IEEE P802.11

Wireless Access Method and Physical Layer Specification

Proposed Text for Section 5.4, Based on responses to Draft D1 Letter Ballot processed at March 1995 Meeting

Michael Fischer
Digital Ocean Inc.
4242-3 Medical Dr
San Antonio TX 78229 USA
Phone: +1 210 614 4096
Fax: +1 210 614 8192
E-Mail: mfischer@child.com

Abstract: This paper presents the changes to section 5.4 in the Draft Standard P802.11/D1 as a result of the Response to Draft D1 Letter Ballot processed at the March 1995 Meeting . Not all Letter Ballot comments were processed at the March 1995 Meeting.

Action: Adopt the changes in this paper to replace the relevent portions of Section 5.4 of P802.11/D1.

5.4 The Wired Equivalent Privacy Algorithm (WEP)

5.4.1 Introduction

Eavesdropping is a familiar problem to users of other types of wireless technology. P802.11 specifies a wired LAN equivalent data confidentiality algorithm. Wired equivalent privacy is defined as protecting authorized users of a wireless LAN from casual eavesdropping. This service is intended to provide functionality for the Wireless LAN equivalent to that provided by the physical security attributes inherent to a wired media. *The WEP can also be used to provide implicit authentication. The commonly held key can be used to encrypt challenges from the access point or another station*

An ESSDwide, commonly held key permits implict authentification and low-overhead BSS-mobility transitions. Any station in possession of the commonly held key is considered to be pre-authenticated. Stations desiring greater security, may maintain receive privacy tables that associate station-specific, keys with station addresses. The commonly held key is used in cases where this table not present or where the table has no station-specific key corresponding to the source address of the received MSDU.

Data confidentiality depends on an external key management service to authenticate users and distribute data enciphering/deciphering keys. P802.11 specifically recommends against running an 802.11 with privacy but without authentication. While this combination is possible, it leaves the system open to significant security threats.

5.4.2 Properties of the WEP Algorithm

The WEP algorithm has the following properties:

Reasonably Strong:

The security afforded by the algorithm relies on the difficulty of discovering the secret key through a brute-force attack. This in turn is related to the length of the secret key and the frequency of changing keys. WEP allows for the changing of the key (*k*) and frequent changing the Initialization Vector (IV).

Self Synchronizing:

WEP is self-synchronizing for each message. This property is critical for a data-link level encryption algorithm, where "best effort" delivery is assumed and packet loss rates can be high.

Efficient:

The WEP algorithm is efficient and can be implemented in either hardware or software.

***Implicit Authentication*:**

A commonly held key can be used to provide implicit authentication without the need for a separate authentication mechanism.

Exportability:

Every effort has been made to design the WEP system operation so as to maximize the chances of approval of export *from the U.S. of products containing a WEP implementation* via the Commerce Department. However, due to the legal and political climate toward cCryptography at the time of publication, no guarantee *can*~~could~~

be made that any specific 802.11 implementations *that* ~~which~~ uses WEP *will*~~would~~ be exportable from the United States.- *Therefore, the implementation and use of WEP is an 802.11 option.*

~~Therefore, the implementation and use of WEP is an 802.11 option.~~

5.4.3 WEP Theory of Operation

The process of disguising (binary) data in order to hide its information content is called **encryption**[1]. Data that is not enciphered is called **plaintext** (denoted by *P*) and data that is enciphered is called **ciphertext** (denoted by *C*). The process of turning ciphertext back into plaintext is called **decryption**. A **cryptographic algorithm**, or cipher, is a mathematical function used for enciphering or deciphering data. Modern cryptographic algorithms use a key *sequence* (denoted by *k*) to modify their output. The encryption function *E* operates on *P* to produce *C:*

$$E_k(P) = C$$

In the reverse process, the decryption function *D* operates on *C* to produce *P:*

$$D_k(C) = P$$

As illustrated in Figure 5-21, note that if the same key is used for encryption and decryption then

$$D_k(E_k(P)) = P$$

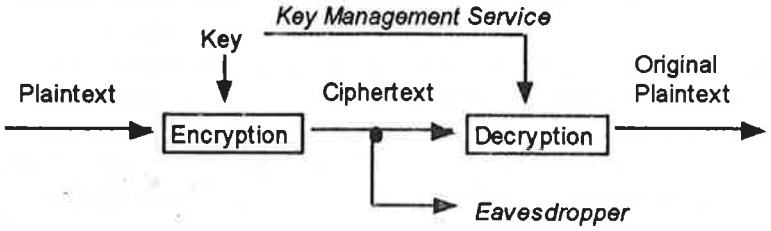


Figure 5-21: A Confidential Data Channel

The WEP algorithm ~~proposed in this submission~~ is a form of electronic code book in which a block of plaintext is bitwise XOR'd with a pseudo random key sequence of equal length. The key sequence is generated by the WEP algorithm.

[1]Bruce Schneier, "Applied Cryptography, Protocols, Algorithms and Source Code in C", John Wiley & Sons, Inc. 1994

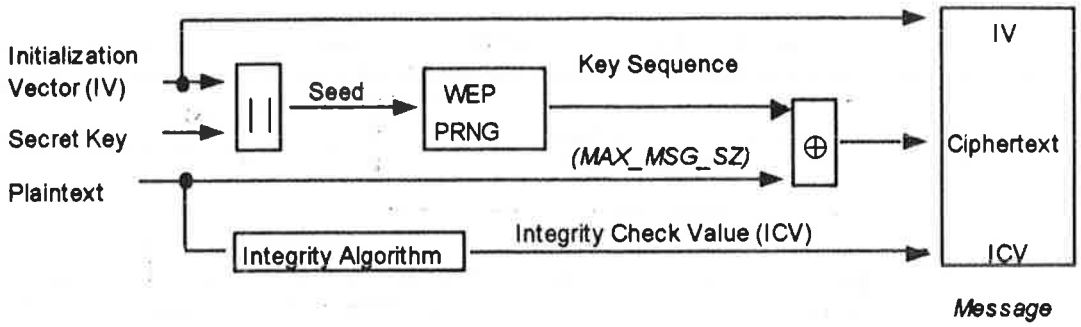


Figure 5-22: WEP Encipherment Block Diagram

Referring to Figure 5-22 and following from left to right, encipherment begins with a **secret key** that has been distributed to cooperating stations by an external key management service. WEP is a symmetric algorithm in which the same key is used for encipherment and decipherment.

The secret key is *concated with* ~~combined with~~ an **initialization vector** (IV) and the resulting **seed** is input to a **pseudo random number generator** (PRNG). The PRNG outputs a **key sequence** *k* of pseudo-random bits equal in length to the largest possible MSDU. Two processes are applied to the plaintext MSDU. To protect against unauthorized data modification, an integrity algorithm operates on *P* to produce an **integrity check value** (ICV). Encipherment is then accomplished by mathematically combining the key sequence with *P*. The output of the process is a **message** containing the resulting ciphertext, the IV, and the ICV.

The WEP PRNG is the critical component of this process, since it transforms a relatively short secret key into an arbitrarily long key sequence. This greatly simplifies the task of key distribution as only the secret key needs to be communicated between stations. The IV extends the useful lifetime of the secret key and provides the self-synchronous property of the algorithm. The secret key remains constant while the IV changes periodically. Each new IV results in a new seed and key sequence, thus there is a one-to-one correspondence between the IV and *k*. The IV may be changed as frequently as every MSDU and, since it travels with the message, the receiver will always be able to decipher any message. The IV may be transmitted in the clear since it does not provide an attacker with any information about the secret key.

Because IV and the ICV must be transmitted with the MSDU, fragmentation may be invoked. The WEP algorithm is applied to an MSDU. The {IV, MSDU, ICV} triplet forms the actual data to be sent in the data frame.

For WEP protected ~~Data~~ frames, the first *four* octets of the frame contain the IV *field* for the MSDU. *This field shall contain two sub-fields: A 1-octet field that contains the confidentiality algorithm ID, followed by a 3-octet field that contains the initialization vector* ~~The WEP IV is 16 bits~~. *The 64-bit PRNG seed is formed using the secret key as the most significant 40 bits and the initialization vector as the least significant 24 bits.* The IV is followed by the MSDU, which is followed by the ICV. The WEP ICV is 32 bits. The WEP Integrity Check algorithm is CRC–32.

The entire {IV, MSDU, ICV} package may be split into several fragments (depending on the realtive values of the MSDU and the active MPDU size).

As stated previously, WEP combines *k* with *P* using bitwise XOR.

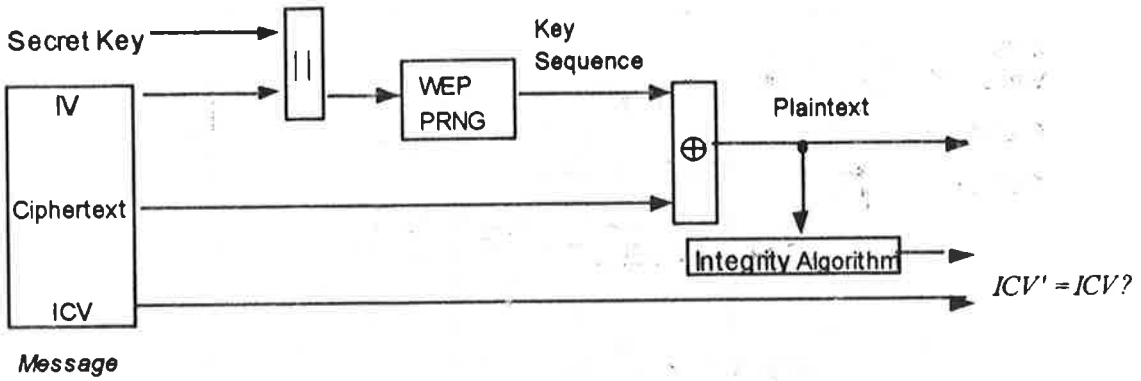


Figure 5-23: WEP Decipherment Block Diagram

Referring to Figure 5-23 and following from left to right, decipherment begins with the arrival of a message. The IV of the incoming message is used to generate the key sequence necessary to decipher the incoming message. Combining the ciphertext with the proper key sequence yields the original plaintext. *Correct decipherment is* ~~If desired, this may be~~ verified by performing the integrity algorithm on the recovered plaintext and comparing the output ICV' to the ICV transmitted with the message. *If ICV' is not equal to ICV, the received MSDU is not passed to LLC and and error indication is sent to MAC management.*

5.4.4 WEP Algorithm Specification

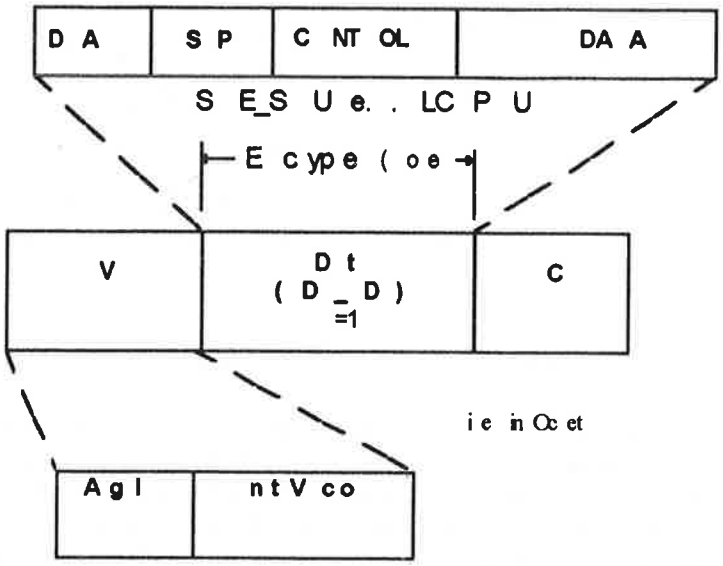
The specific PRNG algorithm is unspecified at present. Reviewers of this draft are encouraged to comment on appropriate PRNG algorithms for adoption by 802.11.

5.4.5 *Relationship of WEP to IEEE 802.10, Secure Data Exchange (SDE):*

The WEP uses a subset of the IEEE 802.10 SDE structure shown in Figure 3-1 of section 3.1.1.3. Figure 5-24 shows the SDE_PDU as constructed by the WEP.

The 802.10 SDE settings for 802.11 WEP shall be clear header length =null, protected header length =null, pad =null, and ICV = 32 bits. The data field shall include a 32Ðbit IV field immediately preceding the MSDU. This field shall contain two sub-fields: A 1-octet field that contains the confidentiality algorithm ID, and a 3-octet field that contains the initialization vector.

The 802.10 WEP mechanism allows for 802.10 SDE entities to be operating in the same protocol stack. If a user chooses to deploy an SDE environment that requires SDE settings more comprehensive than those in the WEP subset, and/or based on an encryption algorithm not supported for the WEP function, that user may disable the WEP function, thereby avoiding the overhead of performing twice on the same MSDU. This is consistent with the 802.10 model, in which lowerÐlayer SDE entities are generally disabled when higherÐlayer SDE entities are present.



Note: The encipherment procedure is expanded to DE_ DU by 8 Octets for the Initialization Vector (IV) field and 4 for the Integrity Check Value (ICV). The ICV is calculated on the Data field only.

Figure 5-24: Construction of WEP SDE_PDU