20-12-2004

Observations on the Message Integrity Code in IEEE802.11Wireless LANs

Jianyong Huang
University of Wollongong, jyh33@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Jennifer Seberry
University of Wollongong, jennie@uow.edu.au

Recommended Citation

Huang, Jianyong; Susilo, Willy; and Seberry, Jennifer: Observations on the Message Integrity Code in IEEE802.11Wireless LANs 2004.
https://ro.uow.edu.au/infopapers/511

Observations on the Message Integrity Code in IEEE802.11Wireless LANs

Abstract

This paper surveys the security of the message integrity code used in the IEEE802.11 Wireless LANs. To address the security flaws of Wired Equivalent Privacy (WEP), the IEEE802.11i draft defines two data confidentiality and integrity protocols, Temporal Key Integrity Protocol (TKIP) and Counter-Mode-CBC-MAC Protocol (CCMP). TKIP is based on RC4, andCCMP is based on the AES cipher. TKIP includes a keyed hash functions, called Michael, as the messag integrity code. The aim of this paper is to summarise the recent research results of Michael and analyse the practicability of two attacks against Michael.

Disciplines

Physical Sciences and Mathematics

Publication Details

This conference paper was originally published as Huang, J, Susilo, W and Seberry, J, Observations on the Message Integrity Code in IEEE802.11Wireless LANs, in Proceedings of the 3rd Workshop on the Internet, Telecommunications and Signal Processing (WITSP'04), Adelaide, Australia, 20-22 December 2004, CD-ROM, 328-332.

Observations on the Message Integrity Code in IEEE802.11 Wireless LANs

Jianyong Huang, Willy Susilo and Jennifer Seberry
School of Information Technology and Computer Science
University of Wollongong
Australia
Email: {jyh33, wsusilo, jennie}@uow.edu.au

Abstract

This paper surveys the security of the message integrity code used in the IEEE802.11 Wireless LANs. To address the security flaws of Wired Equivalent Privacy (WEP), the IEEE802.11i draft defines two data confidentiality and integrity protocols, Temporal Key Integrity Protocol (TKIP) and Counter-Mode-CBC-MAC Protocol (CCMP). TKIP is based on RC4, and CCMP is based on the AES cipher. TKIP includes a keyed hash functions, called Michael, as the message integrity code. The aim of this paper is to summarise the recent research results of Michael and analyse the practicability of two attacks against Michael.

1 Introduction

The IEEE802.11 standard [3] defines a WEP protocol to protect authorised users of a wireless LAN from eavesdropping and other attacks. WEP uses RC4 [11] as the encryption and decryption algorithm, and employs the CRC-32 (Cyclic Redundancy Check) as the Message Integrity Code (MIC). Recent research showed that WEP fails to provide confidentiality, access control and data integrity [5, 7, 12]. The authentication protocol in WEP was attacked by [4]. There are open source tools available on the Internet to break the WEP system, for example, Airsnort [8] and WEPCrack [13].

The IEEE802.11i draft [1] specifies two protocols, TKIP and CCMP, to address the weaknesses of WEP. The WEP protocol is implemented in hardware in most existing IEEE 802.11 devices, and these devices are still used by enterprises and home users at present. TKIP defines a set of algorithms, which wraps WEP, to allow the current WEP implementation to remain unchanged while addressing the security flaws of WEP. Two keyed hash functions, Michael and Temporal Key Hash (TKH), are employed by TKIP to enhance the WEP encryption (illustrated by Figure 1). Designed to prolong the usage of current IEEE802.11 devices, TKIP is subject to the constraints of the legacy hardware implementation. TKIP is considered only as a short-term solution. The long-term solution is CCMP, and the encryption algorithm in CCMP is the Advanced Encryption Standard (AES) [2]. The implementation of CCMP will require new hardware. In this paper, we only focus on the security issues of the message integrity code in TKIP, and refer the reader to [9] for TKH and to [1] for CCMP to find out more details.

Our Contributions: We summarise the recent research results of Michael, and provide remarks and comments on the latest attacks.

Notations

Symbol	Description
TA	Transmitter address
TTAK	TKIP mixed transmitter address and key
TK	Temporal key
IV	Initialisation vector
DA	Destination address
SA	Source address
MAC	Medium access control
MSDU	MAC service data unit
MPDU	MAC protocol data unit
$<<<$	Left rotation
$>>>$	Right rotation
$\oplus$	Exclusive-or
$\boxplus$	addition modulo 2^{32}
$\|\|$	Concatenation

Organisation: The rest of this paper is organised as follows. Section 2 describes the Michael message integrity code. Section 3 documents recent research results of Michael, and contains four subsections: Section 3.1 shows that Michael is not one-way, and Section 3.2 provides a related-message attack. Section 3.3 reveals that Michael is not collision-free, and Section 3.4 demonstrates a packet forgery attack against Michael. Section 4 provides comments on the recent research of Michael. Section 5 concludes this paper.

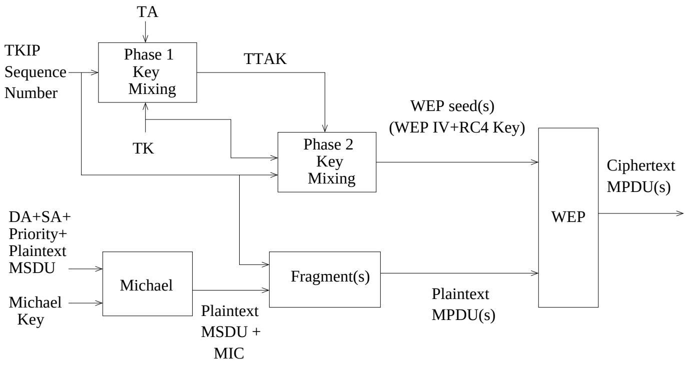

Figure 1. *TKIP Encryption Diagram*

2 The Michael Message Integrity Code

WEP uses CRC-32 as its message integrity code, and the CRC-32 mechanism fails to provide data integrity [5]. TKIP employs **Michael** [6] as its message integrity code.

The inputs to Michael include a 64-bit Michael key and an arbitrarily long message, and the output is a 64-bit Michael value. The message is padded at the end with a single byte with the hexadecimal value *0x5a* and then followed by between 4 and 7 zero bytes. The padding method is to make the total length of the message plus the padding equal to a multiple of 4. The last block of the padded message is 0 while the second last block of the padded message is not 0. The details of Michael are described in Algorithm 2.1 and 2.2.

Algorithm 2.1: MICHAEL$((k_0, k_1), (m_0, ..., m_{n-1}))$

```
Input : Key$(k_0, k_1)$
Input : Padded message $(m_0, ..., m_{n-1})$
Output : MIC value $(L, R)$
$(L, R) \leftarrow (k_0, k_1)$
for $i \leftarrow 0$ to $n-1$
    do $\begin{cases} L \leftarrow L \oplus m_i \\ (L, R) \leftarrow B(L, R)(Algorithm 2.2) \end{cases}$
return $(L, R)$
```

Algorithm 2.2: B(L, R)

```
Input : $(L, R)$
Output : $(L, R)$
$R \leftarrow R \oplus (L <<< 17)$
$L \leftarrow (L + R) \bmod 2^{32}$
$R \leftarrow R \oplus XSWAP(L)$
$L \leftarrow (L + R) \bmod 2^{32}$
$R \leftarrow R \oplus (L <<< 3)$
$L \leftarrow (L + R) \bmod 2^{32}$
$R \leftarrow R \oplus (L >>> 2)$
$L \leftarrow (L + R) \bmod 2^{32}$
return $(L, R)$
```

The Michael value is computed iteratively by beginning with the Michael key value and applying the block function B (given in Algorithm 2.2) for every message block. The block function is an unkeyed 4-round Feistel-type construction. Michael uses several operations, including exclusive-or, left rotation, right rotation, swapping ($XSWAP$), addition modulo 2^{32}. Function $XSWAP$ swaps the position of the two least significant bytes and the position of the two most significant bytes in a word, i.e., $XSWAP(ABCD) = BADC$ where A, B, C, D are bytes.

3 Weaknesses of Michael

This section summarises recent results from two different research groups.

3.1 Michael is Not One-Way

Wool [14] disclosed one serious weakness of Michael: it is *invertible*. Given a known message M and its corresponding Michael value MIC = Michael(K, M), the secret Michael key K can be recovered, and the details are shown in Algorithm 3.1 and 3.2.

Algorithm 3.1: INVMICHAEL$((v_0, v_1), (m_0, ..., m_{n-1}))$

```
Input : Michael value $(v_0, v_1)$
Input : Padded message $(m_0, ..., m_{n-1})$
Output : Key $(k_0, k_1)$
$(L, R) \leftarrow (v_0, v_1)$
for $i \leftarrow n-1$ downto $0$
    do $\begin{cases} (L, R) \leftarrow B^{-1}(L, R)(Algorithm 3.2) \\ L \leftarrow L \oplus m_i \end{cases}$
return $(L, R)$
```

Algorithm 3.2: $B^{-1}(L, R)$

```
**Input** : $(L, R)$
**Output** : $(L, R)$
$L \leftarrow (L - R) \bmod 2^{32}$
$R \leftarrow R \oplus (L >>> 2)$
$L \leftarrow (L - R) \bmod 2^{32}$
$R \leftarrow R \oplus (L <<< 3)$
$L \leftarrow (L - R) \bmod 2^{32}$
$R \leftarrow R \oplus XSWAP(L)$
$L \leftarrow (L - R) \bmod 2^{32}$
$R \leftarrow R \oplus (L <<< 17)$
**return** $(L, R)$
```

3.2 A Related-Message Attack

Wool proposed a related-message attack on Michael [14].

Proposition 3.1 *Suppose an attacker is able to intercept two TKIP message frames (ciphertexts), M_1 and M_2, and the following three conditions hold:*

1. *M_1 and M_2 are encrypted by the same encryption key and the same IV.*
2. *$length(M_1) \geq length(M_2) + 8$*
3. *The plaintext P_1 of the longer message M_1 is known to the attacker.*

Then the attacker can recover the Michael key K of the shorter message M_2.

Wool's attack is illustrated in Figure 2. The attacker can compute K by following three steps listed below.

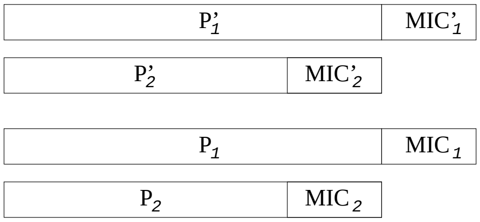

Figure 2. *Wool's Attack*

1. $RC4KEY[i]$ = $M_1[i] \oplus P_1[i]$, $i = 1$, ..., $length(M_1)$-8
2. P_2 || MIC_2 = $RC4KEY[i] \oplus M_2[i]$, $i = 1$, ..., $length(M_2)$
3. K = InvMichael(MIC_2, P_2)

3.3 Michael is Not Collision-Free

Huang, Seberry, Susilo and Bunder (HSSB) [10] proved that the collision status of Michael only depends on the output of its third last round and the second last block message in Theorem 3.1.

Theorem 3.1 *Given two pairs of keys and messages, (Key_1, M_1) and (Key_2, M_2), Michael(Key_1, M_1) = Michael(Key_2, M_2) if and only if the following two conditions hold:*

1. $R_4^{x-3} = R_4'^{y-3}$
2. $L_5^{x-3} \oplus L_5'^{y-3} = m_{x-2} \oplus m'_{y-2}$

where M_1 has x 32-bit blocks, M_2 has y 32-bit blocks, M_1 is distinct from M_2 and both x and y are $\geq$ 3.

(Note: R_4^{x-3} and L_5^{x-3} are the right and left half of the third last round output of Michael(Key_1, M_1) respectively. $R_4'^{y-3}$ and $L_5'^{y-3}$ are the right and left half of the third last round output of Michael(Key_1, M_1) respectively. m_{x-2} is the second last block of M_1, and m'_{y-2} is the second last block of M_2.)

Furthermore, HSSB [10] showed that Michael is not collision-free in Theorem 3.2 and Theorem 3.3.

Theorem 3.2 *Given an arbitrarily length message M and a specific key K, a 96-bit block message M' and a key K' can be computed such that Michael(K, M) = Michael(K', M'), where M has n 32-bit blocks, n is any integer ≥ 3 and M' is distinct from M.*

Theorem 3.3 *Michael is not collision-free (deduced from Theorem 3.2).*

We note that Theorem 3.1 is a necessary and sufficent condition of finding collisions of Michael.

3.4 A Packet Forgery Attack Against Michael

HSSB [10] provided a simple method to find fixed points of Michael, and demonstrated that it is statistically certain that there exist about 2^{32}*0.9 fixed points of Michael. A fixed point of Michael is a triple (L_i, R_i, m_i) such that Michael((L_i, R_i), m_i) = (L_{i+1}, R_{i+1}) = (L_i, R_i), where L_i, R_i and m_i are three input parameters to the block function, and L_{i+1} and R_{i+1} are two output parameters of the block function. The feature of fixed points is illustrated in Figure 3. For example, ((4987c6d0, 1), 7161872) (hexadecimal numbers) is a fixed point of Michael as Michael((4987c6d0, 1), 7161872) = (4987c6d0, 1).

Based on the property of fixed points, a packet forgery attack was proposed against Michael in Theorem 3.4 [10].

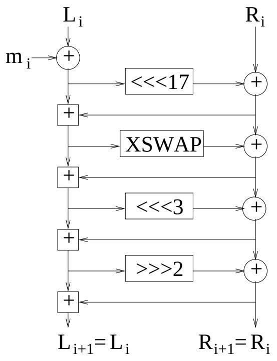

Figure 3. *Fixed Points of Michael*

Theorem 3.4 *Given a message M_1 and an arbitrary key K, an attacker can always construct a message M_2 such that Michael(K, M_1) = Michael(K, M_2) if the following condition holds:*

1. *The output of the block function of Michael(K, M_1) in any round is equal to any of the fixed points*

where M_2 is distinct from M_1.

Figure 4 describes the packet forgery attack. After padding, the original message M_1 has n blocks, written as M_1 = (m_0, m_1, ..., m_{n-1}). Suppose the three inputs to the (i+1)-th round, namely ((L_i, R_i), m_i) in Figure 4, is a fixed point. Then a multiple of 4 blocks of m_i can be inserted to (i+2)-th round without changing the Michael value. In other words, M_2 is constructed as (m_0, m_1, ..., m_i, <m_i, m_i, ..., m_i>, m_{i+1}, ..., m_{n-1}), where the number of inserted blocks of m_i is a multiple of 4. The reason why the inserted blocks of m_i is a multiple of 4 is due to the padding method.

4 Remarks on Recent Attacks on Michael

Michael is a new cryptographic design. Actually, Wool's related-message attack [14] (described in Section 3.2) is known by the designer of Michael, and this attack is mentioned implicitly in the last paragraph of Page 14 in the design document [6], written as " *A known-plaintext attack will reveal the key stream for that IV, and if the second packet encrypted with the same IV is shorter than the first one, the MIC value is revealed, which can then be used to derive the authentication key* ". The related-message attack is based on Condition 1, 2 and 3 in Proposition 3.1. The practicability of the fulfilment of condition 1 is related to the Temporal Key Hash [9] . The aim of Temporal Key Hash is to provide per-packet WEP key, therefore it is not practical for the attacker to achieve Condition 1. So, the related-message attack is still in theory.

The packet forgery attack is also not practical as the attacker needs to fulfil Condition 1 in Theorem 3.4 and the TKIP frames are encrypted by RC4. According to [10], there exist about 2^{32}*0.9 fixed points for Michael. The output of the block function in any round is 64-bit. Suppose a message M has n 32-bit blocks, then the probability of the fulfilment of Condition 1 in Theorem 3.4 is $\frac{n*2^{32}*0.9}{2^{64}}$.

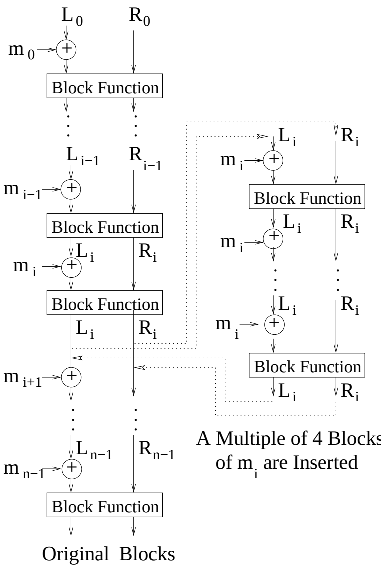

Figure 4. *A Packet Forgery Attack*

5 Conclusions

Michael is the message integrity code used in TKIP in the latest IEEE802.11i draft. Recent research revealed some weaknesses in this keyed hash function. Michael is neither one-way nor collision-free. Although the related-message attack and the packet forgery attack agaist Michael are still not practical, we still would like to point out that designing a keyed hash function is not an easy task.

References

[1] Draft Amendment to Standard For Telecommunications and Information Exchange Between Systems - LAN/MAN Specification Requirements - Part 11: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements. Document Number IEEE Std 802.11i/D7.0. October 2003.

[2] Advanced Encryption Standard. National Institute of Standards and Technology, NIST FIPS PUB 197, U.S. Department of Commerce. November 2001.

[3] ANSI/IEEE Std 802.11, 1999 Edition. 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.

[4] W. Arbaugh, N. Shankar, and Y.C. Wan. Your 802.11 Wireless Network has No Clothes. In *Proceedings of IEEE International Conference on Wireless LANs and Home Networks*, pages 131–144, 2001.

[5] N. Borisov, I. Goldberg, and D. Wagner. Intercepting Mobile Communications: The Insecurity of 802.11. In *Proceedings of the 7th Annual Internaltional Conference on Mobile Computing and Networking (MOBICOM'01)*, 2001.

[6] N. Ferguson. Michael: an improved MIC for 802.11 WEP. *IEEE doc. 802.11-02/020r0*, 17 January 2002. http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-020.zip.

[7] S. Fluhrer, I. Mantin, and A. Shamir. Weaknesses in the Key Scheduling Algorithm of RC4. In *Proceedings of the 8th Annual International Workshop on Selected Areas in Cryptography*, 2001.

[8] The Airsnort Homepage. 10 August 2004. http://airsnort.shmoo.com/.

[9] R. Housley, D. Whiting, and N. Ferguson. Alternate Temporal Key Hash. *IEEE doc. 802.11-02/282r2*, 23 April 2002. http://grouper.ieee.org/groups/802/11/Documents/DocumentHolder/2-282.zip.

[10] J. Huang, J. Seberry, W. Susilo, and M. Bunder. On the Security of the IEEE 802.11i Message Integrity Code: Michael. *Manuscript*, 2004.

[11] R. Rivest. The RC4 Encryption Algorithm, RSA Data Security Inc., (Proprietary). March 1992.

[12] A. Stubblefield, J. Ioannidis, and A. Rubin. Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. In *Proceedings of the 2002 Network and Distributed Systems Security Symposium*, pages 17–22, 2002.

[13] WEPCrack. 10 August 2004. http://wepcrack.sourceforge.net/.

[14] A. Wool. A Note on the Fragility of the “Michael” Message Integrity Code. *Accepted for publication in IEEE Trans. Wireless Communications*, 2004.