

IEEE 802.11/WiFi Security: Report on “The Final Nail in WEP’s Coffin”

Hyung-Joon Kim

Dept. of Electrical and Computer Engineering
Stevens Institute of Technology, Hoboken, New Jersey

1. Introduction

Recently, with rapidly evolving technology, networking has become one of the most important segments of computing environments. Especially, wireless networking has been in high demand by average home users as well as business users due to its characteristic merit. Today, wireless networks are virtually everywhere: home, public area, and enterprise. Furthermore, this trend is expected to boost growth even more in coming years. Meanwhile, as wireless networking is winning popularity, people have started concerning about its security. As a result, the early security scheme, Wired Equivalent Privacy (WEP), was developed to bring security into “unarmed” radio communication as the name implies.

However, it turned out to be one of today’s most infamous security mechanisms. Since WEP was alleged to be theoretically broken, a number of practical attacks have already been reported and tools are now readily available to unskilled attackers. Undeniably, everyone knows WEP is bad. Despite its serious vulnerabilities, WEP is believed to be still widely used. According to [1], Andrea Bittau et al. found that “in London, 76% of the encrypted networks in [their] sample used WEP, and in Seattle 85% of them used WEP.” At this critical moment, the paper [1] gives real evidence for the fact that WEP must be completely dumped away. This report discusses the fundamental ideas and security issues of WEP that are addressed in the paper [1].

2. Fundamental Ideas behind WEP

WEP is a security scheme designed to bring wired equivalent security into attack-prone wireless communications specified by 802.11. At first, the fundamental ideas behind WEP look good enough to make wireless communications as logically secure as wired ones. The logic is simple as usual; that is, secure confidentiality and integrity by employing encryption, security’s old best friend. By encrypting transmitted data, a system can be protected from eavesdropping and modification by perpetrators.

For encryption, WEP uses a stream cipher based on RC4, with a secret key shared between a wireless terminal and an access point (AP). The key distribution is not specified by the standard; therefore, the key could be preloaded by the manufacturer, could be distributed beforehand over the wired network, or could be exchanged by using a public-key scheme. By any means, a 40-bit key (initially used for 802.11) or a 104-bit key (implemented later for 802.11b) must be shared prior to any encryption. The shared key is concatenated with a 24-bit Initialization Vector (IV) to form a *seed* of size of either 64 bits or 128-bit depending on the size of the secret key. The seed is then inserted into a RC4’s *Pseudo Random Generator*, which generate a so-called *keystream*. Finally, the output of RC4, the keystream, is XORed with a plain-text message to generate a cipher-text. In addition to confidentiality, an integrity check using a common checksum algorithm, CRC32, is applied

in order to protect the message against unauthorized modification. In fact, this process is done before encryption by appending a checksum to a packet payload. Thus, the actual message body and checksum data altogether form the plain-text to be encrypted. After encryption, the cipher-text is transmitted to a receiver along with the IV used to encrypt the plain-text message.

For decryption, the process is simply performed backward. The IV is extracted from the transmitted data and used to generate the same keystream as the one used for encryption. The generated keystream is then XORed with the cipher-text, the rest of the transmitted data, to retrieve the original plain-text message. Then, the CRC of the message is checked.

In addition, WEP optionally provides authentication to prevent unauthorized parties from accessing “unbounded” wireless networks. If shared key authentication is enabled, WEP uses a four-way-handshake scheme to control access to an open radio network. When a wireless terminal tries to access an AP, the AP replies the request with a random plain-text challenge. Upon the receipt of the challenge text, the wireless terminal encrypts it with a shared key and sends it back to the AP. The AP then decrypts the encrypted challenge text and compares it to the original message sent earlier. If they match, the AP sends a confirmation message and accepts the access of the wireless terminal to the network. Usually, WEP uses the same shared key for authentication as the one used for encryption.

3. History of Security Issues

Since the debut as part of the 802.11 standard in 1999, there have been numerous criticisms about the weakness of WEP and the lack of effort to enforce a better security solution. Among many, a series of discussions as to vulnerabilities and practical attacks stands out and well represents a history of “insecure” WEP. In 2001, Nikita Borisov et al. published the paper [2] that addresses fairly straightforward threat models to attack WEP. Two major vulnerabilities targeted by the attack models addressed in the paper [2] include the risks of keystream reuse and the improper use of CRC32 for the purpose of cryptographically secure message authentication. In later 2001, Scott Fluhrer et al. published the paper [3] that addresses another devastating attack against WEP, but this time the core of WEP operation, RC4, itself is on target. The paper [3] introduces an efficient way to derive some key bits from the keystream and identifies a lot of weak keys vulnerable to this threat. In the paper [3], Fluhrer et al. asserts that, if an attacker repeats this attack, it would be a matter of time before the entire key is derived. Interestingly, in 2002, Adam Stubblefield, a summer intern at AT&T Labs, et al. [4] actually implemented an attack addressed in [3] in several hours and successfully broke WEP by recovering 128-bit key. The paper [4] ultimately proves that threats to WEP seem to be effective not only in theory but practically achievable in a reasonable amount of time. Besides, the paper [4] claims that there are readily available tools for breaking WEP keys throughout the Internet at the time of publication. Despite the early disclosure of successful attacks, WEP is still widely used mostly because of the exponential increase in the number of personal-use laptops in recent years and the improper security guidelines of vendors. In general, untrained home users tend to prefer plug-and-play installation to setting up configuration parameters.

4. Discussions of Vulnerabilities and Attacks

A typical brute-force attack is unlikely to win against WEP that supports the key size of 104-bit. Indeed, increasing its key size was the initial improvement on weak WEP. However, that falls short of providing stronger security. Many practical threats to WEP are nearly independent of its key size. One of them is an attack against the shared key authentication scheme where a plain-text challenge and its encrypted version are exchanged during a four-way handshaking. A snooper can hijack two messages and simply XOR them to recover a keystream. If a keystream is reused, the snooper can decrypt the transmitted data in future. The immediate response to this threat was using access control techniques such as Service Set Identification (SSID) and Media Access Control (MAC) filtering. However, both methods are not resistant to a common spoofing attack.

Another threat comes from weak IV – 24 bits pre-pended to a secret key. In practice, once a secret key is established, it generally lasts unchanged for a long period of time. Besides, a poorly implemented 802.11 card set IV to 0 when installed and monotonously increment it when sending each packet. Even if the card chooses a random IV for each packet, there are only 2^{24} IVs available. That means, sooner or later, they need to be reused. Statistically, listening for a large amount of packets is reasonably effective to predict weak IVs, which are more frequently used over time. As a consequence, vendors patched hardware to mitigate these vulnerabilities, but it made things even worse because now there are even fewer than 2^{24} IVs. Automated tools for the weak IV attack have been already around and quite successful results have been reported – literally, WEP is completely broken. The paper [1] focuses on accelerating such attack without passively listening for a large amount of packets.

4.1 Fragmentation Attacks for Arbitrary Data Transmission

The paper [1] introduces a novel attack using a critical design flaw in WEP – designed to secure 802.11. Ironically, the property of 802.11, namely fragmentation in data link layer, can be used against WEP. The idea comes from the fact a single logical packet (an encapsulated higher-level datagram) is splitted into smaller packets, limited by Maximum Transmission Unit (MTU), and therefore transmitted in multiple fragments. According to [1], the threats of fragmentation attacks are as follows:

1. Eavesdrop a single packet. Since the initial portion of practically all 802.11 data frames contains LLC/SNAP header, the first 8 bytes of plain-text is virtually known. Then, by XORing the known plain-text with the cipher-text that has been intercepted, the first 8 bytes of keystream can be recovered. Knowing the first 8 bytes of the keystream, it's possible to encrypt at least 8 bytes of payload in future.
2. Since the 802.11 standard allows fragmentation at the MAC layer and specifies each fragment is encrypted independently, it's possible to send multiple fragments (up to 16) using the same keystream whose first 8 bytes have been recovered in Step 1. Since each payload of the fragments requires a CRC (4 bytes), arbitrary 64 bytes (4 bytes excluding CRC \times a maximum of 16 fragments) of data can be immediately injected. Furthermore, IP fragmentation can be used to transmit even larger packets.

3. Since the 802.11 standard specifies only the data portion is encrypted and does not check if a complete payload is replayed as a fragment, the attacker can use fragmentation again in order to pre-pend a forged IP header to the encrypted packet that has been eavesdropped, and replay the packet. Upon reception, the AP will decrypt the packet and then send it in plain-text back to the forged IP address where the attacker is probably waiting. This attack is possible because, in essence, WEP is effective on only a wireless link. Thus, if an AP is connected to non-wireless network (typically the Internet), the AP itself can be used for decryption as above.

4.2 Keystream Attacks for IV Dictionary

The paper [1] introduces even more practical attacks by using fragmentation in 802.11. If a large broadcast frame is transmitted in multiple fragments, upon reception the AP will decrypt the encrypted fragments and reassemble them into a single large frame. Then, the AP will relay it. Based on this, [1] demonstrates keystream attacks as follows:

1. Send large broadcast frames in multiple fragments. Eavesdrop the single large broadcast frame relayed from the AP. Then, XOR the cipher-text with the plain-text to recover the keystream. This time, the keystream recovered from the broadcast frame is not just the first 8 bytes, as discussed earlier in 4.1, but a larger keystream. Repeat the process to expand the keystream (up to 1500 byte).
2. Inject a 1500-byte broadcast frame and intercept the relayed version of it. The AP will most likely use a new IV for the relayed payload. Thus, repeating this process up to 2^{24} times will build a complete IV dictionary.
3. Now the attacker has the IV dictionary on hand, so weak IV attacks can be much more effective to decrypt the cipher-text. Even if the IV used for data is not presented in the dictionary, the keystream for the whole packet can be still recovered by using the first 8-byte plain-text and guessing the next byte in each of multiple multicasts (UDP packets).

5. Potential Applications and Future Opportunities

The fragmentation attack addressed in [1] is fairly effective if an attacker can control an Internet host by IP spoofing. Even if not, it's still very powerful to boost up weak IV attacks by building IV dictionary. Ultimately, these attacks can practically break WEP in near real-time. Moreover, the most significant difference is that the attacks in [1] actively generate traffic on their own whereas other attacks that have been previously presented passively wait for a considerably large amount of packets for successful attacking. The only barrier to those attacks is whether or not an 802.11 card allows an attacker to send raw 802.11 frames. Thus, implementations can be further tested on real hardware available in the market.

The lessons learned from WEP themselves are great opportunities for future security design and guidelines. All the flaws in WEP are in interaction; that is, one flaw creates another flaw and all the flaws together make WEP even more seriously vulnerable. That teaches us designing a security scheme requires the system to be carefully examined as a

whole, not only the security mechanism itself but also all the interacting parts such as lower/higher layer protocols. Even non-cryptographic components may behave against a poorly designed security mechanism and bring unintended consequences as fragmentation in 802.11 does. More importantly, the inappropriate use of a single security component may devastate the whole system.

6. Conclusion

The paper [1] and a series of papers about WEP prove that a poor design of security mechanism can never be improved by patching one after another. They also demonstrate that lack of effort by security community to enforce a better solution can make things even worse. Once again, threats to WEP no longer remain theoretical but can be practically achievable by any unskilled attacker with readily available automated tools. Now, it's really time for WEP to be superseded by WPA or WPA2.

Reference

- [1] Andrea Bittau, Mark Handley, and Joshua Lackey, “The Final Nail in WEP’s Coffin,” *Proceedings of the 2006 IEEE Symposium on Security and Privacy (S&P’06)*, pp. 515-525, 2006.
- [2] Nikita Borisov, Ian Goldberg, and David Wagner, “Intercepting Mobile Communications: The Insecurity of 802.11,” *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, ACM Press, pp. 180-189, 2001.
- [3] Scott Fluhrer, Itsik Mantin, and Adi Shamir, “Weakness in the Key Scheduling Algorithm of RC4,” *Proceedings of the 8th Annual Workshop on Selected Areas in Cryptography*, 2001.
- [4] Adam Stubblefield, John Ioannidis, and Aviel D. Rubin, “Using the Fluhrer, Mantin, and Shamir Attack to Break WEP,” *Proceedings of Network and Distributed Systems Security Symposium*, ISOC, pp. 1-11, 2002