# A Wireless Security Encryption Systems based on Kerberos and Temporal Key Integrity Protocol (TKIP)

**[1]Sandeep Kumar Vishwakarma, [2]Prof. Amit Chouksey**

GGCT, Jabalpur

***Abstract:*** **This paper identifies and summarizes these security concerns and their solutions. Broadly, security concerns in the WLAN world are classified into physical and logical. The paper overviews both physical and logical WLANs security problems followed by a review of the main technologies used to overcome them. It addresses logical security attacks like man in- the-middle attack and Denial of Service attacks as well as physical security attacks like rouge APs. Wired Equivalent Privacy (WEP) was the first logical solution to secure WLANs. However, WEP suffered many problems which were partially solved by IEEE802.1x protocol. Towards perfection in securing WLANs, IEEE802.11i emerged as a new MAC layer standard which permanently fixes most of the security problems found in WEP and other temporary WLANs security solutions. This paper reviews all security solutions starting from WEP to IEEE802.11i and discusses the strength and weakness of these solutions.**

***Keywords:*** **WLAN: Wireless Local Area Networks, Wired Equivalent Privacy (WEP), access node (AP)**

## I-Introduction

There are a lot of doubts and debates in the wireless network community regarding the legality of war chalking and war driving activities. Network security administrators can test the propagation of APs by using special tools to verify to what extent the signals can reach. Accordingly they can control the propagation of APs by lowering the signal strength or by using smart type of antennas to control the direction of the signal or move the AP to a place where it is guaranteed that the signal will not travel beyond the building premises [7]. Some work has been done in the area of smart antennas in APs to direct the propagation of traffic [8]. Directing the propagation of traffic as well as managing the power of signals originating from the APs can be helpful in restricting the coverage of APs to specified regions. Sometimes public and open access to the WLAN is preferable, such public WLANs are called "hot spots" [9]. Implementing hot spots is subject to many of the mentioned security problems. It is important to understand that breaking the security of a hot spot will result in breaking the security of wired network connected to that hot spot. The control and monitoring of APs is minimal because it is installed in a public area like hotel lobbies, coffee shops, and airport lounges so preventing physical access to AP is more difficult as the site has to be monitored all the time. In this case, there is a tradeoff between giving users the mobility and the flexibility to log in to the network in public areas versus the security of the network infrastructure. The network backbone can be highly secured but a breach in the security of the network access node (i.e. AP) can always lead to a breach in the security of the backbone behind the node.

## II-Methodology

Proposed work is design of network where unique combination of Kerberos and Temporal Key Integrity Protocol (TKIP) has been used as found by study**.** TKIP increases the size of the initialization vector (IV) used in the encapsulation process to an effective 48 bits. This significantly decreases the probability of an IV reuse by increasing the size of possible IVs to $2^{48}$ as opposed to $2^{24}$ possible WEP IV values. Increasing the IV length also addresses WEP's weak key vulnerability. It achieves this by implementing a very innovative way of splitting the IV into two parts. The first 16 bits of the least significant part of the IV are padded to create a 24-bit IV in a way that avoids the use of weak keys. This process is called per-packet key mixing. This IV is joined to a mixed key that is calculated using the remaining most significant 32 bits of the TKIP IV as well as the MAC-address of the wireless LAN card to generate the key. It ensures that every packet has a different set of IVs. Thus, one of the main problems of the WEP algorithm is solved, namely that every station belonging to the network is using the same key for encrypting data. As shown in Figure 1, the second phase of TKIP key mixing function reuses the 80-bit TKIP-mixed Transmit Address and Key (TTAK) or phase 1 key (P1K) with MAC Protocol Data Units (MPDUs) associated with the same 32-bits upper IV part, Temporal Key (TK) and Transmitter Address (TA) for the next consecutive $2^{16}$ packets. Hence, the 32-bit high IV becomes known to the receiver when the first encrypted packet is transmitted and this part is cached since it is static for the subsequent $2^{16}$ packets.

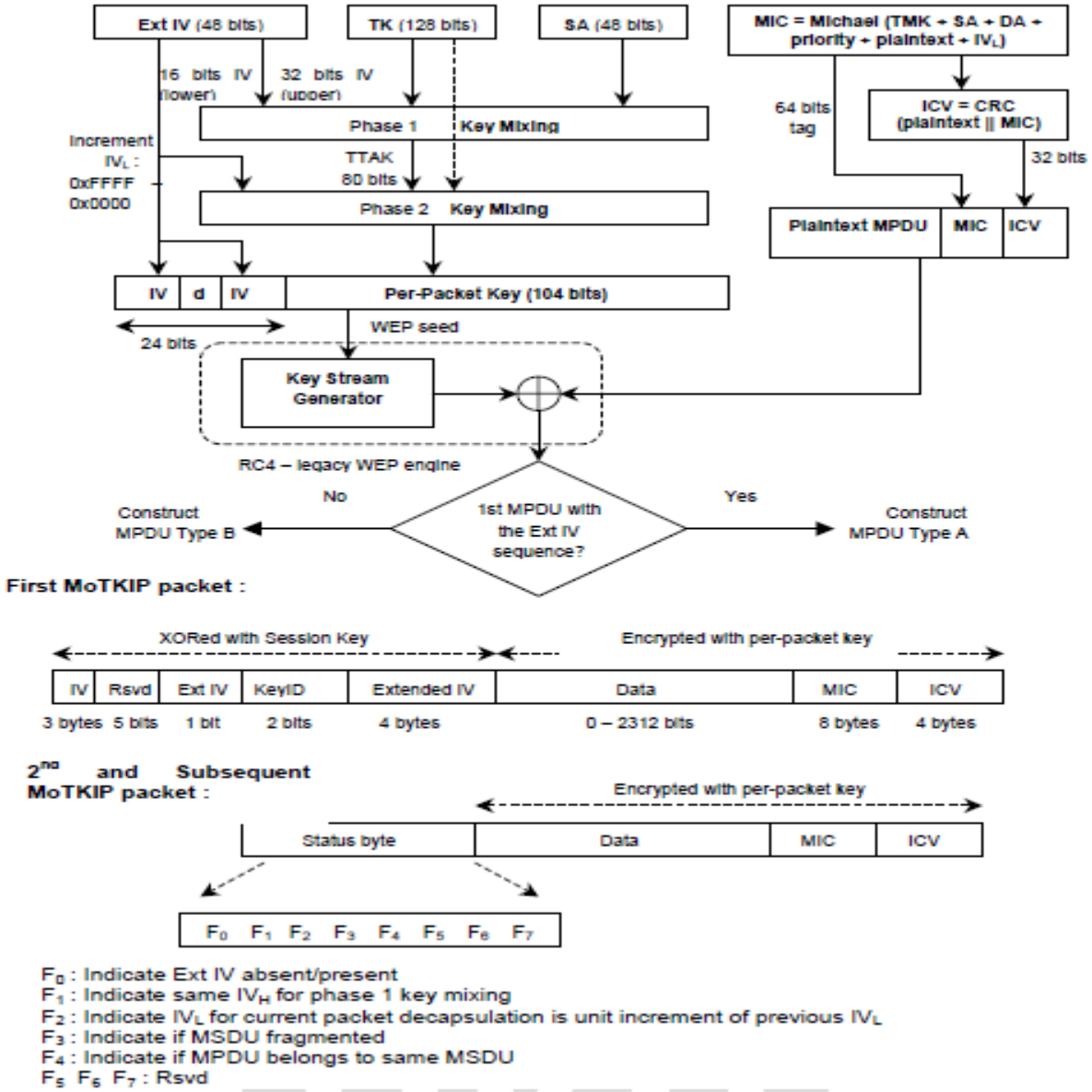


Figure 1: TKIP Authentication in WLAN

The second phase mixes the output of the first phase with the TK and monotonically increments 16-bit low IV part counter (i.e. 0x0000-0xFFFF) to produce the final WEP seed, also called the per-packet key. Since the knowledge of 32-bit high IV and the future sequence of 16-bit low IV is also known to the receiver, it is not necessary to send the full 48-bit extended IV as redundancy again in each packet. Thus, the cached 80-bits TTAK derived from the IV in the first packet at the transmitter will also be that same input to the second phase mixing of the receiver and automatically the next 16-bit IV counter it is just a unit increment of the previous one. Since the IV counter is predictable, phase 2 can be computed in advance while waiting for the next packet(s) to arrive at the receiver. Therefore, in the new Modified TKIP (MoTKIP) frame format, the redundant 4-bytes extended IV is removed from the packet load for packets ranging from the $2^{nd}$ to the $(2^{16})^{th}$ packet. We use the standard code algorithm in (IEEE Standard for Information Technology, 2004) to optimized TKIP key mixing phase.

The function MK16 constructs a 16-bit value from two 8-bit inputs as MK16(X,Y) = (256*X) + Y. The phase 1 output stays the same for $2^{16}$ (i.e. 65, 536) consecutive frames from the same TK and TA. Cheap CPU operations common on 802.11 devices, such as the exclusive-OR operation (⊕), the addition operation (+), the AND operation (&), the OR operation (|), the right bit shift operation (>>), rotate and table look-ups are used in phase 1 and phase 2 key mixing.

Figure 1 also illustrates the general procedure for MoTKIP. For MoTKIP encapsulation process, another major change in TKIP frame format is implemented by calculating the MIC over IV also; and only for first packet transmission the Extended IV XORed with session key is concatenated and sent in the packet. In addition, the MoTKIP encapsulation uses special flag bits for specific control purpose. At the start of secure communication, the transmitter or sender computes a keyed cryptographic message integrity code, or the MIC, over the MSDU source and destination addresses, the priority bits, the MSDU plaintext data and the 48-bit Extended IV also. MoTKIP appends the computed MIC to the MSDU data prior to fragmentation into MPDUs. The receiver obviously has to verify the MIC after decryption with Extended IV, ICV checking, and reassembly of the MPDUs into an MSDU. Invalid MIC naturally leads to discarding of corresponding MSDUs, and this defends against forgery attacks and replay attacks.

MoTKIP uses the cryptographic key mixing function to combine a temporal key (TK), transmitter address (TA), and the extended IV into the WEP seed similar to the classic TKIP. At the start of the session, for the first data packet, the 48-bit extended IV XORed with a session key is appended to the encrypted data.
However, the new MIC for each MoTKIP packet is calculated using the IV, source and destination addresses and data payload at the transmitter and the same is recalculated at the receiver to detect replay attacks.
In the traditional approach, the key reason why the IV is transmitted in the clear is because the 802.11 standard assumes that an adversary does not gain any useful information from its knowledge. The IV is meant to introduce randomness to the key, and appending the clear IV in the transmitted packet helps the receiver to decrypt the information sent from the transmitter station. However, it has been proved that various types of attacks are possible using the IV knowledge as described in the literature (Walker, 2004; Borisov, Goldberg & Wagner, 2001).
In our new approach to strengthen the TKIP security, for the first packet transmitted when a session starts between STA and AP, the extended IV is encrypted with a session key. Hence, first MoTKIP packet sent with encrypted IV uses **C = [ Ks ⊕ IV, P ⊕ RC4 ( IV, TK, SA) ]**. The MoTKIP MIC is computed over: the MSDU destination address (DA); the MSDU source address (SA); the MSDU priority (Reserved for future use); the entire unencrypted MSDU data (payload) and the unencrypted entire IV also. The DA, SA, clear extended IV, 3 octets reserved to 0 and a one octet priority field are used for calculating the MIC and are not transmitted. The TKIP encapsulation mechanism is composed of several sub processes all working continuously to provide the decrypted packets. The process of encapsulating the first MoTKIP packet is the opposite of the encapsulation process of the first packet with the addition of the integrity checks. The MoTKIP decapsulation is designed to be the least computationally intensive.
Since the predictable rule for sequencing the Extended IV for subsequent TKIP packets (from $2^{nd}$ to $65536^{th}$ packet) is to increment the low 16-bit IV part monotonically from 0xFFFF to 0x0000, the 48-bits Ext IV is not included in these MoTKIP packets. Instead, a status byte is appended as shown in Figure 1.

**Kerberos protocol:** An authentication protocol would run before the two communicating parties in the system run some other protocol. The authentication protocol first establishes the identity of the parties to each other's satisfaction; only after authentication does the parties get down to the work at hand. It is a fundamental building block for a secure networked environment.
Kerberos performs authentication under these conditions as a trusted third party authentication service by using conventional cryptography. It trusted in the sense that each of its clients believes Kerberos's judgment as to the identity of each of its other clients to be accurate.
The problem that Kerberos addresses is this: a distributed system in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restricted access to authorized users and to be able to authenticate requests for service. In this system the following three threats exist:
**1.** A user may gain access to a particular workstation and pretend to be another user operating from that workstation.
**2.** A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation.
**3.** A user may eavesdrop on exchanges and use a reply attack to gain entrance to a server or to disrupt operations.

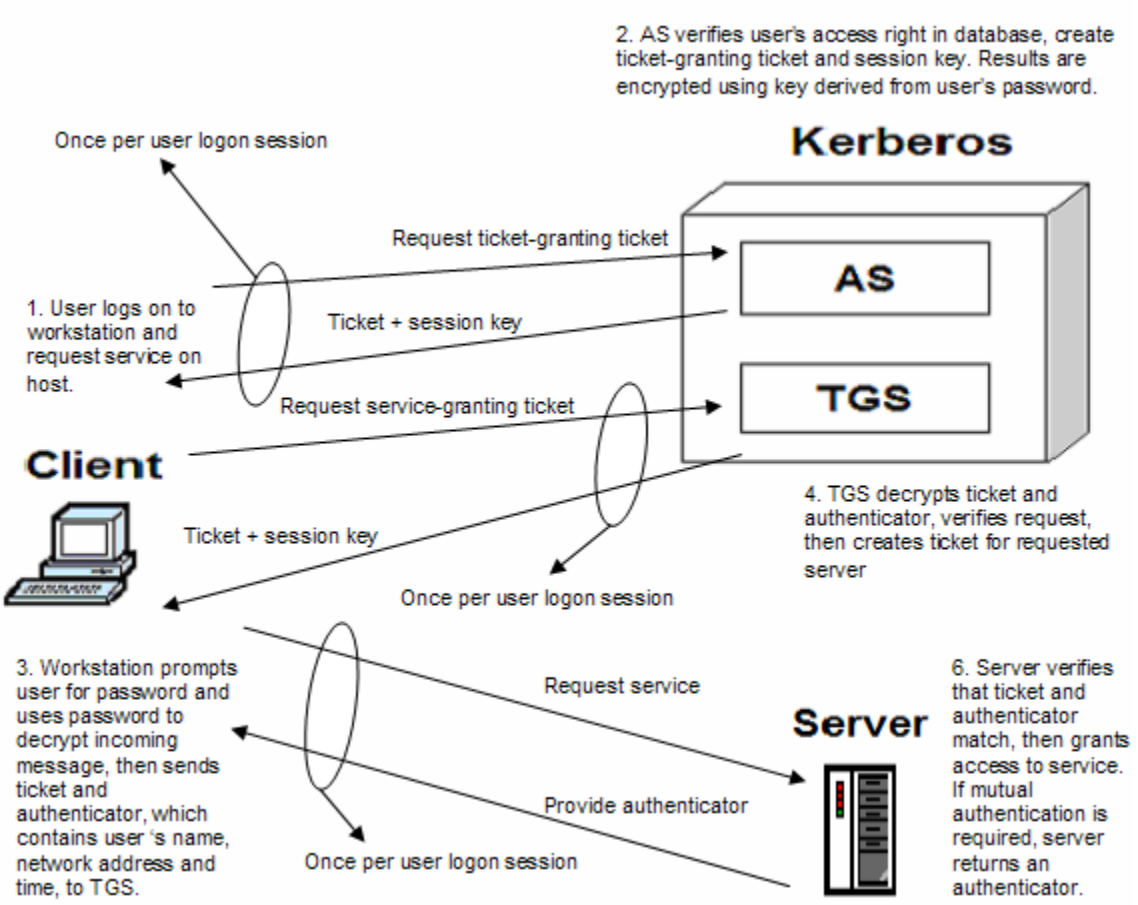


Figure 2: Kerberos Protocol for network security

**III Proposed Design**

CCMP: IEEE802.11i mandates the use of a protocol to protect confidentiality and integrity of data transferred, named Counter mode with CBC-MAC Protocol (CCMP). CCMP provides confidentiality and integrity of the data transferred and authenticity of the sender. It is based on the Advanced Encryption Standard (AES) block cipher. AES is the most reliable block cipher to date, it uses a minimum of 128-bit key length and text blocks of 128- bits as well [4]. This is a great advancement over traditional WEP protocol which is based on weak RC4 stream cipher. CCMP consists of two important protocols, Counter Mode AES encryption (CTR-AES) and Cipher Block Chaining – Massage Authentication Code (CBC-MAC) based on AES. CTR-AES encrypts data transferred (i.e. achieves confidentiality) and CBCMAC provides integrity of data and authentication of the sender by calculating the Message Integrity Code (MIC) of the message. Figure 11 shows how MIC is calculated using CBC-MAC based on AES block cipher.

The cipher text output of the first round of CBC-MAC is fed back as an input to second round of CBC-MAC, this operation continues till the nth round. The output of the nth round is the MIC of the plain text. STA and AP share KCK, 128-bits minimum, derived from PTK and used to calculate MIC. Assume that MIC generated by STA is called MIC (STA); similarly MIC generated by AP is called MIC (AP). MIC (STA) will be sent to the AP as well as the original message. The AP will receive the original message, calculates MIC (AP) based on the message received and compares it with MIC (STA). If both MICs are identical, then this indicates that the message has not been tampered while transmission which also means integrity is preserved. Further, if MIC(STA) = MIC(AP) then there is a very high probability that the message came from STA because only STA holds a shared secret key, KCK in this case, with the AP. Figure 12 shows integrity and authentication protocols of CBC-MAC. CTR-AES is one mode of AES operation, this mode is based on a counter that increment an initial value. CBCMAC requires an IV to start its operation, the counter in CTR-AES and the CBC-MAC IV are constructed from the concatenation of Packet Number (PN) and miscellaneous data like the sender's MAC address and some priority bits reserved for future use. CTR-AES is used to encrypt the traffic between AP and STA and vice versa. Both parties obtain an encryption key from PTK or GTK to encrypt messages as well as generate MIC using CBC-MAC, this key is 128-bits and it is called, Temporal Key (TK). A block diagram of CCMP protocol is illustrated in Figure 13. The diagram shows how CBCMAC IV and CTR-AES counter are constructed. CBCMACIV is fed into the CBC-MAC encryption along with the message, MAC header and TK to generate MIC. Note here that only selected elements of the MAC header are fed into CBC-MAC operation like sender and receiver MAC addresses while other fields are set to Zero. MIC generated is used to preserve the integrity and authenticity of the message and MAC header, MIC will become an input to CTR-AES so it can be protected from modifications.

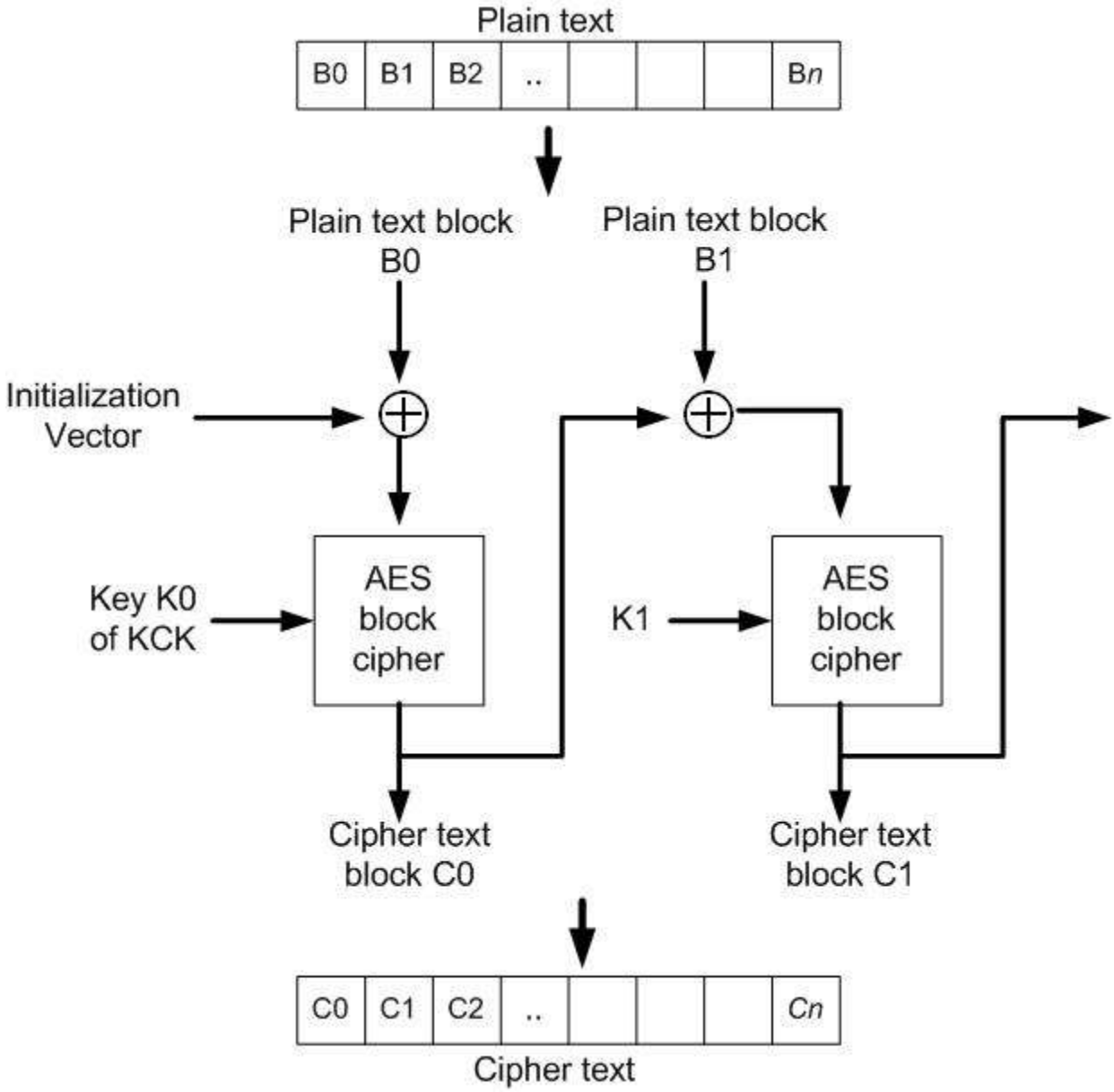


Figure 3 Illustration of Calculation of MIC using CBC-MAC AES based bloc cipher CTR-AES

Counter value is fed into CTR-AES encryption along with TK, the message and MIC. Moreover, a special header is constructed for CCMP. CCMP header contains information like PN which is
Necessary to counter replay attacks. Extended IV is a one bit flag which is always set to one when CCMP protocol is used. The final output from the CCM encryption block is the message and it MIC in an encrypted form, MAC header is added, some parts of MAC is already in MIC to provide authenticity and integrity, then CCMP header is injected between the encrypted message and MAC header. The packet containing the encrypted message with its MIC, CCMP header and MAC header is now sent over the insecure channel. The receiver will decrypt the message and MIC using TK, a new MIC will be generated from the decrypted message and some parts of MAC header, the two MIC's are compared to insure validity of the message as well as authenticity of sender. CCMP uses PN efficiently; PN helps in resolving problems faced by WEP and its successor TKIP encryption protocols. A fresh PN is required for every message, this is achieved by continuously incrementing it. IEEE802.11i specifies that PN should be initialized to one whenever TK is changed. On the receiver side, the PN number is compared to the previous PN number received, if the fresh one is greater than the previous one while using the same TK, this means that the message is not under replay attack. Incrementing PN for each message sent will assure that PN is never repeated with the same TK. In general, IEEE802.11i will overcome all shortcomings of WEP and TKIP, the following points summarizes the advantages of CCMP protocol • Protects the privacy of messages using CTR-AES encryption which is a powerful encryption algorithm. • Protects the integrity of the message, counters forgery packets and proofs the authenticity of the sender using MIC. Additionally, it protects the source and destination addresses from modification hence defending against man-in-the-middle attack and MAC address spoofing. • Protect users from replay attacks because it uses packet sequence numbers. • Prevents key reuse. CCMP uses TK which is derived from the 4-way handshake scheme shown in Figure 10. IEEE802.11i specifies that CBC-MAC IV and counter of CTR-AES are never repeated with the same TK.

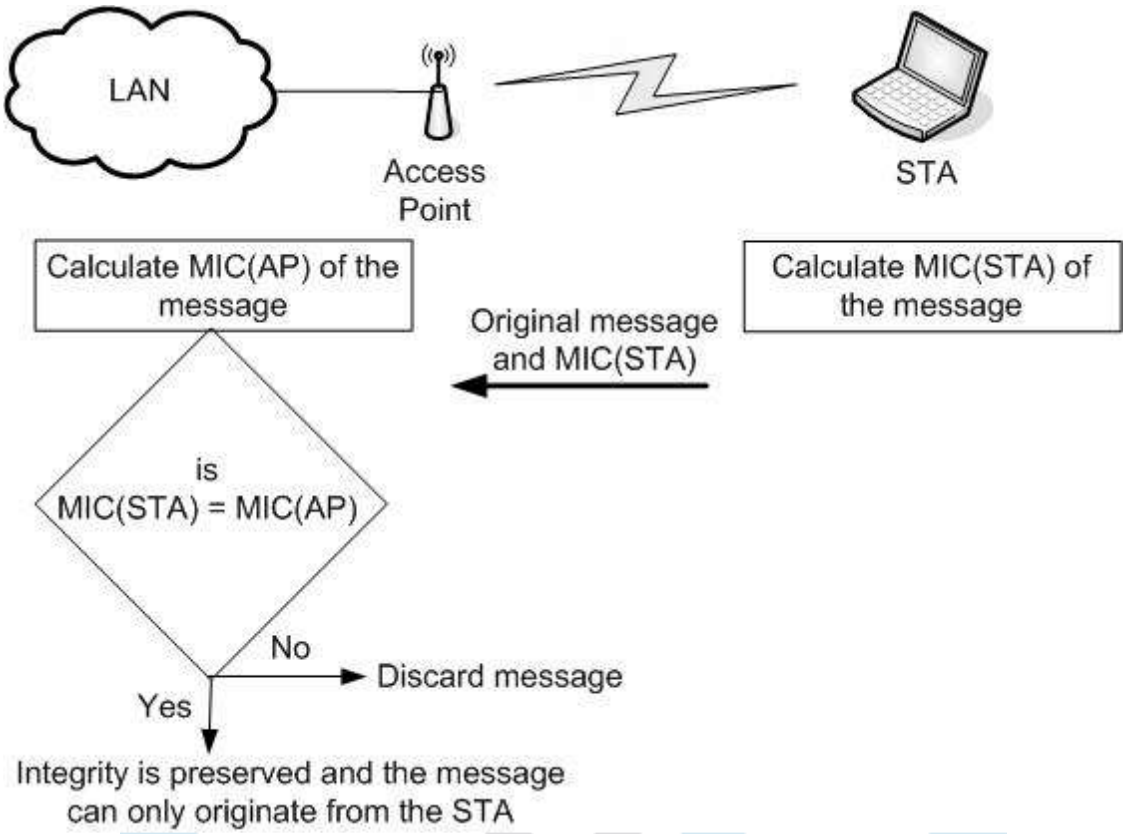


Fig. 4 CBC-MAC protocol used for the calculation of MIC as shown in Fig.11.

## IV-Results

The experimental test setup consisted of establishing and testing secure communication performance via 802.11b wireless network cards and an access point. The nominal data rates were 11Mbps. Encrypted files are transmitted from clients to server. As the server is receiving data packets, it checks for the status byte header to identify the proper MoTKIP algorithm for decryption. Several statistical experiments were performed to verify the MoTKIP operation and WLAN performance throughput.
Parameters to TKIP work are Execution time, throughput & Total Avalanche.

**Time delay:** It is time taken to developing output cipher from input data.

**Throughput:** It is rate to generating cipher per seconds may be compute from formula below

$$\text{throughput} = \frac{\text{No. to Bits}}{\text{Time taken}}$$

**Avalanche:** It is total number to bits change between output ciphers before and after single bits change in key & it may be describe by formula below [15] $\text{Avalanche} = (\text{Cipher}_{\text{key}})\ \text{XOR}\ (\text{Cipher}_{\text{key+1}})$

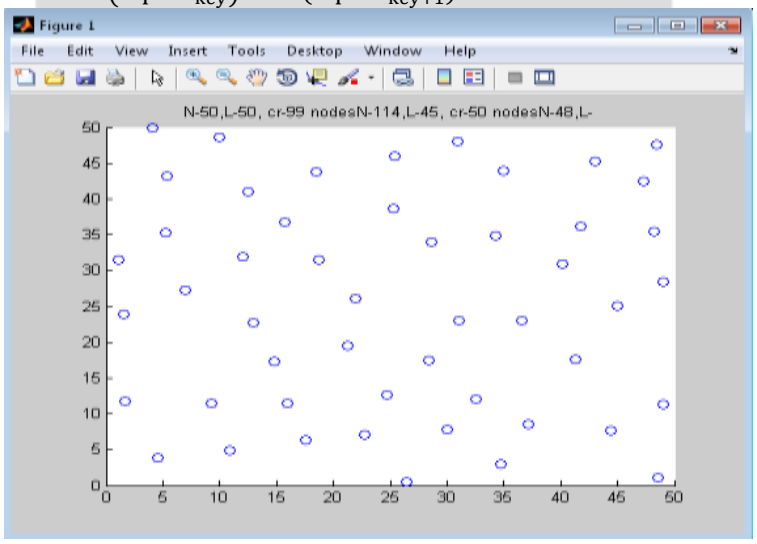


Figure 5: 50 nodes network with data length of 50

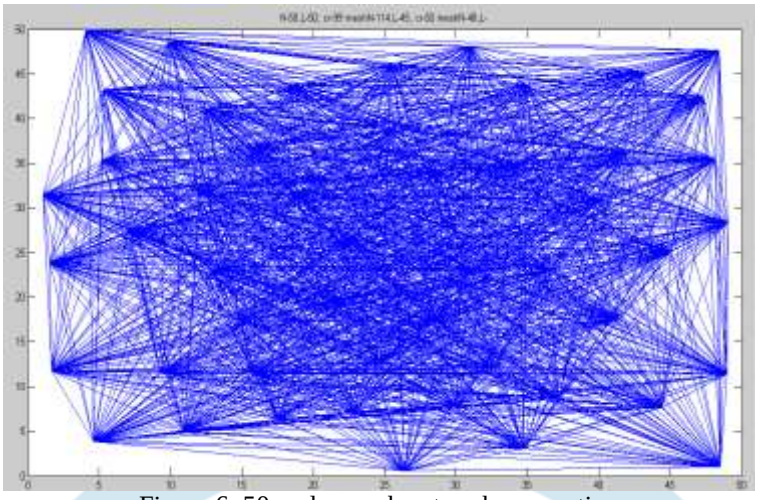

Figure 6: 50 nodes mesh network connection

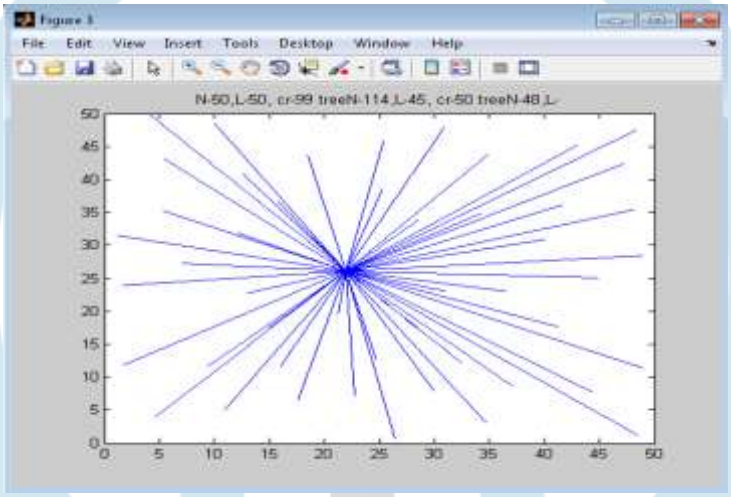


Figure 7: 50 nodes tree network connection

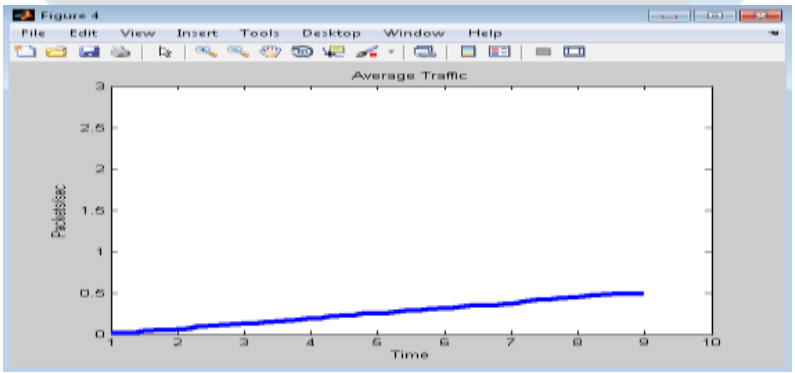


Figure 8: Average traffic in the network of 50 nodes

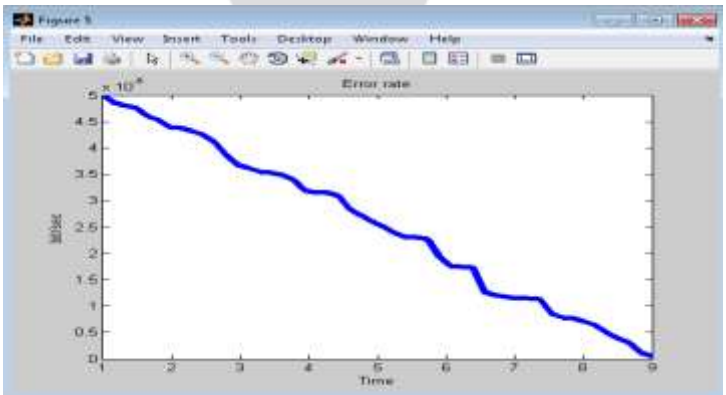


Figure 9: Error rate in the network of 50 nodes

Avalanche Test to Simulation: avalanche is been observed to three various data input -
**Test1:** In first test cipher to 128 bit data "ABDEF1CDEF12EF12" & 128 bit key "0123456789ABCDEF" then cipher text is "EA324AED32E20179" & if we develop cipher to same data & single bit key change new key is "0123456789ABCDEE" new cipher we found is "1D08632DE5B6017C" total change in new & old cipher is 70 bits.
**Test2:** In Second test cipher to 128 bit data "ABDEF1CDEF12EF12 " & 128 bit key "FEDCBA9876543210" then cipher text is "28510AB4CDE97EA2" & if we develop cipher to same data & single bit key change new key is "FEDCBA9876543211" new cipher we found is "3A4C9F7AE297D802" total change in new & old cipher is 69 bits.
**Test3:** In Third test cipher to 128 bit data "ABCDEF0123456789" & 128 bit key "FEDCBA9876543210" then cipher text is "D3D8E2290EACF41A" & if we develop cipher to same data & single bit key change new key is "FEDCBA9876543211" new cipher we found is "B4C982A8C4D5D5E8" total change in new & old cipher is 73 bits.

*Table 1 Avalanche observed to TKIP work*

| SN | Test | Avalanche observed |
|---|---|---|
| **1** | Test-1 | 70 |
| **2** | Test-2 | 69 |
| **3** | Test-3 | 73 |

On behalf to above results we may conclude that minimum Avalanche to TKIP work is 69.

**Time Test to Simulation:** time delay is been observed to three various data input

*Table 2 Time delay observed to TKIP work*

| SN | Test | Time delay observed |
|---|---|---|
| **1** | Test-1 | 0.919 |
| **2** | Test-2 | 0.928 |
| **3** | Test3 | 0.913 |

On behalf to above results we may conclude that maximum time delay is 0.928 seconds to TKIP work.
**Throughput Test to Simulation:** Throughput is been observed to three various data input

*Table.3 Throughput observed to TKIP work*

| SN | Test | Throughput observed |
|---|---|---|
| **1** | Test-1 | 139.2828 |
| **2** | Test-2 | 137.931 |
| **3** | Test3 | 140.1972 |

On behalf to above results we may conclude that minimum throughput is 137.931 kbps to TKIP work.

**Comparative Results:** comparative results are been developed to compare TKIP work with available works here we have compared our work with latest & related work

*Table.4 Comparative results*

| | Time delay in second | Avalanche in db |
|---|---|---|
| **TKIP (Temporal Key Integrity Protocol)** | 0.928 | 69 |
| **Kerberos** | 3.45 | 69 |

Comparative results above shows that as compare with work Kerberos [1] our work is similar in security because avalanche observed in TKIP work is same, however time delay by Kerberos is almost 300% much than our work which makes TKIP work faster than their work. As compare with others two works [2], [3] TKIP with is better in all parameters avalanche, throughput & time delay.

*Table.5 Comparative results Average Traffic*

| | Average Traffic (packets/sec) | | |
|---|---|---|---|
| Time (ms) | EAP TLS [2] | KEAP[1] | Kerberos with TKIP |
| 1 | 0.225 | 0.295 | 0.111 |
| 2 | 0.346 | 0.397 | 0.3925 |
| 3 | 0.339 | 0.436 | 0.4925 |
| 4 | 0.427 | 0.533 | 0.8109 |
| 5 | 0.452 | 0.603 | 0.9277 |
| 6 | 0.475 | 0.652 | 1.0414 |
| 7 | 0.502 | 0.698 | 1.301 |
| 8 | 0.522 | 0.715 | 1.6944 |
| 9 | 0.543 | 0.73 | 1.9583 |

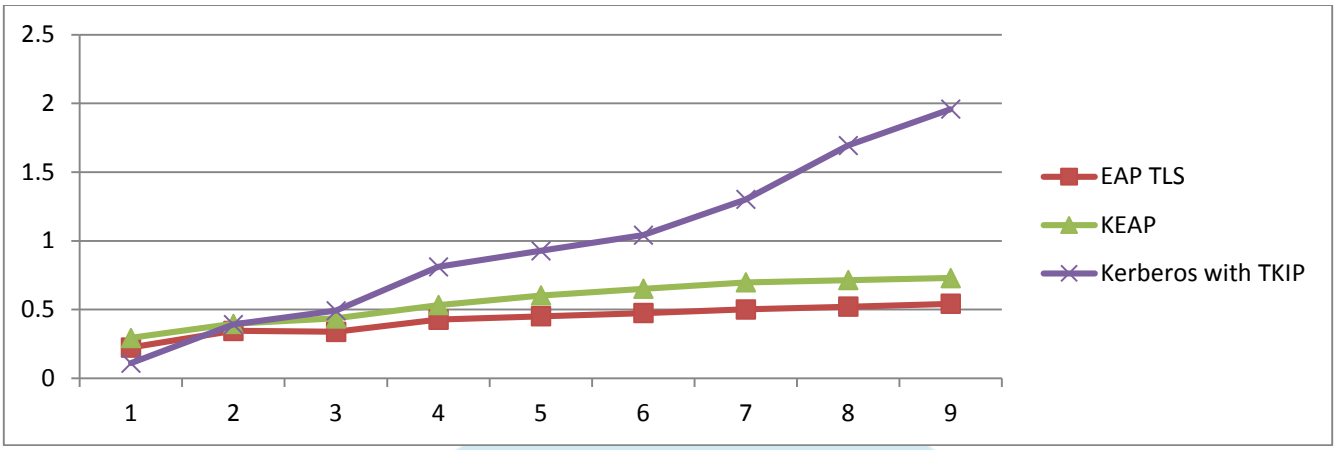


Figure 10 Comparative Results for average traffic

*Table.6 Comparative results Average Bit rate error*

| | Average Bit Rate (bit/sec) | | |
|---|---|---|---|
| Time (ms) | EAP TLS [2] | KEAP[1] | Kerberos with TKIP |
| 1 | 0.0005 | 0.00038 | 0.0001969 |
| 2 | 0.00039 | 0.000232 | 0.00015 |
| 3 | 0.00034 | 0.000164 | 0.0001221 |
| 4 | 0.00029 | 0.000131 | 0.000112 |
| 5 | 0.000275 | 0.000119 | 0.0001051 |
| 6 | 0.000243 | 0.000115 | 0.0000971 |
| 7 | 0.00024 | 0.000112 | 0.0000816 |
| 8 | 0.000228 | 0.000113 | 0.0000616 |
| 9 | 0.000227 | 0.000137 | 0.0000025 |

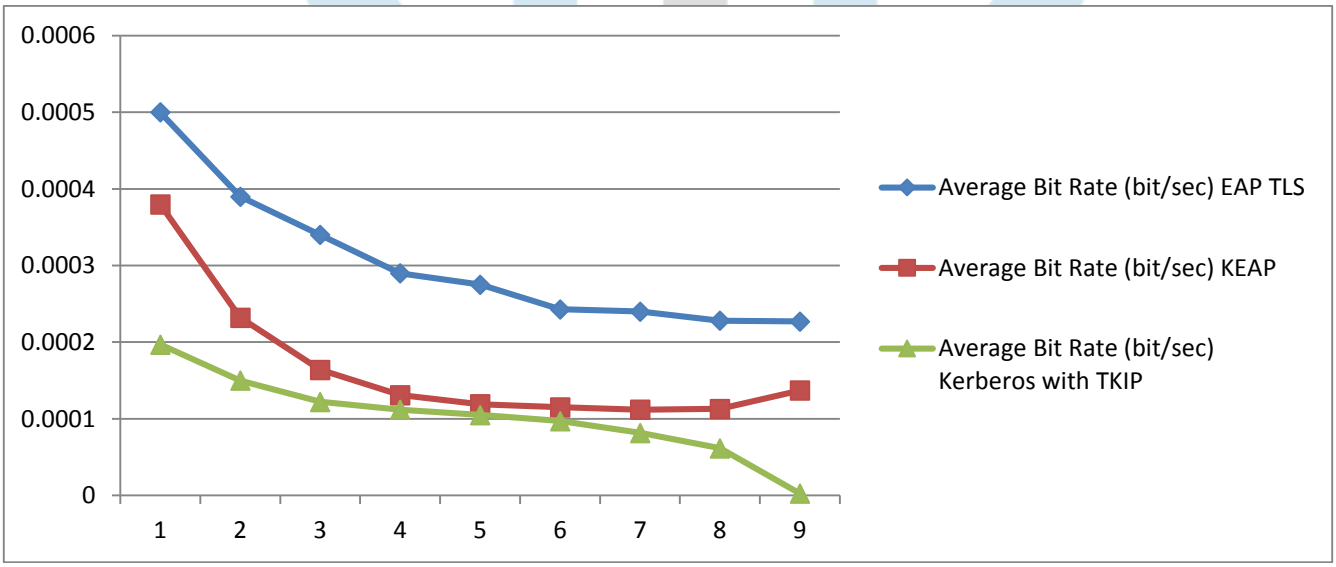


Figure 11: Average Bit Error Rate Comparison

**V-Conclusion**

An upgrade in software is what made TKIP more secured than WEP. However, the core encryption algorithm is still the same, weak RC4 stream cipher, with this encryption algorithm and the design flaws it experiences, TKIP believed to be a short-life solution. IEEE recognized the need for a new protocol that is more secure and long lasting. IEEE finally answered the call by working on a new security standard, IEEE802.11i. The standard was approved in June 2004. This new standard addresses new security protocols and introduces the adoption of strong block encryption algorithm, Advanced Encryption Standard (AES), also introduces a new key management scheme. Attacks on privacy, integrity, and authentication can be overcome by IEEE802.11i. As far as the logical attacks are concerned, IEEE802.11i provides adequate solutions to defend against WEP weaknesses, man-in-the-middle attacks, and forgery packets attacks and replay attacks. However, DoS attack is not addressed properly and there are no solid protocols or

implementations to stop such attacks basically because the attacks target the physical layer of the TCP/IP stack like interfering with the frequency band. Most research activities in wireless security are done on the data link and upper layers. Researchers are working hand to hand with the industry to provide the best solution for logical attacks but there is negligence in the area of physical attacks in which human behavior and human interaction with devices takes place.

**References**

[1] Yi Ma, Hongyun Ning, The Improvement of Wireless LAN Security Authentication Mechanism Based on Kerberos, 2018 International Conference on Electronics Technology, 978-1-5386-5752-2/18/ IEEE

[2] Abhijit Bodhe Mayur Masuti Dr. A.S.Umesh,,WIRELESS LAN SECURITY ATTACKS AND CCM PROTOCOL WITH SOME BEST PRACTICES IN DEPLOYMENT OF SERVICES , International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056 Volume: 03 Issue: 01 | Jan-2016

[3] EMIL SELVAN GSR1, GAYATHRI N1, RAKESH KUMAR S2, ANKUSH RAI3, JAGADEESH KANNAN R, ADVANCED ENCRYPTION AND EXTENDED AUTHENTICATION FOR WIRELESS LOCAL AREA NETWORKS, Advances in Smart Computing and Bioinformatics, AJPCR, Special Issue (April), DOI: http://dx.doi.org/10.22159/ajpcr.2017.v10s1.19987

[4] Matthew S. Gast, 802.11 Wireless Networks, O'REILLY, 2002.

[5] William Stallings, Cryptography and Network Security, Principles and Practices, 3rd Edition, Prentice Hall 2003.

[6] Matija Sorman, Tomislav Kovac and Damir Maurovic, "Implementing Improved WLAN security", 46th International Symposium Electronics in Marine. ELMAR-2004, Zadar. Croatia, 16-18 June 2004.

[7] Joon S.Park and Derrick Dicoi, "WLAN Security: Current and Future". IEEE Computer Society, October 2003.

[8] Nancy R. Mead and Gary McGraw. "Wireless Security's Future". IEEE Computer Society, IEEE Security and Privacy, August 2003.

[9] Joseph Williams, "Providing for Wireless LAN Security, Part 2". IEEE IT Pro, November | December 2002.

[10] IEEE Standard for local and metropolitan area networks, "Wireless LAN Medium Access Control (MAC) and Physical Layer Specifications", ANSI/IEEE Std 802.11, 1999 Edition (R2003).

[11] Shin, M.; Ma, J.; Mishra, A.; Arbaugh, W.A., "Wireless network security and interworking", Proceedings of IEEE, Volume 94, Issue 2, pp 455 – 466, February 2006.

[12] Wang Shunman, TaoRan, WmgYue and ZhangJi, "Wireless LAN and it's security problem". Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2003. PDCAT'2003.