Open Source Implementation of Protocol Carrying an Authentication for Secure Network Access Based on IETF Standards

Abstract— As we know that network authentication to access the network securely is one of the most important procedure for network operator to restrict and control the user access to the network service. Based on this platform many researchers had already done a lot of work but still there is a space for research, also the IETF proposed a PANA protocol to carry out network access authentication regardless the underlying access technology for giving the user a secure network access. This protocol has gaining more interest as a potential candidate for network authentication of both the category (access and service authentication) in both existing and emerging network environments. But due to lack of open source implementation of the protocol standards does not provide the fruitful results during the deployment and testing on different network environments which is the major hurdle in broader expansion and acceptance of this standards. So In this article, we provide the design and open source implementation of protocol based on PANA standard. We also provide the analysis of the performance result of the protocol with the other active open source implementation of PANA (OpenPANA and CPANA) which will definitely help the others developers to create their own implementation by extending this protocol implementation.

I. INTRODUCTION

In network security area this work will be majorly accepted to solve the network security issues by implementing an Protocol based on IETF standards, using the User Datagram Protocol (UDP) Protocol as transport & it act as application protocol to carry the Extensible Authentication Protocol (EAP) in order to support different authentication mechanisms for network access, irrespective of the network access technology. This work provides flexible authentication framework using EAP flexible authentication framework for network access & allow multiple clients to authenticate using Multithreading concept. Using advanced secure key algorithm for the key in implementation of PANA provides more secure solution.

There has been consideration of other alternative solutions to PANA such as AAA protocols, DHCP, and IKEv2 are responsible for EAP transport. After analysis following point is notify that among these protocol some of are having header fields are very large , some are very short & some are not present at all, and so on. But PANA, accepted this all limitation as a challenge & designed its own message format, from start to end and which aims is to matches & satisfy the EAP transport requirements. But when we integrate EAP with DHCP2 then analysis notify that the EAP has creating complexity problems which eliminated it as a candidate for the EAP standard. So this work is considered the EAP with AAA Protocol for the implementation of PANA. Recommendation systems uses the piggybacking of messages which allows either PaC or PAA to send a single PANA message that represents both an answer and a request. This work successfully implemented PANA and recommended that on following area the applicability of PANA is successfully considered.

- Currently used Network (IPV4)
- Forthcoming Generation Network (IPV6)
- Wireless Network

This work create great impact on network security at link layer by combining Protocol based on IETF standards with EAP & completely satisfies the requirements of EAP lower-layer protocol on ordered message delivery and the reliability requirement for messages exchanged after the authentication and authorization phase (e.g., PANA-Notification-Request and PANA-Termination-Request) messages require a response from the communicating peer to complete the notification and session termination operations, respectively) using UDP Protocol. It fully satisfies the EAP lower-layer protocol requirement on ordered message delivery and the reliability requirement for messages exchanged after the authentication and authorization phase using UDP Protocol.
Hence, uses the lighter weight UDP based PANA transport is also less complex, provides strong reliability and more efficient. Thus using UDP, this Protocol provide fruitful secure network solutions & hence there is no need of TCP protocol Steps are, the first extension is the specification of a PAA discovery mechanism used by a PaC to Ascertain the PAA’s IP address to be used during a PaC-initiated PANA authentication. The second extension details the PEMK derivation algorithm that is used for Generating cryptographically independent PEMKs for different EPs. The third extension is the so-called PANA pre-authentication. The objective of Recommended Protocol pre-authentication is to reduce EAP authentication latency during handoff in mobile environments. Some of the cases where this work is considered areas. Since the end of the PANA WG in 2010, there has been increasing interest in the protocol applicability in different areas, due mainly to its robustness, simplicity, degree of flexibility, and the development of open source implementations. In the following, by examining different example scenarios where the Protocol has been considered to control the access not only to the network service but also to application services.

II. TECHNIQUES USED FOR RECOMMENDED SYSTEM

The standard protocol proposed by the IETF is PANA which can done the network access authentication work regardless the underlying access technology in an efficient manner with simplicity & flexibility. The achievement of standardization can be goes through in several network scenarios such as the Internet of Things and machine-to-machine communications, among others. But due to the lack of open source implementations of the standard does not provide the platform on which it will be test and deploy this protocol in different network scenarios, this results in hinder its broader expansion and adoption. So this work presents an OpenPANA, it one of the open source implementation, which implements the PANA as per mention in the standardization. The detail description about the design and implementation decisions around OpenPANA will help other developers to create their own implementations. & also focuses on different usage scenarios where the implementation has been considered and used. At last the work is concluded by proving the performance results and an interoperability test which has been performed with CPANA. The Recommended Protocol based on PANA are used to design so that it can provide the facility of the authentication and authorization of clients in access networks.

PANA is an EAP [RFC3748] lower layer that carries EAP authentication methods encapsulated inside EAP between a client node and a server in the access network. In an authentication process actually PANA has been enabled these two entities which is nothing but a part of the AAA functionality and also an access control framework. A AAA and access control framework using PANA is comprised of four functional entities which are shown in functional model fig 1.

A. PANA Client (PaC)

PaC is the client side entity of Recommended Protocol based on PANA. This entity actually works on the node of PaCs are end hosts, such as laptops, PDAs, cell phones, desktop PCs, or routers that are connected to a network via a wired or wireless interface. This client side entity is responsible for requesting network access and initiating the authentication process using PANA.

RADIUS

| PaC |<-----------------|PAA |<----------------->| AS |
|-----|                 |-----|<--------------------|-----|
|      |<-----+      +-----|      |<---------------------|-----|
|      |                  |      |                        |-----|
|      |<---------+>> +-----|      |<--------------|-----|
|      |                  |      |                        |-----|

IKE, ------->| EP |<-------- ANCP, API, etc.

4-way handshake, +-----+ etc.

\[\begin{array}{c}
\text{Data traffic}
\end{array}\]

Fig. 1: PANA Functional Model

B. PANA Authentication Agent (PAA)

It is the server side entity of Recommended Protocol based on PANA which is implemented in the network. PAA is responsible of interfacing with the PaCs for authenticating and authorizing them for the network access service. The role of PAA is very important, first of all it consults an authentication server for verification of the credentials and rights of a PaC which they have passed.
If the location of authentication server is same as the PAA, that mean both are residing on the same node then an API is sufficient for this interaction. But when they are on different protocols need to work out between the two. AAA protocols like RADIUS [RFC2865] and Diameter [RFC3588] which are most commonly used for the AAA purpose. He is also responsible for updating the access control state (i.e., filters) depending after the creation and deletion of the authorization state. Then the role of PAA is to communicates the updated state to the Enforcement Points (EPs) in the network. If both PAA and EP are residing on the same node, then an API is sufficient for this communication. Otherwise, there is need of the protocol whose role is to carry the authorized client attributes from the PAA to the EP. The node on which PAA resides is actually called as NAS (network access server).

C. Authentication Server (AS)

The implementation of server is mainly focus on the verification of the credentials of a PaC that is requesting the network access service. The role of AS is after receiving the requests from the PAA on behalf of the PaCs, it should be responds with the result of verification together with the authorization parameters such as allowed bandwidth & IP configuration. After completion of his task its prime responsibility to terminate the EAP and the EAP methods. The AS might be hosted on the same node as the PAA, on a dedicated node on the access network, or on a central server somewhere in the Internet.

D. Enforcement Point (EP)

Its role is to implement an access control implementation by considering in charge of allowing access (data traffic) of authorized clients while preventing access by others. An EP learns the attributes of the authorized clients from the PAA. There is involvement of two category of filters in the EAP methods which are as non-cryptographic or cryptographic filters and the EP supports this both categories for selectively allow and discard data packets. After successful deployment of these filters on an EP may be done at the link layer or the IP layer. When cryptographic access control is use then only there is a need of secure association protocol to run between the PaC and EP. After completion of the secure association protocol, link- or network- layer per-packet security is enabled for integrity protection, data origin authentication, replay protection, and optionally confidentiality protection. An EP is located between the access network (the topology within reach of any client) and the accessed network (the topology within reach of only authorized clients).

For minimizing the access of unauthorized clients it must be located strategically in a local area network. The EP should be on the path between the PaC and the PAA for the aforementioned reason is just recommended by PANA not made any compulsion. For example, the EP can be hosted on the switch that is directly connected to the clients in a wired network. That way the EP can drop unauthorized packets before they reach any other client node or nodes beyond the local area network. The decision to collocate these entities PAA and EP are considered in this research work as In fig. 3.3.4.1 is a suggested usage model of PANA where the PAA and EP are tied together in one location but are separated. A secure transportation between PAA and EP is needed for the creation of control lists, which make it possible to authorize and before authenticated clients to send and receive packets on the network. In order to setup filtering policies controlling network access of the PaCs the PANA has to notify whether an existing protocol solution to one or more separated EPs are used or not. Both Pull model and Push model describe the initiator of communication between PAA and EP. The Pull model regards EP as initiator of PAA-to-EP communication: push model communication is initiated by PAA. There is necessity that PAA has to inform to EP regarding which traffic may be generated by a authorized PaC, the filtering rules and very sure that this communication has to be very secure. After this the role of an EP is to bind this rules to each PaC.

E. Authorization Signal Flow

Figure 2 shows the signal flow diagram wherein the complete flow of authorizing a client for network access is shown in fig 2. The general data traffic of any PaC are permitted to access the network at any Enforcement Point after the successfully authentication & it also restricted the traffic control of type such as PANA, DHCP, Discovery of router) for the PaC which is unauthorized still. This may leads to attachment of new clients which are having the minimum access service to busy with PANA and try to immediately complete the authorization process so that it can access the unlimited service. The static or dynamic configuration of an IP address can be done by PaC before running the PANA. There is need of to re-configure its IP address or configure additional IP address(es) of PaC after knowing that it has been successfully authenticated depending on scenario wherein it is the deployed. As far as when considering the forthcoming network IPV6 is concerned the addressing technique used by PANA is the a link-local IPv6 address and there is permission of PaC that they can configure additional global IPv6 address(es) once it has completed the authentication process successfully.
Another example: while in IPV4 network there is restriction on PaC for using link-local address during PANA are limited and are permitted to reconfigure its interface with a non-link-local IPv4 address after the successful completion of authentication process. This interface are not permitted to use for General-purpose applications cannot use the interface until PANA authentication succeeds and appropriate IP address configuration takes place. At the initial stage the unauthorized PaC starts the authentication of PANA by discovering the PAA by following the steps of each EAP methods that can be exchange over the PANA. After discovery there is interaction of the PAA with the AS, as the next step of process. It is mandatory that the PAA has to informs the PaC related to the result of its request for network access after getting the result from the AS upon successful completion of the authentication and authorization process. After gaining the network access in terms of authorized PaC there is necessarily that the, the PAA has to sends the some attribute such as IP address, cryptographic keys, which is called as PaC-specific attributes to the EP with the help of another protocol. After receiving the information the permission has been give to PaC

<table>
<thead>
<tr>
<th>PaC</th>
<th>EP</th>
<th>PAA</th>
<th>AS</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address</td>
<td>Provisioning</td>
<td>AAA</td>
<td></td>
</tr>
<tr>
<td>Config</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fig. 2: PANA Signaling Flow

EP to use this information for modification of its filters to allow data traffic from and to the PaC to pass through. when the secure association protocol runs between the PaC and the EP during the PANA authentication there is a need of enabling the cryptographic access control. Some parameters are required which will be dynamically available to some protocol for example endpoint identity, shared secret which can be derived upon the successful PANA authentication. Since for authenticating the PaC to the EP and vice-versa these parameters are used which will be act as a essential part of creation of security association.

The PANA-IPSEC uses IKE which is based on using a specific EAP method to generate a key which is required for PANA to establish a secure association between the PaC and PAA by enabling cryptographic data traffic protection through message exchange. Inclusion of cryptographic data traffic protection per-packet has leads to an additional burden on per-packet but this burden issue will remain only in communication between the PaC and EP and also given the assurance of not affecting the communications beyond the EP. Now at last there is successful installation of filters on EP for allowing data traffic for general purpose to flow between the PaC and the access network.

F. Environments

PANA can be used on any network environment whether there is a lower-layer secure channel between the PaC and the EP prior to PANA after successful PANA authentication process. For successful network access authentication process there are two types of networks has been considered: first the networks where there is assurance of secure channel & it is easily available before running the PANA. The category of such type of network is characterized by the protection of existence network from spoofing and eavesdropping. As per the demand of user who wants secure network access then the authentication and authorization is mandatory for him so that his purpose has to be achieved If there is guaranty of presence of secure channel before the initiation of messages under PANA has been eliminates the compulsory requirement of execution of a secure association protocol after PANA. The PANA has responsible for taking initiation of development of session in association with the communication channel over it was carried. Similarly as far as the selection of EAP authentication method is concerned it is totally depends on the assurance of security must be present when PANA in the running phase & are applied on the networks where there is no lower-layer protection prior to running PANA. The generation of cryptographic keys that are used with a secure association protocol to enable per-packet & the cryptographic protection is an indication that the successful authentication of PANA has been done. The choice of EAP method must be resilient to the possible attacks associated with such an environment.
Sequential Steps

In general, any PANA implementation needs to interface with an EAP implementation that accomplishes the standard defined in RFC 3748. Typically, this EAP implementation follows the reference model for the EAP state machine defined in RFC 4137. Moreover, given that an AAA client is usually required in the PAA, Recommended Protocol based on OpenPANA standards integrates a RADIUS client as defined in the RADIUS standard (RFC 2865) & also using the existing third-party open source libraries for the RADIUS client and EAP implementation including several EAP methods such as EAP-TLS and EAP-PSK.

III. DESIGN OF PANA

Recommended Protocol is an open source PANA implementation in C which manages an efficient management of devices’ resources such as memory, data, and execution threads. The other software packages and libraries for network access security are used which is already based on C, such as OpenSSL, wpa_supplicant, hostapd, and FreeRADIUS, among others. In general, PANA implementation needs to interface with an EAP implementation as per the standard mentioned in RFC 3748 & the EAP implementation follows the reference model as per mentioned in the RFC 4137. Recommended Protocol based on PANA has been integrates a RADIUS client as defined in the RADIUS standard (RFC 2865). The implementation of the protocol is totally based on OpenPANA uses existing open source libraries for the RADIUS client and EAP implementation as third party source libraries which included several EAP methods such as EAP-TLS and EAP-PSK client mentioned in the RADIUS standard (RFC 2865).

A. General Architecture

The general architecture (Fig.3.3.8.1) of protocol is a multithreaded framework, which is implemented in the PaC and PAA, since both entities may need to process multiple PANA authentication sessions (especially the PAA) needs to be handled at a time, whereas the PaC does not need RADIUS message processing. The architecture handles the three main data structure which are shown in fig 3, PANA session list; This list stores an active PANA sessions which is going on PANA.

- Task list: This list indicates tasks which is pending (e.g., management of a received message, transition in the state machine or update of a PANA session). Each and every task contains a pointer to the corresponding PANA session.

- Alarm list: This list indicates a linked list holding active alarms (e.g., retransmissions or session lifetime). Each alarm contains a pointer to its corresponding PANA session. These structures are implemented by mainly three components that form the core of our architecture:

- Network manager: The network manager is a single thread that listens for incoming PANA messages, and RADIUS messages in the PAA of PANA.
- Network manager: The network manager is a single thread that listens for incoming PANA messages, and RADIUS messages in the case of the PAA of PANA.
- Alarm manager: Alarm manager is nothing but a single thread that wakes up every certain period of time (e.g., 1 s) and checks the alarms available at the alarm list.

- Workers: Worker has defined a configurable number of threads to carry out the tasks which is specified in the task list, it is defined as a simple stateless forwarder of PANA messages between the PaC and the PAA, so the multi-threaded architecture has not been deployed in this entity.

After arrival of every new message the task is added in the task list by network manager using the adds (addTask()) after executing some checking process over the message header format. In case if there is possibility of existence of pending tasks then at that time, each free worker obtains one task using (getTask()). After the task is retrieved it has been started to process a message that intimate the beginning of a new PANA authentication, then the role of worker is to generate a PANA session structure and includes it in the PANA session list. This way, it creates the delay of any further state until it is really required. In addition a worker also include new alarms to
B. General Operation

The alarm list during the processing if it is needed. After activation of alarm the alarm manager can also add a new task to the task list. This implementation of protocol internally manages its own alarm framework for handling alarms.

C. PANA Session Structure

Each task has an associated PANA session through which the workers will operate successfully. The PANA session is a structure that contains an essential variables as specified in (e.g., PaC and PAA IP addresses, session lifetime or cryptographic material) as well as the current state of the PANA state machine. But the PaC an EAP peer state machine has added in addition with the PANA session that each session should be associated with the instance of PaC whereas in comparison with the PAA, it includes an instance of an EAP authenticator state machine. The EAP authenticator state machine instance also has a pointer to both the category are specified as PANA Session in the PaC b) PANA session in the PAA a single RADIUS client instance that handles the interaction with a RADIUS server. Sharing of the same RADIUS client instance among all PANA sessions defined in the PAA (singleton pattern), and ability to multiplex and handles the different RADIUS communications for different PANA sessions over the same connection is the speciality of this research . This way the RADIUS client installed in the PAA and the RADIUS server use the same ports irrespective of the number of PaCs connected to the PAA. For identifying the PANA sessions, the identifier of the RADIUS message sent is always associated with the PANA session. To avoid depleting the available free ports in the access device and more utilization of resources, even though it may reduce the general performance.

D. PANA State Machine

The PANA state machine operates internally as follows: given a current state, an event happens when certain conditions are met. Then the state machine executes the action associated with the event and moves to the next state. To implement this schema, the PANA state machine is designed as a two-dimensional array where the rows represent the states and the columns the events. Each cell, represented by the pair (current state, event), contains a pointer to a function (callback). This callback verifies the conditions that trigger the event in the column and, if met, executes the corresponding action and returns the next state of the state machine after the transition. Hence, when a worker processes a task, it accesses the PANA session to know the current state. Then the worker starts executing each callback in the row of the current state in order to trigger a transition to a next state when possible.

E. Implementation Details

The implementation of the protocol based on PANA is divided into two main implementation parts. On one hand, there is a implementation of basic components of PANA from scratch: PANA state machine, PANA session, PANA messages, and the multi-threaded architecture. In addition, we have used a set of libraries to implement , which required other parts such as cryptography and the reading of configuration information:

- libssl and libcrypto: cryptographic libraries from the OpenSSL project. This library covers all the cryptographic needs to build the PANASA.
- libxml2: This is the library that adds the functionality needed to handle the configuration file, which has been defined in XML format.
- pthread: This is the POSIX library, available in most Linux operating systems, including the management of threads. On the other hand, we have reused existing and well-known open source implementations of the EAP state machines, and the RADIUS client and server.
- wpa_supplicant: It contains a library that implements the EAP peer state machine. The details of this library is defined in RFC 4137. It has been used to implement an operational PaC.
- hostapd: It contains the EAP authenticator state machine and a RADIUS client. The protocol based on PANA also provides a wrapper following RFC 4137 to implement an operational PAA.

![Fig. 4: PANA session design](image-url)
FreeRADIUS: Its role is to implement a RADIUS-based server which can easily interact with the PAA for secure access network.

F. Evaluation of PANA

The performance evaluation in terms of shows the LoC (measured with the sloc command) and memory footprint (measured with ps) respectively for OpenPANA, CPANA, and the modified PANA are shown in Figs. 5a and 5b when we have adapted the existing network and wireless network. In the general case (first two columns), the LoC in PANA are lower than OpenPANA and CPANA (excluding the external libraries). However, OpenPANA has to load them in memory, so the memory footprint is higher than that of PANA and CPANA. For authentication/re-authentication performance, Figs.5c and 5d show the average time for both processes (after running around 200 tests) distinguishing between the parts dedicated to PANA used local host as compared to PANA for the (re-) authentication by implementing on the local host, eth0 and wlan0 for communication. The reason for using this network interface is that we are not interested in the communication link delay but in the average processing time. To measure it, we have used clock_gettime to observe the delay between the first and last messages of the (re-)authentication in CPU time. After observation, the OpenPANA implementation takes longer than PANA, indicating that OpenPANA is, in general, implementation while CPANA does not support this AAA interaction. To achieve this, OpenPANA shares a RADIUS client instance between multiple PANA sessions to avoid a more complex implementation than CPANA. There are several reasons for that. Firstly PANA implements a multi-threading framework as same as OpenPANA while as CPANA is a mono-process implementation. In fact, CPANA was unable to process simultaneous authentications as compared with OpenPANA and PANA.

Secondly PANA and Open PANA both relays in external open source libraries for EAP implementation to provide multiple authentication mechanisms. In Open Source implementation while CPANA integrates EAP implementation with the PANA source code itself and only implements one EAP method (EAP-PSK), optimizing the overall processing. Third, There is interaction of PANA with an external RADIUS server based on FreeRADIUS for adding the flexibility same as of OpenPANA depleting available ports in the PAA. But this will be reduced the overall performance & produces a bottleneck. At last, the FreeRADIUS supports an experimental implementation of the EAP-PSK method that adds one unnecessary exchange before starting EAP-PSK. This exchange can be negligible in a more optimized implementation of EAP-PSK & would imply a reduction of ~2 ms in the overall authentication time. When we summarized it is as follows, the major issue of CPANA are related to the lack of implemented features, such as an AAA client in the CPANA-PAA or support for non-EAP-piggyback. PaC-initiated re-authentication, or IPv6. But on the other side PANA has opted for many features which includes an authentication flexibility, integration with existing AAA infrastructures, and multiple-user support at the cost of a more complex implementation as compared with CPANA and OpenPANA. This can proves that PANA implementation support a large set of features for testing PANA than CPANA and OpenPANA ,EAP, and RADIUS. For evaluation of PANA protocol we have used a PC Core duo 2.16 GHZ with 3 GBytes RAM used for (re) authentication.

IV. PROTOCOL OPERATION

As per the commitment of providing a secure access to network service for every user it need to use the flexible framework where either the user or the network is responsible for taking initiation of secure network access control process is the prime proposal of PANA.
For that purpose both the PaC or the PAA can be initiated the PANA authentication process. To start the PANA execution, both entities must first know or discover each other’s IP address. For example, either static configuration or with Dynamic Host Configuration Protocol (DHCP), where the PaC is not only provided with a valid IP address but also with a list of available PAAAs are used by PAA discovery mechanism are shown in fig. 5. The execution of protocol start only if the PaC knows the PAA’s IP address. As observed in Fig 5, a Steps of PANA execution which consist of the execution of four different phases. At the beginning during the authentication and authorization phase, the PaC and the PAA has compromises certain parameters like the integrity algorithms requires to protect PANA messages (1) In addition In working of this phase, the user authentication is taking place by using EAP, which is transported within both PANA messages (between PaC and PAA) and an AAA protocol (between PAA and AS). As an example, Fig. 6 indicates an authentication process based on the widely employed EAP-TLS method. The monitoring of this shown in the fig 7 to notify that PAA forwards the EAP messages to the backend AS (where the EAP server is collocated) for verification of the user’s credentials. A PANA session is established with an associated lifetime after the authentication is successfully completed. The next phase of protocol start that is nothing but the access phase, in which the protocol has give the permission to PaC to access the network by transmitting and receiving data through the EP(s). After a consequence of the successful authentication ,the PANA SA derived from the MSK can be protected all PANA messages transmitted from the beginning of the access phase to the end of the PANA session with integrity by using MSK obtained by EAP if and only if the underlying network are unable to provide the security protection. After successful establishment of PANA SA, a PaC-EP SA is established for data traffic protection between PaC and EP to provide confidentiality and integrity to user data exchanged between both entities. Now the PAA can installed the required PEMK on the specific EP. The session of PANA in which the data transmission and receiving is going, if it is about to expire, then the PaC or PAA can initiate the re-authentication phase to extend the current session. For example, in Fig. 5 this phase is now work as a based phase on which the re-authentication phase will start. The re-authentication mechanism of EAP-TLS will generate a new key (MSK’) to rebuild the PANA SA and potential PaC-EP SAs.

Now at the end the PANA session can end with the help of these entities by entering into the next phase called as termination phase to remove the resources allocated by the network for the PaC. when the PANA session lifetime is reached the additional reserved network resources are also released. After analysis of working of each phase then the research work can notify certain points as, during each phase can send the different set of messages which is as per the rule mentioned in the PANA state machines which already has been proven to be well designed and deadlock-free .As mention below, the protocol operation uses four types of PANA messages which are as follows:

- **PANA Client Initiation (PCI)**
  This is actual starting process of PANA wherein a PaC-initiated PANA authentication is going with the help of this message, which is sent by the PaC for requesting the PAA. to start the authentication process, so that the PaC can access the network. This message is used if the PAA is initiating the PANA authentication process.

- **PANA-Auth-Request/Answer (PAR/PAN):**
  The authentication and authorization phase and the re-authentication phase uses these messages during the working of the above mentioned phase. This phase allow for compromising the certain parameters between the PaC and the PAA (by activating the Start flag [S] in the message header of the first exchange) and authentication information to be carried in the format of EAP packets. The authentication or re-authentication process has ended by setting the Completion flag (C) activated.

- **PANA-Notification-Request/Answer (PNR/PNA):**
  Once the authentication of PaC has been completed then these messages may be exchanged between PaC and PAA. These messages are used only to notify that as a PANA session keep-alive mechanism or to signal by A-Flag activation at the beginning of a re-authentication process.

- **PANA-Termination-Request/Answer (PTR/PTA):**
  These messages can be used to provoke a controlled finalization of the PANA session. Either the PaC or the PAA can initiate the termination of a session. The general format of these messages includes a defined PANA header and a list of attribute value pairs (AVPs) as information containers. Each AVP carries an explicit tag identifying its content type.
For example, the EAP-Payload AVP is used to transport the EAP packets in PAR or PAN messages; and AUTH AVP includes a message authentication code to provide integrity to every PANA message except the PCI. This explicit tagging has been conceived for extensibility of the PANA relies on EAP method for Producing the keying material for PANA SA which are shown in fig. 6(a)(b). A) PANA session in the PaC  B) PANA session in the PAA

Fig. 5 PANA Workflow

Fig. 6(a)(b): PANA Secure association

Fig. 7 (a)(b): PANA Client / Server Architecture

V. CONCLUSION

We have provided a detail explanation of open source implementation of protocol which is based on IETF standard & analysis of PANA protocol, which is the contribution made by the IETF in the field of network access authentication for achieving the visibility and maturity of the protocol.
In this article, we have provided a comprehensive explanation of the design and implementation of PANA, our open source implementation. We have explained details about the internal general architecture and operation, which has considered reference of other implementations. To complete the study, we have presented usage scenarios where the implementation is being tested. Finally, we have carried out an interoperability test and performance evaluation between OpenPANA, CPANA and PANA which has allowed us to discover that there is still room for improvement in the implementations of PANA. We have obtained from the interop and with some improvements to optimize the general performance. Moreover, again their is also need of extending the PANA project with PaC implementations for smart phones (i.e., iOS, Android).

Acknowledgment

The authors are very much thankful to Mr. Firoz Kazi, Dubai because of their motivation and support we are trying to present a paper on very interesting and very innovative topic analysis of PANA for achieving the network security. We are also thankful to the PANA workgroup who have initiated this topic and presented an excellent article on this topic.

We are also thankful to Pedro Moreno Sanchez (co-developer of OpenPANA) for his excellent articles from that i have learned the detail concept of PANA.

REFERENCES