Vulnerabilities of IEEE 802.11i Wireless LAN CCMP Protocol

M. Junaid , Dr Muid Mufti, M.Umar Ilyas

***Abstract*—**IEEE has recently incorporated CCMP protocol to provide robust security to IEEE 802.11 wireless LANs. It is found that CCMP has been designed with a weak nonce construction and transmission mechanism, which leads to the exposure of initial counter value. This weak construction of nonce renders the protocol vulnerable to attacks by intruders. This paper presents how the initial counter can be pre-computed by the intruder. This vulnerability of counter block value leads to pre-computation attack on the counter mode encryption of CCMP. The failure of the counter mode will result in the collapse of the whole security mechanism of 802.11 WLAN.

***Keywords*—** Information Security, Cryptography, IEEE 802.11i, Computer security, Wireless LAN.

I. INTRODUCTION

THE IEEE 802.11i [1] incorporates authentication, data integrity and data encryption mechanisms to address security concerns for legacy and new wireless LANs in infrastructure and ad-hoc (peer-to-peer) based 802.11 networks. 802.11i specifies device authentication through IEEE 802.1X [2] and data security through the Wire Equivalent Privacy (WEP), *Temporal Key Integrity Protocol (TKIP) or Counter Mode with CBC-MAC Protocol (CCMP)*.

WEP and TKIP target legacy 802.11 equipment. Various academic and commercial studies have shown that WEP based WLAN Security can be breached by intruders. Vulnerabilities of WEP include weak encryption (short keys), static encryption keys and lack of key distribution mechanism. TKIP [1] provides counter-measures to possible attacks on WEP and reduces the rate at which a hacker can make message forgery attempts, down to two packets every 60 seconds; after which new encryption keys are generated. The counter-measures reduce the probability of successful forgery and amount of information an attacker can learn about a key.

By contrast, CCMP requires new 802.11 hardware with greater processing power. CCMP is based on the *Advanced Encryption Standard (AES)* [3], a FIPS-197 certified algorithm approved by NIST. AES (128 bits key length) operates in a counter mode (AES-128-CM) within 802.11i with CBC-MAC (CCM) [4] [5]. Counter mode is used for data confidentiality and Cipher Block Chaining -Message Authentication Code (CBC-MAC) is used for data integrity and authentication.

Counter mode operates by encrypting the initial counter and the resulting output is XORed with the plaintext to produce the cipher text [4]. The initial counter is constructed from the flags field, length of the payload and the nonce. The nonce is constructed from the packet number (PN), MAC layer A2 Address field (A2) and MAC layer priority field.

In this paper, it is described that the initial counter value used in the CCMP of 802.11 Wireless LANs can be predicted. Since the nonce value can be pre-computed, the only thing required to predict the counter value is length of payload. The length of the payload can be obtained through a priori information e.g. 802.11 maximum Payload length is 2296 bytes (2312 bytes total payload length – 8 bytes MIC – 8 bytes CCMP Header) and if the data is more than maximum length of Payload then MSDU is fragmented into MPDUs. If larger data than the maximum payload length is to be transmitted, then the first fragment's (MPDU) Payload length will be 2296 bytes. In [6], it is iterated that if initial counter value is predictable, then attacks using pre-computation can be used to lower the security level of AES-128-CM below the recommended strength for block ciphers. In this paper, we have shown that initial counter value of 802.11i CCMP is considerably predictable and is vulnerable to time memory trade off (TMTO) pre-computation attack.

The rest of the paper is organized as follows. Section II describes the threat model. Section III explains CCMP security mechanism, Section IV and V shows how the nonce and counter block value can be pre-computed by adversary. Section VI describes the TMTO precomputation attack on CCMP and Section VII concludes the paper.

II. THREAT MODEL

Wireless networks are prone to different kind of security threats. Ubiquitous RF signals provide conducive environment for malicious and well planned information warfare, where attackers can use the advance technology to mount attacks

Dr muid mufti is an associate professor at the University of Engineering and Technology ,Taxila ,Pakistan.(email: muid@uetaxila.edu.pk) . M.Junaid and Umar Ilyas are at National University of Sciences and Technology, Pakistan.(email: junaid-mcs@nust.edu.pk; muillyas@wol.net.pk)

with the ease to sniff the MPDUs traversing the air. Generally the threats can be classified into the following:

- *Leakage of Information:* Information dissemination to anyone who is not authorized to access it.
- *Alteration of Information:* Un-authorized or malicious alteration of data while in transit between autonomous systems, injection of spurious information using spoofing, replay of packets etc.
- *Repudiation:* A party involved in the communication denies its involvement.
- *Impersonation:* An adversary pretends to be an authorized entity.
- *Service Stealing:* Unauthorized use of network or domain services without degrading the services to other users.
- *Denial of Service:* Illegitimate access and intentional degradation or blocking of internetwork communication links or services.

III. Introduction to CCMP Security Mechanism

CCMP requires a fresh temporal key for every session. CCMP also requires a unique nonce value for each frame protected by a given temporal key, and CCMP uses a 48-bit packet number (PN) for this purpose [1]. The CCMP header is concatenated with the MAC header, the encrypted payload, the
encrypted MIC and the FCS field. These fields form the MPDU as illustrated in Fig. 1 [1].

The CCMP encapsulation process is depicted in Fig. 2 [1]. CCMP encrypts the payload of a plaintext MPDU and encapsulates the resulting cipher text using the following steps [1]:

a) Increment the PN, to obtain a fresh PN for each MPDU, so that the PN never repeats for the same temporal key. Note that retransmitted MPDUs are not modified on retransmission.

b) Use the fields in the MPDU header to construct the additional authentication data (AAD) for CCM. The CCM algorithm provides integrity protection for the fields included in the AAD. MPDU header fields that may change when retransmitted are muted by being masked to 0 when calculating the AAD.

c) Construct the CCM Nonce block from the PN, A2, and the Priority field of the MPDU where A2 is MPDU Address 2. The Priority field has a reserved value set to 0.

d) Place the new PN and the key identifier into the 8-octet CCMP header.

e) Use the temporal key, AAD, nonce, and MPDU data to form the cipher text and MIC. This step is known as CCM originator processing.

f) Form the encrypted MPDU by combining the original MPDU header, the CCMP header, the encrypted data and MIC.

CCMP decrypts the payload of a cipher text MPDU and decapsulates a plaintext MPDU as shown in Fig. 3 [1], using the following steps:

a) The encrypted MPDU is parsed to construct the AAD and nonce values.

b) The AAD is formed from the MPDU header of the encrypted MPDU.

c) The nonce value is constructed from the A2, PN, and Priority Octet fields (reserved and set to 0).

d) The MIC is extracted for use in the CCM integrity checking.

e) The CCM recipient processing uses the temporal key, AAD, nonce, MIC, and MPDU cipher text data to recover the MPDU plaintext data as well as to check the integrity of the AAD and MPDU plaintext data.

f) The received MPDU header and the MPDU plaintext data from the CCM recipient processing may be concatenated to form a plaintext MPDU.

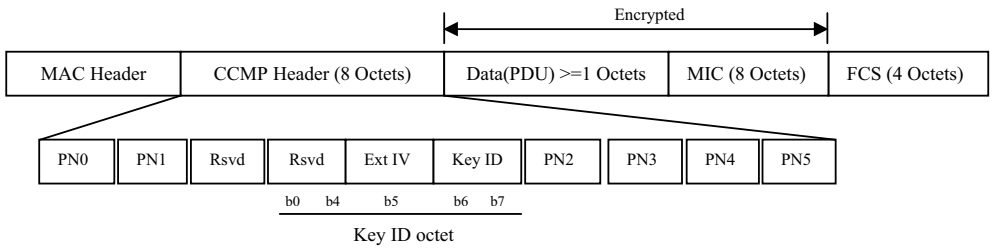

Fig. 1 CCMP MPDU

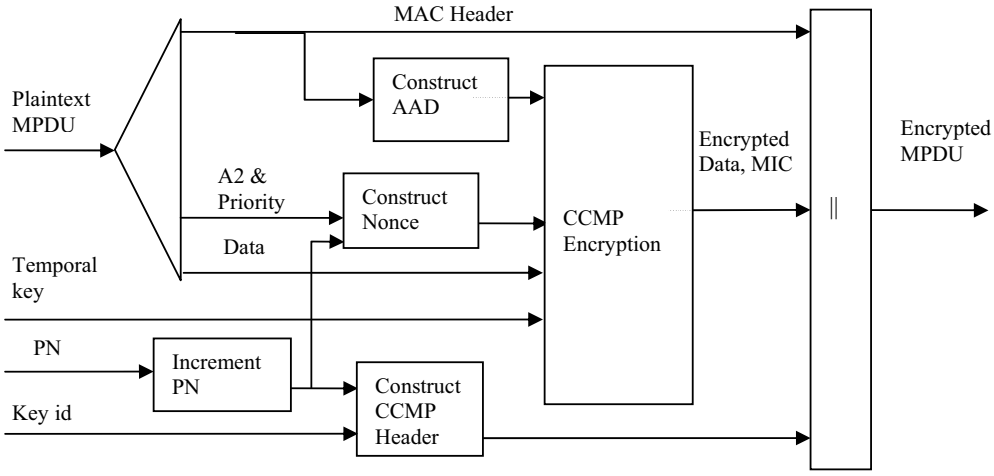

Fig. 2 CCMP encapsulation block diagram

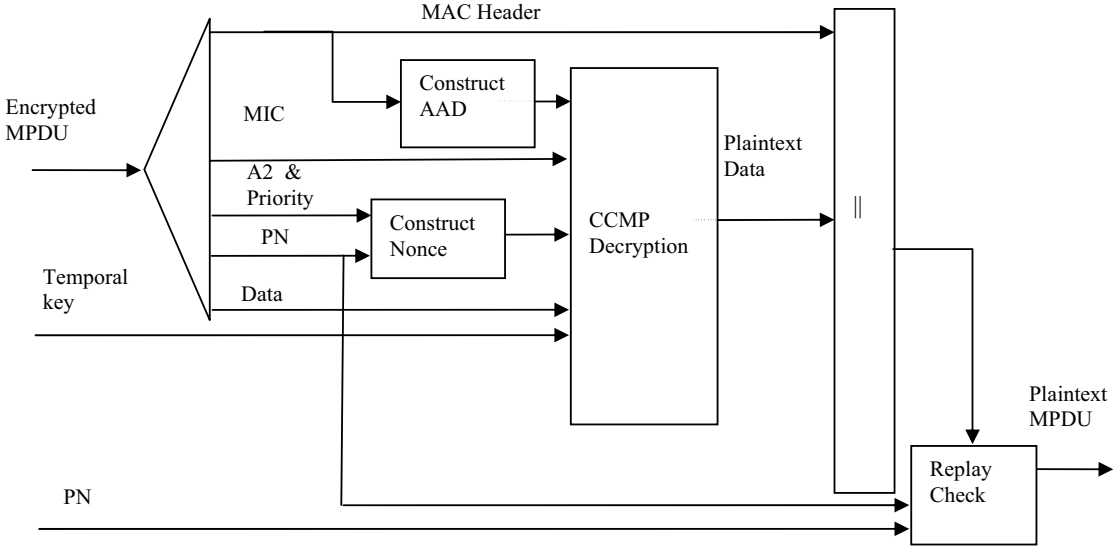

Fig. 3 CCMP decapsulation Block Diagram

IV. Reconstruction of Nonce

The nonce block constitutes three fields. The first field is A2 address of MAC header (A2), second is priority field which is set to '0' by default and the third field is PN field.

Priority Field || Address (A2) || Packet Number (PN) = Nonce

The construction of nonce has been devised in such a manner that its reconstruction by an adversary is possible. The first 8 bits of nonce is the priority field which is presently kept as '0', this field will be used in future for 802.11 frame prioritization. The A2 field, which is 48 bits, is extracted from the MAC header field and is concatenated with the priority field. The only dynamic field, which is monotonically increasing per MPDU, is the PN field. [1] specifies in its subclause 8.3.3.4.3 that PN should be initialized to Value '1' when corresponding temporal key is initialized or refreshed.

Keeping in view, the nature of wireless medium, anyone in possession of compatible equipment, could easily sniff the MPDUs. Since the MAC header and CCMP header are transmitted in plaintext as shown in Fig. 1. and their field loation is also fixed within the MPDU, therefore, anyone with

the intention of verifying the pre-computed nonce could easily be able to extract the priority and A2 field from the MAC header. Furthermore, the PN field in CCMP plaintext header is monotonically increasing, so its initial value as well as future value can be calculated after little deliberation. Therefore nonce can be pre-computed and verified successfully as illustrated in Fig. 4.

V. Reconstruction of Initial Counter

In 802.11i, the payload and message integrity code (MIC) is encrypted using counter mode encryption. The encryption process occurs by computing keystream blocks (S_i) as:

$S_i = e_K(Ctr_i)$

where,

$Ctr_i = (Ctr1 + i - 1) \bmod 2^n \ (1 \le i \le b)$

Ctr_i = counter block value of the i^{th} iteration

e_k = Encryption with 128bit AES Key(k)

n = number of bits in a block.

b = number of key stream blocks to be exclusive-OR with Plaintext block.

The ciphertext 'C' is computed as follows:

$$C = P \oplus (S_1 || \ldots\ldots || S_b)$$

On the receiving side, the plaintext 'P' is computed as follows:

$$P = C \oplus (S_1 || \ldots\ldots || S_b)$$

The counter block value (Ctr) consists of three data values:

- Flag field
- Nonce
- Length of length of Payload

The counter blocks (Ctr_i) having counter index 'i' are formatted as shown in Table 1. Flags field is a one octet field and consists of 2 reserved bits for future use, next 3 bits having value 0 each and the last three bits are the encoding of octet length of binary representation of octet length of payload (q) in bits and computed as $[q-1]_3$.

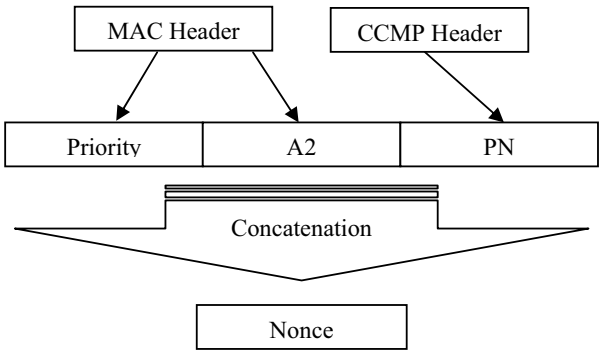

Fig. 4 Nonce reconstruction scheme

TABLE I
FORMATTING OF COUNTER BLOCKS

Octet number	0	1…..15-q	16-q……..15
Contents	Flags	Nonce	$[i]_{8q}$

The nonce field is the same field that has been discussed in Section IV. The bit length of each input string, i.e., nonce (N) and Payload (P), is a multiple of 8 bits [5]. The octet lengths of these strings are denoted as n and p respectively. Thus, n and p are integers. The octet length of P is represented within the first block of the formatted data as an octet string denoted Q. The octet length of Q, denoted q, is a parameter of the formatting function. Thus Q is equivalent to $[p]_{8q}$, the binary representation of p in q octets.

It is observed that Flag field is a known constant value. The reconstruction of nonce has already been shown in Section IV. Now, to find out the counter block value, length of the payload is required. In case of IEEE 802.11 MPDUs, the max payload length is defined to be 2312 bytes (2296 Data + 08 MIC + 08 CCMP Header). 802.11 also specifies that if MSDU has larger data than 2296 bytes, then MSDU is fragmented into MPDUs. Since the payload of the MPDU also contains TCP Header, IP Header and SNAP Header, it is observed that fragmentation is required in almost all MSDUs. In case of fragmentation, the first packet will be of maximum size. Hence, the length of payload length can be pre-computed. This will lead to the prediction of the initial counter value and subsequently all counter values can also be computed. The payload computation is given as:

p = 2296 octets

if q = 2, then

$Q = [p]_{8q} = [2296]_{8\times 2}$

Q = 00001000 11111000

Where,

p = octet length of Payload.

q = The octet length of the binary representation of octet length of payload.

Q = A bit string representation of the octet length of P.

The extraction of fields to pre-compute the initial counter value is illustrated in Fig. 5. Any unauthorized user may calculate the counter value irrespective of undergoing through the successful authentication process.

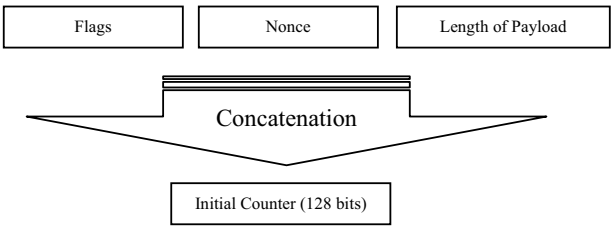

Fig. 5Reconstruction of Initial counter

VI. TMTO Precomputation Attack

In the Sections IV and V, we have shown that an unauthorized person can compute the A2, priority field, PN, and length of length of payload. By concatenating these values we get the initial counter value. This counter value provides the basis for TMTO precomputation attack.

The TMTO attack [7] is a shortcut over exhaustive key search that trade a storage requirement against decreased computational effort. It can be used against any cipher, even ones that are not statistically defective. In these attacks, the adversary computes a large database prior to attacking any secret keys, then using this database during the attack stage, it potentially attacks many different secret keys. An important property of this method is that it does not require any knowledge of the plaintext during the pre-computation stage. In fact, this attack can be used even when there is uncertainty in the plaintext during the attack stage, using techniques from error-correcting codes [8]. The usefulness of the TMTO is demonstrated by the fact that its use was crucial in the subversion of the A5/1 cipher [9]. Pre-computation attacks are useful for attacking a system in which many keys will be used. Cryptographic systems typically use many traffic-encryption keys.

In many cases, a system should be considered subverted if even a small fraction of the traffic-encryption keys are found by an adversary [6]. These cases provide fruitful ground for pre-computation attacks.

Success of TMTO depends heavily on the available amount of data, so devising an appropriate scenario of attack is also crucial. In IEEE 802.11i CCMP protocol, if we focus on the 2296 bytes payload only, then it is observed that the counter of the counter mode encryption increments monotonically during the same session. And it is also noted in 802.11 networks that there is no upper bound on the number of MPDUs per session. Therefore the amount of available data is sufficient to launch TMTO attack.

In [10], counter mode is stated as vulnerable to TMTO precomputation attack if counter update is predictable. It is shown in this paper that both the initial counter and its update are predictable, therefore TMTO attack is possible. TMTO has an effective key size of 2n/3 [7]. Where 'n' is the cipher key size. The AES counter mode key size is 128 bits in 802.11i and after TMTO attack the effective key size will be:

$$\text{Effective key size} = 2n/3 \text{ bits}$$
$$\text{where, } n = 128 \text{ bits}$$
$$\text{Effective key size} = (2\times 128)/3 \text{ bits}$$
$$\text{Effective key size} \approx 85 \text{ bits}$$

The 1996 ad-hoc report on minimal key lengths [11] recommended 75 bits key length for symmetric ciphers to provide adequate security at that time. [11] also recommends to add 14 bits to keep it secure for next 20 years atleast. Applying Moore's laws [12], if we add key bits for 8 years (1996 to 2004) and 5 more years for the validity of [1], then the recommended current strength for the cipher is 97 bits. From TMTO perspective, we deduced that effective key size of IEEE 802.11i CCMP protocol AES counter mode (TMTO scenario) is 85 bits, whereas it should be atleast 97 bits to thwart the TMTO precomputation attack. This exposes the vulnerability of IEEE 802.11 wireless LAN security mechanism to TMTO attack.

Furthermore, [6] recommends atleast one of the following points for effective defense against TMTO precomputation attack:

- There must be 64 bits unpredictable value to the initial counter, which is considered as part of the AES CM key, or
- Use a predictable but uniformly distributed component in the initial counter, or
- The key length should be larger than 128 bits.

We have observed that none of these recommendations has been incorporated in the IEEE 802.11i standard, resulting in exposure to TMTO precomputation attack. Depending only on the strength of underlying algorithm (AES) and ignoring modes of operation and associated protocols may create weak links in the security mechanism. It is recommended that these weak links may be strengthened before any exploitation by the adversaries that may result in the collapse of the whole system as seen in the case of WEP.

VII. Conclusions

IEEE 802.11i has been well analyzed and recently CCMP protocol has been incorporated providing encryption, integrity and authentication. The counter mode has been used with AES to provide the confidentiality services. The mechanism, devised, is using the PN, A2, priority field and length of payload length to compute the counter value. We have shown in this paper that these values can be pre-computed by an unauthorized user leading to TMTO precomputation attack. Efforts are in hand at our side to provide an enhanced security mechanism to counter the possible TMTO precomputation attack.

References

[1] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.11i-2004, Amendment to Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: "Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications: Medium Access Control (MAC) Security Enhancements", July, 2004.

[2] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.1X-2001, "IEEE Standard for Local and metropolitan area networks – Port-Based Network Access Control" June, 2001.
[3] Specification for the Advanced Encryption Standard (AES), FIPS 197, U.S. National Institute of Standards and Technology. November 26, 2001. [Online] Available: http://www.nist.gov/aes
[4] D. Whiting, R. Housley, and N. Ferguson. "Counter with CBC-MAC (CCM)". RFC 3610, September 2003.
[5] NIST Special Publication 800-38C, "Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality". May 2004. [Online] Available: http://csrc.nist.gov/publications/
[6] David A. McGrew, "Counter Mode Security: Analysis and Recommendations", Cisco Systems, November, 2002.
[7] M.E. Hellman, "A cryptanalytic time-memory trade-off", IEEE Transactions on Information Theory, July, 1980, pp. 401-406.
[8] D. A. McGrew and S. R. Fluhrer, "Attacks on Additive Encryption of Redundant Plaintext and Implications on Internet Security", The Proceedings of the Seventh Annual Workshop on Selected Areas in Cryptography (SAC 2000), Springer-Verlag, August, 2000. [Online] Available: http://www.mindspring.com/~dmcgrew/dam-srf-sac00.pdf
[9] A. Biryukov, A. Shamir, D. Wagner , "Real Time Cryptanalysis of A5/1 on a PC", Proceedings of the Fast Software Encryption Workshop 2000, Springer-Verlag, Lecture Notes in Computer Science, 2000.
[10] Jin Hong, Palash Sarkar, "Rediscovery of Time Memory Tradeoffs", 2005. [Online] Available: http://cr.yp.to/2005-590/hong.pdf
[11] M. Blaze, W. Die, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Weiner, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security", January 1996. [Online] Available: http://www.counterpane.com/keylength.html
[12] Moore's law [Online] Available: http://www.Webopedia.com/TERM/M/Moores_Law.html

