International Journal of Engineering Research and Development
***e*-ISSN: 2278-067X, *p*-ISSN: 2278-800X, www.ijerd.com**
Volume 3, Issue 10 (September 2012), PP. 29-34

Linux Based Implementation of MACSec Key Agreement (MKA)

Mr. Anand G S[1], Mrs.Sridevi K N[2], Dr. Jitendranath Mungara[3]
[1]*M.Tech, Computer Network Engineering.*
[2]*Associate Professor, Dept of CSE, CMR Institute of Technology, Bangalore*
[3]*Professor / Dean P.G Program, Dept of CSE/ISE, CMR Institute of Technology, Bangalore*

Abstract—IEEE 802.1AE and IEEE 802.1AF are two IEEE 802.1X standards providing security to the data link layer. The IEEE 802.1AE is the IEEE MAC Security standard (also known as MACSec), it defines data confidentiality and integrity for media access independent protocols but lacks in providing key management and the establishment of secure associations. IEEE 802.1AF MACSec Key Agreement will facilitate secure communication over publicly accessible LAN/MAN, with key management and establishment of secure associations, by exclusive use of secret key cryptographic algorithms .This paper will give us an insight of MACSec Key Agreement (MKA) is implemented in the Linux environment.

Keywords—MKA, CA, SA, CAK, SAK, ICK, KEK, EAPOL

I. INTRODUCTION

Without a stable and secure Ethernet, it is hard to assure the service security of whole telecommunication network. International Standard Organization has instituted several Ethernet security standards including IEEE 802.1AE and IEEE 802.1AF. IEEE 802.1AE had defined the infrastructure of secure MAC transmission. MACSec does not directly address how keys are obtained for encryption. IEEE 802.1AF, a secure key agreement and management schemes, including secure key generation and distribution, identification of Live Peer Lists and Potential Live Peer Lists. Key agreement means that the sender and the receiver negotiate on how to generate shared keys. Asymmetric or symmetric key technology can be used during this process. The main issues in key management are who is responsible for generating keys, how to generate keys and when keys generate. Connectivity Association Key (CAK), is a master secret key. The possession of the CAK is suitable for proof that it has been authenticated using an IEEE 802.1X framework, and it is authorized to participate on a particular LAN.A key establishment protocol is required to generate one or more Secure Association Keys (SAKs), which are the secret keys that IEEE 802.1AE uses to encrypt data packets on the LAN and for secure transmission.

II. TERMINOLOGY

Advanced Encryption Standard (AES): FIPS approved symmetric block cipher cryptographic algorithm used for protection of data.
Association Number (AN): It is concatenated with secure channel identifier, to identify a secure Association.
Connectivity Association (CA): A security relationship which is established and maintained by key agreement protocols, which consists of fully connected service access points attached to single LAN.
Connectivity Association Key (CAK): The key associated with the CA.
CAK Identifier (CKI): An identifier for a particular CAK.
Secure Association Key (SAK): The secret key used by a SA.
SAK Identifier (SKI): For identifying a particular SAK.
Secure Channel (SC): A security relationship used to provide security guarantees for the transmission of data between the members of CA.
Key Establishment: Process where cryptographic keys are securely established and exchanged among cryptographic modules.
Key wrapping: A method of encrypting keys which provide confidentiality and integrity protection using symmetric key.
Random Number Generator (RNG): An algorithm used for producing/generating random number, which is used in generation of keys and cryptographic application.

Acronyms

AES Advanced Encryption Standard
AN Association Number
CA Connectivity Association
CAK Connectivity Association Key
CKI Connectivity Association Key Identifier
RNG Random Number Generator
SAK Secure Association Key
KEK Key Encrypting Key

ICK Integrity Check Value Key
SC Secure Channel
SKI Secure Association Key Identifier

III. EXISTING SYSTEM

The IEEE 802.1AE Standard for Local and Metropolitan Area Networks (LAN/MANs): MAC Security specifies how all or a part of a network can be secured transparently to peer protocol entities that use the MAC Service provided by IEEE 802 LANs to communicate. The standard defines MAC security (MACSec) entities in end stations that provide connectionless user data confidentiality, frame data integrity, and data origin authenticity utilizing the IEEE Standard 802.1X.
MACSec (IEEE 802.1AF) was designed with the following salient features:

- MACSec defines the layer 2 security protocols that provide origin authentication, data integrity checking, and data confidentiality. It defines a frame format.
- Includes data encapsulation, encryption, and authentication.
- KeySec defines the key management protocol for MACSec. MACSec supports point-to-point connections in a hop-by-hop architecture

However, MACSec does not specify how the relationships between MACSec protocol peers are discovered and authenticated, as supported by key management or key distribution protocols. To overcome these drawbacks of MACSec a new standard called IEEE 802.1AF: MACSec Key Agreement was defined.

IV. PROPOSED SYSTEM

The MACSec Key Agreement (MKA) protocol allows PAEs, each associated with a Port that is an authenticated

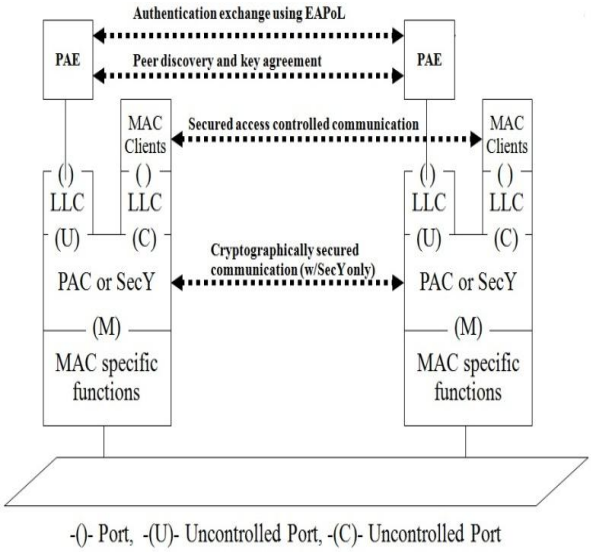

Figure 1: Interaction of Two PAE's

member of a secure connectivity association (CA) or a potential CA as shown in Fig 1, to discover other PAEs attached to the same LAN, to confirm mutual possession of a CAK and hence to prove a past mutual authentication, to agree the secret keys (SAKs) used by MACSec for symmetric shared key cryptography, and to ensure that the data protected by MACSec has not been delayed.

MKA comprises a secure fully distributed multipoint-to-multipoint transport and a number of applications of that transport, including the distribution of SAKs by an elected key server using AES Key Wrap. In addition to distributing fresh SAKs, MKA manages their installation and use by the SecY that secure the data transmitted and received by each Controlled Port, ensuring that each is capable of receiving data protected by that SAK before it is used for transmission.

A) Key Server & Key Hierarchy

The MKA participant can assume either the role a Key server (who will have some special responsibilities) or a normal participant. Whenever the LAN network comes up for the first time OR there is any change in the nodes of the LAN, the key server is determined by a process called as Key Server Election. As soon as the MKA participant is create it send outs the live participant EAPoL MKA MKPDU with the encoded Key server priority (an 8-bit unit). The participant's keeps sending out these messages until it receives any other live participant message. As soon as it receives any the participants with highest Key Server priority will be elected as the Key server.
Elected Key server has the following special responsibilities:

- Deciding on the use of MACSec
- Cipher suite selection
- SAK generation and distribution

- SA Assignment

MKA module (process) will be triggered when a CAK, CKN tuple becomes available as a result of the configuration of a pre-shared key or the SecY module completion of the initial dot1x authentication phase. The Key Encryption Key (KEK) & ICV Key (ICK) are 128-bit keys derived from CAK using the AES Cipher in CMAC mode as shown in Fig 2. The CAK is not used directly. The derived keys are tied to the identity of the CAK.

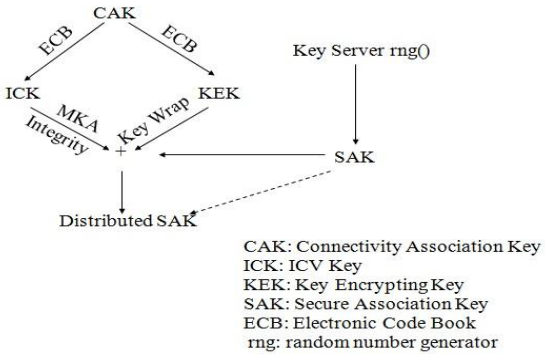

Figure 2: Key Hierarchy

B) SAK Generation & Distribution.

The Key Server is responsible for generating and distributing MACSec SAKs, using AES Key Wrap. Each SAK is identified by a 128-bit Key Identifier (KI), comprising the Key Server's MI (providing the more significant bits) and a 32-bit Key Number (KN) assigned by that Key Server (sequentially, beginning with 1). Each KI is used to identify the corresponding SAK for the purposes of SAI assignment.

MI is the random number generated using a strong random number generator. MKPDUs should enforce in-order delivery and message numbers (MN) in MKPDUs are used for this purpose. When a MKPDU is received, all the prior Message Numbers (MN) received should be discarded. MN initially starts with the value 1. When MN reaches its upper limit, the PAE chooses a new random number for Member Identifier (MI) and start MN with value 1.

Each participant that considers it to be the current Key Server can distribute an SAK by encoding the following information in transmitted MKPDUs:

- Distributed SAK, the SAK, protected by AES Key Wrap.
- The KN, 32 bits.

C) EAPoL PDU

The EAPOL PDUs exchanged between peer PAEs to support authentication using EAP to support the MACSec Key Agreement protocol and to announce network identities and other access point capabilities. The Figure 3 shows the frame format of the EAPoL.

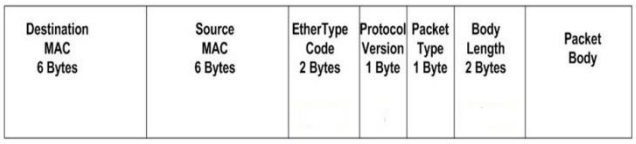

Figure 3: EAPoL Frame Format

The Table 1 shows the packet type and the values corresponding to them.

Packet Type	Value
EAP-Packet	0000 0000
EAPoL-Start	0000 0001
EAPoL-Logoff	0000 0010
EAPoL-Key	0000 0011

EAPoL-ASF Alert	0000 0100
EAPoL-MKA	0000 0101

Table 1: Packet Type

V. CRYPTOGRAPHIC OPERATION

Stations using this protocol must have the following capabilities:

- AES protocol with 128 bit key and Electronic Code Book (ECB) and cipher based message authentication code (CMAC) modes of operation.
- Strong Random Number Generator (RNG).

A) SAK Generation

SAK's are generated using the strong RNG, approved by FIPS. The SKI identifying a SAK is also generated using RNG.

B) Deriving Keys from the CAK

CAK are the long term shared secret key available between the two stations. This key is used for two purposes: to encrypt SAK's and provide an integrity check. In order to use CAK for these two keys called KEK and ICK are derived from it.

C) SAK Distribution

SAKs are distributed from station generating the key to their station on LAN. This SAK must be encrypted during the transmit so that only authorized stations can recover the key. The SAK is encrypted using KEK, which is derived from CAK. The KEK is given as input to the key wrapping algorithm to protect the SAK between stations. The default algorithm is AES key wrap.

D) Message Authentication

It is achieved by using the ICV present in each message. The ICV is computed as a cryptographic operation over the bytes of the message with secret key. The key used in the ICV generation is ICK derived from the CKA. The default algorithm is CMAC using AES-128 key.

VI. MKA IMPLEMENTATION

The Figure 4 shows the components involved in the MKA module and their interactions with other Linux software. MKA module communicates with each of the components using an abstraction layer. The MKA module will be responsible for Key Generation, Key distribution and SA installation to SecY entity. The MKA module interacts with crypto toolkit module for all Key derivations, Random number generation, encryption and data integrity APIs. The MACSec layer will provide APIs to send & receive EAPoL MKA messages.

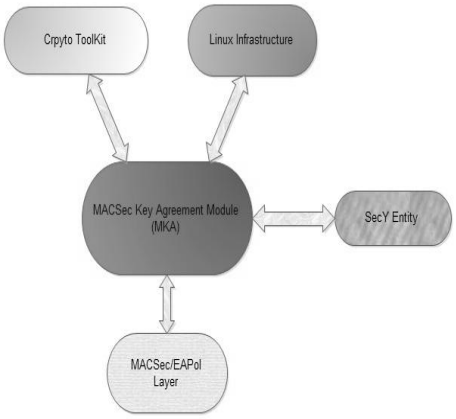

Figure 4: MKA Implementation

The MKA module interacts with SecY module in two different ways, one is that SecY module triggers the MKA module for the creation of the MKA entity when the CAK/CKN is available (this will be the result of initial authentication phase during the dot1x initial exchanges) and that the MKA modules installs the SA which contains encryption, ICV keys to the SecY entity. The module will also use of some of the Linux infrastructure for timers.

"The Linux Based Implementation of MKA" consist of total three modules. The implementation of these three modules is described as follows:-

A) Identity Verification Module

This module facilitates the identity verification of the peer which wants to exchange data. In this module there will be exchange of EAPoL PDU between peers. The server peer will then authenticate if the peer entity provides a valid identity that was provided during registration.

Algorithm:
Step 1: Server will be waiting for the connection by sending out EAPoL Announcement by telling what kind of service it provides along with NID.
Step 2: Client on receiving the EAPoL Announcement and sends EAPoL Start along with its NID.
Step 3: Server will then process the packet and will construct an EAPoL EAP packet requesting for clients Identity.
Step 4: Client will then provide with identity through the EAPoL EAP response packet and waits for the server conformation.
Step 5: Server will then authenticate the client's identity. If the information provided by the client is false the server will then send a EAPoL EAP Failure packet.
Step 6: If the identity provided by the client is true ,the server will send EAPoL EAP SUCCESS packet and initiates the MKA Establishment module.

B) MKA Establishment
This module is invoked after the identity verification is done. In this module the two peer entities will come to agreement on the key server, use of MACSec and the cipher suite to use during the data exchange.
Algorithm:
Step1: MKA starts on client being authenticated by the server.
Step 2: Server and client will now exchange EAPoL MKA deciding on all the parameters.
Step 3: The key server is elected based on the KEY SERVER Priority (8bit).This is encoded into the MKPDU and distributed. The highest key server priority is selected as key server or if the key server priority is same SCI MAC address as priority
Step 4: Key server elected will have to do the following:

- Use of MACSec: participant will advertise if they want to protect data using the MACSec.
- MKPDU encodes following MACSec capability and MACSec desired flag if participant wants to use MACSec.
- Cipher Suit Selection: Key sever will encode the MKPDU with each distributed SAK: Default cipher suit GCM AES 128.

Step 5: After the establishment, both the peers will use CAK for deriving the two keys called ICV and KEK using a key derivation module.
Step 6: The key server is responsible for SAK generation.SAK is also generated by initiating key generating module and it is encrypted by AES key wrap using KEK.
Step 7: The SAK protected wit AES along with the ICV, are encoded into the EAPoL MKA and then transmitted.SAK is used for encoding the data exchanged between the two entities.

C) Key Generation and Cryptography
This module is invoked during the MKA establishment module for the generation of keys and cryptography. This module will generate KEK, ICK, and SAK. These keys are used provide secure communication between the two peer entities.
Input: Key, a key derivation key of 128 or 256 bits
Label, a string identifying the purpose of the keys derived using this KDF.
Context, a bit string providing context in identifying the derived key.
Length, the output length in bits encoded in two octets with the most significant octet first.
Output: a Length-bit derived value.
Fixed values:
h, the length of the output of the PRF in bits
r, denoting the length of the binary representation of the counter i

Code:
iterations ← (Length + (h-1))/h
if iterations > 2r-1, then indicate an error and stop.
result ← ""
do i = 1 to iterations
Result ← result | PRF(Key, i | Label | 0x00 | Context | Length)
do
Return first Length bits of result, and securely delete all unused bits

VII. RESULT

"Linux Based Implementation of MACSec Key Agreement" was successfully tested between the peers in Linux environment and the SAK was successfully exchanged between them.

VIII. CONCLUSION

IEEE 802.1AF MACSec Key Agreement (MKA) protocol allows an authenticated member of a secure connectivity association, to discover other member attached to the same LAN, confirm mutual possession of a CAK and to agree upon the secret keys (SAKs) used by MACSec for symmetric shared key cryptography, and to ensure that the data protected by MACSec .

REFERENCES

[1]. Hayriye C. Altunbasak: Layer 2 Security Inter-Layering In Networks, Georgia Institute of Technology December 2006.
[2]. Security in the Data Link Layer (Layer 2) : Hayriye Altunbasak, Sven Krasser, HenryL.Owen,Jochen Grimminger,Hans-Peter Huth,Joachim Sokol
[3]. Romanow, A.: Media Access Control (MAC) Security. IEEE 802.1 AE (2006).
[4]. Weis.B: Security considerations and proposal for MACSec key establishment. (2009).
[5]. IEEE 802.1AF, Draft Standard for Local and Metropolitan Area Networks Port-Based Network Access Control-Amendment 1: Authenticated Key Agreement for Media Access Control (MAC) Security, Nov 2007.
[6]. Mishra, A., Arbaugh, W.: An initial security analysis of the IEEE 802.1 X standard. (2002).
[7]. IEEE STD 802.1X-2010, IEEE Standard for Local and Metropolitan Area Networks—Port-Based Network Access
[8]. Control.
[9]. "IEEE 802.1AE-Media Access Control (MAC) Security," July 2006.
[10]. Jyh-Cheng Chen and Yu-ping Wang, National Tsing Hua university: Extensible Authentication Protocol (EAP) and IEEE 802.1x: Tutorial and Empirical Experience.
[11]. Design of MACSec (802.1AE) Jun-Won Lee1, Seon-Ho Park1, Ki-Ho Gum, and Tai-Myoung Chung
[12]. IETF RFC 3268, Advanced Encryption Standard (AES) Cipher suites for Layer Security (LS), Chown, P., June 2002