Wired Equivalent Privacy (WEP)

Aldrich Asuncion & Brian Guadalupe

1 Introduction

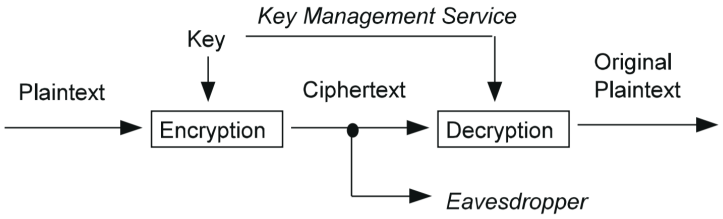
General idea

Since the introduction of wireless networks, its open-air communication makes it susceptible to eavesdropping and remote sniffing by anyone within range of a wireless access point. Now, this begs the question: How to send protocol data units securely from one entity to another?

In order for this to happen, it requires both cryptography (for security) and encoding (for data integrity).

What is WEP?

WEP is a security protocol for wireless networks, based on the IEEE 802.11-1997 standard. Its main goal is to provide security that equates the security mechanisms of wired local access networks (LANs), hence the name.



WEP is a symmetric-key algorithm. It assumes that there is some Key Management Service which securely provides a copy of the same secret key to both the sender and recipient. One such way to allow secure key exchange is the Diffie-Hellman key exchange.

Properties of WEP

From §8.1.2 of the IEEE 802.11-1997 standard, 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. WEP allows for changing of the key and frequent changing of the initialization vector (IV). But this property is soon debunked (more on that later).

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

Efficient. May be implemented in either hardware or software.

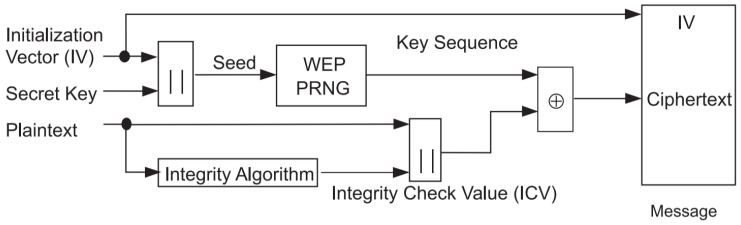
May be exportable. Due to the political climate regarding the export of cryptography from the US at that time, the key size was made limited (WEP-40). Soon after, a variant (WEP-104) was developed that uses a longer key size.

Optional. The implementation and use of WEP is an IEEE 802.11 option.

2 Encryption and Encoding

Encryption process

The following shows the block diagram of the encryption process. The concatenation operator is denoted by ||.



We will take a look at two of the components in the process, namely, the pseudorandom number generator (PRNG) and the integrity check algorithm.

Pseudorandom number generator (PRNG)

WEP uses the **RC4** algorithm for its PRNG. RC4 is a stream cipher designed by Ron Rivest in 1987. Originally a trade secret, the source code of RC4 was anonymously posted in 1994, and then was reverse-engineered within days.

Overview of RC4

RC4 generates a keystream, which is a pseudorandom stream of bits. To generate the keystream, the cipher uses an internal state which consists of two parts:

1. A permutation of all 256 possible bytes S
2. Two 8-bit index-pointers i and j

Key-scheduling algorithm

KSA is the RC4 key-scheduling algorithm, which initializes i, j, and the permutation S based on a supplied key K.

```
1: procedure KSA(K)
2:     for i = 0 to 255 do
3:         S[i] = i
4:     end for
5:     j = 0
6:     for i = 0 to 255 do
7:         j = (j + S[i] + K[i mod K.length]) mod 256
8:         Swap(S[i], S[j])
9:     end for
10: end procedure
```

Pseudorandom generator

For as many iterations as needed, the PRG updates the state and produces one byte (modulo 256) of output.

```
1: i = 0
2: j = 0
3: procedure PRG
4:     i = (i + 1) mod 256
5:     j = (j + S[i]) mod 256
6:     Swap(S[i], S[j])
7:     z = S[(S[i] + S[j]) mod 256]
8:     return z
9: end procedure
```

Integrity check algorithm

The WEP integrity check algorithm uses **CRC-32**, a 32-bit *cyclic redundancy check*. From the name itself, CRCs are based on cyclic codes, and their popularity arises from ease of implementation in binary hardware.

CRC-32 algorithm

CRC-32 uses the generating polynomial of degree 32 as defined below:

$$G(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$$

Definition. *Suppose that the plaintext is expressed as a polynomial $P(x)$. Then*

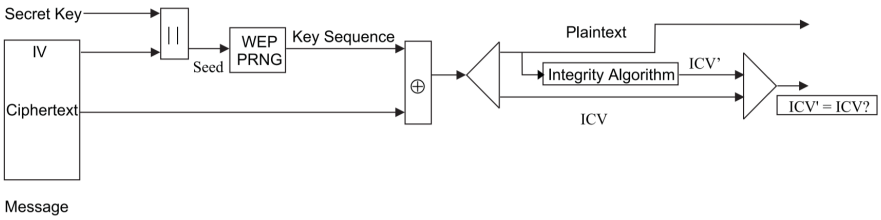
$$\mathrm{CRC}\left(P(x)\right) = x^{32}P(x) \bmod G(x)$$

A special property of the CRC function is that it is linear.

$$\text{CRC}\left(P(x) \oplus Q(x)\right) = \text{CRC}\left(P(x)\right) \oplus \text{CRC}\left(Q(x)\right)$$

3 Decryption and Verification

Using the initialization vector, the secret key and RC4, the recipient first decrypts the ciphertext.



The recipient then verifies the recovered plaintext.

4 Vulnerabilities

In this section, we explore two types of vulnerabilities, namely, IV reuse and unauthorized plaintext modification. But first, we recap on how WEP encrypts data.

Recap: How WEP encrypts data

Given the plaintext (as a polynomial) $P(x)$, we first attach the CRC-32 checksum to it:

$$x^{32}P(x) \oplus \text{CRC}\left(P(x)\right)$$

Recall that $\text{CRC}\left(P(x)\right) = x^{32}P(x) \bmod G(x)$. Then the RC4 algorithm generates a key $\text{RC4}_{\text{IV, Secret Message}}(x)$, and we encrypt the plaintext by adding the key to it:

$$C(x) = x^{32}P(x) \oplus \text{CRC}\left(P(x)\right) \oplus \text{RC4}_{\text{IV, Secret Message}}(x)$$

IV reuse

If two packets of data use the same, publicly viewable IV, then any outside party can view the message. Recall that in WEP, the secret message is not frequently changed, only the IV is frequently changed!

Suppose an attacker acquires the following:

$$C_1(x) = x^{32}P(x) \oplus \text{CRC}\left(P(x)\right) \oplus \text{RC4}_{\text{IV, Secret Message}}(x)$$
$$C_2(x) = x^{32}Q(x) \oplus \text{CRC}\left(Q(x)\right) \oplus \text{RC4}_{\text{IV, Secret Message}}(x)$$

Note that by simply adding the intercepted blocks together:

$$\begin{aligned} C_1(x) \oplus C_2(x) &= x^{32}P(x) \oplus \text{CRC}\left(P(x)\right) \oplus \text{RC4}_{\text{IV, Secret Message}}(x) \\ &\quad \oplus x^{32}Q(x) \oplus \text{CRC}\left(Q(x)\right) \oplus \text{RC4}_{\text{IV, Secret Message}}(x) \\ C_1(x) \oplus C_2(x) &= x^{32}P(x) \oplus \text{CRC}\left(P(x)\right) \oplus x^{32}Q(x) \oplus \text{CRC}\left(Q(x)\right) \end{aligned}$$

The packets are now vulnerable to a **known-plaintext attack**. If the attacker knows either $P(x)$ or $Q(x)$ then it is possible to recover the other message.

Since the initialization vector is only 24 bits long, there are only $2^{24} = 16777216$ choices for the IV. Because of this, an attacker can build a decryption dictionary for all keystreams.

Unauthorized plaintext modification

Suppose the attacker wants to modify the sent message. Specifically, the attacker wants to add a polynomial $Q(x)$ to the message. All the attacker has to do is intercept the message and add $x^{32}Q(x) \oplus \mathrm{CRC}\left(Q(x)\right)$ to it.

We now prove that this still produces a valid message.

Proof.

$$
\begin{aligned}
& x^{32}P(x) \oplus \mathrm{CRC}\left(P(x)\right) \oplus \mathrm{RC4}_{\text{IV, Secret Message}}(x) \oplus x^{32}Q(x) \oplus \mathrm{CRC}\left(Q(x)\right) \\
&= \left(P(x) \oplus Q(x)\right) x^{32} \oplus \mathrm{CRC}\left(P(x)\right) \oplus \mathrm{CRC}\left(Q(x)\right) \oplus \mathrm{RC4}_{\text{IV, Secret Message}}(x) \\
&= \left(P(x) \oplus Q(x)\right) x^{32} \oplus \left(x^{32}P(x) \bmod G(x)\right) \oplus \left(x^{32}Q(x) \bmod G(x)\right) \oplus \mathrm{RC4}_{\text{IV, Secret Message}}(x) \\
&= \left(P(x) \oplus Q(x)\right) x^{32} \oplus \left(\left(x^{32}P(x) \oplus x^{32}Q(x)\right) \bmod G(x)\right) \oplus \mathrm{RC4}_{\text{IV, Secret Message}}(x) \\
&= \left(P(x) \oplus Q(x)\right) x^{32} \oplus \mathrm{CRC}\left(P(x) \oplus Q(x)\right) \oplus \mathrm{RC4}_{\text{IV, Secret Message}}(x)
\end{aligned}
$$

□

The attacker has tampered with the message **and the checksum**! The key vulnerability is that the XOR operation used by the RC4 stream cipher is associative, and that the CRC-32 operation is linear.

It is also important to note that CRC-32 is **not a cryptographic hash function**, so it doesn't offer the security of being one-way.

5 Attacks on WEP

Attacks based on statistical analysis of attempted packets have been found since 2001.

FMS attack (Fluhrer, Mantin, Shamir) The attack needs to intercept 4 to 6 million packets to succeed in obtaining the full secret key with a success probability of at least 50%.

KoreK attack The number of captured packets is reduced to about 700,000 for 50% success probability.

PTW attack (Pyshkin, Tews, Weinmann) The attack needs just about 35,000 to 40,000 packets for 50% success probability, which can be collected in less than 60 seconds on a fast network. Only a few seconds of CPU time is needed to execute the attack.

Aftermath

The Wi-Fi Alliance announced that WEP had been superseded by Wi-Fi Protected Access (WPA) in 2003. Both WEP-40 and WEP-104 were deprecated in 2004, with the ratification of the full IEEE 802.11i standard, where WPA2 was made available.

References

[1] IEEE standard for wireless LAN medium access control (MAC) and physical layer (PHY) specifications 1–445.

[2] Tews, E. & Beck, M. Practical attacks against WEP and WPA. WiSec '09, 79–86 (ACM Press). URL `http://portal.acm.org/citation.cfm?doid=1514274.1514286`.

[3] Kong, J., Gerla, M., Prabhu, B. & Gadh, R. An overview of network security in WLANs. In *Handbook of Wireless Local Area Networks*, 476–496 (CRC Press).