Design of Multithreaded Security Algorithm using Session Keys to Secure Initialization Vector (IV) for Enhancing the Wireless Protocol WEP

D.J.Evanjaline
Dept. of Comp. Science,
Nehru Memorial College,
Puthanampatti

E. Kirubakaran
Bharath Heavy Electricals Ltd,
Tiruchirappalli

ABSTRACT
The 802.11 standard defines the Wired Equivalent Privacy (WEP) and encapsulation of data frames. It is intended to provide data privacy to the level of a wired network. WEP suffered threat of attacks from hackers owing to certain security shortcomings in the WEP protocol. Lately, many new protocols like WiFi Protected Access (WPA), WPA2, Robust Secure Network (RSN) and 802.11i have come into being, yet their implementation is fairly limited. Despite its shortcomings one cannot undermine the importance of WEP as it still remains the most widely used system and we chose to address certain security issues and propose some modifications to make it more secure. In this paper we have proposed a modification to the existing WEP protocol to make it more secure. We achieve Message Privacy by ensuring that the encryption is not breached. The proposed enhancements attempt to rectify the vulnerabilities to enhance the WEP with Private IV and Session Time for improved authentication process. In the proposed algorithm we can use all possible 2^{24} different IVs without making them predictable for an attacker, eliminates the IV collision ensuring Message Privacy that further strengthens security of the existing WEP.

Key Words—Wireless networks, security, Session Time, IV, WEP.

1. INTRODUCTION
With the widespread use of wireless networks, securing data transmission becomes a basic requirement. The IEEE 802.11 standard which defines wireless networks communication, has proposed in its second version IEEE 802.11b a new protocol to offer some wired-like security services, such as: data privacy, data integrity, and authentication. Unfortunately, this protocol falls short these objectives, and has shown many threats which were exploited by intruders. The Task Group I started developing a more secured standard: the IEEE 802.11i. Meanwhile, the Wi-Fi alliance Group together with IEEE proposed the Wi-Fi Protected Access (WPA), which enhances security model of WEP using the well known authentication. Despite their efficiency, these two standards, and especially 802.11i, need hardware renew and reconsideration of security architecture. This paper begins with an introduction of WEP's well-known vulnerability, followed by a description of our solution, and a comparaison between the two.

2. THE WEP PROTOCOL
2.1. The WEP Mechanism
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. XORing the key stream with the cipher text yields the original plaintext.

This mode of operation makes stream ciphers vulnerable to several attacks. 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[1].

WEP has defenses 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 Cyclic Redundancy Check-32 (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[1].

The IV 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.

In the first and foremost stage each member of the Basic Service Set (BSS) is initialized with a shared secret key *K*, (The details of initialization are not known. It could be either end user contacting the network administrator for the shared key or network administrator distributing the keys to the legitimate user). Before sending the frame the sender calculates the (CRC) of the frame payload and appends it to the frame, which now becomes the plaintext.

2.2 Security Flaws in WEP

WEP uses the RC4 encryption algorithm [2], 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. XORing the key stream with the cipher text yields the original plaintext.

This mode of operation makes stream ciphers vulnerable to several attacks. 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 defenses 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[5].

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[14].

Weaknesses in the RC4 were discovered due to which WEP failed [3] to achieve its security goals. Some of its vulnerabilities are listed below:

- Subtle weaknesses that can be exploited to crack keys.
- The presence of relatively short IVs and keys that remains static. If an individual collects enough frames based on the same IV, he can determine the shared values among them, i.e., the key stream or the shared secret key. This leads to decrypting any of the 802.11 frames.
- The implementation has so far been based on 64-bit with 24-bit initialization vector resulting in only a 40-bit encryption.
- RC4 is susceptible to brute force and word list attacks. Providing a stronger encryption mechanism at higher levels might improve the security of WEP[2].

2.3 Proposed Mechanism

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. To prevent key sequence reuse, the WEP recommends varying key sequences for payload so that the WEP uses a 24-bit IV , 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 thereby exposing the system to key sequence reuse attacks. Thus, a popular pitfall of stream ciphers servers is the compromise in the WEP recommendations. The secret key K always remain the same, but the change in the key sequence is due to the change in the IV every time. We observe that there exist chances 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.

$C1 = \{P1, ICV (P1)\} \oplus RC4 (IV, K)$
$C2 = \{P2, ICV (P2)\} \oplus RC4 (IV, K)$

In the above example, RC4 (IV, K) are reused. When the same IV is used for encrypting two different plaintexts, it is called a collision.

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

$P2 = (C1 \oplus C2) \oplus P1$

To find the key sequence reuse is easy and described as follows. The IVs are public and when they are sent with the cipher texts,

the intruder 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 224. Experimental result depicts that the 1st collision occurs after transmitting 5000 packets which are few minutes after the data transmission. Considering the above, 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 already.

The following procedure is to ensure longer IV/Key reuse period.

1. The IV is initialized with a 24 bit random number each time the station is started.
2. The Session key is derived at the time of encryption
3. For every new frame to be encrypted, the IV encrypted with session key without using the same static key.

This guarantees that each IV will be unique.

2.4. Session Key Derivation

J.Walker [4] recommends a session key derivation algorithm in the case of a manually configured base key, as used by WEP today. It does not recommend an algorithm for session key derivation when dynamic keying is available, because the scheme should incorporate state from the dynamic keying operation, to tie the key to the particular session that negotiated the key.
This algorithm produces two session keys, one for sending and the other for receiving.

i. Concatenate the (a) BSSID, (b) the sender's MAC address, and (c) the receiver's MAC address to produce a string. The order is important, as the two MAC addresses are reversed for sending and receiving.

ii. Using the base key (manually configured key) and an IV of 128 zero bits, run the OCB-AES algorithm on the concatenated string. The session key is the authentication tag output by this:

$session\text{-}key \leftarrow \text{OCB-AES-tag}_{base\text{-}key}(0, BSSID \mid sender\text{-}mac\text{-}addr \mid receiver\text{-}mac\text{-}address)$

Here '*a* | *b*' means the concatenation of strings *a* and *b*.

The motives for this algorithm are (a) to remove the base-key from direct attack and (b) weakly tie the session key to the particular parties using it. Under this algorithm different sets of peers use different session keys, even though all the members of the BSS share the same base key. Note that the keys produced by this algorithm are still subject to dictionary attack when the base key is a password or derived from a password by techniques such as PCKS #5. And all the keys are subject to spoofing if the base key is revealed to an adversary. There is no magic that can avoid these weaknesses[11].

2.5 Encryption of IV

The proposed enhancements attempt to rectify the vulnerabilities to enhance the WEP with Private IV and Session Time for improved authentication process. Encrypt the IV by the Session key will disable an intruder's ability to easily map IVs to known key sequences.

Specifically, the WEP's cipher text C is
(IV, P ⊕ RC4 (IV, K)),

whereas the WEP with private IV uses
C= (K1⊕ IV, P ⊕ RC4 (IV, K)) where K1 is the Session key.

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 reuse can be largely avoided.

The KO/EO and KE/EE threads in the above figure refer to the processing fashion while the KO thread generates a sequence of keystream bits based on the odd increment of the key k, and EO threads use those keystream bits to encrypt bits at the odd position of the IV. Meanwhile, KE and EE threads of the keystream generation and encryption processes work in the same processing fashion but with even increment of k for KE and even bit position of the IV for EE[9].

The encryption phase of our model is designed to encrypt the given plaintext by applying XOR operation between the keystream bits and the plaintext bits. The encryption process is accomplished by dividing the activities into two parts handled by two threads and synchronized with their analogue threads in the keystream generator component. The two threads which control the encryption phase have an extra job of monitoring key stream bits availability.

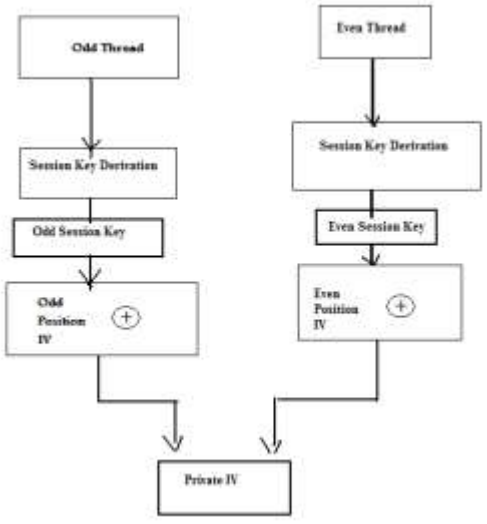

Figure 1: Encryption of IV using Two threads

The aim of designing a multithreaded model is to increase the performance.

2.6 Analysis

As mentioned earlier in this document, the aim of designing a multithreaded model for stream cipher is to increase the performance since the current stream ciphers tend to use intensive calculations for keystream generation in order to increase the provided security level. Single thread execution (sequential) is a one-path execution whereby the work flow of threads associated with each core will start the execution at time *t0* and finish at *tm.*

The total time required is computed as the following

$$t_m = R \sum_{i-1}^{m} + timeC(i)$$

where *R* is the number of rounds, *i* is the number of components to execute in each round, *m* is the total number of components in the overall rounds, and *time(Ci)* is the time required to execute each component. In contrast to the single-thread execution, multi-thread execution provides multiple-paths to accomplish the same task. The total time required to accomplish the same work flow as discussed above is divided among multiple cores, resulting in higher throughput and performance[9].

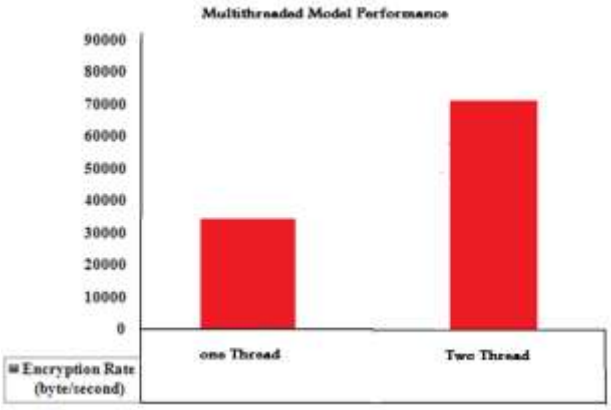

Figure 2: Comparison of Multithreaded Model Performance

3. CONCLUSIONS

Existing WEP protocol has been shown to be vulnerable to different kinds of cryptanalytic attacks [6]. These stem from inappropriate usage of cryptography and not because of the key size. The possible drawback one can identify with our method is the computational overhead associated with generating, and transmitting the session keys at the access point. In this paper we have shown that our proposed modification to the existing WEP protocol makes it more secure and robust in terms of Message Privacy. The fact that we frequently change the shared secret keys through the WEP mechanism makes any kind of cryptanalytic attack futile. The IV collision problem has been successfully resolved by our proposed private IV generation algorithm that further enhanced the security of WEP. IEEE 802.11i standards have explicitly talked about key management which is must for its security but comes with the overhead of upgrading the hardware. Our proposed solution is a very efficient alternative till actual hardware is available and deployed for 802.11i. Our proposed system works well with the existing hardware and gives an edge over the present WEP protocol.

4. REFERENCES

[01] N.Borisov, I. Goldberg, and D.Wagner. Intercepting mobile communications: The insecurity of 802.11. In *MOBICOM 2001*, Rome, Italy, July 2001.

[2] Jim Geier, "802.11 WEP: Concepts and Vulnerability", http://www.wifiplanet.com/tutorials/article.php/1368661

[3]"An Examination of Security Algorithm flaws in Wireless Networks", University of Maryland,, OCT 2004

[04] J.R.Walker. Unsafe at any key size; an analysis of the WEP encapsulation. IEEE Document 802.11-00/362, Oct 2000.

[05] W.A.Arbaugh, N.Shankar, and Y.J Wan. Your 802.11 wireless network has no clothes. In *IEEE International Conference on Wireless LANs and Home Networks*.

[06] J.Edney and W.A.Arabaugh, "Real 802.11 Security Wi-Fi Protected Access and 802.11i", 2004, Pearsons Education Inc.

[07] William Stallings, Cryptography and Network Security, Principles and Practices", 3rd Edition, 2003, Pearsons Educations.

[08] A.Stubblefield, J.Ioannaidis, and A.D.Rubin. Using the Fluhrer Mantin and Shamir attack to break WEP. *ACM Transactions on Information and Security Security, Vol. 7, No. 2, May 2004, Pages 319-332*.

[9] Khaled M. Suwais and Azman Bin Samsudin, High Performance Multithreaded Model for Stream Cipher, IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008.

[10] N.Chandran, D.Sampath. Strengthening WEP Protocol for Wireless Networks using Block Chaining Algorithm with Variable Encrypting function Mechanism. 2004 IEEE Sarnoff Symposium on Advances in Wired and Wireless Communications, Princeton, NJ, USA.

[11] D.Purandare, R.Guha. Enhancing Message Privacy in WEP. To appear in International Workshop on Wireless Information Systems, ICEIS 2005, Miami USA.

[12]. Hisham A. Kholidy, Khaled S. Alghathbar, "Adapting and accelerating the Stream Cipher Algorithm "RC4" using "Ultra Gridsec" and "HIMAN" and use it to secure "HIMAN" Data", Journal of Information Assurance and Security 4 (2009) 474-483.

[13]. Atul Kahate, "Cryptography & Network Security ", Tata McGraw Hill, 2003.

[14]. Jim Geier, "802.11 WEP: Concepts and Vulnerability", http://www.wifiplanet.com/tutorials/article.php/1368661