (WEP) algorithm is used to protect wireless communication from eavesdropping. A secondary function of WEP is to prevent unauthorized access to a wireless network; this function is not an explicit goal in the 802.11 standard, but it is frequently considered to be a feature of WEP.
WEP relies on a secret key that is shared between a mobile station (e.g. a laptop with a wireless Ethernet card) and an access point (i.e. a base station) [2]. The secret key is used to encrypt packets before they are transmitted, and an integrity check is used to ensure that packets are not modified in transit. The standard does not discuss how the shared key is established. In practice, most installations use a single key that is shared between all mobile stations and access points. More sophisticated key management techniques can be used to help defend from the attacks; however, no commercial systems are in aware to support such techniques.

B. WEP Problems

WEP uses the RC4 encryption algorithm, which is known as a stream cipher. 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.
XOR-ing the key stream with the cipher-text yields the original plaintext.
This mode of operation makes stream ciphers vulnerable to several attacks [3]. If an attacker flips a bit in the cipher-text, then upon decryption, the corresponding bit in the plaintext will be flipped.
Also, if an eavesdropper intercepts two cipher-texts encrypted with the same key stream, it is possible to obtain the XOR of the two plaintexts.
Knowledge of this XOR can enable statistical attacks to recover the plaintexts. The statistical attacks become increasingly practical as more cipher-texts that use the same key stream are known. Once one of the plaintexts becomes known, it is trivial to recover all of the others.
WEP has defences against both of these attacks.
To ensure that a packet has not been modified in transit, it uses an Integrity Check (IC) field in the packet. To avoid encrypting two cipher-texts with the same key stream, an Initialization Vector (IV) is used to augment the shared secret key and produce a different RC4 key for each packet. The IV is also included in the packet.
However, both of these measures are implemented incorrectly, resulting in poor security. The integrity check field is implemented as a CRC-32 checksum, which is part of the encrypted payload of the packet. However, CRC-32 is linear, which means that it is possible to compute the bit difference of two CRCs based on the bit difference of the messages over which they are taken. In other words, flipping bit "n" in the message results in a deterministic set of bits in the CRC that must be flipped to produce a correct checksum on the modified message.
Because flipping bits carries through after an RC4 decryption, this allows the attacker to flip arbitrary bits in an encrypted message and correctly adjust the checksum so that the resulting message appears valid.

The initialization vector in WEP is a 24-bit field, which is sent in the clear-text part of a message. Such a small space of initialization vectors guarantees the reuse of the same key stream. A busy access point, which constantly sends 1500 byte packets at 11Mbps, will exhaust the space of IVs after 1500*8/(11*10^6)*2^24 = ~18000 seconds, or 5 hours. (The amount of time may be even smaller, since many packets are smaller than 1500 bytes.) This allows an attacker to collect two cipher-texts that are encrypted with the same key stream and perform statistical attacks to recover the plaintext. Worse, when the same key is used by all mobile stations, there are even more chances of IV collision. For example, a common wireless card from Lucent resets the IV to 0 each time a card is initialized, and increments the IV by 1 with each packet. This means that two cards inserted at roughly the same time will provide an abundance of IV collisions for an attacker. (Worse still, the 802.11 standard specifies that changing the IV with each packet is optional!)

II. Proposed Methodology

The proposed method works by the basis of a Code-book (CB), without the needing of a key to be shared at each and every point of the Wireless Network. Maintaining of sessions is done to renew the code-book from time-to-time. The proposed Methodology works as follows:

- Generation of CB.
- Encryption of CB using WEP and exchange of CB.
- Maintaining of sessions for generation of new CB.

A. Generation of CB

The code-book is generated as follows:

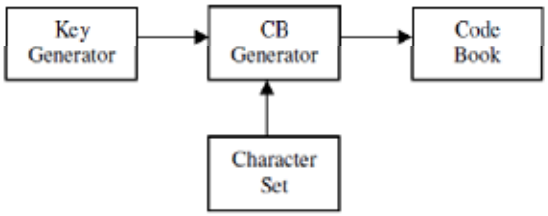

Fig. 1 :

The Key Generator generates a key, based on which the Code-Book generator generates a Code-Book for the given Character-Set. The working is as follows:

The Key-Generator generates a key K ('g', 'mg'), where 'g' denotes number of groups and 'mg' denotes members (characters) per group.

Now, the Code-Book Generator permutes the given character set into 'g' groups. Each group is identified or denoted by a single member (character) of that group. For example let us suppose the character set C= {set of alphabets (a - z)} and the key K = (9, 3), now the code generator algorithm permutes the set of alphabet into a 9-group and each group having 3 members or less. After the grouping from each group an element is selected as a special code. Now, the generated CB may look like Table-1.

Table 1 :

Group	Group Code
AJS	J
BKT	T
CLU	C
DMV	V
ENW	E
FOX	O
GPY	P
HQZ	Z
IR	R

These special codes are used for representing the character set (here alphabet). Let us suppose that "HELLO ARE YOU THERE" is the message, and then it is coded as:

- H is coded as Z1, since 'H' belongs to the group that is having a group code. 'Z' and 'H' is in 1st position from right.
- E is coded as E1, since 'E' belongs to the group that is having a group code 'E' and 'E' is in 1st position from right.
- L is coded as C2, since 'L' belongs to the group that is having a group code 'C' and 'L' is in 2nd position from right.

Likewise the total message is coded as -
"Z1E1C2C2O2 J1R2 E1 P3O2C3 T3Z1E1R2E1".

The message is coded in such a way that a group code is followed by the position of that character in that group. The selection of group code may vary, like it can be from the same character-set that is being encoded or it can be from a different character set.

Suppose if special symbols are used to represent the special code of each group and instead of the positions are being given, we can repeat the special symbol the number of times equal to the position value of that character in that group.

Another example code-book is illustrated in Table-2.

Table 2 :

Group	Group Code
AJS	!
BKT	@
CLU	#
DMV	$
ENW	%
FOX	^
GPY	&
HQZ	*
IR	+

Now the encoded message consists of only symbols that are used as group codes.

The decryption is the mirror process of what is done in encryption.

B. Encryption of CB using WEP and exchange of CB

Now after the generation of the CB, the CB has to be exchanged between the two communication parties. So, here the WEP is used to encrypt the code book and is sent to the receiver. Since, initially CB will be only at one side, so there is a need of exchange of CB between both the parties.

Fig. 2 :

Fig.2 illustrates the packet that contains CB and is encapsulated using WEP encryption and sent to the other party. Once the CB is available at both ends, communication may take place in two forms.

1. Method-1

In this method the data is first encrypted using WEP algorithm and then the resulting cipher is given to the Code-Book algorithm to generate final cipher that is transmitted. The method is illustrated in Fig. 3.

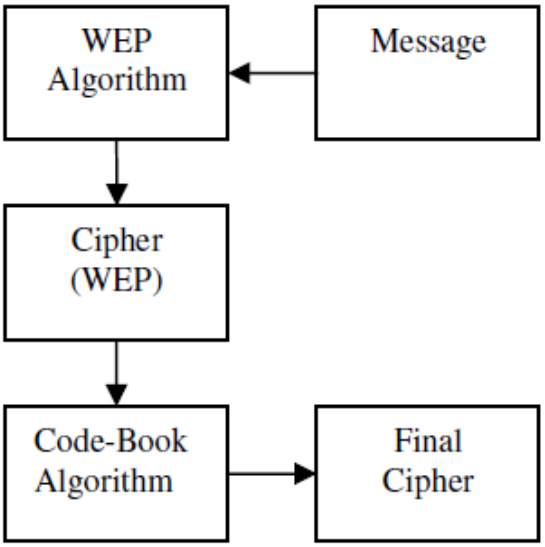

Fig. 3 :

2. Method-2

In this method the message is first encoded using the Code-Book and then the WEP encryption is applied on it to get the final cipher. This method is illustrated in Fig.4.

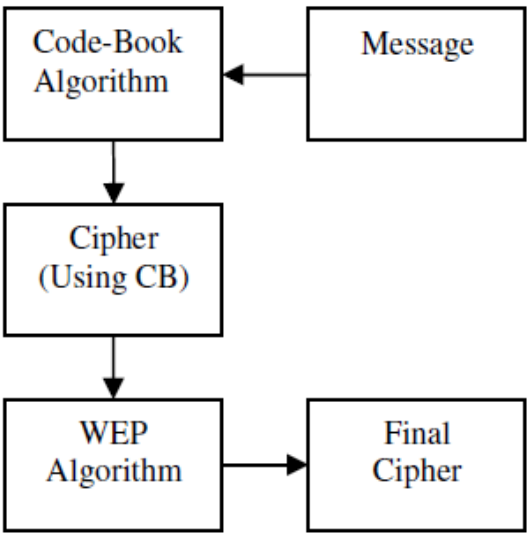

Fig. 4 :

C. Maintaining of sessions for generation of new CB

Though everything is fine up to now, the problem with the Code-Book ciphers is that if once the code-book is exposed or got into hands of a cracker or an attacker then all the communications can be decoded. To protect the code-book from getting exposed, both the parties must choose or install a "Session Manager" program which maintains the sessions of the parties' communication.

1. Types of Sessions

The sessions can be implemented by the session manager program different ways:

- A new session can be started at the morning of a day and end at evening.
- Random Sessions can be maintained.
- The session manager maintains some thresholds for communication that trigger for a new session whenever a certain amount of data is either transmitted or received.

2. Generating of Code-Book

The issue here is that, not only the sender party always generates the CB; sometimes the receiver party may also generate a CB and exchange it with sender. For this a mechanism is employed in session manager to change the CB from time to time basis, on sessions. At the start of each session the two session managers can exchanges messages and decide which party has to generate the CB. Once the decision is made then the CB is sent to the other party and then the encryption will be based on this new CB.

III. Discussion

There are several pros and cons that are identified in the methodology, which are discussed below:

A. Pros

- The cipher that is created here is a strong cipher to break since it is encrypted twice and every time the code-book changes from session to session which creates more confusion and diffusion for the attacker.
- The introduction of session manager makes it difficult for the attacker to implement a Dictionary-building attack that, after analysis of about a day's worth of traffic, allows real-time automated decryption of all traffic.

B. Cons

The methodology is a slightly long process that may take additional time for processing before the message is sent.

IV. Conclusion

Though the basic issues like authentication and integrity that are not accurately answered by the methodology and it mostly concentrates on the encryption, the real time implementation of the methodology will help in resolving the other problems discussed. Author is still in the analysis stage of the algorithm and the preliminary simulation results are encouraging.

References

[1] Brian Carter, Russell Shumway, 2002, "Wireless Security End to End". Wiley Publishing, Inc.
[2] Bruce Schneier 2003, "Applied Cryptography". John Wiley & Sons, Inc.
[3] J. Walker, "Unsafe at any key size: an analysis of the WEP encapsulation," Tech. Rep. 03628E, IEEE 802.11 committee, March 2000.

Vikram Jain received his M.Tech degree in IT (Networking) from VIT University, Vellore, India in 2007 and B.E. degree in Information Technology from Shri Ram Institute of Technology, Jabalpur, India. He is currently working with Shri Ram Institute of Technology, Jabalpur, India as an Assistant Professor and Head, with Department of Information Technology. He has worked with Honeywell Technology Solutions, Bangalore, India, as a Senior Aerospace Engineer from 2007 to December 2009. His research interests include security and QOS in Wireless Networks, Network Protocols & Programming, Software Engineering for Velocity Product Development & Automation, and Object Oriented Programming & Framework.