Brief yet Scintillating Article about WEP and WPA2

Ashley Oudenne
The University of Texas at Austin
aoudenne@cs.utexas.edu

Jerremy Adams
The University of Texas at Austin
ja7872@cs.utexas.edu

December 12, 2012

Abstract

Interesting vague things about our paper. 200 words or less, and should say something like: In this paper, we present the WEP and WPA Four-Way Handshake protocols in detail and formally model them using the ProVerif Cryptographic Verifier.

1 Introduction

Wireless communication is a staple of 21st century daily life, and security is a serious concern for individuals, businesses, and organizations looking to communicate quickly and securely. To ensure this, different standards have been proposed that define protocols for secure communication.

One such standard was the 802.11 standard ratified in September of 1999, which intended to provide data confidentiality through Wired Equivalent Privacy, or WEP [1]. WEP uses a 40- or 104-bit encryption key that is manually entered into access points and devices. This key never changes, so if it is compromised, all future messages on the network are compromised until every device is manually rekeyed. The intention of WEP was to provide the same level of confidentiality as that of a traditional wired network.

Unfortunately, WEP relied on the RC4 stream cipher, which was thought to be secure but which was actually vulnerable to attacks. The Cyclic Redundancy Check (CRC) checksum algorithm used by WEP does not provide a strong enough integrity guarantee, because it permits the guessing of individual bytes of a packet [8]. Since the CRC is simple a linear function of the message, an attacker can modify an encrypted message and fix the checksum so that the message appears not to have been modified.

To remedy this, IEEE released WPA in 2003 as a temporary remedy for WEP until a new standard could be ratified. WPA was designed to work on devices that were currently using WEP until new hardware was created, so it also uses the RC4 cipher and the CRC checksum mechanism [2]. However, it uses the Temporal Key Integrity Protocol, which implements a key mixing function that combines the root key with an initialization vector *before* passing it to RC4, instead of just concatenating it as in WEP. This prevents related key attacks, to which WEP is susceptible. TKIP also enforces rekeying and sequence counters to thwart replay attacks. It also includes an additional message integrity check called Michael that increases security. Although

WPA provides greater security than WEP, it is still susceptible to some of the same attacks as WEP, because of the insecurity of the RC4 cipher and the CRC checksum algorithm. As a result, the 802.11i standard was ratified in June of 2004 to replace the use of the TKIP protocol (which uses RC4) with AES-based CCMP encryption[3]. This provides strong security for key generation.

Both WPA and WPA2 rely on the computation of a secure key to encrypt data. To avoid the insecurity of WEP (in which the means of calculating the key are sent over the network), both parties begin with the same Pairwise Master Key (PMK) and use this key to compute the Pairwise Transient Key (PTK) which is actually used to encrypt data. Neither of these keys are ever sent over the network. Instead, the wireless access point and the supplicant device engage in the Four-Way Handshake protocol in order to transmit the data necessary to calculate the PTK.

While the Four-Way Handshake is immune to most attacks, it is vulnerable to a Denial-of-Service (DOS) attack. This attack prevents the station and the access point from ever fully authenticating with one another. While an attack of this nature is not particularly devastating, in that it does not leak keys or permit the attacker to corrupt data, the availability of a network should not be influenced by an attacker.

In this paper, we present the WEP and WPA Four-Way Handshake protocols in detail and formally model them using the ProVerif Cryptographic Verifier. In Section 2, we present work by other researchers on the security of WEP and 802.11i. We particularly focus on the security of the Four-Way Handshake protocol. We explain the protocols and how we model them in Section 3. We also describe ProVerif and our attacker model. In Section 4, we show the possible attacks on the WEP protocol and what particular aspects of the protocol lead to these attacks. We will then explain how these aspects have been eliminated in WPA/WPA2 due to the Four-Way Handshake Protocol, and then demonstrate how a Denial-Of-Service attack is still possible. Finally, in Section 5, we summarize our conclusions and suggest future work that could be done on proving the correctness of wireless security.

2 Related Work

There has been much work, both with automatic verifiers and without, on proving the insecurity of WEP. In [7], a large number of insecurities in WEP are discussed without the aid of an automatic verifier. The authors highlight possible attacks resulting from the risk of keystream reuse, due to the fact that encrypting two messages under the same keystream can reveal information about both messages. Since the initialization vectors used to compute the keystreams are re-initialized every time a wireless card is reinserted into a device, there are many opportunities for an attack of this nature.

The authors also discuss the risk of message modification in WEP due to the CRC checksum being a linear function of the message, whihc means that it distributes over XOR. An attacker can arbitrarily modify even messages he hasn't decrypted by simply XORing the ciphertext with some bitstream and the checksum of the bitstream. Messages can be injected into a network because CRC is not dependent on the keystream used to encode a message. Once an attacker learns an initialization vector and its corresponding keystream, the keystream can be reused indefinitely to insert new messages into

the network, because initialization vectors are never checked for freshness. This also allows an attacker to authenticate himself to the network.

The security of 802.11i has also been concluded without the use of automatic verifiers. In [10], He and Mitchell consider each stage of the protocol and conclude its security from a number of possible threats, including malicious access points, session hijacking, eavesdropping, and message deletion. They conclude that it provides effective confidentiality and integrity when the CCMP protocol is used, and that it may provide satisfactory mutual authentication and key management. However, they identify several possible Denial of Service attacks, since the protocol is not designed to ensure liveness.

One of these Denial of Service attacks, identified in [9] and [12], deals with the Four-Way Handshake protocol that is responsible for establishing the Pairwise Transient Key (PTK). Using automatic verification tools, both groups of researchers prove that it is possible to launch a Denial of Service attack on a supplicant in which the supplicant is continually forced to regenerate a new but incorrect PTK, preventing it from ever communicating with the server. [9] solves this problem by suggesting the reuse of the Automatic verification of WEP and 802.11i has been conducted by several researchers. This is particularly important to do because the security flaws in cryptographic protocols are often non-obvious and can rely on particular sequences of messages that isn't usually generated. Automatic verification can explore the entire state space of a protocol and identify possible attacks.

Lafourcade et al. found an attack similar to [**?**] described above using the ProVerif Cryptographic Verifier. By placing multiple messages on the channel encrypted with the same keystream, the attacker is able to recover the contents of encrypted messages. To prevent this attack, they then implemented a version of WEP in which all intialization vectors were guaranteed to be unique. This protocol was considered secure by ProVerif.

802.11i has also been automatically verified, in particular by Changhua He and John C. Mitchell.

3 Modeling of Protocols

3.1 Protocol Descriptions

3.1.1 WEP

WEP, described in [7], uses a one-message protocol to transmit data between two parties that relies on a secret key k that has been shared between them. The intention of this protocol is to provide the same confidentiality as that of a wired network [1]. Before a message is sent, an integrity checksum *C(m)* is computed on the message so that the recipient can verify that the message has not been altered in transit. The message is concatenated to this checksum to form the plaintext P.

The plaintext is now encrypted using the RC4 cipher. The sender chooses an initialization vector IV and uses this vector along with the secret key k to generate a keystream, which is a long sequence of pseudorandom bits. The sender then uses exclusive-or (or XOR, denoted by $\oplus$) to XOR P and the keystream to generate ciphertext C. The complete encryption process can be represented by:

$$C = P \oplus RC4(IV, k)$$

The message is then ready to be transmitted from the sender to the receiver. We will rep-

resent this transmission symbolically as:

$$A \rightarrow B : IV, (P \oplus RC4(IV, k)),$$

$$\text{where} P = \langle m, C(m) \rangle$$

Notice that the initialization vector IV is sent in the clear, meaning that anyone can read the value of the initialization vector. This should not matter because an attacker would need boh the IV and the secret key k to recover the keystream used to decrypt the message, but we will show later in this paper that this is enough to break WEP.

Decryption works by reversing the encryption process described above. The recipient first regenerates the keystream used to encode the message with the secret key k and the IV sent along with the encrypted message. He can XOR the ciphertext with the keystream to recover the plaintext P:

$$\begin{aligned} P &= C \oplus RC4(IV, k) \\ &= (P \oplus RC4(IV, k)) \oplus RC4(IV, k) \\ &= P \end{aligned}$$

The recipient can then verify the integrity of the message by splitting P into $\langle m, C(m) \rangle$ and recomputing the checksum of m. If the computed checksum matches the sent checksum, then the message has not been tampered with.

3.1.2 WPA/WPA2 Four-Way Handshake

The Four-Way Handshake protocol is used in both WPA and WPA2 to authenticate a station to an access point, to compute a pairwise transient key (PTK) to be used in future communication between these parties, and to distribute a group transient key (GTK) to be used by the station to communicate with other devices connected to the access point [12]. The PTK actually consists of five different keys, but for the purposes of modeling the protocol it is sufficient to think of it as a single key. We can assume that both the station and the access point begin by knowing the Pairwise Master Key (PMK), which will be used to compute the PTK. The PMK is either computed by both parties in enterprise mode, or known ahead of time in pre-shared key mode (used for personal networks).

TO-DO:Finsih this part!!!!

3.2 ProVerif

To formally prove the correctness of WEP and the Four-Way Handshake, we use the ProVerif Cryptographic Verifier created by Bruno Blanchet. ProVerif uses prolog rules to encode the protocol and abstracts away fresh values and the number of steps in the protocol [5]. Instead, Proverif treats each fresh value as a function of other messages in the protocol, meaning that different values are used for each pair of protocol participants. Each step in the protocol can be completed any number of times, and past steps can be re-executed arbitrarily. This permits ProVerif to execute an unlimited number of runs of a protocol. A protocol is proved to be secure by querying ProVerif whether an attacker knows a key or the content of some encrypted message. Since we assume that messages are placed on a channel c, and that the attacker has access to all messages on c, security is proved if the attacker cannot learn the security property despite access to all the messages. This verifier has been used to successfully prove the insecurity of protocols such as the Diffe-Hellman key exchange protocol, Initial Key Agreement, and the Needham-Schroeder symmetric-key protocol

[11, 4].

3.2.1 Horn Clauses

ProVerif can take protocols in either horn clauses or pi-calculus. In our protocol implementations, we chose to use typed horn clauses. An untyped horn clause is a disjunction of literals with at most one positive literal (ex. $\neg p \vee \neg q \vee t$) [6]. They can be written as implications (ex. $(p \wedge q) \rightarrow t$), as they are in Prolog, on which ProVerif is based. Typed horn clauses merely allow the user to add a type system to the program for clarity, but the underlying logic is the same as that of untyped horn clauses.

3.2.2 Attacker Model

In ProVerif, it is not necessary to explicitly model the attacker. Rather, the user constructs a list of clauses detailing what anyone, including the attacker, can do with messages that are put on the channel c. Such abilities include separating a message $c(a, b)$ into its separate parts and placing those parts on the channel $((c(a), c(b))$, decrypting messages if the encryption key of the message is known, encrypting messages using a known key, XOR-ing messages, and computing checksums of messages. We use the Dolev-Yao attacker model, in which the attacker can intercept, overhear, and create new messages, since this corresponds closely to the capabilities a WEP or Four-Way Handshake protocol attacker would have.

3.3 Modeling Security Properties

4 Analysis of Protocol Models

4.1 WEP

4.2 WPA/WPA2 Four-Way Handshake

5 Conclusions and Future Work

A ProVerif Implementation of WEP

B ProVerif Implementation of Four-Way Handshake

References

[1] Ieee standard for information technology-telecommunications and information exchange between systems-local and metropolitan area networks-specific requirements-part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-1997*, pages i–445, 1997.

[2] Wi-fi protected access: Strong, standards-based, interoperable security for today's Wi-Fi networks. White paper, Wi-Fi Alliance, 2003.

[3] Ieee standard for information technology-telecommunications and information exchange between systems- local and metropolitan area networks- specific requirements part 11: Wireless lan medium access control (mac) and physical layer

(phy) specifications amendment 6: Medium access control (mac) security enhancements. *IEEE Std 802.11i-2004*, pages 1 –175, 2004.

[4] Martín Abadi. Security protocols: Principles and calculi tutorial notes.

[5] Bruno Blanchet. An Efficient Cryptographic Protocol Verifier Based on Prolog Rules. In *14th IEEE Computer Security Foundations Workshop (CSFW-14)*, pages 82–96. IEEE Computer Society, 2001.

[6] Bruno Blanchet. Using Horn clauses for analyzing security protocols. In Véronique Cortier and Steve Kremer, editors, *Formal Models and Techniques for Analyzing Security Protocols*, volume 5 of *Cryptology and Information Security Series*, pages 86–111. IOS Press, 2011.

[7] Nikita Borisov, Ian Goldberg, and David Wagner. Intercepting mobile communications: The insecurity of 802.11. In *MobiCom '01: Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pages 180–189, 2001.

[8] H.I. Bulbul, I. Batmaz, and M. Ozel. Wireless network security: Comparison of wep (wired equivalent privacy) mechanism, wpa (wi-fi protected access) and rsn (robust security network) security protocols. In *Proceedings of the 1st international conference on Forensic applications and techniques in telecommunications, information, and multimedia and workshop*. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2008.

[9] Changhua He and John C. Mitchell. Analysis of the 802.11i 4-way handshake. In *Proceeding of the Third ACM International Workshop on Wireless Security (WiSe '04*, pages 43–50, 2004.

[10] Changhua He and John C. Mitchell. Security Analysis and Improvements for IEEE 802.11i. In *Proceedings of the 12th ACM Conference on Computer and Communications Security (CCS '05)*, 2005.

[11] Pascal Lafourcade, Vanessa Terrade, and Sylvain Vigier. Comparison of cryptographic verification tools dealing with algebraic properties. In *FAST '09: Proceedings of the 6th International Conference on Formal Aspects in Security and Trust*, 2010.

[12] Jong Liu, Jun Zhang, and Jun Li. Security Verification of 802.11i 4-Way Handshake Protocol. In *ICC '08: IEEE International Conference on Communications, 2008*, pages 1642–1647, 2008.