I. J. Computer Network and Information Security, **2012, 3, 31-46**
Published Online April 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.03.05

Attacks Due to Flaw of Protocols Used In Network Access Control (NAC), Their Solutions and Issues: A Survey

Snehasish Parhi
Department of Computer Center
National Institute of Technology, Rourkela, Odisha, India, 769008
E-mail: sparhi@nitrkl.ac.in

***Abstract*—**In order to ensure and enforce endpoint security, Network Access Control (NAC) is attracting considerable interest from the research community. Most NAC architectures are based on 802.1x, EAP (Extensible Authentication Protocol), EAPoL (EAP over LAN) 802.11i, 802.11w, and RADIUS (Remote Authentication Dial-In User Service) protocols. Unprotected management and control frames in some of above protocols lead to several attacks. Eliminating flaws completely in design of each protocol is a challenge. These flaws help malicious user and infected endpoint to intrude into the NAC architecture to make damage into it. Many researches have been carried out to address this issue. In this paper, we have made an attempt to explain attacks in above protocols and present a survey and analysis of different solution approaches proposed by researchers. The affect of vulnerability and attack of above protocols in NAC is also discussed. The finding of this review will provide useful insights into the vulnerabilities, attacks in above protocols, and their proposed solutions with issues, which may create a research scope for strengthening security in NAC.

***Index Terms*—** NAC, 802.1x, EAP 802.11i, 802.11w, RADIUS

I. Introduction

Network Access Control (NAC) is designed to meet the demand of securing network though several security solutions like IDS (Intrusion Detection System), IPS (Intrusion Prevention System), antivirus, host or gateway based firewall. It unifies endpoint security technology (such as vulnerability assessment, host intrusion prevention, and antivirus), user or system authentication and network security enforcement. In order to control secured network access, it uses security policy, which includes endpoint's security, policy checking before access, location and behavior control after access. This will minimize attacks from endpoints, strengthen policy enforcement, and manage the identity and access [1].

The days are gone when users are satisfied putting firewall to stop potentially hostile traffic from penetrating the perimeter was felt sufficient. Now, security threats are much more numerous and sophisticated. Malware spyware and phishing attacks now arise in disguises and it is hard to guard against them. Security threats can be internal or external. There are several measures to fight against external threats, such as IDS, IPS, and firewall. In internal network, the danger comes from systems that are already infected and are vulnerable. Such systems might not be compliant with the organization's internal security policies or up-to-date with operating system patches or antivirus update. This infected host on a network can infect other hosts. In order to guard against internal threats, corporate world require a compliance policy for internal clients/hosts, so that healthy environment inside network can be maintained by isolating unhealthy clients/hosts. NAC promises the best alternative to curtail these hazards. It ensures a much cleaner and less risky host environment by stopping any infected or noncompliant device from accessing the network, and by providing remediation method to bring those devices into compliance. NAC is not the be-all and end-all for internal security. It cannot fully protect against phishing attacks or remove malware completely. It is increasingly considered as the best first line of defense for enterprise security by researchers [2].

Network Admission Control of Cisco (CNAC), Network Access Protection (NAP) of Microsoft, and Trusted Network Connect (TNC) of Trusted Computing Group (TCG) are three representative solutions of NAC [1]. These NAC solutions are based on 802.1x and RADIUS [3]. Both wired and wireless network use 802.1x architecture [1,4,5]. Implementation of secure NAC solution depends on the strength and weakness of protocols, such as 802.1x, EAPoL, EAP, 802.11i, 802.11w, and RADIUS. Unprotected management and control frames or weak encryption method in above protocols lead to several attacks. Eliminating flaws completely in design of each protocol is a challenge.

Rest of the paper is organized as follows. Section II-VI defines above six protocols and their attacks, solutions, and their issues. Section VII summarizes all solutions of attacks and their issues. Section VII illustrates basic communication flow of generic NAC, affect of attacks over NAC, and Section IX concludes.

II. 802.1x Framework, Attacks and Solutions with Issues

The authentication framework in 802.1x, exchange of EAP frames encapsulated under EAPoL frame in 802.1x, attacks in 802.1x, and their solutions with issues are mentioned below.

A. 802.1x framework

Purpose of 802.1x framework is to provide an authentication framework for devices and users attempting to connect in wired and wireless network, so that only authorized connections are allowed. The authentication system based on 802.1x protocol does not have its own algorithms for authentication, thus, relies on an EAP and EAPoL to deal with authentication algorithms. 802.1x divides every physical ethernet port into two logical, controlled and uncontrolled ports. Initially, it works as uncontrolled port, where only EAP and EAPoL messages are allowed. Once, the device is successfully authenticated and authorized, the port works like controlled port and the device can fully communicate with other devices. The key components of 802.1x are supplicant (i.e. laptop or a computer or any other device), authenticator (i.e. be switch or access point) and authentication server (AS). AS can be RADIUS [3] server or DIAMETER [6] server. Message communication between supplicant and authenticator based on EAPoL protocol and between authenticator and AS is based on EAPoR (EAP over RADIUS if RADIUS is used as AS) protocol. Both EAP and EAPoL protocols do not contain any integrity measures or privacy protection. The RADIUS protocol contain mechanism for per packet authenticity and integrity between authenticator and AS [7,8,9]. Most managed switches and access points support 802.1x. NAC depends on 802.1x for policy enforcement if user in endpoint is not authenticated.

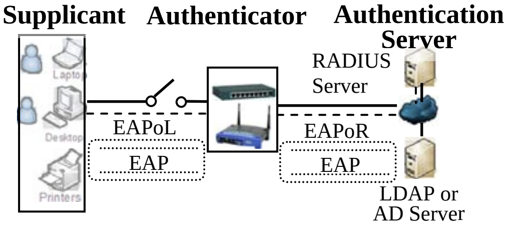


Figure 1: Authentication framework based on 802.1x

Figure 1 represents authentication framework based on 802.1x with three discussed components. Line represented with dash shows uncontrolled connection and line represented with bold shows controlled connection in the figure.

B. Attacks in 802.1x protocol

Merit of 802.1x is that it is a dual-port model, thus, can split network dataflow and authentication information, but demerit of 802.1x is that it does not supervise access port after successful authentication, thus, problems, such as *user name embezzlement* and *user name lift* occurs [7], which is explained as follows.

Scenario 1: Computer 1 is connected to Port A and authenticated via Switch1. After successful authentication, Computer1 changes its IP address manually and keeps its access status to LAN (802.1x protocol should not allow access in this case, because machine identity is changed). Illegal user Computer2 is connected to Port B and authenticated via Switch1 using user name and password of the user using Computer1. In this situation, Computer 2 will successfully pass the authentication because it carries the same user name and password of Computer1, although with different IP address (802.1x protocol should check that same user should not be allowed from machine with different identity than original when same user is already logged on). This is called *user name embezzlement.*

Scenario 2: Hub1 is connected to Port A in Switch1. Computer1 is a legal user and is connected to Hub1. Computer 2 is an illegal user and connected in another port of Hub1. As Computer1 is a legal user, it can easily pass the authentication check via switch1. As soon as authentication is completed, Port A in Switch1 is in the Controlled status, and Port A will allow the supplicant from Computer1. Once authentication of computer1 is successful, authenticator also allow the illegal user using Computer2 to freely access to LAN without any further authentication. This is very dangerous in terms of network security. This is called *user name lift*. Here, authenticator should not allow other users who have different machine identity from original user. This attack occurs due to absence of supervision on access port after authentication in 802.1x protocol.

By-passing upper layer EAP methods like EAP-TLS (Transport Layer Security): When authenticator sends *EAP-Success* message to supplicant after receiving *RADIUS-Access-Accept* message from AS, unconditional transfer to authenticated state occurs (switch port opens) irrespective of current state. Network port in supplicant is always available in authenticated state. Thus, when authenticator port return to controlled state, network connectivity starts. Though EAP-TLS provides strong mutual authentication, but this design error (one way authentication) can bypass the entire EAP-TLS (explained in Section III.A) authentication and cause MITM attack [8].

Session hijacking: In [8], the author has explained session hijacking. Robust Security Network (RSN) provides mechanisms to restrict network connectivity (at MAC layer) to authorized entities via 802.1x. With IEEE 802.1x, higher layer authentication takes place after RSN association/reassociation. Thus, there are two state machines, the RSN and the 802.1x state machine. Their combined action dictates the state of authentication. Because of a lack of clear communication between these state machines and message authenticity, it possible to perform a simple *session hijacking* taking advantage of the loose coupling. It has following steps, which is represented in Figure 2.

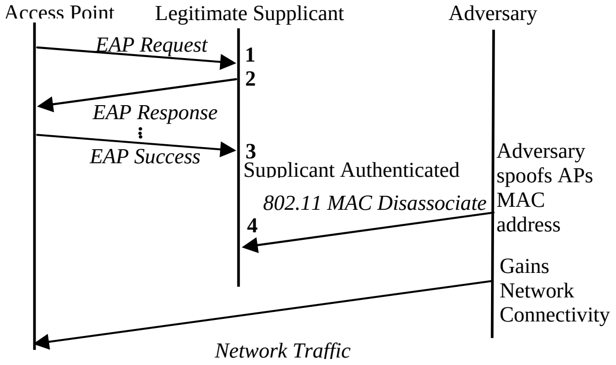


Figure 2: The session hijack by spoofing a 802.11 MAC disassociate message [8]

a) Message 1-3 are example of message communication among legitimate supplicant and AP.
b) An adversary sends a disassociation frame by spoofing MAC address of AP after supplicant is successfully authenticated. Thus, supplicant is disassociated from AP while 802.1x state machine of authenticator still remains in authenticated state.
c) The adversary gains network access using MAC address of authenticated supplicant because the 802.1x state machine in the authenticator is still in the authenticated state.

In 802.1x, supplicant sends only response message to authenticator and authenticator only sends request message to supplicant. Supplicant does not send any request message to authenticator and authenticator does not send any response message to supplicant, which develops asymmetrical relationship, where it supposed to be symmetrical relationship (represented in Figure 5). According to the 802.1x standard, the authenticator port is in the controlled state (switch port is open) only when session is authenticated. This is untrue for supplicant, whose port is always in the authenticated state. Thus, only one way authentication of supplicant to the access point (AP) exists. The one-way authentication of supplicant to the access point can expose the supplicant to potential Man-In-The-Middle (MITM) attack with an adversary acting as an access point to the supplicant and as a client to the network access point [8,10]. MITM attack is represented in Figure 3.

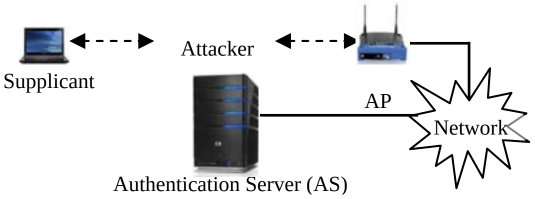


Figure 3: MITM attack

C. Solution of attacks in 802.1x and issues

The solutions above attacks are summarized below.

Modification of EAP payload by adding extra fields for identification: The Type-Data field of EAP represented in Figure 6 which contains payload information. EAP packet is modified, where type is “identity” and identity values are kept in Type-Data. The extra information added are fixed IP address, user name, MAC address, IP messages reaching the access point (IPin), and the number of IP messages departing from the access port (IPout). This information is then sent in *Response/identify* message to AS. Addition of this information helps 802.1x to supervise a machine identity uniquely. Thus, computers using other user’s user name and other computer’s IP address cannot by-pass authentication in 802.1x (this bypass occurs in hub) [7]. *Issues:* Solution may fail if IP and MAC address can be spoofed and changed by Attacker using several tools, such as Spoof-MAC, Airsnarf and MAC Changer [11] to create fake identity. It also requires major modification in EAP frame.

Addition of EAP Authenticator in EAP-success frame like Request Authenticator in RADIUS packet: In RADIUS packet, *Request Authenticator* is kept in *authenticator* field by client and *Response Authenticator* is kept in *authenticator* field by RADIUS server (explained in Section VI.A). Both *Request Authenticator* and *Response Authenticator* are protected by MD5 hash. In [8], the author proposed to use *EAP-Authenticator* in *EAP-Success*, so only legitimate client can accept it. As switch port is opened or closed based on *EAP-Success* or *EAP-Failure* respectively, thus, protection of *EAP-Success* frame by *EAP-Authenticator* only allows legitimate client to accept (not by Attacker) and switch port will be opened for legitimate client only (not for Attacker). Encryption key for *EAP-Authenticator* can be collected from higher-layer authentication protocol, such as EAP-TLS session key, so that Attacker can not spoof the frame which can prevent MITM attack. Author also proposed *EAPoL-key* in place of *EAP-Success* as an indication of success at EAP layer. *Issues:* The proposed solution requires major modifications of EAP frame. Putting hashing algorithm require encryption in AP side, and decryption and verification in supplicant side. This may develop delay in communication.

Building symmetrical relationship: Both the supplicant and AP should be treaded in same way. A mechanism for mutual authentication should be established between them. A supplicant machine should be similar to authenticator including the dual port model. This can be achieved by implementing control logic similar to port access entity (PAE) in addition to authentication of IEEE 802.1x message [10]. *Issues:* This may require major change in architecture based on 802.1x.

III. EAP AND EAPOL PROTOCOLS, THEIR ATTACKS AND SOLUTIONS WITH ISSUES

The authentication process in EAP, EAPoL, frame structure of EAP and EAPoL, their attacks, and solutions with issues are mentioned below.

A. EAP and EAPoL packets in 802.1x authentication process

The communication flow of 802.1x authentication framework among supplicant, authenticator, and AS has following five steps [12] explained in Figure 5.

1. *Initialization*: All APs transmit a *Beacon* frame at fixed intervals. Wireless station (supplicant) listens

for *Beacon* packets to identify AP within range. Alternatively, probe request frames are generated by stations actively searching for existing wireless networks. AP periodically broadcast its security capabilities, indicated by RSN IE (Robust Security Network Information Element) in a specified channel through *Beacon* frame, which is represented in Figure 5, where AA and SPA stands for authenticator and supplicant respectively. RSN IE is used for packet integrity. Station selects one AP out of list of APs available and tries to authenticate and associate with that AP by exchanging messages numbered (4) to (7) in Figure 5. A wireless station (supplicant) can be authenticated by multiple APs. However, it should be associated by only one AP at a time. Authentication can be open or Shared Key Authentication [10]. Once AP is selected, the switch port is enabled and set to "unauthorized" state. 802.11 association must complete before the 802.1x negotiation begins because the 802.1x state machine requires an active link. Supplicant starts authentication process by sending *EAPoL-Start* frame to AP [13]. Steps (1) to (7) in Figure 5 represent association and authentication of wireless device. It is represented by dotted rectangle.

2. *Initiation*: Supplicant responds to *EAP-Request-Identity* frames of authenticator with *EAP-Response-Identity* frame containing an identifier for the supplicant, such as User-ID. The authenticator forward to AS through *RADIUS-Access-Request* packet.
3. *Negotiation*: The AS sends reply encapsulated in a *RADIUS-Access-Challenge* packet to the authenticator, containing an EAP request specifying the EAP method, such as EAP-MD5, EAP-TLS (Transport Layer Security), EAP-TTLS (Tunnel TLS) or PEAP (Protected EAP) etc. The authenticator encapsulates the EAP Request in an EAPoL frame and transmits it to the supplicant through *EAP-Request-Authentication* frame. The supplicant can send *NAK* packet with EAP method it supports or responds with its credentials information using *EAPoL-Response-Authentication* frame using requested EAP method.

In order to provide strong encryption to prevent eavesdropping and mutual authentication between supplicant and AS, 802.1x has provided framework into which a particular EAP method out of several EAP methods can be used. Organization can adopt any EAP method depending on their suitability. Comparison among several EAP methods are mentioned in [2,14]. Figure 4 represents basic EAP architecture in 802.1x environment.

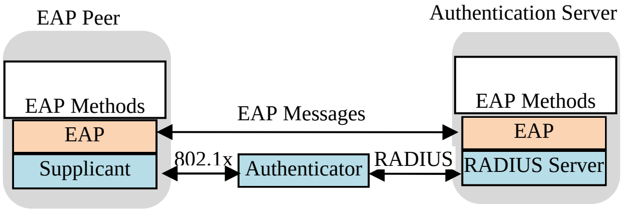


Figure 4: Framework of 802.1x and RADIUS where EAP methods are used

4. *Authentication*: The authenticator sends *RADIUS-Access-Request* frame with supplicant's credential information to AS. If the supplicant's credentials are valid, *RADIUS-Access-Accept* frame is sent otherwise, *RADIUS-Access-Reject* frame is sent back to the authenticator by AS. Authenticator sends *EAP-Success* to supplicant if authentication successful otherwise, *EAP-Failure* if authentication unsuccessful. Once, authentication is successful, port is open for accessing network. Whenever supplicant desires to terminate connection, it sends an *EAPoL-Logoff* frame to AP.
5. *Generate keys*: After successful authentication between supplicant and AS, they generate some common secret, called Master Session Key (MSK). Supplicant uses MSK to derive Pairwise Master Key (PMK). The AAA (authentication, authorization, and accounting) key material on AS side is securely transferred to the authenticator to derive same PMK (indicated by Message 18 in Figure 5). This stage is skipped if both supplicant and authenticator are configured using static Pre-Shared Key (PSK). AS afterwards, detach itself after successful transfer of key materials to authenticator then, supplicant and authenticator proceed with further key generation. After successful 802.1x authentication between supplicant and AS, PMKSA (PMK Security Association) consists of PMK, PMKID (PMK Identity), which identifies PMK, lifetime etc. is created for supplicant and authenticator. After completion of above steps, 4-way handshake (represented in Figure 7) based on PMK is performed between supplicant and authenticator for mutual authentication and session key derivation. A session key i.e. Pairwise Transient Key (PTK) is derived, which is used to protect data frames exchanged between supplicant and authenticator [15].

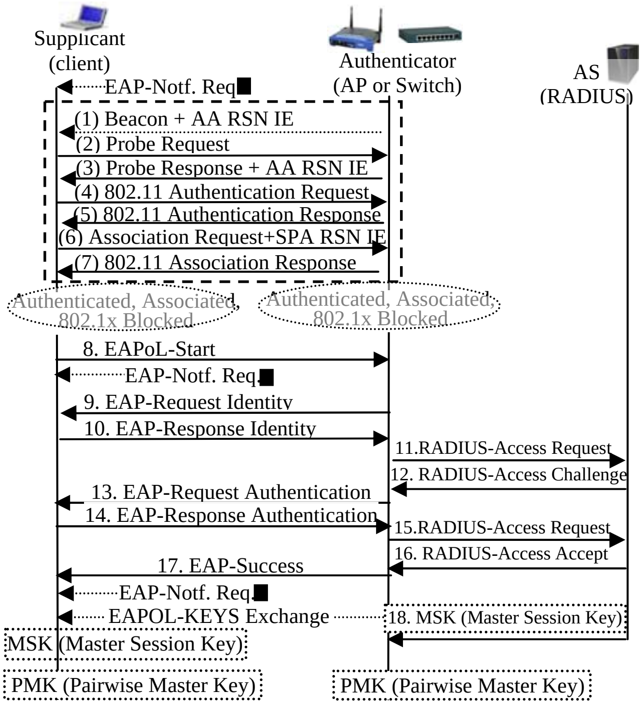
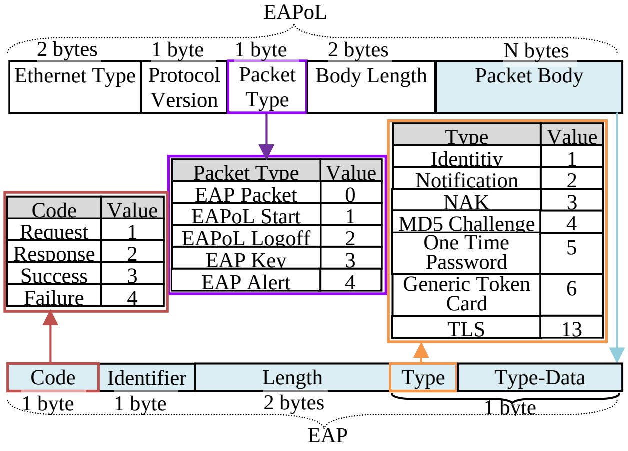

Supplicant (client)
Authenticator (AP or Switch)
AS (RADIUS)
EAP-Notf. Req
(1) Beacon + AA RSN IE
(2) Probe Request
(3) Probe Response + AA RSN IE
(4) 802.11 Authentication Request
(5) 802.11 Authentication Response
(6) Association Request+SPA RSN IE
(7) 802.11 Association Response
Authenticated, Associated, 802.1x Blocked
Authenticated, Associated, 802.1x Blocked
8. EAPoL-Start
EAP-Notf. Req
9. EAP-Request Identity
10. EAP-Response Identity
11.RADIUS-Access Request
12. RADIUS-Access Challenge
13. EAP-Request Authentication
14. EAP-Response Authentication
15.RADIUS-Access Request
16. RADIUS-Access Accept
17. EAP-Success
EAP-Notf. Req
EAPOL-KEYS Exchange
18. MSK (Master Session Key)
MSK (Master Session Key)
PMK (Pairwise Master Key)
PMK (Pairwise Master Key)

Figure 5: Communication flow in 802.1x authentication framework

B. *EAP and EAPoL frame structure*

The frame structure of EAPoL and EAP [12,16] are mentioned in Figure 6.

Type and Type-Data in EAP frame is coming under one category called, "Data". Data field is 0 in Success and Failure packets. For Request and Response message, data field is dual group one is Type and other is Type-Data. The Type field in EAP frame is used to indicate the type of information, which is being transmitted. The Type-Data field contains the payload information. If customization is required, this field is modified to include new information as illustrated in [7,17].



Code	Value
Request	1
Response	2
Success	3
Failure	4

Packet Type	Value
EAP Packet	0
EAPoL Start	1
EAPoL Logoff	2
EAP Key	3
EAP Alert	4

Type	Value
Identity	1
Notification	2
NAK	3
MD5 Challenge	4
One Time Password	5
Generic Token Card	6
TLS	13

Figure 6: EAP and EAPoL frame format

C. *Attacks in EAP and EAPoL packets used in 802.1x protocol*

Attacks in EAPoL and EAP frames are mentioned below.

Some unprotected management frames of EAPoL are "*EAPoL-Start*" and "*EAPoL-Logoff*". Unprotected management frames of EAP are "*EAP-Success*", "*EAP-Failure*", "*EAP-NAK*", and "*EAP-Notification*". These frames are transmitted in clear text, thus, Attacker can learn the identity of the user and can generate attacks. Attacker continuously sends *EAPoL-Start* request making an AP busy with authentication dialog and unable to handle legitimate traffic or Attacker continuously send *EAP-Failure* to supplicant and dissociates legitimate clients or Attacker send *EAPoL-Logoff* request to AP making disassociation from client, which develops into *DoS attack*. [16,18].

When AS receives *EAP-NAK* frame with desired EAP method from supplicant and AS does not support the EAP method, AS sends *EAP-Failure* packet. Attacker spoofs *EAP-NAK* frame and can send fake *EAP-NAK* frame with unsupported EAP method, so that AS will be busy with discarding packets, which develops into *DoS attack*. *EAP-Notification request* message (represented as square bullet in Figure 5) supply some useful information to supplicant during authentication phase they are: expiration time of a password and cautioning authentication failure sent by AS. Attackers can spoof MAC address of AP and can generate fake *EAP-Notification* message. If supplicant is already authenticated and associated to AP then, supplicant became busy with responding these notification messages, thus, *flooding attack* is developed [19].

D. *Solution of attacks in EAP and EAPoL packets used in 802.1x protocol with issues*

The solutions above attacks are summarized below.

Use of EAP Key and secure EAP methods: EAPoL key can be used for indication of successful authentication instead of *EAP-Success*. Secured EAP methods like TTLS or PEAP can be used to protect identity [10]. *Issues*: Both EAP methods are not so secure like EAP-TLS [8,12,20].

Storing AP Identity (ID) and time stamp in EAP packets: APID and timestamp information are stored in *Request-Identity* messages and *EAPoL-Start* respectively. Both supplicant and AP check these fields before accepting packets. Initially, *EAPoL-Start* having length field is zero and do not contain any data field. The time when *EAP-Start* packet is sent, time is recorded in the data field of the packet and length in EAP packet is updated. It is sent with encryption by an encryption function, which is agreed in advance between the supplicant and AP. AP decrypts it by decryption function already agreed with the supplicant, records time data in a table and reply with *Request-Identity* message after adding APID (AP serial ID) and records time in a table with a request to supplicant to send identity information. Supplicant send identity information to AS through AP and communication continues. AP transfers *EAP-Success/EAP-Failure* to supplicant after successful checking of both SSID and start-time stored. When supplicant receives the *EAP-Success* message, it checks both APID information and start-time information stored in the table. If found different (in case of attack), it informs the AP access point with connection exception otherwise, continue with normal communication. Like *EAP-Start*, *EAP-Logoff* is also dealt with similarly [16].

Issues: The solution may require major modifications of protocol and may affect speed of communication among supplicant and authenticator due to burden of checking APID and timestamp before accepting packet.

Using Central Manager (CM): CM can be used to dynamically manage large number of APs and their clients and protect from DoS attacks. All management frames are forwarded to CM by AP. AP does not respond to management frame until CM instructs it. CM maintains authentication and unauthentication status of clients, priority and timestamp information [18]. *Issues*: Forwarding packet to CM by AP and verification of above information in database delay communication.

Wireless IDS (WIDS): WIDS can alert *EAP-NAK*, *EAP-Notification* flooding attacks based on addition of signatures into WIDS database. WIDS can also generate alarm to 802.1x flooding, based on configured rate threshold [21]. *Issues*: Very costly to implement.

Prioritize process of packets based on cost of packet: In [22], the author proposed two solutions. They are: a) each EAP packet has a known cost that indicates the precedence of packet based on sequence in entire protocol flow process. For example, packets with a lower cost, the receiver must process it prior to packets that have a higher cost. By using such an approach, the authentication process between the legitimate supplicant and the AS will usually succeed, where other forged attempts, such as *EAP-NAK* attack, would be eliminated. b) Another approach of processing packet after certain delay of time is proposed. *Issues*: May not fully protect attack and arise delay in accepting or responding legitimate packet.

Using MIC: MIC is used to protect 4-way handshake message (illustrated in Section IV.B). It can also be used protecting unprotected EAP packets [23]. *Issues*: Including MIC, require extra steps to generate keys and its verification, thus, adds burden on client and authenticator.

IV. 802.11I PROTOCOL, ITS ATTACKS AND SOLUTIONS WITH ISSUES

The brief background of 802.11i, its attacks, and solutions with issues are mentioned below.

A Brief idea about 802.11 and 802.11i

In September 1999, original IEEE 802.11 standard ratified to support WEP (Wired Equivalent Privacy) to provide data confidentiality for wireless LAN. Rapid growth of Wi-Fi networks and much vulnerability in WEP demands robust security solution. WEP uses the stream cipher RC4 (Rivest's Cipher 4) for confidentiality and the CRC-32 (Cyclic Redundancy Check) checksum for integrity. Open System Authentication and Shared Key Authentication are two methods used for authentication in WEP. WEP was designed not to meet high level of security demand, but to provide relatively smaller level of data protection. WEP combines its 40/104-bit key with 24 bit Initialization Vector (IV) to encrypt data. IV has its vulnerabilities [24,25]. WEP has no protection against weak keys, which can be used to recover secret key [26], this vulnerability was also successfully implemented by [27]. Attack tools, such as Airsnort [28], Aircrack [29] and WepLab [30] use these vulnerabilities to crack WEP keys. CRC is a linear function that can be used for error detection, but not for data integrity [31,32,33]. Open System Authentication has no importance for authentication and Shared Key Authentication uses challenge/response system that rests on the knowledge of secret shared key. Both challenge and response are sent in clear management frames, which is easily exploited [34]. Several other management and control frames are neither encrypted nor authenticated, thus, MAC address can be spoofed from a legitimate user/node for an attack. Researchers found serious weaknesses in WEP, thus, in 2004, IEEE declared them deprecated as they fail to meet security goals [35,36]. Therefore, attack and solution for WEP in detail is not discussed here.

As WEP has above vulnerabilities, thus, IEEE 802.11 task group i developed a new wireless security standard, IEEE 802.11i by providing initially with Wi-Fi Protected Access (WPA) and its improved version, Wi-Fi Protected Access II (WPA2) protocol also known as RSN (Robust Security Network) on 24th June, 2004. WPA/WPA2 addresses most of the known WEP vulnerabilities. WPA and WPA2 support two authentication modes: WPA-PSK (Pre-shared key) mode (designed for home/office network) and WPA-802.1X mode (designed for enterprise network). WPA uses TLS as default authentication method and WPA2 uses authentication methods (explained in Section III.A), such as TLS, PEAP and TTLS. Two types of encryption protocols used for data confidentiality and message integrity: TKIP (Temporal Key Integrity Protocol) and CCMP (Counter Mode with Cipher Block Chaining Message Authentication Code Protocol) also called AES (Advanced Encryption Standard). CCMP was created to address the vulnerabilities presented by researchers in TKIP. CCMP provides more secure and scalable solution than TKIP. WPA2 though provides higher security, but cause significant CPU overhead. The 802.11i specification defines two classes of security algorithms: Robust Security Network Association (RSNA) and Pre-RSNA. Pre-RSNA security consists of WEP and 802.11 entity authentication. RSNA provides two data confidentiality protocols, called TKIP and CCMP [37,38]. PSK authentication suffers from key management problem for large networks and also prone to attack by breaking keys using coWPAtty [39]. Handshakes of RSNA establishment are represented in Figure 5 and Figure 7 together. RSNA has following six stages [40].

Stage 1. Network and Security Capability Discovery: This stage consists of messages numbered (1) to (3). It is illustrated in Step 1 of Section III.A.

Stage 2. 802.11 Authentication and Association: This stage consists of messages numbered (4) to (7). Station selects one AP from list of available APs and tries to authenticate and associate with that AP. It is illustrated in Step 1 of Section III.A.

Stage 3. EAP/802.1X/RADIUS Authentication: This stage consists of messages numbered (8) to (18). It is illustrated in Step 2-5 of Section III.A.

Stage 4. 4-Way Handshake: This stage consists of messages numbered (1) to (4) illustrated in Section IV.B and represented in Figure 7.

Stage 5. Group Key Handshake: For multicast application, GTK (Group Transient Key) is generated otherwise distributed in Stage 4.

Stage 6. Secure Data Communication: Supplicant and authenticator exchange protected data packets using PTK (or GTK) and negotiated cipher suite from above stages. In 802.11i, supplicant is client or station and authenticator is called access point (AP).

B. *4-way handshake*

The messages of 4-way handshake are conveyed in 802.1x *EAPoL-key* frames. In order to deliver the *EAPoL-Key* frames over a wireless medium, IEEE 802.11 MAC encapsulates each *EAPoL-key* frame in a Data frame [15]. The 4-way handshake [15,40,41,42] occurred in 802.11i is explained below.

The four-way handshake is represented in Figure 7 (Message 1-4). AP sends Message-1 to the client. Message-1 consists *AA* (MAC address of AP), *ANonce* (a random number generated by AP), *SN* (sequence number from AP to prevent from replay attack), and Msg1 (type of message). After receiving Message-1, client generates three values. They are: *SPA* (MAC address of client), *SNonce* (a random number generated by client), and *PTK* (Pairwise Transient Key) from five parameters, i.e. *AA*, *ANonce*, *SPA*, *SNonce*, and *PMK* (illustrated in step 5 of Section III A). PTK is computed as follows:

PTK = prf (PMK, SNonce, ANonce, AA, SPA), where prf is a key derivation function.

Client stores *SNonce* for further reference in Message 3. The PTK is used to calculate Message Integrity Code (MIC), which is used to provide message integrity protection of Message 2. Client sends Message-2 with five parameters, such as *SPA, SNonce, SN, Msg2, and* MIC_{PTK} to authenticator. MIC_{PTK} denotes a MIC value computed over all preceding field values using PTK. Since AP obtains *SNonce* from client, it can also compute PTK same way as client and verify MIC_{PTK}. If verification successful, AP can confirm that client has same PTK. The MIC is a cryptographic digest used to provide the integrity of messages. Therefore, the MIC can be used not only to verify the integrity of the message, but also to make sure that the client has the same *PTK* as the AP. Message-3 is sent by AP similar manner to Message-2. After receiving Message 3, client verifies MIC_{PTK} sent by AP. If verification unsuccessful, client rejects message otherwise, install PTK. Message-4 acts as an acknowledgment of Message-3 without any cryptographic functionality. Message 4 is required to ensure reliability and inform AP that client has installed PTK and ready to receive encrypted data frame. If verification of MIC_{PTK} after receiving Message 4 is successful, AP also installs PTK. Due to requirement of multicast applications, the AP is able to generate and distribute fresh GTK to clients in Message-3. Once, PTK verification in either side is confirmed, It is used to protect data frames.

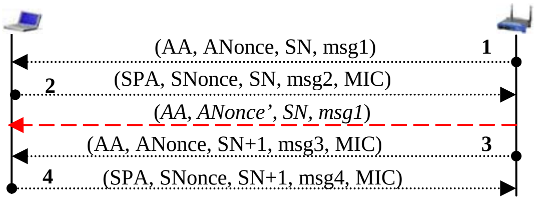

(AA, ANonce, SN, msg1)
1
2
(SPA, SNonce, SN, msg2, MIC)
(AA, ANonce', SN, msg1)
(AA, ANonce, SN+1, msg3, MIC)
3
4
(SPA, SNonce, SN+1, msg4, MIC)

Figure 7: 4-way handshake in 802.11i

C. *Attacks in 802.11i*

Attacks in 802.11 are mentioned below.

RSN IE Poisoning: Unprotected management frames like Beacon, Probe Response and (Re)Association Request (represented in Figure 5) exploited by Attackers to use RSN IE. Authenticator share RSN IE to supplicant and supplicant uses it in 4 way handshake. Supplicant and authenticator use same RSN IE in Message 2 and Message 3 respectively. Authenticator when accept Message 2 from client, it first checks MIC then check RSN IE, where as supplicant check first RSN IE then MIC in Message 3 (client aborts if RSN IE is unmatched). Attacker eavesdrop Beacon frames of legitimate authenticator and modify certain insignificant bits of RSN IE fields, so that it will not affect validity of the frame and selection of authenticator cipher suits. Then, Attacker broadcast this frame to supplicants (poison RSN IE). Communication is not affected till Message 3 is accepted by client. Supplicant verifies forged RSN IE (which it has) with RSN IE sent by authenticator when Message 3 is accepted. The verification will not be successful as it mismatches, thus, 4 way handshake is failed [40].

Vulnerability of Michael Algorithm in MIC: TKIP uses the Michael algorithm to provide MIC protection. It is designed to provide only 20 bits of security, thus, it is possible for an adversary to construct a successful forgery after 2^{19} attempts. In order to prevent this vulnerability, FCS (Frame Check Sequence), ICV (Integrity Check Value), TSC (TKIP Sequence Counter), and MIC are checked sequentially. However, Attacker can intercept a message and can obtain valid TSC value. Keeping the TSC field unchanged, the Attacker can modify some bits of the packet and update the corresponding FCS and ICV fields to make them consistent, due to weakness in the ICV algorithm. The packet can pass the check and Attacker could force a Michael MIC failure in the receiver side, and eventually launch a *DoS attack* [40].

Guessing attack and DoS attack in 4 way handshake: The Message-1 is not protected by PTK, thus, not secure and prone to attack. Some passive wireless network eavesdropping analyzers, such as Kismet and Wireshark can be used to capture 4-way handshake message and generate *guessing attack* [42]. Attacker can forge Message 1 with same AA, SN, Msg1, but different random number ANonce' and sends it to supplicant after Message 2, but before Message 3. Supplicant treats forged Message 1 as retransmission from authenticator. Supplicant generates new PTK value and sends to authenticator, which does not match and discarded by authenticator. Supplicant stores all ANonces and derived PTKs. When supplicant receives valid MIC in Message 3, supplicant installs corresponding PTK and rejects others. Attacker consistently sends forged message to supplicant, which develops into *DoS attack* by performing CPU exhaustion [42,44,45]. This is represented in dash line with single headed arrow in Figure 7.

Failure of 4-way handshake: In order to show failure of 4-way handshake precisely, supplicant is subdivided into an IEEE 802.1x supplicant and IEEE 802.11 MAC, while AP is subdivided into an IEEE 802.1x authenticator and IEEE 802.11 MAC as represented in Figure 8. In Figure 8, Mobile Station (MS) is explained as supplicant here. Suppose, the Message 4 received by authenticator has an invalid MIC or data frame not received by authenticator before *Timeout$_2$* expires (due to noisy or busy wireless channel) then, according to IEEE 802.11 standard [46], authenticator silently discards Message 4 and Message 3 is resent after *Timeout$_2$*. In this case, as supplicant successfully received Message 3 earlier and it has already installed PTK after sending Message 4, thus, expects to receive encrypted data frames from AP. Suppose *Timeout$_2$* of authenticator expires after sending Message 3 to the supplicant then, PTK is not installed in IEEE 802.11 MAC of AP, thus, the data frame containing the second Message 3 is not encrypted. Therefore, the second Message 3 is proven as invalid and discarded by supplicant. Even though Message 3 is repeatedly resent after *Timeout$_2$* expires, it is continuously dropped by supplicant and 4-way handshake is terminated unsuccessfully [15].

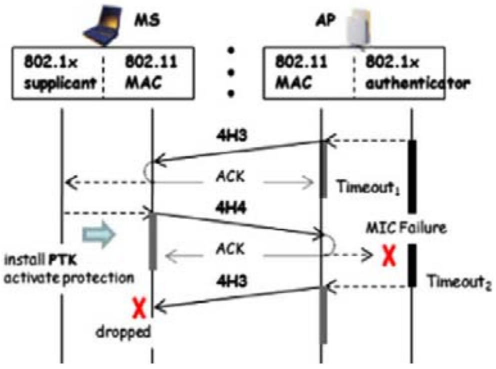


Figure 8: Failure of 4-way handshake [15]

DoS attack due to deauthentication and disassociation frames:

Authentication and association management frames are unprotected and prone to attack. Authentication and association has four states (represented in Figure 9) [47]. They are as follows:

a. State 1: unauthenticated and unassociated.
b. State 2: authenticated and unassociated.
c. State 3: authenticated and associated.
d. State 4: authenticated, associated and 802.1x authenticated.

Following three scenarios illustrated below:

Scenario1: Disasssociation and deauthentication will bring station into State 2 and State 1 respectively though state of client and AP are in State 4. If the Attacker spoofs a deauthentication or a disassociation frame of an AP with a broadcast destination MAC address (i.e. FF:FF:FF:FF:FF:FF), then effectively all clients associated to the AP will be disconnected. This is called *farewell attack* [11,48].

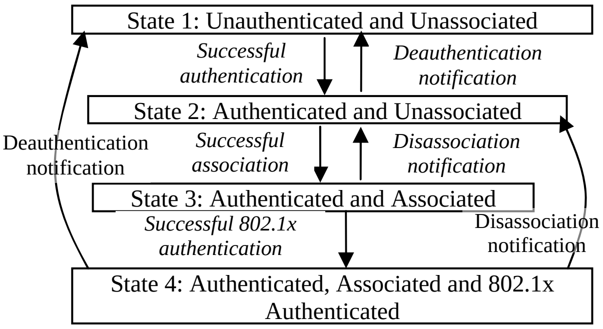

State 1: Unauthenticated and Unassociated
Successful authentication
Deauthentication notification
State 2: Authenticated and Unassociated
Deauthentication notification
Successful association
Disassociation notification
State 3: Authenticated and Associated
Successful 802.1x authentication
Disassociation notification
State 4: Authenticated, Associated and 802.1x Authenticated

Figure 9: Authentication and association in 802.11 before 802.1x authentication

Scenario 2: Single wireless station in 802.11i network make large number of associations using random MAC address to prevent other stations from joining the AP as result, develops *DoS attack* [11].

Scenario 3: Suppose client roams from AP$_1$ to AP$_2$ and performs reassociation with AP$_2$ then, AP$_2$ notifies the layer-2 devices, such as hub, switch or bridge, of a new association between client and AP$_2$. If an Attacker impersonating client and sends a forged reassociation request frame to AP$_1$, it is successfully accepted by AP$_1$, even though the 4-way handshake can not be performed between the Attacker and AP$_1$. However, the data communication meant for legitimate client is redirected from AP$_2$ to AP$_1$ [15].

Attack due to unprotected null frames: Null data frames contain an empty frame body, which is used in communication between wireless client and AP for conserving energy of the client. It carries special control information, such as client informing AP that client is going into sleep state (so that AP can start buffering inbound traffic). Occasionally, client awakes and polls AP for any pending traffic, AP informs client about presence of buffered packets by a broadcast packet called traffic indication map (TIM). Key synchronization information, such as the period of TIM packets and a timestamp broadcast by the AP are sent unauthenticated and clear state. By spoofing polling messages, an Attacker can cause AP to discard the client packet, while it is asleep. If TIM message is spoofed, an Attacker may convince client that no pending data exist and client will revert back to sleep state. By forging timestamp and period of TIM packets, an Attacker can cause a client node to fall out of synchronization with AP and failed to wakeup at appropriate time [49].

Some more *DoS attacks* occur in physical and MAC layer of 802.11 are discussed in [50].

D. Solution of attacks in 802.11i and issues

Solutions of attacks in 802.11i (mentioned above) are illustrated below.

Solution based on factorization problem: In [11], the author applied factorization problem (N=p x q, where p and q are large prime numbers) to develop letter-envelop protocol to defend against *firewall attack*. Initially, client share N1 (N1= p1 x q1) to AP and AP shares N2 (N2 = p2 x q2) to client during authentication process. When deauthentication or disassociation frame is sent by client or AP they share their respective p value to each other and find whether divisible or not. If divisible then, dissociation or deauthentication is accepted. They used ComView software for Wifi in their experiment to capture, modify frame, and generate frame to launch attack. *Issues:* Modifying protocol to send integer value and checking are extra work AP and client has to perform.

2-way handshake solution using loosely synchronized sequence number: In [15], the author proposed 2-way handshake solution to provide protection against *DoS attack* due to forged association and reassociation frames, and to provide faster reassociation during roaming scenario as comparison to existing 4-way handshake in 802.11i. Author proposed two sequence numbers SN_{MS} and SN_{AP}. SN_{MS} is used when reassociation request (RR) is sent by supplicant to AP and SN_{AP} is used when reassociation response (RP) is sent by AP to supplicant. Author used MS instead of SPA and AP instead of AA (SPA and AA illustrated in Section IV.B). Client roams into a new AP and sends RR frame. Supplicant increments SN_{MS} by one, then derives *PTK= prf* (*PMK, SN_{MS}, AP, MS*). The 802.11 MAC of the client sends an RR frame to 802.11 MAC of the AP, consisting of (*msg1, PMKID, SN_{MS}, MIC_{PTK}*), where *msg1* denotes list of parameters i.e service set identifier (SSID), listen interval and current AP etc. (PMKID is illustrated in Step 5 of Section III.A). RR frame is protected by MIC_{PTK}. Authenticator verifies MIC of RR frame by above function for generating PTK. If verification is unsuccessful, authenticator rejects frame otherwise, further checks whether SN_{MS} greater than SN_{AP}. If SN_{MS} is greater than SN_{AP}, status code is set to “success” (means, reassociation requested by client is successful) or it is set to “SN-fail” (means reassociation requested by client is unsuccessful). *Issues*: The proposed solution requires modification of 802.11 protocol. No solution has been provided during initial association of client with AP, where unprotected Message 1 in 4-way handshake is still used.

Updating TSC: In [40], the author has proposed solution of *DoS attack* and *RSNIE poisoning*. *DoS attack* due to failure of Michael algorithm in TKIP is mitigated if TSC could be updated once a packet passes the check of FCS, ICV, and TSC even if the Michael MIC failure occurs. Once two failures are detected within 60 seconds, the transmission and reception will cease for 60 seconds. They also suggested that there is no need to use fields in RSN IE having insignificant bits. *Issues*: This requires extra modification in checking design of protocol, thus, may develop delay in communication.

Asymmetric cryptography approach: In [42], the author proposed improved authentication mechanism, which adopts an asymmetric cryptography approach to accomplish effective protection for management frames, null data frames, and EAPoL frames as well as protection from *DoS attacks* and *offline guessing attacks*. This authentication mechanism installs a pair of public-private keys, which can be achieved by using Certificate-Authentication-signed (CA-signed) certificate or a self-signed certificate. *Issues*: AP verifies timestamp and MAC address of client in list of registered station before, responding to client. Registering MAC address is an administrative burden and also MAC address can be spoofed easily.

2-way handshake solution using temporary PTK: In [43], the author proposed 2-way handshake protocol solution for *CPU exhaustion attack* and *DoS attack*. As Message 1 serves as both Message 1 and Message 3 in original 4-way handshake protocol, they suggested that it can be removed, so that computational load of CPU will be less. Acknowledgement role of Message 4 can be replaced by timer method. They use insertion of new authentication information to protect Message 1. *Issues*: Solution requires major modification in message format for keeping new encryption, require too many changes in hardware, can not mitigate *DoS flooding attack* as supplicant has to verify encryption field for each received message 1. Thus, in [44], the author proposed solution, which does not require too many change in current message format and hardware. By protecting Message 1 using temporary PTK value, which is calculated based on only *ANonce,* PMK, supplicant MAC address (SPA) collected in early state of authentication process. Once, supplicant receives, it makes same calculation and confirm its legitimacy. If verification is successful, Message 3-4 steps are followed as like in original 4-way handshake. To avoid deadlock situation raised by DoS attack, they provided following solutions: a) client blocks port after receiving Message 1 and b) client starts timer after sending Message 2 and expect to receive Message 3 after timer expires. If Message 3 received before timeout, it continues to send Message 4 otherwise, it reopen port (which it has blocked after receiving Message 1) to receive Message 1. *Issues*: As dissociation and deauthentication packets still unprotected, thus, the *DoS attack* can be generated by Attacker [35].

In [47], the author had summarized several other solutions to *DoS attack* due to dissociation or deauthentication frames, such as queue approach, Reverse Address Resolution Protocol (RARP) approach, sequence number based detection, authentication before association, and lightweight authentication.

Some more countermeasures against DoS attacks occur in physical and MAC layer of 802.11 are discussed in [50].

Authenticating ID (identity) by message: In [51], the author proposed for ID authentication encrypted by MIC for Message 1 in 4-way handshake. They proposed to use same PMK generated after successful 802.1x authentication with ANonce. They proposed only modification in Message 1, not whole protocol and encrypted ANonce in Message 1. Thus, Message 1 is protected. *Issues*: a) The author has not explained how client collect correct ANonce before accepting encrypted ANonce in Message 1 for its verification, b) vulnerability of Michel algorithm in MIC [40], and c) mechanism to protect deauthentication and disassociation frame is not addressed.

V. 802.11w Protocol, Its Attacks and Solutions with Issues

Brief idea about 802.11w, its attacks and solutions with issues are mentioned below.

A. *Brief idea about 802.11w*

Brief idea about 802.11w protocol, its attacks and solutions are mentioned below.

On 30th September 2009, IEEE 802.11w was ratified as an amendment to the 802.11i. 802.11w provides a mechanism to protect the management frames [52]. The 802.11w uses BIP (Broadcast/Multicast Integrity Protocol) to provide integrity to broadcast/multicast frames. In 802.11w, CCMP's protection scope is expanded to the unicast management frames [35]. The IEEE 802.11w provides protection in three categories: 1) protection for unicast management frames, 2) generic broadcast management frames, and 3) de-authentication/ disassociation frames. The security offered by 802.11w can not protect attacks occurs before EAP/802.1x/RADIUS authentication and 4-way handshake. It also does not support any protection scheme for null data frames [42]. Association and reassociation frames are not protected by 802.11w since these frames are exchanged prior to the establishment of the session key PTK between client and AP [15].

B. *Attacks in 802.11w*

The two scenarios of *4-way handshake blocking attack* [35] are mentioned below.

Scenario-1: Attacker is capable of forging the de-authentication/authentication frames and sending it to the client (wireless station) just after the message 1 of the Four-Way Handshake (explained in Section IV.B). Since it will take some times to calculate PTK and MIC after receiving the message 1, the client will receive the forged deauthentication or authentication frame before sending message 2. Then, client will be tricked to terminate the association or authentication and the message 2 would not be send, which will essentially reset the connection.

Scenario-2: Attacker spoofs Message 1 of 4-way handshake and send Message 1 to supplicant by only changing ANonce before client prepare PTK and MIC, thus, forced to generate new PTK to replace correct one and recalculate MIC for Message 2. When AP receives Message 2 with wrong MIC, it discards, thus, handshake could not compete and successful *DoS attack* occurs.

C. *Solution of attacks in 802.11w and issues*

The solutions above attacks are summarized below.

Solution using encryption technique: In [35], the author proposed Temporary Safe Tunnel (TST) to provide solution of 4-way handshake blocking attack. TST protects the process of authentication and association. TST key (TSTK) is a one-time pad used to encrypt next frame. Different frames are protected by different TST keys (TSTK). TST calculates an integrity check value (ICV) for every frame to provide an integrity protection. The mixture of TSTK and original PMK prevents the attack. *Issues*: The solution proposed by author is not supported by any experimental analysis. Therefore, cost of a) calculating and checking the ICV, b) cost of process of matching the TST entity (TSTE) by MAC address, and c) the encryption and decryption in the solution proposed are not evaluated.

Solution using cryptographic mechanism: In [53], the author proposed solution by implementing 802.11w cryptographic mechanism, such as MD5 hashing, RC4 encryption, and AES encryption. Author also has used integrated approach to apply traffic pattern filtering (TPF) over 802.11w. They added two information elements (IE), such as crypto IE to hold cryptographic information and timestamp IE to prevent reply attack in deauthentication and disassociation frames. Their simulation based experimental analysis using NS2 and DoS attacking tool using void 11 shows that the proposed approach is able to prevent DoS attacks. *Issues*: The process generates more computational load for checking timestamp and hashing result.

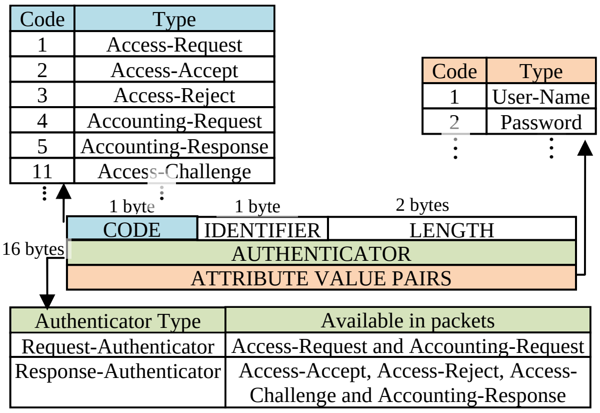
VI. RADIUS Protocol, Its Attacks and Solutions with Issues

The brief idea about RADIUS communication, its attack and solutions with issues are mentioned below.

A. *RADIUS communication*

RADIUS protocol is used in EAPoR frame. RADIUS frame is represented in Figure 10. Out of five fields in RADIUS packet [54], following three fields are relevant from attack point of view. They are as follows:

1. *Code*: It establishes the type of RADIUS packet. Out of value and attribute pairs, those relevant to attack are: 1: Access-Request, 2: Access-Accept, and 3: Access-Reject.
2. *Authenticator*: It is used to authenticate the reply from the RADIUS server. It is used to encrypt passwords. Its length is 128 bits.
3. *Attributes*: The only relevant attributes discussed here are, the User-Name and User-Password attributes.



Code	Type
1	Access-Request
2	Access-Accept
3	Access-Reject
4	Accounting-Request
5	Accounting-Response
11	Access-Challenge
⋮	⋮

Code	Type
1	User-Name
2	Password
⋮	⋮

CODE	IDENTIFIER	LENGTH
AUTHENTICATOR		
ATTRIBUTE VALUE PAIRS		

Authenticator Type	Available in packets
Request-Authenticator	Access-Request and Accounting-Request
Response-Authenticator	Access-Accept, Access-Reject, Access-Challenge and Accounting-Response

Figure 10: RADIUS packet format

The RADIUS communication between supplicant (or client) and AS [55] is briefly explained below. Supplicant sends *Access-Request* RADIUS packet with User-Name and User-Password attributes to RADIUS server. RADIUS server sends *Access-Accept* or *Access-Reject* packet to supplicant if authentication successful or failed respectively. Both packets use the same identifier value from the supplicant's *Access-Request* packet, and put a *Response Authenticator* in the *Authenticator* field.

B. Attack in RADIUS protocol

Attack in RADIUS protocol [55] is explained as follows. The *Response Authenticator* is essentially an MD5 based keyed hash. Attacker tries to authenticate to the client with a known password. The Attacker captures the resulting *Access-Request* packet and XORs the protected portion of the User-Password attribute with the user name and password he or she provided to client. This provides results of MD5 (Shared Secret + *Request Authenticator*) operation. The *Request Authenticator* is known so the Attacker can replay modified *Access-Request* packet using same *Request Authenticator* and MD5 value with only changing the password, which allow Attacker to efficiently perform an exhaustive search for correct user password. It develops into an *off-line exhaustive attack.*

C. Solutions of attack in RADIUS protocol and issues

In [55], the author has suggested following solutions.

1. Author stated that for compatibility reasons, the block cipher would not be keyed independently from the shared secret. Thus, author suggested to key the cipher from some derived value of the shared secret and the *Request Authenticator*. Author also suggested that cipher could be keyed from the output of an HMAC (Hash-based Message Authentication Code) of the *Request Authenticator* (the HMAC is keyed by the shared secret) or by seeding a cryptographic PRNG (pseudo-random number generator) with the shared secret and the *Request Authenticator*.
2. Looking into attack with MD5 cryptographic hash, author suggested to use SHA-1 (Secure Hash Algorithm) instead of MD5 for HMACs.
3. In order to protect the *Access-Request, Access-Accept,* and *Access-Deny* packets, author suggested a new attribute. This attribute should be created which contains a SHA-1-HMAC of the entire RADIUS packet. If this attribute is present, the receiving client or server should compute the HMAC for the entire RADIUS packet and verify that the result is the same as the stored HMAC in packet. If the result is not the same, the packet should be discarded.
4. Author also suggested that the RADIUS specification should require a strong cryptographic PRNG for generation of the *Request-Authenticator* and each RADIUS client should use a different Shared Secret. It should also require the shared secret to be a random bit string at least 128 bit long that was generated by a strong cryptographic PRNG.
5. Author also suggested to use DIAMETER instead of RADIUS to eliminate many vulnerability of RADIUS. The comparative study between RADIUS and DIAMETER, and how DIAMETER is superior than RADIUS are illustrated in [56,57].

Issues: Using cryptographic mechanisms may require modification in frame and also affect response time of server, which can be validated after proper experimental analysis.

VII. SUMMARY OF SOLUTIONS AND ISSUES OF ATTACKS

Table I represents the summery of solution for attacks in 802.1x, EAP, EAPoL, 802.11i, 802.11w, and RADIUS with their issues.

TABLE I. SUMMERY OF SOLUTION OF ATTACKS AND THEIR ISSUES

Solution	Issues
Solution of attacks in 802.1x protocol and their issues	
Modification of EAP payload by adding extra fields for identification to protect *MITM attack* [7].	Solution may fail if IP and MAC address can be spoofed and changed by Attacker [11].
Addition of *EAP Authenticator* in *EAP-Success* frame like *Request Authenticator* in RADIUS packet [8].	Require major modification in EAP frame. Supplicant has to perform extra steps like decryption and verification, which may develop delay in communication.
Building symmetrical relationship to protect *user name embezzlement* and *user name lift* [10].	Require major changes in 802.1x architecture.
Solution of attacks in EAP and EAPoL protocols and their issues	
Secured EAP methods like TTLS or PEAP can be used to protect identity [10].	Both EAP methods are not so secure like EAP-TLS [8,12,20]
Storing AP Identity (ID) and time stamp in EAP packets to protect *DoS attack* [16].	It requires modification of several EAP frames. Due to burden of checking APID and time stamp, it may affect speed of communication.
All frames are routed through CM to protect *DoS attack* [18].	Forwarding packet and verification make delay in network communication.
Use of Wireless IDS to alert about attack [21]	It is very costly to implement.
Prioritize process of packet based on cost of packet to protect from *flooding attack* [22].	May not fully protect attack and delay replying legitimate packet.
MIC can be used for protection of EAP packets [23].	It requires extra steps to generate keys and its verification, thus, extra burden both on client and authenticator.
Solution of attacks in 802.11i protocol and their issues	
Solution based on factorization problem to solve *DoS attack* [11].	Modifying protocol to send integer value and checking are extra work AP and client has to perform, which may slow down communication.
2-way handshake solution using loosely synchronized sequence number to protect reassociation packets and faster handover during roaming scenario [15].	It requires modification of 802.11 protocol. No solution has been provided during initial association of client with AP, where unprotected Message 1 in 4-way handshake is still used.
Updating TSC to mitigate RSNIE poisoning and *DoS attack* [40].	This requires extra modification in checking design of protocol, thus, may develop delay in communication.

Asymmetric cryptography approach to protect for management frames, null data frames, *DoS* and offline *guessing attack* [42].	Registering MAC address is an administrative burden and also MAC address can be spoofed easily.
2-way handshake solution using temporary PTK for *CPU exhaustion attack* and *DoS attack* [43].	It requires major modification in message format and changes in hardware, thus, temporary PTK is proposed to protect Message 1 [44]. However, unprotected dissociation and deauthentication packets can also generate *DoS attack* [35].
Authenticating ID (identity) by message using PMK generated in end of 802.1x authentication and encrypted ANonce [51].	a) The author has not explained how client collect correct ANonce before accepting encrypted ANonce in Message 1 for its verification, b) vulnerability of Michel algorithm in MIC [40], and mechanism to protect deauthentication and disassociation frame is not addressed.
Solution of attacks in 802.11w protocol and their issues	
Using encryption technique to protect *DoS attack* [35].	The solution proposed by author is not supported by any experimental analysis.
Cryptographic mechanism to protect *DoS attack* [53].	It generates more computational load for checking timestamp and hashing result.
Solution of attack in RADIUS protocol and their issues	
Solutions such as use of SHA-1 and cryptographic PRNG [55].	May affect response time of server, which yet to be validated after proper experimental analysis.

VIII. HOW NAC IS AFFECTED

Before understanding how above attacks affect network access control (NAC), understanding basic communication flow in generic NAC is necessary. Basic communication flow in generic NAC, affect of NAC due to above attacks and summary are mentioned below.

A. Basic communication flow in generic NAC

When a new host (or endpoint) tries to connect to the network, NAC solution identifies its presence by certain detection techniques, such as DHCP proxy, broadcast listener, sniffing to IP traffic, client-based NAC software, SNMP traps, and enable authentication procedure based on 802.1x (mentioned in Section II.A). Both the user and the machine authentication should be performed in order to identify authorized users and machines. Every endpoint have NAC agent, which collect user authentication and machine integrity related information and provide to verifier in NAC. After successful authentication of the user and the machine, NAC performs set of security checks by gathering knowledge regarding endpoint's operating system, the list of installed patches, the presence of antivirus, and antispam software and their patches, signature date etc. (i.e integrity measures are checked). If authentication is not successful, switch port is blocked and no further integrity checking occurs because user is not authorized to access network. If compliant endpoints are infected by virus and spam, it can affect other endpoints of network. Therefore, integrity checking for compliant endpoints is important part of organization's security policy. There can be several rounds of message communication occurs from endpoint to NAC during course of gathering above information (machine integrity checking). Once information is collected, they are sent to verifier for taking policy decision whether the endpoint is compliant or noncompliant to policies already set (also called endpoint security assessment). For example, verifier checks whether the endpoint has latest version and patches of operating system and antivirus etc. If policies are not met, endpoint is declared noncompliant otherwise, NAC still performs security checks periodically on the activities or behavior shown by user and the machine. NAC declare the endpoint noncompliant when violation of policy detected. If the client is failed in machine integrity checking of NAC, then procedure should be followed to increase health status of the client until it is fully compliant with organization's policy. This is called remediation of the client.

Verifier sends instruction to noncompliant endpoint, how it should connect for a remediation process. Since, noncompliant endpoint is considered to be at risk, thus, the machine is placed on isolated network (or quarantined) or provided with very limited access to network resources in order to protect compliant endpoints from threats or vulnerabilities that it may introduce. One of the solutions for non compliant computers is, they should be kept under separate VLAN (Virtual LAN) called remediation VLAN. In remediation VLAN, servers like antivirus server, windows server update service (WSUS) etc. should be kept, so that clients can communicate and download patches and updates. After successful remediation, client can request again to NAC to connect to network. Noncompliant endpoints (working in remediation VLAN) fix their compliance issues by installing operating system (OS), antivirus patches, and updates. They may connect to remediation servers (also part of remediation and production VLAN), which allow a user to remedy the issues that prevented it from being allowed on the network. It then, installs the appropriate software stored in remediation servers. Once they became compliant, they again send request to NAC to work under production VLAN [58]. The basic communication flow in generic NAC is represented in Figure 11, which is adopted by three representative solutions of NAC i.e CNAC [59], NAP [60], and TNC [61].

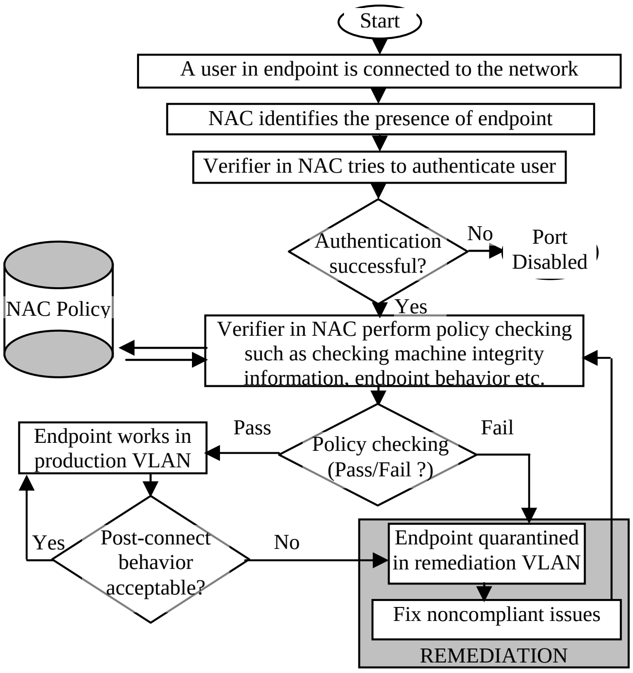

Start
A user in endpoint is connected to the network
NAC identifies the presence of endpoint
Verifier in NAC tries to authenticate user
Authentication successful?
No
Port Disabled
Yes
NAC Policy
Verifier in NAC perform policy checking such as checking machine integrity information, endpoint behavior etc.
Pass
Fail
Endpoint works in production VLAN
Policy checking (Pass/Fail ?)
Yes
Post-connect behavior acceptable?
No
Endpoint quarantined in remediation VLAN
Fix noncompliant issues
REMEDIATION

Figure 11: Flowchart represents basic communication flow in generic NAC

The NAC assessment, taking decision, enforcement, and remediation are four basic tasks of all NAC solutions. In a generic NAC architecture, several major tasks (also called roles) and miner tasks (also called functions) are involved to complete these four basic tasks. TNC is most acceptable as comparison to NAP and CNAC [62,63]. Comparison of three frameworks for NAC is available in [62,64,65,66,67]. One of most acceptable NAC is TCG's TNC. In [68], several roles and functions of TNC are discussed in detail.

B. *Affect of NAC due to attack*

Above attacks (mentioned in Section II-VI) immensely affect network access control. It also severely affects continuous assessment of endpoint, taking policy decision, enforcement for noncompliant clients, and remediating endpoints. The affect of above attacks in NAC are mentioned below.

1. NAC agents, such as CNAC agent in Cisco NAC, NAP agent in Microsoft's NAP, and TNC client in TCG's TNC etc. can not communicate with centralized verifier (NAC device) due to choke of network bandwidth or resource exhaustion by *DoS* and *flooding attacks* occurred in EAP, EAPoL, 802.11i, and 802.11w protocols. Therefore, user authentication and machine integrity related information could not be reached to verifier and verifier can not take appropriate policy decision.
2. Generation of more attack from endpoint (or client) may develop noncompliant status looking into strict NAC policy in an organization, thus, client is isolated from network to protect healthy compliant endpoints.
3. The weak hashing algorithm in RADIUS protocol may expose user credential information, such as user account and password, which can be spoofed easily by Attacker and can make a compliant endpoint isolated from network based on fake or spoofed data. Before completion of 802.1x authentication, user credentials shared to authentication server (AS) through EAPoR protocol. The weak hashing algorithm used in RADIUS protocol (illustrated in Section VI) may expose user credentials to Attacker and Attacker can make authentication unsuccessful consistently, thus, endpoint will be isolated from network (as unsuccessful authentication cause switch port to be blocked due to 802.1x methodology).
4. The unprotected packets may expose machine integrity information, such as IP address, MAC address etc., which can be spoofed easily by Attacker and can make a compliant endpoint into noncompliant based on fake or spoofed data even though endpoint succeeded in 802.1x authentication. IP address and MAC address can be easily known to Attacker by spoofing any unprotected EAP or EAPoL or 802.11 or 802.11w packet discussed in Section II-V. Attacker may modify IP address and MAC address to IP address and MAC address of any noncompliant endpoint, so that the legitimate endpoint can be declared noncompliant by NAC and forced to stay in remediation VLAN (explained in Section VIII.A).
5. Unprotected Message 1 in 4-way handshake in 802.11i (discussed in Section IV.C) may develop *DoS attack* and NAC agent in an endpoint can not provide machine integrity information to verifier (a NAC device) though endpoint is successful in 802.1x authentication. In this case, endpoint is virtually isolated from network. Unprotected disassociation and deauthentication frame (Discussed in Section IV.C) can also detach a wireless endpoint at any state (represented in Figure 9), which make endpoint to again start association or authentication steps and 802.1x authentication steps to send machine integrity information. By this way, the endpoint will struggle to send machine integrity information to verifier and due to policy decision of NAC, the endpoint may be send into remediation VLAN after several attempts.
6. Attacker can know user credential information, such as user account, password etc. by spoofing RADIUS packets and machine identity information, such as IP address, MAC address etc. by spoofing unprotected EAP, EAPoL, 802.11, 802.11w packets. Then, Attacker can deceive NAC by using spoofed user credentials and providing machine identity information of a compliant endpoint to bypass a noncompliant endpoint. Attacker can also bypass NAC easily by putting a hub instead of switch (as mentioned in Section II.B). Once, a noncompliant endpoint intrudes into production VLAN by an Attacker, it can make damage into the compliant endpoints and servers in network easily. This damage can be making compliant endpoints noncompliant or choke the network bandwidth etc.

C. *Summary*

From solution mentioned in Section II-VI, it is clear that the frames, such as 802.1x, EAP, EAPoL, 802.11i, 802.11w, and RADIUS should be modified to provide higher security. In NAC, maintaining security is equally important like providing faster network access. Today, most of endpoints use antivirus and anti-spam and gateway devices use gateway antivirus, anti-spam, IDS, IPS and content filtering agents etc. Still network is found unprotected, thus, requirement of NAC is felt essential by researchers. Until security of NAC is enhanced by adopting secure protocols, basic purpose of NAC will not be fulfilled and NAC may not succeed in its goal.

IX. Conclusion

We have presented vulnerabilities, attacks and solutions with issues on 802.1x, EAP, EAPoL, 802.11, 802.11w, and RADIUS protocols. Howsoever enforcement policy occurs in NAC to force endpoints complaint with organization's policy, enforcement can not be fulfilled until unprotected management and control frames discussed in this paper are secured properly. The affect of vulnerability and attacks of above protocols in NAC are also discussed. The solution to above design flaws in protocols should balance minimum change in frame and hardware with higher response time for network access by client. The discussed solution approaches by researchers may help NAC researchers to build secure NAC.

References

[1] Yabin, L., Huanguo, Z., Liqiang, Z., and Bo, Z. (2009). Research on Unified Network Access Control Architecture. *International Conference on Computer and Information Technology*. (Oct. 2009), pp. 295-299.

[2] Chiornita, A., Gheorghe, L., and Rosner, D. (2010). A practical analysis of EAP authentication methods. *Roedunet International Conference (RoEduNet)*. (June 2010), pp. 31-35.

[3] Rigney C. and et. al. Remote Authentication Dial In User Service (RADIUS). *RFC 2138*, April 1997.

[4] Serrao, G. J. (2010). Network access control (NAC): An open source analysis of architectures and requirements. *Security Technology (ICCST). IEEE International Carnahan Conference.*, (Oct. 2010), pp. 94-102.

[5] Wang, J., and Wu, Z. (2009). A New Model for Continuous Network Access Control of Trusted Network Connection. *5th International Conference on Wireless Communications, Networking and Mobile Computing*, 2009. WiCom '09. (Sep. 2009), pp. 24-26.

[6] Calhoun P., Loughney J., Guttman E., Zorn G., and Arkko J. Diameter base protocol. Internet Engineering Task Force, (Sep. 2003). Request for Comments (RFC) 3588.

[7] Qian, Q., Li, C., and Zhang, X. (2010). On Authentication System Based on 802.1X Protocol in LAN. *International Conference on Internet Technology and Applications*. (Aug. 2010), pp. 1-4.

[8] Arbaugh, W. and Mishra, A. A. (2002). An Initial Security Analysis of the 802.1X Standard. (Feb. 2002). pp. 1-12. URL: http://www.cs.umd.edu/%7Ewaa /1x.pdf.

[9] IEEE 802.1x, IEEE Standards for Local and Metropolitan Area Networks: Port based Network Access Control, IEEE Standard, 2001.

[10] Ali, K. M., and Owens, T. J. (2010). Access mechanisms in Wi-Fi networks state of art, flaws and proposed solutions. *17th IEEE International Conference on Telecommunications (ICT).* (Apr. 2010) pp. 280-287.

[11] Nguyen, T. D., Nguyen, D. H. M., Tran, B. N., Vu, H., and Mittal, N. (2008). A Lightweight Solution for Defending Against Deauthentication/Disassociation Attacks on 802.11 Networks. *Proceedings of 17th International Conference Computer Communications and Networks*, 2008. ICCCN '08. (Aug. 2008), pp. 1-6.

[12] Malekzadeh, M., Azim, A., Ghani, A., Desa, J., and Subramaniam, S. (2009). Vulnerability Analysis of Extensible Authentication Protocol (EAP) DoS Attack over Wireless Networks. *ICGST International Journal on Computer Network and Internet Research CNIR*. vol. 9, (July 2009), pp. 39-46.

[13] Matthew, G. (2002). 802.11 Wireless Networks: The Definitive Guide. O'Reilly. pp. 1-436.

[14] Dantu, R., Clothier, G., and Atri, A. (2007). EAP methods for wireless networks, *Computer Standards & Interfaces*, vol. 29, issue 3, (Mar. 2007), pp. 289-301.

[15] Park, C. (2010) Two-way Handshake protocol for improved security in IEEE 802.11 wireless LANs. *Computer Communications*, vol. 33, Issue 9, (June 2010), pp. 1133-1140.

[16] Kong, F. and Huang, W. (2010). IEEE 802.1x of protocol analysis and improvement. *3rd International Conference on Advanced Computer Theory and Engineering (ICACTE)*. (Aug. 2010), pp. V3-282-V3-285.

[17] Zrelli, S., and Shinoda, Y. (2007). Specifying Kerberos over EAP: Towards an integrated network access and Kerberos single sign-on process, *21st International Conference on Advanced Information Networking and Applications*, AINA '07., (May 2007) pp. 490-497.

[18] Ding, P. Q., Holliday, J. N., and Celik, A. (2004). Improving the security of wireless LANs by managing 802.1x disassociation. *Consumer Communications and Networking Conference (CCNC)*. (Jan. 2004), pp. 53- 58.

[19] Alruban, A. and Everitt, E. (2011). Two Novel 802.1x Denial of Service Attacks. *Intelligence and Security Informatics Conference (EISIC)*, 2011 European (Sept. 2011), pp. 183-190.

[20] Bhakti, M. A. C., Abdullah, A. J., and Jung, L. T. (2007). EAP-based Authentication with EAP Method Selection Mechanism: Simulation Design. *5th Student Conference on Research and Development.* (Dec. 2007), pp. 1-4.

[21] Phifer, L. (2006). Fighting wireless DoS attacks. URL:http://searchnetworking.techtarget.com/generic/0,295582,sid7_gci1169024,00.html.

[22] Zhao, Y., Vemuri, S., Chen, J., Chen, Y., Zhou, H., and Fu, Z. (2009). Exception triggered DoS attacks on wireless networks. *Dependable Systems & Networks. DSN '09*. IEEE/IFIP International, Lisbon.

[23] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and Levkowetz. H. (2004). Extensible Authentication Protocol (EAP). URL: http://tools.ietf.org/html/rfc3748.

[24] Walker, J. (2000). Unsafe At Any Key Size: An Analysis of the WEP Encapsulation, tech. report 03628E, IEEE 802.11 Committee, (Mar. 2000).

[25] Borisov, N., Goldberg, I., and Wagner, D. (2001). Intercepting mobile communications: The insecurity of 802.11. *In Proc. of the 7th Annual ACMIIEEE International Conf on Mobile Computing and Networking* - Mobicom'01, Rome, Italy, (July 2001), pp. 180-189.

[26] Fluhrer, S., Mantin, l., and Shamir, A. (2001). Weaknesses in the key scheduling algorithm of RC4. *The 8th Annual International Workshop on Selected Areas in Cryptography*, pp. 1-24.

[27] Stubblefield, A., Ioannidis, J., and Rubin, A. (2001). Using the Fluhrer, Mantin, and Shamir Attack to Break WEP. *Technical Report TD-4ZCPZZ, AT&T Labs*.

[28] Airsnort: airsnort.shmoo.com.

[29] Aircrack: www.cr0.net:8040/code/network/.

[30] WepLab: weplab.sourceforge.net/.

[31] Jueneman, R., Matyas, S., and Meyer, C. (1985). Message authentication. IEEE Comm. Magazine, 23(9), (Sept. 1985), pp. 29-40.

[32] Stubblebine S. G. and Gligor V. D. (1992) On message integrity in cryptographic protocols. *In Proc. of the IEEE Symposium on Research in Security and Privacy*, pp. 85-105.

[33] Core SDI. CRC32 compensation attack against ssh-1.5, (July 1998). Website: www.coresdi.com.

[34] Arbaugh, W. A., Shankar, N. and Wang J. (2001) Your 802.11 Network has no Clothes. *In Proc. of the First IEEE International Conf on Wireless LANs and Home Networks*, (Dec. 2001) pp. 131-144.

[35] Wang, W., and Wang, H. (2011). Weakness in 802.11w and an improved mechanism on protection of management frame. *International Conference on Wireless Communications and Signal Processing (WCSP)*, (Nov 2011), pp. 1-4.

[36] URL: http://en.wikipedia.org/wiki/Wired_ Equivalent _Privacy.

[37] URL: http://en.wikipedia.org/wiki/IEEE_ 802.11i-2004.

[38] URL: http://en.wikipedia.org/wiki/WiFi_ Protected _Access.

[39] coWPAtty: new.remote-exploit.org/

[40] He, C., and Mitchell, J. C. (2005). Security analysis and improvements for IEEE 802.11i. *The 12th Annual Network and Distributed System Security Symposium*, pp. 1-20.

[41] Eum, S., Cho, S., Choi, H., and Choo, H. (2008). A Robust Session Key Distribution in 802.11i. *International Conference on Computational Sciences and Its Applications*, ICCSA '08. (June 2008), pp. 201-206.

[42] Xing, X., Shakshuki, E., Benoit, D., and Sheltami, T. (2008). Security Analysis and Authentication Improvement for IEEE 802.11i Specification. *Global Telecommunications Conference*, IEEE GLOBECOM (Nov. 2008), pp. 1-5.

[43] Liu, J., Ye, X., Zhang J. and Li, J. (2008). Security verification of 802.11i 4-way handshake protocol. *Proceedings of the IEEE International Conference on Communications*, pp. 1642-1647.

[44] Xiaodong, Z., and Maode, M. (2010). Security improvements of IEEE 802.11i, 4-way handshake scheme, *International Conference on Communication Systems (ICCS)*, IEEE, (Nov. 2010), pp. 667-671.

[45] He, C. and Mitchell C. (2004) Analysis of the 802.11i 4-way Handshake, *In Proceedings of the ACM Workshop on Wireless Security*, Philadelphia, PA, USA, (Oct. 2004), pp. 43–50.

[46] IEEE 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, IEEE Standard, (2007).

[47] Aslam, B., Islam, M., H., and Khan, S. A. (2006). 802.11 Disassociation DoS Attack and Its Solutions: A Survey. *The First Proceedings of Mobile Computing and Wireless Communication (MCWC) International Conference*. (Sept. 2006), pp. 221-226.

[48] Wang, L. and Srinivasan, B., (2010). Analysis and Improvements over DoS Attacks against IEEE 80.11i Standard, *second International Conference on Network Security, Wireless Communications and Trusted Computing*, pp. 109-113.

[49] Bellardo, J. and Savage S. 802.11 Denial-of-Service attacks: real vulnerabilities and practical solutions. *In Proc. of the USENIX Security Symposium*, (Aug. 2003), pp. 15-28.

[50] Bicakci K., and Tavli, B. (2009). Denial-of-Service attacks and countermeasures in IEEE 802.11 wireless networks, *Computer Standards & Interfaces*, vol. 31, issue 5, (Sep. 2009), pp. 931-941.

[51] Zhao S., Shoniregun, C. A., and Imafidon, C., (2008). Addressing the vulnerability of the 4-way handshake of 802.11i, *Third International Conference on Digital Information Management, ICDIM 2008, (Nov. 2008)*, pp. 351-356.

[52] IEEE Standard 802.11w-2007, IEEE-SA Standards Board, (Sep. 2009).

[53] Liu., C. and Yu, J. (2008). Rogue Access Point Based DoS Attacks against 802.11 WLANs. *Fourth Advanced International Conference on Telecommunications, AICT '08*. (June 2008), pp. 271-276.

[54] URL: http://en.wikipedia.org/wiki/RADIUS

[55] Hill, J. (2001). An Analysis of the RADIUS Authentication Protocol.

URL:http://www.untruth.org/~josh/security/radius/radius-auth.html.

[56] Hosia, A. (2003). Comparison between RADIUS and Diameter”, Helsinki University of Technology, Telecommunications Software and Multimedia Laboratory,pp. 1-15. (www.tml.tkk.fi/Studies/ T110.551/2003/papers/11.pdf).

[57] López, G., Cánovas, O., Gómez, A. F., Jiménez, J. D., and Marín R., (2007). A network access control approach based on the AAA architecture and authorization attributes, *Journal of Network and Computer Applications*, vol. 30, issue 3, (Aug. 2007), pp. 900-919.

[58] Bypassing Network Access Control Systems. URL: http://www.blackhat.com/presentations/bh-dc-07/Arkin/Paper/bh-dc-07-Arkin-WP.pdf.

[59] NAC Architecture. URL: http://www.cisco.com/en /US/netsol/ns466/networking_solutions_package.html.

[60] NAP Architecture. URL: http://www.microsoft.com/ technet/network/nap/ naparch.mspx.

[61] TCG Specification Trusted Network Connect -TNC Architecture for Interoperability Revision 1.3, Trusted Computing Group, (2008). URL: http://www.trustedcomputinggroup.org.

[62] What's Up With NAC Standards? http://www.wwpi.com/index.php?option=com_content&view=article&catid=99:coverstory&id=6893:whats-up-with-nac-standards&Itemid=2701018.

[63] IETA NEA. URL: http://www.ietf.org/dyn/wg/ charter/nea-charter.

[64] Network Access Control Technologies. URL: http://www.opswat.com/sites/default/files/ Network_ Access_Control_Technologies.pdf.

[65] Tutorial: Network Access Control (NAC). URL: http://www.networkcomputing.com/data-protection/tutorial-network-access-control-nac.php?printer_friendly=this-page.

[66] 802.1x and NAC: Best Practices for Effective Network access Control: http://www.wavelink.com.au/downloads/bradford-networks/Network_Access_Control_802.1X.pdf.

[67] Network Access Control Technologies. URL: http://www.bizforum.org/whitepapers/sygate-4.htm

[68] TCG Trusted Network Connect TNC Architecture for interoperability. URL: www.opus1.com/nac/tnc/ tnc_architecture_v1_1_r2.pdf.

Parhi Snehasish, received MCA degree on 2003. Currently, he is working as Scientific Officer in National Institute of Technology (NIT), Rourkela since March 2007. He has total 12 years of experience on teaching, server administration and network administration in datacenter environment. His main research interests include network access control, Intrusion detection and prevention system and email security.