The comparative cost analysis of EAP Re-authentication Protocol and EAP TLS Protocol

Seema Mehla
Computer Department
N.C.College of Engineering
Panipat, India

Bhawna Gupta
Computer Department
N.C.College of Engineering
Panipat, India

***Abstract*—the Extensible Authentication Protocol (EAP) is a generic framework supporting multiple types of authentication methods. In systems where EAP is used for authentication, it is desirable to not repeat the entire EAP exchange with another authenticator. The EAP re-authentication Protocol provides a consistent, method-independent and low-latency re-authentication. It is extension to current EAP mechanism to support intra-domain handoff authentication. This paper analyzed the performance of the EAP re-authentication protocol, and compared it with that of the EAP-TLS protocol. The result shows that the authentication delay of EAP re-authentication protocol is only twentieth of that in the EAP-TLS protocol.**

Keywords- ERP; EAP-TLS; EMSK; RADIUS

I. INTRODUCTION

While accessing the wireless n/w the security is always concerned with mobility management. Mobile terminal when roams to the new network access server (NAS), it needs to be authenticated by the new NAS. The new NAS usually does not have trust relationship with the mobile terminal. So, a three-party authentication technique such as Extensible Authentication Protocol (EAP) is used for addressing it, where the new NAS acts as a pass through authenticator, and it relays the authentication messages to the home authentication server of the mobile terminal through the foreign authentication server (figure.1).

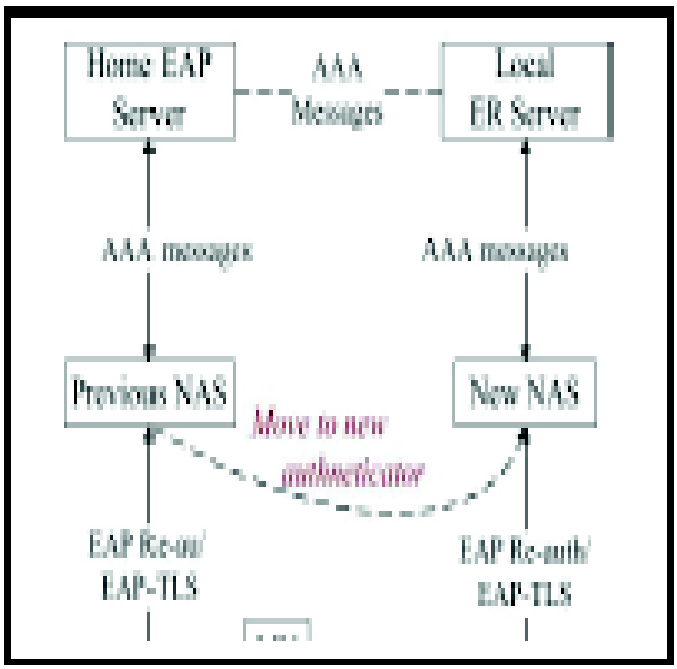

Figure 1 Three party authentications

When the peer moves and attach to a new authenticator, the latter will consult the home authentication server for authentication, which is "proxied" by the local authentication server. Taking EAP-MD5 as example, the re-execution of full EAP exchange involves an EAP-Response/Identity message, followed by one round trip to perform authentication, more than 3-round trips are needed plus with the EAP-Finish message. It causes costly distributions of key material, multiple message delivery and more significant security computation. For the implementation of EAP-MD5 more than 70ms [20] are not suitable for latency sensitive services such as VOIP. To optimize the performance of intra-domain re-authentication, some EAP methods have been designed such as the EAP-TLS, which is an EAP integration of the TLS protocol supporting either one-way or mutual authentication by using digital certificates. A per-session WEP key could be set up to implement the re-authentication and re-keyed on the peer. However, the problem with EAP-TLS is that it requires the PKI

infrastructure to handle certificates, so it is difficult for many private users to deploy. In addition to that the way certificates issued requires multiple rounds of message delivery between the peer and the server. To overcome this problem, EAP-TTLS and PEAP aiming at achieving a similar level of security without client certificates. They both rely on EAP-TLS tunnels with server. But these methods are rather insecure to the peer and more complicated for private users [19]. The Extended Master Session Key (EMSK) generated in most EAP exchange are not presently used for any re-authentication specifications, its most common usage is only to derive Transient Session Keys (TSKs) to provide access link security in networks (e.g., IEEE 802.11i, IEEE 802.16e). In the EAP re-authentication Protocol (ERP), designed by the Handover Keying Working Group of IETF, an Extended Master Session Key (EMSK) was derived in initial EAP exchange, the peer and the ER Server use the EMSK to derive a re-authentication Root Key (rRK) for subsequent handover authentication. Each time the peer authenticates to a new authenticator, a re-authentication MSK (rMSK) generated from the rRK at the local server, would be carried by an EAP-Re-auth message via the new authenticator and also distributed to it. Meanwhile, with the rMSK derived at the peer after ERP exchange, the local server has low-latency connectivity to the peer, allowing the peer re-authenticate locally without communicate with its home server, thus the ERP specifies a method-independent and efficient re-authentication. The key elements in managing mobility and optimizing efficiency of re-authentication in wireless access mainly focus on the two aspects, (1) the time consumed in the message exchange, (2) the security burden of EAP Server result from computation and verification. These are the main aspects we consider to analysis the fast re-authentication scheme versus the EAP-TLS.

This paper is organized as follows: section II introduces the ERP protocol. Section III gives the general calculation of consumption result from HMAC-SHA algorithm, which is used for key generation and integrity checksum in ERP. Section IV gives the comparison between the ERP and the EAP-TLS stand on their packet size, the encryption and verify cost. After that comparison of runs of message delivery in section V, section VI gives the conclusion and show the time saved by the ERP protocol is more.

II. ERP EXCHANGE

When a peer first attaches to the network and performs a full EAP exchange with the EAP server. There is difference between the EAP and ERP in key usage and that is the latter prescribes the generation and deliberation of EMSK, to generate rRK and rIK for subsequent efficient re-authentication [5]. Another important key is the Domain Specific Root Key. Then the domain-specified keys generated from DSRK would be used to derive DS-rMSK for efficient re-authentication. Where there was a local ER server present between the initial authenticator and the home server [4].That is the establishment of the trust relationship between the Local ER server and the peer via the new authenticator. As in figure 2, the handover begins with EAP-Initiate/Re-auth-Start and EAP-Initiate/Re-auth messages between the mobile terminal (peer) and the new authenticator. Detailed content of the message packets will be shown in latter tables for security analysis. The Local server derives an rMSK using HMAC algorithms in the third step.

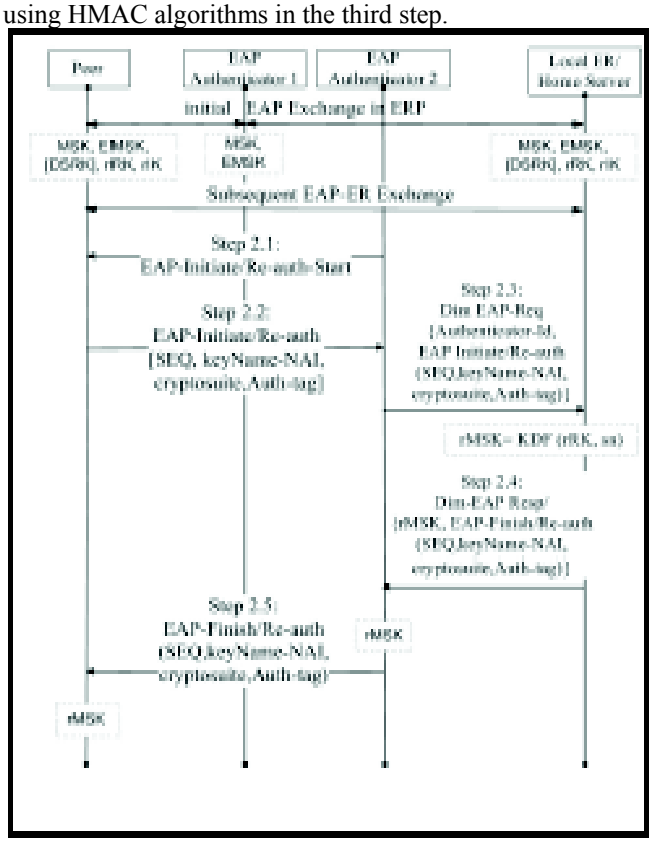

Figure 2. The whole ERP Exchange

The authenticator extracts the rMSK and forwards an EAP-Finish/Re-auth message to the peer. Then the peer uses sequence number to compute the rMSK as the final step. For transmitting EAP information between the authenticator and the server or among the servers, AAA protocol such as RADIUS or DIAMETER [2] should encapsulate the ERP message as an EAP-Payload AVP attribute (i.e. the rMSK encapsulated as an EAP-Master-Session-Key AVP).

III. EFFICIENCY ANALYSIS OF HMAC-SHA256 SECURITY ALGORITHM

Secure Hash algorithms SHA256 is an iterative, one-way hash function that could be used in Conjunction with cryptographic algorithms for performing authentication. It process input text blocks of B =512 bits to generate L=128 bit hash values, for the verification of the correct message transfer, apply padding to make the plaintext a multiple of 512 bits. However, the SHA256 cannot be directly used as "message authentication codes" (MAC) algorithm, as it does not include a secret key. Therefore, combine the SHA256 with HMAC [10], a mechanism that provide integrity check based on a secret key: *HMAC-SHA256 (K, S) = SHA256 (K .opad, SHA256 (K .ipad, 'data')).* *(1)*

The K denotes an input secret key, the ipad and opad are two fixed bytes (0x36 and 0x5C) repeated B times respectively. Plus 210 operations per block for initiation and termination [11].

Table 1 SHA-256 operations

Basic functions in SHA256 & their operations	
$Ch(x,y,z) = (x \wedge y) \oplus (\_x \wedge z)$	12
$Maj(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$	13
$\Sigma_0(x) = ROTR^2(x) \oplus ROTR^{13}(x) \oplus ROTR^{22}(x)$	37
$\Sigma_1(x) = ROTR^6(x) \oplus ROTR^{11}(x) \oplus ROTR^{25}(x)$	36
$\sigma_0(x) = ROTR^7(x) \oplus ROTR^{18}(x) \oplus SHR^3(x)$	36
$\sigma_1(x) = ROTR^{17}(x) \oplus ROTR^{19}(x) \oplus SHR^{10}(x)$	50
Steps in SHA256:	
Prepare message schedule $\{Wt\}$	200
initialize a,b,c,d,e,f,g,h:	24
Computation from 0 to 63	
$T_1 = h + \Sigma_0(e) + Ch(e,f,g) + Kt + Wt$ $= h + (ROTR^2(e) \oplus ROTR^{13}(e) \oplus ROTR^{22}(e)) + ((e \wedge f) \oplus (\_e \wedge g)) + Kt + Wt$	181
$T_2 = \Sigma_0(a) + Maj(a,b,c)$ $= ROTR^2(a) \oplus ROTR^{13}(a) \oplus ROTR^{22}(a) + ((a \wedge b) \oplus (a \wedge c))$	72
h=g, g=f, f=e, e=d+T, d=c,	47
c=b, b=a, a= $T_1 + T_2$	41
Total of Computation	21824
Compute intermediate hash value H	8
Total of n_k times	$22084n_k$

According to (1), the total number of operations needed for HMAC-SHA256 is:

$T(nk) = 210 + 22084\,nk$ (2)

The nk represents the nk -block input data to be encrypted. The required authentication and verification time for HMAC-SHA-256, T (nk, Cp), as a function of the number of input blocks and the processor speed is:

$T(n_k, Cp) = (210 + 22084 n_k)/\mathrm{Cp}.$ (3)

$n_k = N/512 = (8 \times Sd + Sp + Ss + K)/512.$ (4)

Takes Sd-byte data to be encrypted as an example, N is the N-bit total encryption data, Sp-bit is the length of padding field, Ss-bit is the length of the Size filed and the K-bit denotes the extra appended inner form of the key.

IV. SECURITY COST COMPARISON BETWEEN THE ERP AND THE EAP-TLS PROTOCOLS

This section focuses on authentication / encryption space complexity and computation time complexity. In general, the ciphering delay depends on the packet overhead related with the packet format and selected security algorism. We use the default security algorithms for comparison in followed paragraphs.

(A). ERP Security Cost of Message Integrity Calculation

Table 2. Notation Definition

Symbol	Description
S_{dn}	the delivery cost, n denote the n^{th} step of message delivery
D_{pan}, D_{pas}	the average distance between the peer & authenticator, the peer & Server
C_{tr}	the total packet delivery cost in re-authentication
C_{tt}	the total transmission of data packet in EAP-TLS procedure
S_{msk}	the key generation of rMSK at the peer and the server
S_{rik}	the key generation of rIK at the peer
S_{ic}	the computation of integrity checksum at the peer and the server
S_{erp}	the total security cost in Re-authentication
S_{tls}	the total security cost in EAP-TLS

The EAP-Initiate/Re-auth-Start Packet takes at least one Domain-Name-NAI TLV (Table 3), plus its header, the total length should be:

L_{ER} *(EAP-Initiate/Re-auth-Start)* $= L_H + L_{TLV}$ *(Domain-Name-NAI) = 48 bits +* $L_{TLV} \sim 100$ *bits*

Except the Start message, EAP Re-auth messages all have the same length of packet header and integrity protected with The Authentication Tag [4]. The header length, LEH is 64 bits. The length of crypto suite field, Lc is 1 octet. The authentication tag contains the integrity checksum over the ERP packet excluding the AT filed itself, thus the LAT should always be 272 bits in our paper. As described in [4], the key Name-NAI should be present in the EAP-Initiate-Re-auth message and the EAP-Finish Re-auth message to identify the ER server's domain and the rIK. The username portion of the key Name-NAI takes up128 octets and the realm portion should be the domain nameof the rIK's parent key (EMSK/DSRK). L_{keyame}

=1-octet (type) +1-octet (length) + value payload
=16 bits + username length + realm length
=18bits+16×8bits+ realm portion length
= 144 bits +L' $\sim$ *200bits.* (5)

Thus the general formulae to calculate the length of different EAP /Re-auth packets should be:

L_{ER}

$= L_H + L_{TLV}$ *(one or more)* $+ L_{AT} + L_C$
= 64 bits + L_{TLV} *+ 272 bits+128 bits*
$= 464$ *bits +* L_{TLVs}. (6)

While the variable T_{LV} lengths, L_{TLV} here as in table 3,

Table 3. TLV Attribute Format

Value	Notation	Attribute Type	NAI length (bit)
'1'	keyName-NAI	TLV	≈200
'4'	domain name	TLV	≥64
'6'	Authorization Indication	TLV	272

Submitting the LTLV to (6), we could calculate the lengths of the EAP-Re-auth messages (table 4). The size of the Diameter Packet, LD includes the length of the Diameter Header LDH and the length of the ERP AVPs:

$L_D = L_{DH} + L_{AVPs} = 5\times32$ *octets* $+ L_{AVPs} = 160$ *bits* $+ L_{AVPs}$.

The L_{AVPs} represents the length of the EAP AVPs such as the rMSK AVP, add the AVP header to its data filed, the general Diameter Packet length is at least:

$L_D = L_{DH} + L_{AVP} = 160$ *bits* $+ L_{AVP}$

$\tilde{}$ *160bits + 256bits + 64bits = 480 bits.*

And with the length of the EAP /Re-auth message involved in the AAA message, the packets transport between the NAS and the Server should be: $L_D + L_{ER} = 480$*bits* $+100$*bits* $+ L_{TLVs} \tilde{} 580$ *bits* $+ L_{TLVs}$ Upon the receipt of message, the peer should demonstrate possession

Table 4. EAP Re-auth packet size & CPU cycles

Packet	Total Length (bits)	n_k	Key Generate
EAP-Initiate/Re-auth-Start	100	1	rIK
EAP-Initiate/Re-auth	680	2	
Dim EAP (EAP-Initiate/Re-auth)	1052	3	rMSK
Dim EAP (rMSK, EAP-Initiate/Re-auth)	1308	3	
EAP-Finish/Re-auth	680	2	rMSK
Total cpu cycles	243974		

Of the rIK by computing the integrity checksum over the EAP-Initiate/Re-auth message. After the peer chooses the integrity algorithm, the server will use the same algorithm with a given rIK for all ERP messages. Since the HMACE-SHA256 as the default integrity algorithm of ERP [1], according to *(3)* and *(4)*, the computation of the checksum should be:

Checksum = K (K, S)

= HMAC-SHA256 (rIK, EAP-Initiate/Re-auth message)

We defined Sic represents the computation cost of integrity check at the peer and the server, the input of the computation and time cost (n_k, $T(n_k)$) were calculated in table IV.

(B). ERP Security Cost of Key Generation

1. SrIK: The calculation of rIK is generated in HMAC algorithm , the rIK Label is length of 40 octets, the length field refers to the length of the rIK is 2 octets, adding with the crypto suite encoded as a 1 octet number, the length of the S is *(40+1+2+1) octets = 352 bits*. Because the length of the K is the length of the rRK, which should be the same as the EMSK/DSRK, at least *64 octets = 512 bits*. As (4) and evaluation of the rIK generating total input, we gave the number of the SHA256 operation nk and the function of the number of input blocks and processor speed T(nk): *nk = (352+160+8+64×8)/512 ≈2.*

T(nk) =210 + 2×22084 = 44378.

2. SrMK: Meanwhile, as the rMSK Generation [4], the length of the S consists of the lengths of the rMSK label, the SEQ and the derived rMSK. We could see the rMSK label as an 8-bit ASCII string, length of 35×8 bits; The SEQ encoded as a 16-bit number and the "\0" is a NULL octet. The length tag is in length of 16 bits. Thus the length of the S should be:

Sd = (35 ×8 + 16 +1 ×8 + 16) bits = 320 bits.

The rMSK is a kind of USRK[9], which is a 2-octet unsigned integer in network byte order, thus the length of the rRK/rIK should be the same as 2 octets. Similar as the evaluation of rIK key generating equation, the number of the SHA256 operation nk and the time of rMSK generation SrMSK should be:

$n_k = 2$, $S_{rMSK} = T(n_k) = 44378$.

3. The security Cost of ERP: Because the Security cost per step in a network node is: *Security cost = time of key generating + time of encryption / decryption + time of verify* Then the Cs-erp, represent the total security cost in the procedure of Re-authentication according to above section, involving two times of rMSK generation (at the peer and the server respectively) , five times of verify of authentication tag (each step in the re authentication exchange), should be:

Ss-erp = 2×SrMSK +SrIK + Sic= 3×44378/Cp + 243974 /Cp = 377108/Cp. (7)

(C). EAP-TLS Security Cost

EAP-TLS supports two methods for generating keying material. One is RSA encryption based (RSA case) and the other is based on a Diffie Hellman key exchange (DHE case). In the DHE case, the server uses a Server-Certificate of type DHE-RSA or DHE-DSS and following with a Server-Key-Exchange message, including the server's public DH value. Because in RSA case the server uses a certificate of type RSA without sending Server-Key-Exchange, we select this scheme to calculate the security cost of EAP-TLS. In RSA case, neglect the encryption implementation for its small exponent (mostly only 16 bits); we only consider the time cost of RSA decryption. The time of a 1024-bit modular exponentiation (decryption side of 2048-bit RSA), is about 450,000 CPU cycles on a 64-bit computer [15], which is equivalent to that of the 256 bits modular exponentiation on a 32-bit computer. As the description in [15], the 256 bit exponentiations costs one of sixteenth of the 1024-bit exponentiations ((256/1024) ^2=1/16), thus we gain the cost of 2048-bit RSA decryption on a 32-bit computer is *450,000×16 = 7,200,000 CPU cycles*. In a conclusion,The total security cost of EAP-TLS (RSA case), including two times of encryption and one time of decryption, on a Dual Pentium II-350 (350MHz) is about:

Stls=7,200,000/ 350,000,000 = 20.57 ms .For easy to compare, we used the Cp in (7) equal to 350MHz, thus the Ss-erp should

be *377,108/Cp =1.08ms.* This saved about 19 ms than the EAP-TLS in term of security cost.

V. MESSAGE DELIVERY COST COMPARISON BETWEEN THE ERP AND THE EAP-TLS PROTOCOLS

The ERP requires only five EAP messages (figure.2) as opposed to the nine messages in EAP-TLS in RSA case and ten messages in DHE case. As the time cost of message delivery, which is related with the number of message and the hops between network nodes. For the sake of simplicity we assume the transmission cost per hop and per packet is proportional to a constant Ct, thus the transportation costs over a distance of D should be D×Ct. And we use time Sdn represent the message delivering cost of each step (figure 2). Because the transmission cost between the peer and the authenticator in the first, second and the fifth step should be the same, that are $Sd1 = Sd2 = Sd5 = Dpau \times Ct$. Similarly, the transmission cost between the peer and the authenticator in the third and the fourth steps are $Sd3 = Sd4 = Dpas \times Ct$. Thus, the total transmission delay of data packet in re authentication procedure Ctr is: $Ctr = Sd1+Sd2+Sd3+Sd4+Sd5 = 2 \times Dpas \times Ct + 3 \times Dpau \times Ct$. As the mutual implementation of EAP-TLS[14], the authenticator transmission cost between the peer and the server are: $Ctt = 9 \times Sd = 9 \times (Dpau+Dpas) \times Ct$, which is much longer than the *Ctr*. Compare to EAP-TLS, the ERP significantly reduces the transmit cost more than half.

VI. CONCLUSION

The EAP protocol is a three-party authentication framework, while the ERP protocol is an extension of EAP which aims to reduce the transmissions and computation costs of EAP. This paper analyzed the efficiency of the ERP protocol and compared it with that of the EAP-TLS protocol (The most important EAP method standardized by the IETF EMU work group). The result shows that the computation cost of ERP protocol is around 1ms, which is one twentieth of the EAP-TLS. And it meets the requirement of VOIP.

REFERENCES

[1] H. Krawczyk, M. Bellare, "HMAC: Keyed-Hashing for Message Authentication," IETF RFC 2104, February 1997
[2] O. Elkeelany, M.M. Matalgah, Performance analysis of IPSec protocol: encryption and authentication, in : IEEE Communications Conference (ICC 2002), 2002, pp. 1164 1168
[3] Federal Information Processing Standards Publication Specifications for the Secure Hash Standard, August 1, 2002
[4] H. Orman, Purple Streak Dev., RFC 3766 Determining Strengths For Public Keys Used For Exchanging Symmetric Keys, April 2004
[5] B. Aboba, L. Blunk, J. Vollbrecht, Extensible
Authentication Protocol (EAP), IETF RFC 3748June 2004
[6] P. Calhoun, J. Loughney, E. Guttman, Diameter Base Protocol , IETFRFC 3588, September 2004
[7] P. Eronen, Diameter Extensible Authentication Protocol Application, IETF RFC 4072, August 2005
[8] B. Aboba, M. Beadles, Network Access Identifier (NAI), IETF RFC4282, December 2005
[9] C. Kaufman, Internet Key Exchange (IKEv2) Protocol, RFC 4306,December 2005
[10] T. Clancy, draft-ietf-hokey-reauth-ps-02, "Handover Key Management and Re-authentication Problem Statement", July 24,2007
[11] R. Housley, B. Aboba, Guidance for AAA Key Management, RFC4962, July 2007
[12] Bernard Aboba, Dan Simon, "Extensible Authentication Protocol(EAP) Key Management Framework", IETF draft-ietf-eap-keying-22, 11 November 2007
[13] V. Narayanan, L. Dondeti, "Diameter Support for EAP
Re-authentication Protocol "IETF draft-dondeti-dime-erp-diameter-01, November 19, 2007
[14] "benchmarks for cryptographic algorithms" unpublished
http://www.eskimo.com/~weidai/benchmarks.html
[15] M. Nakhjiri, Y. Ohba, "Derivation, delivery and management of EAP based keys for handover and re-authentication ", IETF draft-ietf-hokey-key-mgm-03, February 25, 2008
[16] D. Simon,B. Aboba, R. Hurst, RFC 5216 The EAP-TLS Authentication Protocol, March 2008
[17] Carolin Latze, Ulrich Ultes-Nitsche, Strong Mutual Authentication in a User-Friendly Way in EAP-TLS.
[18] V. Narayanan, "EAP Extensions for EAP Re-authentication Protocol (ERP)", IETF draft-ietf-hokey-erx-14, March 29, 2008
[19] Kaouthar Sethom, "Requirements and Adaptation Solutions for Transparent Handover between Wifi and Bluetooth", Mobile Computing and Communications Review, Volume 8, Number 1, pp. 61-83, San Jose State University, San Jose, CA, USA