Jurnal Teknologi Maklumat & Multimedia **5**(2008): 13-40

EAP Authentication Mechanism for Ad Hoc Wireless LAN

M. Agni Catur Bhakti, Azween Abdullah,
Low Tan Jung & Halabi Hasbullah

ABSTRACT

Wireless networks have some security issues, mainly due to its wireless nature (open and non-physical), that need to be addressed. One of the solutions to improve wireless network security is the IEEE 802.1X specification, which is based on Extensible Authentication Protocol (EAP). EAP is an authentication framework that can support multiple authentication methods. With its advantage of being flexible, EAP has been used in many types of networks, wired and wireless, including the infrastructure model of wireless LAN. With EAP's flexibility, it might be also possible to use EAP as a generic authentication mechanism in other types of networks. This paper studies and explores the feasibility of using EAP in ad-hoc model of WLAN and proposes a mechanism to implement EAP in ad hoc WLAN based on EAP multiplexing model. We also propose a mechanism to select a suitable EAP method out of a set of EAP methods to be used in authentication process in heterogeneous wireless network, in which there are different types of nodes with different specifications and capabilities in the network. Thus, each node may support a different type of EAP authentication method. Toward the end of this paper, we formally specify and verify the proposed authentication mechanism in order to proof and obtain strong beliefs of the authentication.

Keywords: Extensible authentication protocol (EAP), IEEE 802.1X, authentication, wireless local area network.

ABSTRAK

Pada masa kini, rangkaian teknologi tanpa wayar semakin berkembang kerana kelebihannya berbanding rangkaian berwayar dari segi mobiliti dan kemudahannya. Walau bagaimanapun, masih terdapat kelemahan dari segi limitasi dalam rangkaian teknologi tanpa wayar. IEEE 802.1X adalah satu mekanisma untuk mengawal akses berdasarkan EAP. EAP mampu menyokong pelbagai kaedah dalam mengenalpasti identiti. Atas kelebihan ini EAP banyak digunakan di dalam pelbagai jenis rangkaian. Kajian ini bertujuan untuk mengkaji kesesuaian menggunakan EAP dalam rangkaian tanpa wayar adhoc *dan mencadangkan satu mekanisma baru untuk melaksanakan EAP dalam rangkaian tanpa wayar ad-hoc berdasarkan model EAP multipleks. Satu kelebihan menggunakan kaedah pengawalan akses berdasarkan EAP ialah kemampuan untuk digunakan di dalam pelbagai jenis rangkaian. Kajian ini juga mencadangkan satu mekanisma untuk memilih teknik EAP yang sesuai dalam proses pengesahan identiti untuk rangkaian tanpa wayar pelbagai jenis. Dalam rangkaian tanpa wayar pelbagai jenis, terdapat banyak jenis terminal (komputer) dengan pelbagai spesifikasi dan kemampuan. Oleh itu terminal berkemungkinan memerlukan teknik EAP yang berlainan. Pada akhir kajian ini, spesifikasi dan pengesahan formal mekanisma pengesahan yang dicadangkan dibincangkan.*

Katakunci: Perluasan Protokol Pembuktian (EAP), IEEE 802.1X, pembuktian, rangkaian tanpa wayar kawasan setempat.

INTRODUCTION

For the past few years, the popularity and use of wireless network has grown rapidly. Report by Horrigan (2007) shows significant growth of wireless access or connectivity to the Internet using wireless devices (i.e. laptop, cell phone, wireless-enabled personal digital assistant). One contributing factor of this growth is wireless network's advantages over the wired network, such as convenience, mobility, scalability, and rapid deployment. However, wireless network also introduces new security issues. The wireless channel itself suffers from poor protection and susceptible to attacks. Unauthorized wireless devices (rouge devices) are relatively easier to connect to the network because they do not need any physical access. Therefore, providing a way for the communicating parties in wireless network to validate each other's identity, i.e. authentication, is crucial and important.

One of the solutions to overcome the limitation of wireless network security and providing authentication is the IEEE 802.1X (2004) specification, a mechanism for port-based network access control, which is based on Extensible Authentication Protocol by Aboba et al. (2004), an authentication framework that can support multiple authentication methods. EAP can run over many types of data-link layer and it is relatively flexible in its implementation.

Extensible Authentication Protocol (EAP) as one of the authentication mechanisms has been used in many types of networks, both wired and wireless networks, including the infrastructure model of wireless LAN / Wi-Fi (IEEE 802.11). However, the typical EAP authentication mechanism might not be able to be implemented in the ad hoc model of WLAN due to ad hoc network characteristics, e.g. infrastructure-less. Thus new scheme or mechanism of EAP-based authentication has to be designed for ad hoc WLAN.

This motivated researchers to carry out this work, i.e. to study how EAP can be implemented in ad hoc WLAN and to design a mechanism of an EAP-based authentication mechanism for ad hoc WLAN. One promising advantage of using EAP-based authentication mechanism in ad hoc WLAN is interoperability with other types of networks since EAP has already become the platform of many authentication mechanisms, and that is one step closer towards interoperability across heterogeneous networks in the near future.

In this paper, we limited the ad hoc wireless network model to the single-hop ad hoc or peer-to-peer model of IEEE 802.11 WLAN/Wi-Fi network devices only. We did not emphasize on the routing protocol aspect of ad hoc WLAN yet.

BACKGROUND

Wireless LAN provides greater convenience and mobility than the wired LAN with its range, reaching tens to hundreds of meters. The international standards for wireless LAN is the IEEE 802.11 family, providing transmission speeds ranging from 1 – 54 Mbps typically in 2.4 or 5 GHz frequency bands. The standard includes the v 802.11a/b/g/i. The latter, i.e. IEEE 802.11i (2004), defines a framework and means for supporting security over WLAN. It adds Medium Access Control (MAC) security enhancements for WLAN using IEEE 802.1X standard and EAP.

EXTENSIBLE AUTHENTICATION PROTOCOL (EAP)

IEEE 802.1X standard is a port-based network access control that makes use of the physical access characteristics of IEEE 802 LAN infrastructures in order to provide a means of authenticating and authorising devices attached to a LAN port, and of preventing access to that port in cases which the authentication and authorisation fails. This standard makes use of Extensible Authentication Protocol.

EAP is one of the authentication mechanisms that have been widely used currently. It has been used in Point-to-Point Protocol (PPP), wired networks, and wireless networks. It is an authentication framework which supports multiple authentication methods. EAP was initially used for PPP authentication, but it can also run over other data-link layer such as the IEEE 802 LAN family.

One of the advantages of EAP framework is flexibility. EAP may be used on dedicated links, switched circuit links, wired and wireless links. EAP also permits the use of back-end authentication server to implement some or all the authentication methods. It is proven that EAP is able to be implemented in various network access technologies as the following:

- 2G technology: Global System for Mobile communication (GSM)
- 3G technology: Universal Mobile Telecommunication System (UMTS)

- IEEE 802.11 wireless LAN / Wi-Fi
- 3G – WLAN internetworking based on the 3rd Generation Partnership Project (3GPP) specification (3GPP 2006)
- EAP also has been ratified as one of authentication mechanism for WiMAX (Worldwide Interoperability for Microwave Access) (IEEE 802.16e 2006).

EAP METHODS

EAP methods are the authentication methods used in EAP. There are many types of EAP methods available today using different kinds of mechanisms or technologies such as passwords, digital certificates, challenge-response, hash message, smart card, etc. Some of the available EAP methods are the following:

- EAP with MD5 hash (EAP-MD5) by Aboba, et al. (2004) uses Message-Digest algorithm 5 (MD5) hash to authenticate client. This is the basic EAP method and in most situations this method gives inappropriate wireless network security.
- EAP with Transport Layer Security (EAP-TLS) by Aboba & Simon (1999) uses TLS (Dierks & Allen 1999), successor of Secure Socket Layer version 3 (SSLv3), and requires both the client-side and server-side to have Public Key Infrastructure (PKI) digital certificates in order to provide secure mutual authentication. This method considered as the strongest (security wise) EAP method currently (Ali & Owens 2007; Microsoft 2007).
- EAP with Tunneled TLS (EAP-TTLS) by Funk & Blake-Wilson (2007) requires server-side certificate while user-side can use an extensible set of user authentication such as Windows login and password and legacy user authentication methods. EAP-TTLS uses secure TLS record layer channel to set up tunnel to exchange information between client and server. EAP-TTLS offers strong security while avoiding the complexities of PKI implementation on client's side. It was co-developed by Funk Software and Certicom.
- Protected EAP (PEAP) by Kamath et al. (2002) is similar to EAP-TTLS in the way that it only requires server-side certificate and using other way to authenticate client, uses TLS tunnel, and offers strong security. The main difference is in compatibility with legacy (older) methods and platforms which PEAP is less compatible compared to EAP-TTLS. It was jointly developed by Microsoft, Cisco, and RSA Security.
- Lightweight EAP (LEAP) (Sankar et al. 2005) is a proprietary EAP method developed by Cisco Systems for their wireless LAN devices. LEAP supports mutual authentication and dynamic security keys changes in every (re)authentication to improve security.

- EAP with Subscriber Identity Module (EAP-SIM) by Haverinen & Saloway (2006) uses 2nd Generation (2G) GSM network SIM.
- EAP-AKA (Authentication and Key Agreement) by Arkko & Haverinen (2006) uses 3rd Generation (3G) UMTS Subscriber Identity Module (USIM).

EAP ENTITIES

There are three entities defined in the IEEE 802.1X standard that involved in the EAP authentication process: *Supplicant*, *Authenticator*, and *Authentication Server*. Supplicant is an entity in the network that seeks to be authenticated. Authenticator is an entity that facilitates authentication of the supplicant. Authentication Server is an entity that provides the authentication service to the authenticator.

In Wi-Fi environment, typically mobile/wireless station or device will act as the supplicant. Wireless access point acts as the authenticator. Authentication, Authorization, and Accounting (AAA) server, such as RADIUS (Remote Authentication Dial-In User Service) or Diameter server acts as the authentication server.

EAP permits the authentication server to implement some or all authentication methods while the authenticator only acts as a pass-through entity. The Authenticator and Authentication Server may reside in different devices or collocated in one device. This will be explained further in EAP implementation model.

EAP MODEL

Based on its current specification, i.e. RFC 3748, EAP consists of the following components:

- Lower layer. It is responsible for transmitting and receiving EAP frames between the peer and authenticator. This layer includes Point-to-Point Protocol (PPP), Ethernet wired LAN, wireless LAN, etc.
- EAP layer. It receives and transmits EAP packets via the lower layer; implements duplicate detection and retransmission, and deliver and receive EAP messages to and from the EAP peer and authenticator layers.
- EAP peer or EAP authenticator layer. The EAP layer demultiplexes incoming EAP packets to the EAP peer and authenticator layers. Typically implementation on a host only will support either peer or authenticator functionality, but it is possible for a host to act as both.
- EAP method layer. It implements the authentication algorithms, receives and transmits EAP messages via the EAP peer and authenticator layers.

EAP IMPLEMENTATION MODEL

In the typical EAP implementation, the three entities reside in three separated devices, and the authenticator acts as a "pass-through-authenticator". It forwards packets from the peer and destined to its authenticator layer to the back-end authentication server; and vice versa packets received from the back-end authentication server destined to the peer are forwarded to it.

Another model specified in EAP's RFC is the EAP multiplexing model. In the multiplexing model, there is no authentication server entity since the authenticator will implement all the authentication methods. In other words, the authentication service is embedded into the authenticator. However, this may require the node/host that acts as the authenticator to have more computational capabilities in order to support functionality of both authenticator and authentication server and implement all the authentication methods.

RELATED WORKS

Most research efforts in ad hoc wireless network have been focused on the development of the network architecture itself, particularly in the network routing protocol and medium access control (MAC) protocol design. Lou & Fang (2004) claimed that most of the ad hoc routing protocols assume that all nodes are cooperative and trustworthy, and there are relatively little works that have been carried out with security consideration.

As far as we know, currently there is no specific EAP method that has been developed for ad hoc wireless network and there are only few EAP mechanisms that have been proposed (in open literature) for ad hoc wireless network.

Lee & Park (2003) proposed a user authentication mechanism for mobile ad hoc networks using EAP and Ad-hoc On-demand Distance Vector (AODV) routing protocol. The mechanism defines master node for authentication server and how other nodes acquire authentication from it using MD5 Challenge. This mechanism requires modification / expansion of EAP and AODV hello packet format. In today's wireless network environment, MD5 considered to provide weak / poor security protection.

Moustafa et al. (2005) proposed architecture for vehicular communication on highways with ad hoc networking support. The architecture adapts an Authorisation, Authentication, and Accounting (AAA) scheme using Kerberos, instead of RADIUS server, and EAP-Kerberos method for vehicular communication on highways environment.

Khan & Akbar (2006) proposed the use of EAP-TTLS and Protocol for carrying Authentication for Network Access (PANA) by Forsberg et al. (2007) in multi-hop wireless mesh networks (WMN) that can be extended by ad hoc wireless network. This method requires PANA which is still under development and exists only as Internet Engineering Task Force (IETF) draft.

Nidjam & Scholten (2006) proposed the use of virtual Authentication Server in Wi-Fi ad hoc implementation of Access Point Security Service (APSS) with a scenario that comprises of two people communicating for the first time at a conference, both having subscriptions with network service providers. This method requires the existence of infrastructures (such as access points) of the network service providers (both telecommunication and wireless / hotspot service providers).

Most of the work above are still in architecture or mechanisms proposal stage. As far as we know, except for Nidjam & Scholten (2006), those researchers have not provided any proof of concept of their proposed mechanisms whether in formal verification, simulation, or test bed development.

Related to our proposed mechanism for EAP method selection, Ali & Owens (2007) has pointed out the need of selecting the most suitable authentication method for a particular wireless LAN network environment. The work is useful in selecting one EAP method suitable for a network, prior to implementing EAP framework in that particular network. It identified the factors to be considered when employing EAP in wireless LAN, and it would serve as a foundation for our selection mechanism.

PROPOSED MECHANISM

We designed the mechanism of EAP-based authentication for ad hoc wireless LAN using EAP multiplexing model where there is no separate authenticator or access point entity as described in previous section. This model is more suitable with the infrastructure-less nature of ad hoc communication (no access points) and the authentication is carried out between two nodes.

For the EAP method, we used the existing EAP methods. There are already numerous EAP methods available and some of them are sufficient to provide secure authentication in wireless network environments. We believe that this can reduce the development, implementation, and deployment time of the mechanism significantly compared to designing and developing a new EAP method.

The authentication processes in our mechanism consists of two phases: initial phase authentication and operational phase authentication. The ad hoc network configuration consists of one or some master node(s) and several mobile nodes. Master node is the node that will act as authentication server that will provide authentication service in the initial phase. The mobile nodes are the nodes that seek to be authenticated, whether to the master node in the initial phase or to each other in the operational phase. Master node should also have digital certificate service installed to issue certificates for the mobile nodes. Therefore, master node should have more computational capabilities compared to mobile nodes.

INITIAL PHASE (NODE-TO-MASTER-NODE) AUTHENTICATION

In the initial phase, the mobile nodes are needed to be authenticated to the master node that has the authentication server and digital certificate services installed. The mobile node will prove its identity using user name, ID number, serial number, etc and password. This method is chosen because mobile node may not have any certificate yet since Public Key Infrastructure (PKI) may not always be available in ad hoc wireless network. We can use EAP types that only require server-side certificate such as EAP-TTLS and PEAP. After successful authentication, the mobile node will receive a type of digital certificate generated and signed by the master node. This certificate will have expiry timestamp and will be used in operational phase authentication. The initial phase authentication is illustrated in Figure 1 and the EAP messages exchange is illustrated in Figure 2.

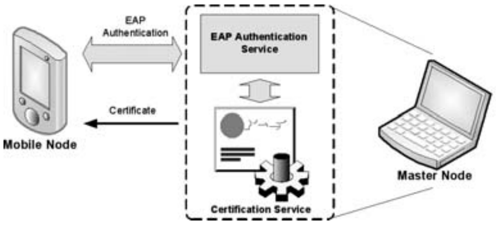

FIGURE 1. Node-to-Master-Node authentication

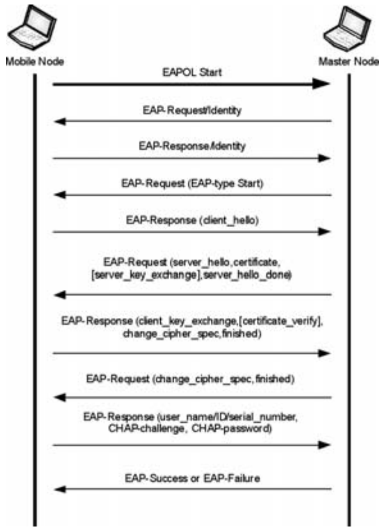
Mobile Node
Master Node
EAPOL Start
EAP-Request/Identity
EAP-Response/Identity
EAP-Request (EAP-type Start)
EAP-Response (client_hello)
EAP-Request (server_hello,certificate, [server_key_exchange],server_hello_done)
EAP-Response (client_key_exchange,[certificate_verify], change_cipher_spec,finished)
EAP-Request (change_cipher_spec,finished)
EAP-Response (user_name/ID/serial_number, CHAP-challenge, CHAP-password)
EAP-Success or EAP-Failure

FIGURE 2. Node-to-Master-Node authentication messages exchange

The following are the processes executed in the initial phase authentication:

- Master node/authentication server sends an EAP-Request/Identity packet to the mobile node/client.
- Client responds with an EAP-Response/Identity packet to the server (containing the client's userId / sessionID).
- Server responds with an EAP-type Start packet (EAP-TTLS or PEAP).
- Execute TLS handshake process:
  - Client sends a "Client hello" message to the server, along with the client's random value (nonce).
  - Server responds by sending a "Server hello" message to the client, along with the server's random value (nonce), its certificate, and the "Server hello done" message.
  - Client creates a random Pre-Master Secret (PMS).
  - Client encrypts the PMS with the public key from the server's certificate.
  - Client sends the encrypted PMS to the server.
  - Server receives the PMS.
  - Server and client each generate the Master Secret and session keys based on the Pre-Master Secret and the nonces using pseudo-random-number function (PRF).
  - Client sends "Change cipher spec" notification to server (to indicate that the client will start using the new session keys for hashing and encrypting messages).
  - Client sends "Client finished" message.
  - Server receives "Change cipher spec" and switches its record layer security state to symmetric encryption using the session keys.
  - Server sends "Server finished" message to the client.
- Execute client authentication:
  - Server sends challenge to client.
  - Client responds with username, password, and challenge, encrypted with the session key.
  - If username, password, and challenge are validated by server then "EAP-Success" else "EAP-Failure".
- EAP-Success:
  - Client creates its private and public keys.
  - Server creates certificate for client (containing client's identity, public key, and validity period/expiry).
  - Server signs client's certificate with server's private key.
- EAP-Failure:
  - Abort authentication.
  - Server disconnects from client.

OPERATIONAL PHASE (NODE-TO-NODE) AUTHENTICATION

In the operational phase, the mobile nodes that have been authenticated can authenticate each other using their certificates received from master node. The mobile nodes will exchange their certificates, checking the validity and expiry of the certificates, thus proving their identities. We can use EAP types that employ authentication requiring or supporting certificates of both sides such as EAP-TLS, EAP-TTLS, and PEAP. The operational phase authentication is implemented using EAP multiplexing model without the need of authentication server/master node support.

The operational phase authentication is illustrated in Figure 3 and the EAP messages exchange is illustrated in Figure 4.

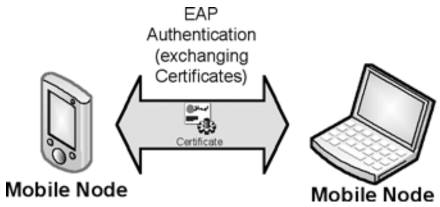

FIGURE 3. Node-to-Node authentication

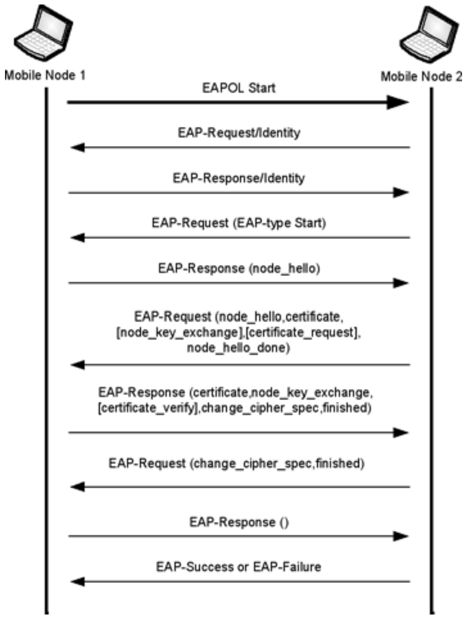
Mobile Node 1
Mobile Node 2
EAPOL Start
EAP-Request/Identity
EAP-Response/Identity
EAP-Request (EAP-type Start)
EAP-Response (node_hello)
EAP-Request (node_hello,certificate, [node_key_exchange],[certificate_request], node_hello_done)
EAP-Response (certificate,node_key_exchange, [certificate_verify],change_cipher_spec,finished)
EAP-Request (change_cipher_spec,finished)
EAP-Response ()
EAP-Success or EAP-Failure

FIGURE 4. Node-to-Node authentication messages exchange

The following are the processes executed in the operational phase authentication:

- Mobile node 2 sends an EAP-Request/Identity packet to the mobile node 1.
- Node 1 responds with an EAP-Response/Identity packet to node 2 (containing the node 1 userId / sessionID).
- Node 2 responds with an EAP-TLS Start packet.
- Execute TLS handshake process:
  - Node 1 sends a "Client hello" message to the Node 1, along with the Node 1's random value (nonce).
  - Node 2 responds by sending a "Server hello" message to Node 1, along with the Node 2's random value (nonce), its certificate, request for Node 1's certificate, and the "Server hello done" message.
  - Node 1 validates Node 2's certificate (using Master Node's public key obtained in initial phase).
  - If Node 2's certificate is validated then continue else "EAP-Failure".
  - Node 1 creates a random Pre-Master Secret (PMS).
  - Node 1 encrypts the PMS using Node 2's public key.
  - Node 1 sends the encrypted PMS along with its certificate, and certificate verify.
  - Node 2 receives Node 1's response.
  - Node 2 decrypts the response and validate Node 1's certificate (using Master Node's public key obtained in initial phase).
  - If Node 1's certificate is validated then continue else "EAP-Failure".
  - Node 1 and Node 2 each generates the Master Secret and session keys based on the Pre-Master Secret and the nonces using pseudo-random-number function (PRF).
  - Node 1 sends "Change cipher spec" notification to Node 2 (to indicate that Node 1 will start using the new session keys for hashing and encrypting messages).
  - Node 1 sends "Client finished" message.
  - Node 2 receives "Change cipher spec" and switches its record layer security state to symmetric encryption using the session keys.
  - Node 2 sends "Server finished" message to the client.
- EAP-Success:
  - Node 1 and Node 2 can start data communication.
- EAP-Failure:
  - Abort authentication.
  - Node 1 and Node 2 are disconnected.

We also propose an extension to the existing EAP architecture by adding the EAP Method Selection process which is executed prior to the EAP authentication process, as illustrated in Figure 5. The diagram of the extended EAP architecture is illustrated in Figure 6.

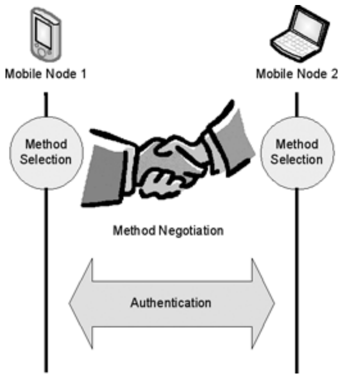

FIGURE 5. EAP method selection prior to EAP authentication

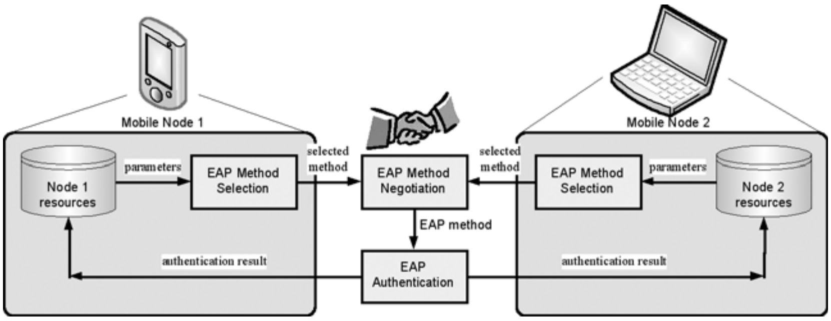

FIGURE 6. EAP authentication with EAP method selection

The function of the EAP method selection module is to recommend which EAP method is suitable to be used for authentication. The selection is based on the following criteria:

- The node’s resources: digital certificates availability, platform or operating system, node specifications (hardware, software, etc). Each EAP method may require different resources; therefore it is important to select the suitable EAP method based on this criterion.
- Previous authentication records: these records will provide data about previous authentications, such as list of nodes that have tried to authenticate

or tried to be authenticated, time of authentications, the used methods, authentication results, etc. These data can be used to obtain information useful in the method selection, such as the last successful authentication method and the most successful authentication method.

- Previous communication records: these records will provide data about previous network communications, such as previous malicious packets / traffic from other nodes. These records could be an output or a log file from a network communication analyser program. That program will monitor and analyse the network activities and records them in a log file. The collected data can be used to obtain information whether a node trying to be authenticated is considered harmless or not by examining whether that node has history of sending malicious packets.

For this work, we used a set of EAP methods that have been widely used and considered to give strong security protections; they are EAP-TLS, EAP-TTLS, and PEAP, as claimed by Ali & Owens (2007). We put the highest priority to EAP-TLS because it provides the strongest security protection. The requirement of using EAP-TLS is the availability of digital certificates in both nodes, thus we have to check them first.

If only one party has digital certificate, then we have to use EAP-TTLS or PEAP. EAP-TTLS and PEAP give strong level of protection but EAP-TTLS has the advantage of more flexibility because it can be used in different operating systems and supports many mechanisms, such as CHAP, PAP, MS-CHAP, and MS-CHAPv2, while PEAP only supports newer Microsoft Windows operating systems (Windows XP and above) and Microsoft mechanisms. For better compatibility, we put EAP-TTLS as the last option of method when conditions for the other methods are not met. We can do the selection by checking the operating system of the node. Thus, the flow diagram of the EAP authentication with EAP method selection in our study case can be illustrated as in Figure 7.

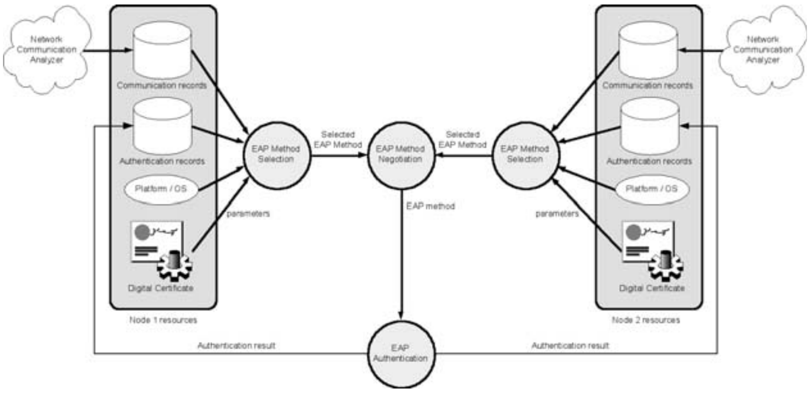

FIGURE 7. Flow diagram of EAP authentication with EAP method selection

Figure 8 illustrates the algorithm of the EAP method selection mechanism and Figure 9 illustrates the EAP method selection process based on the node's current resources. In this case, if there is any history of malicious packet from the other node trying to be authenticated, the authentication process will be aborted immediately. The authentication records information used here are the last successful and most successful methods.

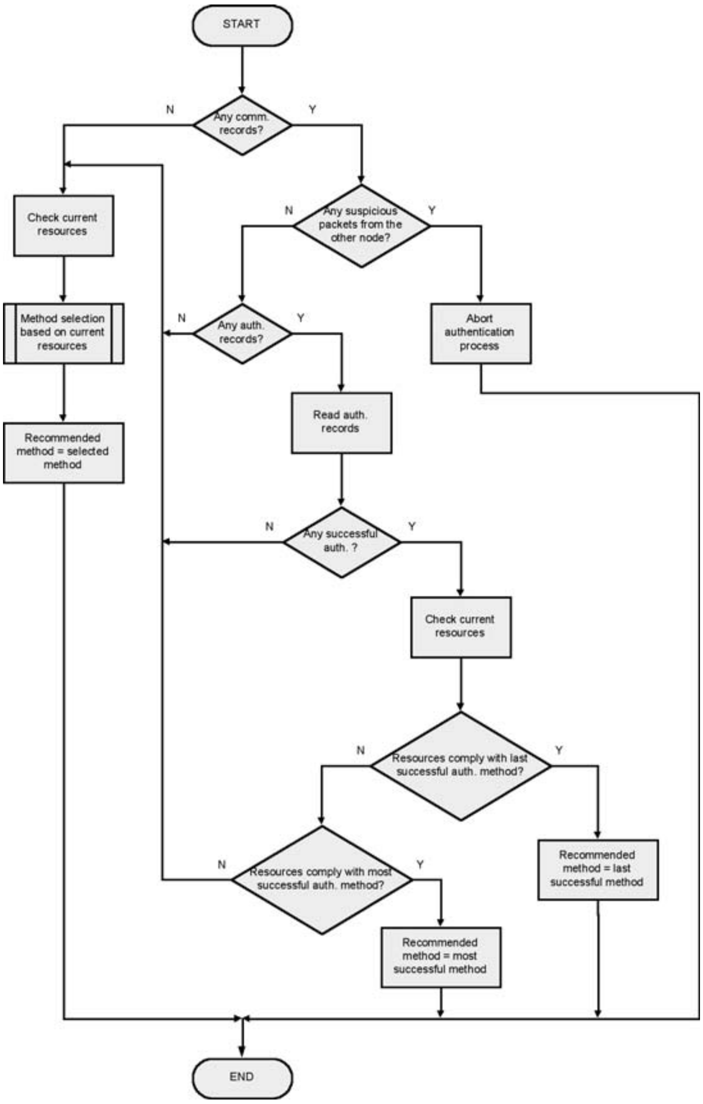
START
N
Any comm. records?
Y
Check current resources
N
Any suspicious packets from the other node?
Y
Method selection based on current resources
N
Any auth. records?
Y
Abort authentication process
Recommended method = selected method
Read auth. records
N
Any successful auth. ?
Y
Check current resources
N
Resources comply with last successful auth. method?
Y
N
Resources comply with most successful auth. method?
Y
Recommended method = last successful method
Recommended method = most successful method
END

FIGURE 8. Flow chart of the EAP method selection mechanism

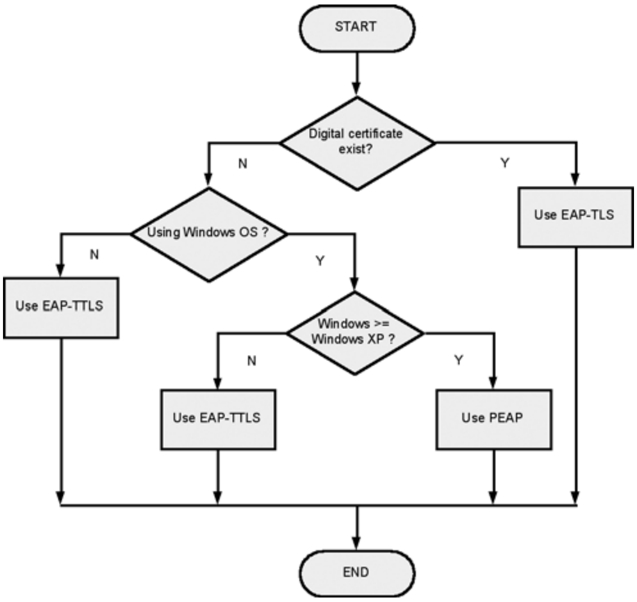

FIGURE 9. Flow chart of the EAP method selection based on the node's current resources

If both nodes selected the same method, then authentication process will be carried out using that selected method. If they select different methods, then authentication process will be carried out using the lower method in hierarchy out of the selected methods. As described in previous section, the hierarchy is as the following: EAP-TLS, then PEAP, and then EAP-TTLS. The selection of the lower method in hierarchy will not compromise the authentication security since the last method in hierarchy (and the most compatible method), i.e. EAP-TTLS, still provides a strong level of security.

Formal Specification and Verification

In this section, BAN Logic by Burrows et al. (1990) is used to specify and prove the abstract model of the proposed authentication mechanism. BAN Logic is logic of authentication that used logic, rules, and postulates, instead of mathematical expression, to express and describe the beliefs of entities/ parties/*principals* (people, computers, services) involved in authentication process. BAN Logic focuses on the beliefs of trustworthy parties involved in authentication protocol and on the evolution of these beliefs as a consequence of communication during authentication process. If proof that a protocol is correct cannot be obtained, then the protocol deserves to be treated with a precaution. BAN Logic has been used to analyse and improve many authentication protocols such as in (Anderson 1997; Agray et al. 2002; Chiu-Man 2002). Please refer to the Appendix and (Burrows et al. 1990) for more details about BAN Logic, including its rules and postulates.

The steps in analysing a protocol using BAN Logic are as the following:

- The idealised form of the protocol is derived from the original form.
- Assumptions about the initial state are written.
- Logical formulas are attached to the statements/messages of the protocol, as assertions about the state of the system after each message.
- The logical postulates of BAN Logic are applied to the assumptions and the assertions in order to discover the beliefs held by the parties in the protocol.

The BAN Logic considers that authentication process is complete between *A* and *B* if there is a *K* such that:

A believes $A \xleftrightarrow{K} B$, B believes $A \xleftrightarrow{K} B$

Or possibly more than the above state can be achieved, as the following:

A believes B believes $A \xleftrightarrow{K} B$, B believes A believes $A \xleftrightarrow{K} B$

Some public key protocols are not intended to result in the exchange of shared key, but instead transfer other data. For example, the interaction of a principal with certification authority (CA) might be intended to transfer a public key, or to establish shared secret or nonces. In our case, the goal of the authentication is to establish shared session key (K) generated from shared nonces, pre-master secret (PMS), and master secret (M). Thus the goals are:

A believes $A \xleftrightarrow{Kab} B$, B believes $A \xleftrightarrow{Kab} B$

In this section we analyse the proposed authentication mechanism using BAN Logic by following the steps mentioned above, in order to discover the beliefs held by the parties in the authentication mechanism. Paulson (1998) formally analysed one of the underlying protocols used in EAP, i.e. TLS; while in our case, we would like to analyse the overall proposed extended EAP authentication.

INITIAL PHASE (NODE-TO-MASTER-NODE) AUTHENTICATION

We refer to the client mobile node as ‘A’ and the master node as ‘S’ (Server). Sid is Session ID; Na, Ns, are nonces (random numbers/values); Ka and Ks are public keys;

Ka^{-1} and Ks^{-1} are the related private keys; PMS is pre-master secret, a random string generated by ‘A’; Kas is shared session key generated from master secret (M) and the nonces. The master secret is a 48-bytes secret calculated from PMS and the nonces. These are the messages in node to master node authentication:

1. A → S: EAPOL_start
2. S → A: req_id
3. A → S: A, Sid
4. S → A: EAP_start
5. A → S: A, Na
6. S → A: Ns, Sid, cert(S, Ks)
7. A → S: $\{PMS\}_{Ks}$, $\{finished\}_{Kas}$
8. S → A: $\{finished, challenge\}_{Kas}$
9. A → S: $\{user\_name, password, challenge\}_{Kas}$
10. S → A: EAP_success / EAP_failure

Burrows et al. (1990) stated that we can omit the messages that are not contributing to the logical properties of the mechanism and any clear text communications since they provide no guarantees of any kind, thus we omitted messages 1 – 5, and 10. In message 6, server sends its certificate signed by itself (self-signed certificate) with an assumption that the server is already known to and trusted by the clients as the certificate authority. Pre-master secret (PMS) is transformed into N'a as PMS is a random string generated by A. The 'finished' message should contain the hashed of the master secret and all previous handshake messages as specified by Dierks & Allen (1999), thus it is transformed into H(M, Sid, A, Na, S, Ns). N's is a random challenge, considered as another nonce generated by S to be used in subsequent messages. Xa and Ya are user data which is the user name and password pair to be used in client authentication. If the authentication succeeds, the process will continue with key and certificate generation. The certificate will contain the identity of A, the public key of A, and the certificate validity time, signed with S's private key. Thus we can obtain the idealised form as the following:

6. S → A: $\{\xrightarrow{Ks} S\}_{Ks^{-1}}$
7. A → S: $\{ N'a\}_{Ks}$, $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$
8. S → A: $\{H(M, Sid, A, Na, S, Ns), N's\}_{Kas}$
9. A → S: $\{\langle Xa\rangle_{Ya}, N' s\}_{Kas}$

Initial assumptions:

A believes $\xrightarrow{Ks} S$
A believes fresh (Na)
A believes fresh N'a)

S believes $\xrightarrow{Ks} S$
S believes fresh (Ns)
S believes fresh N's)

S believes (A controls $A \xleftrightarrow{N} S$)

A believes $A \overset{Ya}{\Leftrightarrow} S$ S believes $A \overset{Ya}{\Leftrightarrow} S$

A knows the public key of certification agent S, and S knows its own keys. Each principal believes that the nonce they generate is fresh. A will invent a new nonce as pre-master secret and S trusts A to invent good nonce. Each principal believes that they shared a secret. The authentication analysed:

Message 6:

Message 6 will give a belief that A can be assured that it is communicating with S since only S can encrypt the message with Ks^{-1}.

A believes $\xrightarrow{Ks} S$

Message 7:

Then A sends message 7. A can be sure that only S can decrypt $\{N'a\}_{Ks}$ and see N'a since only S knows the Ks^{-1}. Therefore A believes that it shares N'a as a secret with S.

A believes $A \overset{N'a}{\Leftrightarrow} S$

Since N'a is the PMS and from it A can calculate master secret M and Kas, therefore we obtain:

A believes $A \overset{M}{\Leftrightarrow} S$

A believes $A \xleftrightarrow{Kas} S$

S receives message 7 which will give:

S sees $\{N'a\}_{Ks}$

S sees $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$

S sees N'a, since S can decrypt $\{N'a\}_{Ks}$ using its private key Ks^{-1}. S then can calculate the master secret M and Kas. Using Kas, S can decrypt $\{H(M, Sid, A, Na, S, Ns)\}_{Kas}$, thus:

S sees $A \overset{N'a}{\Leftrightarrow} S$

S sees $A \overset{M}{\Leftrightarrow} S$

S sees $A \xleftrightarrow{Kas} S$

S sees H(M, Sid, A, Na, S, Ns)

At this point, we cannot obtain better belief for S yet. S still cannot be sure yet that it is communicating with A, that message 7 was sent by A recently, since other party, e.g. C, who is able to intercept the messages exchanged between A and S (acts as Man-in-The-Middle), can replace N'a with N'c, encrypt it with Ks, and send it along with hashed of the intercepted messages. Ks is the master node's public key which is likely available in public. Actually, this problem can be addressed using certificate verify from A, a hash of messages signed by A. However, in this initial phase, the node might not have a certificate yet.

Message 8:

S will response with message 8, sending its finished message and N's. A supposed to receive message 8 and we can obtain:

A sees $\{H(M, Sid, A, Na, S, Ns), N's\}_{Kas}$

Using message-meaning rule, we can obtain:

A believes S said $\{H(M, Sid, A, Na, S, Ns), N's\}$

Since A believes fresh(Na), thus:

A believes fresh$\{H(M, Sid, A, Na, S, Ns), N's\}$

Using nonce-verification rule, we can obtain:

A believes S believes $\{H(M, Sid, A, Na, S, Ns), N's\}$

A believes S believes $A \overset{M}{\Leftrightarrow} S$

A believes S believes $A \xleftrightarrow{Kas} S$

If an attacker, C, intercepts message 8, C will not be able to pass this message to A since the finished message will be different with the one calculated by A because C can not obtain the value of N'a from message 7 (it is encrypted with Ks and can only be decrypted with Ks^{-1}), thus C can not calculate the correct M and finished message. If A does not receive the correct finished message within time, it will disconnect and abort the authentication session. Thus there is no security breach.

Message 9:

A responses with message 9, sending its identity and secret along with N's. S receives it, which will give:

S sees $\{\langle Xa\rangle_{Ya}, N's\}_{Kas}$

Since S believes $A \overset{Ya}{\Leftrightarrow} S$, $\langle Xa\rangle_{Ya}$ serves as proof of A's identity. Thus we obtain:

S believes A said $(\{\langle Xa\rangle_{Ya}, N's, A \xleftrightarrow{Kas} S)$

S believes A said$(A \xleftrightarrow{Kas} S)$

Since S believes fresh (N's) then

S believes A believes$(A \xleftrightarrow{Kas} S)$

Using nonce-verification rule, we obtain:

S believes A believes $A \xleftrightarrow{Kas} S$

And finally, using juridication rule we obtain:

S believes $A \xleftrightarrow{Kas} S$

Attacker C will not be able to create message 9 since C needs to provide A's identity and secret, i.e. $\langle Xa\rangle_{Ya}$, which is unlikely to be obtained by C. Therefore, Man-in-The-Middle (MiTM) and replay attacks will not work in this authentication scheme.

The final beliefs of the initial phase authentication are:

A believes $A \xleftrightarrow{Kas} S$

S believes $A \xleftrightarrow{Kas} S$

A believes S believes $A \xleftrightarrow{Kas} S$

S believes A believes $A \xleftrightarrow{Kas} S$

We refer to the client mobile nodes as ‘A’ and ‘B’, and the master node as ‘S’ (Server). Sid is Session ID; Na, Nb are nonces; Ka, Kb, Ks are public keys; Ka^{-1}, Kb^{-1}, Ks^{-1} are the related private keys; PMS is pre-master secret, a random string generated by ‘A’; Kab is shared session key generated from master secret (M) and the nonces. The master secret is a 48-bytes secret calculated from PMS and the nonces. These are the messages in operational phase authentication:

1. A → B: EAPOL_start
2. B → A: req_id
3. A → B: A, Sid
4. B → A: EAP_start
5. A → B: A, Na
6. B → A: Nb, Sid, cert(B, Kb, Tb)
7. A → B: cert(A, Ka, Ta), $\{PMS\}_{Kb}$, certificate_verify, $\{finished\}_{Kab}$
8. B → A: $\{finished\}_{Kab}$
9. A → B: EAP_response()
10. B → A: EAP_success / EAP_failure

Again, we omitted the messages that are not contributing to the logical properties of the mechanism and any clear text communications, i.e. messages 1 – 5 and messages 9 – 10. In message 6, B sends its certificate signed by S in initial phase. In message 7, A must send its certificate along with its certificate verify message and the pre-master secret it generated encrypted with B’s public key, and followed by the finished message. In message 8, B also responses with finished message to confirm that both parties have agreed on the same parameters (secrets and keys). Thus we can obtain the idealised form as the following:

6. B → A: $\{B, Kb, Tb\}_{Ks}{}^{-1}$
7. A → B: $\{A, Ka, Ta\}_{Ks}{}^{-1}$, $\{N'a\}_{Kb}$, $\{H(A, Na, N'a, B, Nb, Sid)\}_{Ka}{}^{-1}$, $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$
8. B → A: $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$

Initial assumptions:

A believes $\xrightarrow{Kas} A$	B believes $\xrightarrow{Kb} B$
A believes $\xrightarrow{Ks} S$	B believes $\xrightarrow{Ks} S$
A believes (S controls $\xrightarrow{K} B$)	B believes (S controls $\xrightarrow{K} A$)
A believes fresh (Na)	B believes fresh (Nb)
A believes fresh (N’a)	B believes fresh (N’b)

A believes (A controls $A \xleftrightarrow{N} B$

Each principal knows the public key of certification agent S, and each knows its own keys. Each principal trusts the certification agent to sign digital certificates. A will invent a new nonce as pre-master secret and S trusts A to invent good nonce. Each principal believes that the nonces they generate are fresh. The authentication analysed:

Message 6:

We apply message-meaning and jurisdiction rules to message 6 and obtain:

A believes $\xrightarrow{Kb} B$

Message 7:

A responses with message 7, containing its digital certificate, pre-master secret (N’a), and its certificate verify. A can be sure that only B can decrypt $\{N'a\}_{Kb}$ and see N’a since only B knows the Kb^{-1}. Therefore A believes that it shares N’a as a secret with B.

A believes $A \overset{N'a}{\Leftrightarrow} B$

A believes $A \overset{M}{\Leftrightarrow} S$

A believes $A \overset{Kab}{\longleftrightarrow} B$

B receives message 7. Using message-meaning and jurisdiction rules we obtain:

B believes $\xrightarrow{Ka} A$

B sees N’a, since B can decrypt $\{N'a\}_{Kb}$ using its private key Kb^{-1}. B then can calculate the master secret M and Kab. Using Kab, B can decrypt $\{H(M, Sid, A, Na, S, Ns)\}_{Kab}$, thus:

B sees $A \overset{N'a}{\longleftrightarrow} B$

B sees $A \overset{M}{\longleftrightarrow} B$

B sees $A \overset{Kab}{\longleftrightarrow} B$

Using message-meaning rule, we obtain:

B believes A said H(A, Na, N’a, B, Nb, Sid)

B believes A said (A, Na, N’a, B, Nb, Sid)

Since B believes fresh(Nb):

B believes fresh(A, Na, N’a, B, Nb, Sid)

Using nonce-verification rule, we obtain:

B believes A believes(A, Na, N’a, B, Nb, Sid)

B believes A believes $A \overset{N'a}{\Leftrightarrow} B$

B believes A believes $A \overset{M}{\Leftrightarrow} S$

B believes A believes $A \overset{Kab}{\longleftrightarrow} B$

And using jurisdiction rule, we obtain:

B believes $A \overset{Kab}{\longleftrightarrow} B$

As seen in the belief obtained from message 7, we can obtain stronger belief for B due to the certificate verify from A. Unlike the initial phase, in this operational phase the node already has digital certificate thus it can produce certificate verify.

Message 8:

B will response with message 8, sending its finished message. A supposed to receive message 8 and we can obtain:

A sees $\{H(M, Sid, A, Na, B, Nb)\}_{Kab}$

Using message-meaning rule, we obtain:

A believes B said H(M, Sid, A, Na, B, Nb)

A believes B said (M, Sid, A, Na, B, Nb)

Since A believes fresh(Na), thus:

A believes fresh (M, Sid, A, Na, B, Nb)

Using nonce-verification rule, we obtain:

A believes B believes (M, Sid, A, Na, B, Nb)

A believes B believes $A \overset{M}{\Leftrightarrow} B$

A believes B believes $A \xleftrightarrow{Kab} B$

The final beliefs of the operational phase authentication are:

A believes $A \xleftrightarrow{Kab} B$

B believes $A \xleftrightarrow{Kab} B$

A believes B believes $A \xleftrightarrow{Kab} B$

B believes A believes $A \xleftrightarrow{Kab} B$

Conclusion and Future Works

This paper presents our study on EAP-based authentication for ad hoc wireless LAN. We designed authentication mechanism for ad hoc wireless LAN based on EAP multiplexing model. As implied by the name, Extensible Authentication Protocol, EAP is able to be extended in order to support the growing and expanding needs. The EAP framework provides extensible environment where it is possible to customize or grow the framework. We have extended EAP framework by adding a mechanism to select an EAP method out of a set of EAP methods based on some parameters.

As a proof of our concept, the authentication mechanism is specified and verified using BAN Logic. Strong final beliefs for the proposed authentication mechanism were able to be obtained. It indicates that the proposed authentication mechanism can provide secure authentication and after authentication the two parties entitled to believe that they are communicating with each other and not with intruder.

By enabling EAP in network authentication, in our case is in ad hoc wireless LAN authentication, it will enable the network users to be authenticated across heterogeneous network types using EAP as the enabling technology, though more works and other protocols will be required. One of the solutions might involve the use Protocol for carrying Authentication for Network Access (PANA) as network-layer transport for EAP. PANA will carry EAP which can carry various authentication methods. By PANA's feature of enabling transport of EAP above internet protocol (IP), any authentication method that can be carried as an EAP method is made available to PANA, thus to any data link-layer technology.

The work done in this paper need to be tested in simulation, test-bed, or real network environment. Agni et al. (2008) simulated the mechanism and the results showed that the method selection and negotiation algorithm is able to select and negotiate the suitable method for the authenticating nodes. The simulation also showed that in order to execute secure node-to-node authentication, a successful node-to-master-node authentication is required. Otherwise, the authentication process will have to use weak authentication method which should be avoided. Future works should focus on implementation of EAP authentication in test-bed or even in real ad hoc wireless network environment.

REFERENCES

3rd Generation Partnership Project (3GPP) Technical Specification Group Service and System Aspects, 3G Security, Wireless Local Area Network (WLAN) internetworking security (Release 7). 2006. 3GPP TS 33.234.

Aboba, B. & Simon, D. 1999. RFC 2716, PPP EAP TLS Authentication Protocol. The Internet Society.

Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J. & Levkowetz, H. (Ed.) 2004. RFC 3748, Extensible Authentication Protocol (EAP). The Internet Society.

Agni, M., Abdullah, Azween & Low Tan Jung. 2008. Simulation of EAP Method Selection and Negotiation Mechanism. *Proceedings of the 3rd International Symposium on Information Technology (ITSIM)*, 26-29 August. Kuala Lumpur, Malaysia.

Agray, N., Wiebe van der Hoek & Erik de Vink. 2002. On BAN Logic for Industrial Security Protocols. *Proceeding of the 2nd International Workshop of Central and Eastern Europe on Multi-Agent Systems (CEEMAS)*. Berlin, Heidelberg: Springer-Verlag.

Ali, K. M. & Owens, T. J. 2007. Selection of an EAP Authentication Method for a WLAN. *Int. J. Information and Computer Security*, 1(1/2): pp. 210-233.

Anderson, R. J. 1997. The Formal Verification of a Payment System. Cambridge, UK: Computer Laboratory.

Arkko, J. & Haverinen, H. 2006. RFC 4187, Extensible Authentication Protocol Method for 3rd Generation Authentication and Key Agreement (EAP-AKA). The Internet Society.

Burrows, M., Abadi, M. & Needham, R. 1990. A Logic of Authentication. *ACM Transaction on Computer Systems,* 8(1): 18-36

Chiu-Man, Yu. 2002. Secure Execution of Mobile Agents on Open Networks using Cooperative Agents. Master Thesis, the Chinese University of Hongkong.

Dierks, T. & Allen, C. 1999. RFC 2246, The TLS Protocol Version 1.0. The Internet Society.

Forsberg, D., Ohba, Y. (Ed.), Patil, B., Tschofenig, H. & Yegin, A. 2007. Protocol for Carrying Authentication for Network Access (PANA). Internet-Draft. The IETF Trust.

Funk, P. & Blake-Wilson, S. 2007. EAP Tunneled TLS Authentication Protocol Version 0 (EAP-TTLSv0). Internet-Draft. The Internet Trust.

Haverinen, H. (Ed.) & Saloway, J., Ed. 2006. RFC 4186, Extensible Authentication Protocol Method for Global System for Mobile Communications (GSM) Subscriber Identity Modules (EAP-SIM). The Internet Society.

Horrigan, J. 2007. Wireless Internet Access, December 2006 Tracking Survey. Princeton Survey Research Associates International for Pew Internet & American Life Project.

IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks – Specific requirements, Part 11, Amendment 6: Medium Access Control (MAC) Security Enhancements. 2004. IEEE Standard 802.11i.

IEEE Standard for Local and metropolitan area networks, Part 16, Amendment 2: Physical and Medium Access Control Layers for Combined Fixed and Mobile Operations in Licensed Bands. 2006. IEEE Standard 802.16e.

IEEE Standard for Local and metropolitan area networks, Port-Based Network Access Control, 2004. IEEE Standard 802.1X.

Kamath, V., Palekar, A. & Wodrich, M. 2002. Microsoft's PEAP version 0 (Implementation in Windows XP SP1). Internet-Draft. The Internet Society.

Khan, K. and Akbar, M. 2006. Authentication in Multi-Hop Wireless Mesh Networks. *Transactions on Engineering, Computing and Technology*, Vol. 16, November. World Enformatika Society.

Lee, Jong-Hoon & Park, Ho Jin. 2003. A User Authentication Protocol Using EAP for Mobile Ad Hoc Networks. *Proceedings of the IASTED International Conference: Communication, Network, and Information Security*. New York, USA.

Lou, W. & Fang, Y. 2004. A Survey of Wireless Security in Mobile Ad Hoc Networks: Challenges and Available Solutions. In *Ad Hoc Wireless Networking, Network Theory and Applications*, Vol. 14. Kluwer Academic Publishers.

Microsoft Corp. 2007. IEEE 802.11 Wireless LAN Security with Microsoft Windows, January. (online). http://www.microsoft.com/downloads/

Moustafa, H., Bourdon, G. & Gourhant, Y. 2005. AAA in Vehicular Communication on Highways with Ad hoc Networking Support: A Proposed Architecture. *Proceedings of the 2nd ACM International Workshop on Vehicular Ad Hoc Networks (VANET 2005)*. Cologne, Germany.

Nidjam, M. & Scholten, H. 2006. Access Point Security Service for wireless ad-hoc communication. Technical Report TR-CTIT-06-66 Centre for Telematics and Information Technology. Enschede, Netherlands: University of Twente.

Palekar, A., Simon, D., Salowey, J., Zhou, H., Zorn, G. & Josefsson, S. 2004. Protected EAP Protocol (PEAP) Version 2. Internet-Draft, The Internet Society.

Paulson, L.C. 1998. Inductive Analysis of the Internet Protocol TLS. *Security Protocols*.Berlin, Heidelberg: Springer-Verlag.

Sankar, K., Sundaralingam, S., Miller, D. & Balinsky, A. 2005. *Cisco Wireless LAN Security*. N.York: Cisco Press.

M. Agni Catur Bhakti, Azween Abdullah
Universiti Teknologi PETRONAS
Bandar Seri Iskandar, 31750 Tronoh, Perak.
Agni_catur_bhakti@utp.edu.my
azweenabdullah@yahoo.com

APPENDIX: BAN LOGIC

BASIC SYMBOLS / NOTATION

In BAN logic, there are three objects distinguished: principals (parties involved in authentication protocol), encryption/decryption keys, and formulas (called statements). Messages are identified with statements.
Typically, BAN logic uses these symbols:

-	A, B, S	: denote specific principals
-	K_{ab}, K_{as}, K_{bs}	: denote shared key between principals
-	K_a, K_b, K_s	: denote public key of principal
-	K_a^{-1}, K_b^{-1}, K_s^{-1}	: denote the corresponding secret / private key
-	N_a, N_b, N_s	: denote specific statements (nonce, etc)

The symbols P, Q, and R range over principals; X and Y range over statements; and K ranges over encryption / decryption keys.

The only propositional connective in BAN Logic is conjunction, denoted by a comma, and properties such as associative and commutative are also taken for granted. In addition to conjunction, the following constructs are used:

- **P *believes* X**: P believes X or P would be entitled to believe X. The principal X may act as though X is true.
- $P \Delta X$: **P *sees* X**: P can read and repeat X (possibly after doing some decryption).
- **P *said* X**: P once said X. The principal P at some time sent a message including the statement X.
- P ⇒ X**: P *controls* X**: P has jurisdiction over X. The principal P is an authority on X and should be trusted on this matter.
- ***#(X): fresh*(X)**: the formula X is fresh, in a way that X has not been sent in message at any time before the current of the protocol.
- $P \xleftrightarrow{K} Q$: P and Q may use the shared-key K to communicate. The key K is good, in that it will never be discovered by any principal except P or Q, or a principal trusted by either P or Q.
- $\overset{K}{\mapsto} P$: has K as public key. The matching private key (K^{-1}) will never be discovered by any principal except P, or a principal trusted by P.
- $P \overset{X}{\Leftrightarrow} Q$: The formula X is a secret known only to P and Q, and possibly to principals trusted by them. An example of a secret is a password.
- $\{X\}_K$: This represents formula X encrypted under the key K.
- $\langle X \rangle_Y$: Thie represents X combined with formula Y. It is intended that Y be a secreat and that its presence is proof of origin for X.

(1) The *message-meaning rules* concern the interpretation of messages. They explain how to derive beliefs about the origin of messages.
For shared keys:

$$\frac{P\ \textit{believes}\ Q \xleftrightarrow{K} P,\ P\ \Delta\{\mathrm{X}\}_K}{P\ \textit{believes}\ Q\ \textit{said}\ X}$$

For public keys:

$$\frac{P\ \textit{believes}\ \overset{K}{\mapsto} Q,\ P\ \Delta\{X\}_{K^{-1}}}{P\ \textit{believes}\ Q\ \textit{said}\ X}$$

For shared secrets:

$$\frac{P\ \textit{believes}\ Q \overset{Y}{\Leftrightarrow} P,\ P\ \Delta\langle \mathrm{X}\rangle_{\mathrm{Y}}}{P\ \textit{believes}\ Q\ \textit{said}\ X}$$

(2) The *nonce-verification rule* expresses the check that a message is recent, thus the sender still believes in it:

$$\frac{P\ \textit{believes fresh}\ (X),\ P\ \textit{believes}\ Q\ \textit{said}\ X}{P\ \textit{believes}\ Q\ \textit{believes}\ X}$$

(3) The *jurisdiction rule* states that if P believes that Q has jurisdiction over X, then P trusts Q on the truth of X:

$$\frac{P\ \textit{believes}\ Q \Rightarrow X,\ P\ \textit{believes}\ Q\ \textit{believes}\ X}{P\ \textit{believes}\ X}$$

(4) A necessary property of the ***belief*** operator is that P believes a set of statements if and only if P believes each individual statement separately. This justifies the following rules:

$$\frac{P\ \textit{believes}\ X, P\ \textit{believes}\ Y}{P\ \textit{believes}\ (X,\ Y)}\ ,\ \frac{P\ \textit{believes}\ (X,Y)}{P\ \textit{believes}\ X}\ ,\ \frac{P\ \textit{believes}\ Q\ \textit{belives}\ (,Y)}{P\ \textit{believes}\ Q\ \textit{believes}\ X}$$

(5) Similar rule applies to the said operator:

$$\frac{P\ \textit{believes}\ Q\ \textit{said}\ (X,\ Y)}{P\ \textit{believes}\ Q\ \textit{said}\ X}$$

(6) If a principal see a formula, then he also sees its components, provided he knows the necessary keys:

$$\frac{P\Delta\ (X,\ Y)\ ,}{P\ \Delta\ X}\qquad \frac{P\ \Delta\ \langle X\rangle_{Y}\ ,}{P\ \Delta\ X}\qquad \frac{P\ \textit{believes}\ \ Q\xleftrightarrow{K}P,\ P\ \Delta\ \{X\}_{K}}{P\ \Delta\ X}$$

$$\frac{P\ \textit{believes}\ \overset{K}{\mapsto}P,\ P\ \Delta\{X\}_{K}\ ,}{P\ \Delta\ X}\qquad \frac{P\ \textit{believes}\ \overset{K}{\mapsto}Q,\ P\ \Delta\ \{X\}_{K\text{-}1}}{P\ \Delta\ X}$$

(7) If one part of a formula is fresh, then the entire formula must also be fresh:

$$\frac{P\ \textit{believes}\ \#(X)}{P\ \textit{believes}\ \#\ (X,\ Y)}$$

(8) The same key is used between a pair of principals in either direction. The following two rules reflect this property:

$$\frac{P\ \textit{believes}\ \ R\xleftrightarrow{K}R'\ \ ,}{P\ \textit{believes}\ \ R'\xleftrightarrow{K}R}\qquad \frac{P\ \textit{believes}\ Q\ \textit{believes}\ \ R\xleftrightarrow{K}R'}{P\ \textit{believes}\ Q\ \textit{believes}\ \ R'\xleftrightarrow{K}R}$$

(9) A secret can also be used between a pair of principals in either direction. The following two rules reflect this property:

$$\frac{P\ \textit{believes}\ R\overset{X}{\Leftrightarrow}R'\ \ ,}{P\ \textit{believes}\ \ R'\overset{X}{\Leftrightarrow}R}\qquad \frac{P\ \textit{believes}\ Q\ \textit{believes}\ R\overset{X}{\Leftrightarrow}R'}{P\ \textit{believes}\ Q\ \textit{believes}\ \ R'\overset{X}{\Leftrightarrow}R}$$