3GPP TSG SA WG3 Security — S3#26 **S3-020611**
19- 22 November 2002
Oxford, UK

Title: ~~—~~**Enhancing EAP/SIM and EAP/AKA ~~-~~Authentication with PEAP (A~~a~~lternative to using Temp ID for IMSI privacy)**

Source: **Intel, Cisco, AT&T Wireless, Gemplus, Transat**

Agenda item: **7.9~~9.6~~**
Document for: **[Discussion, Decision]**

1 Introduction

~~This proposal addresses security-related issues that are of primary interest to 3GPP SA3, and we are pursuing parallel submissions to SA3. However, there are significant architectural implications to this proposal, making it of interest to SA2 as well.~~

~~T~~The present SA3~~2~~ TS document 33.234 v0.2.0 specifies the use of (U)SIM for UE authentication. Given the rising popularity of 802.11-based WLANs, many GSM operators are becoming interested in providing WLAN services to their subscribers to complement their 2.5G and 3G data services. For user convenience and billing infrastructure reasons, it is desirable to support the use of (U)SIM for WLAN authentication. However, security threats in WLAN networks are substantially worse than in GSM networks, because it is relatively easy for an attacker to eavesdrop on a WLAN or pose as a WLAN network operator. Since the original GSM SIM authentication mechanism has known vulnerabilities (primarily due to insufficient key lengths), it is generally accepted that legacy SIM authentication protocols should not be used on a WLAN without additional security measures.

To address some of the security vulnerabilities of legacy SIM authentication, several companies, including Nokia, Cisco, Intel, Gemp~~P~~lus, and Transat, are working together to enhance the ~~working together on a proposal for enhanced~~ SIM-based WLAN authentication proposals~~using~~ EAP/SIM [1] and EAP/AKA [3]. However, since the underlying authentication mechanisms of (U)SIM are based on shared secrets, these proposals do not provide a complete solution for user identity protection. Temporary IDs provide some protection, but it is easy to envision a scenario where an attacker could induce the UE to divulge its IMSI by claiming that the temporary ID is not valid. Fortunately, it is possible to perform EAP/SIM or EAP/AKA authentication over a secure tunneling protocol such as PEAP [2] and thereby gain additional security benefits (additional benefits include IMSI privacy, avoidance of denial-of-service attacks through unauthenticated EAP Success/Failure messages, better protection against network impersonation attacks if triplets are compromised, and reduced consumption of authentication triplets).

With PEAP, common session key derivation, distribution, and configuration solutions can be defined for a variety of credential types, including certificates, username/password, and (U)SIM. We specifically mention PEAP rather than alternative tunneling protocols, because we expect PEAP to eventually be the most widely deployed EAP protection protocol for WLAN clients. If industry alignment can be achieved in these areas, it will be much easier for network operators to support a variety of roaming scenarios across different network types. PEAP helps address all of the user security requirements listed in Section 4.2~~6.3.3~~ of the 3GPP TS33~~R 22~~.2~~9~~34 document. Furthermore, TLS and the PKI infrastructure used by PEAP can also help address many of the requirements for network operator security features listed in that section. It is also interesting to note that 3GPP SA3 also has started discussing Operator PKI issues in the context of digital subscriber certificates, making this effort timely. The PEAP PKI requirements also relate to network domain security using PKI in the context of Diameter using TLS or IPSec [7].

In this document, we recommend the use of PEAP in conjunction with EAP/SIM and EAP/AKA as a long-term solution to provide enhanced security and user privacy for WLAN authentication. There are some important technical issues that must be resolved before PEAP is ready for widespread deployment. These issues are being worked on in the IETF. ~~-~~ Assuming the technical issues outlined below will be

resolved, we recommend that 3GPP consider adding PEAP as an example authentication method to use along with EAP/SIM and EAP/AKA and also other potential user credential types Network Operators may wish to deploy in the future.~~are not advocating immediate adoption of PEAP by 3GPP is that PEAP is still only a draft standard in the IETF, and~~ However, we want to emphasize that deployments without PEAP should be supported as well, at least in the near term. Because PEAP is designed to support arbitrary legacy EAP [6] protocols, it should be very straightforward to migrate EAP/SIM and EAP/AKA deployments to PEAP at the appropriate time.

2 PEAP Issues

PEAP must be used correctly if its full security benefits are to be obtained. Particular attention must be paid to the following issues:

- WLAN clients must be configured with an appropriate PKI trust policy so that user identities are never divulged to an untrusted authentication server and connections are not established with untrusted networks.
- The endpoints of the PEAP tunnel must be cryptographically bound to the EAP/SIM or EAP/AKA authentication exchange and session keys to prevent man-in-the-middle attacks. [4] [5]
- WLAN session keys derived from PEAP and the inner EAP/SIM or EAP/AKA authentication must be distributed securely from the home AAA server to the WLAN access point. This issue is not PEAP-specific, since the same problem needs to be solved if EAP/SIM or EAP/AKA are run without PEAP.

~~2.1~~3 Benefits of using PEAP with EAP/SIM and EAP/AKA

The primary benefits of using PEAP with EAP/SIM rather than using EAP/SIM alone are:

- PEAP is based on the TLS protocol, which is a proven and widely-deployed security technology that has undergone extensive review and is considered to be cryptographically strong. If EAP/SIM and EAP/AKA runs over PEAP, it can benefit from the additional security provided by the PEAP channel.

- PEAP can provide complete protection for user identities, since ~~the permanent (IMSI) or temporary~~ identities are sent through an encrypted TLS tunnel. With PEAP, it is unnecessary for clients or authentication servers to maintain or update temporary identities. This feature can permit substantial simplifications in UE and AAA server implementations, especially considering the severe reliability requirements currently being discussed in the context of Temp ID management. The simplification itself has security value since security and complexity are inversely related.

- It is possible to run a variety of different EAP methods over PEAP, so PEAP clients are not restricted to using only one type of credential or EAP method. It is likely that for WLAN value added services other EAP methods could be defined which can benefit from the PEAP protection when run inside it.

- When EAP/SIM is run without PEAP, multiple triplets (2 or 3) must be used to yield adequate session key entropy. When run over PEAP, the TLS channel provides key entropy and protection against eavesdropping and brute force attacks against the SIM algorithm, so only a single SIM triplet is needed.

- PEAP is likely to be very widely deployed on client platforms such as laptop computers and PDAs running some of the ~~Microsoft~~popular operating systems.

- PEAP provides additional protection against attackers who might try to impersonate the network if they are able to obtain a set of SIM triplets for a particular user. If PEAP is used, an attacker would also need to obtain an X.509 certificate declaring it to be a legitimate authentication server trusted by the user's home network. This is a significant obstacle to a would-be attacker.

- Once PEAP is deployed it provides the operators and vendors with a Protected EAP protocol foundation on which to add other EAP methods for different services and their authorization aspects if needed. This will require only changes on the Client and the Server entities and no changes in other network elements.

4 ~~Conclusion/Recommendations~~Proposals:

~~The supporting companies identified in the source would like to recommend adding the PEAP Protocol described in the Annex A below as an example authentication method to enhance the security of EAP/SIM and EAP/AKA in the Annex A3 of the TS23.234 v1.1.0.~~

4.1 Proposal 1

The supporting companies identified in the source propose adding the following text to section ~~5.3.3~~6.1 of TS 3~~2~~3.234.

6.1.3 PEAP based authentication with EAP/SIM and EAP/AKA

Due to
~~Because of~~ the heightened security risks associated with WLAN networks, the EAP/SIM and EAP/AKA authentication protocols should be run inside a TLS channel established using the PEAP protocol. PEAP can provide complete user identity protection, more robust network authentication, and stronger cryptographic keys using fewer triplets than EAP/SIM or EAP/AKA alone. Because it runs on top of EAP, PEAP can be transported over the Wr reference point. Note that if PEAP is used with EAP/SIM or EAP/AKA, cryptographic binding between the PEAP session and (U)SIM-derived session keys must be provided to prevent man-in-the-middle attacks.

4.2 Proposal 2

The supporting companies identified in the source propose adding the following text to a new subsection of TS ~~2~~33.234 section 5.1~~.4.~~

5.1.4 User Identity Privacy

The subscriber identity IMSI, needs to be protected on the WLAN access network when using EAP based authentication methods. PEAP can be used as a tunnelling protocol for providing strong user identity privacy when using SIM or USIM credentials with EAP/SIM or EAP/AKA respectively. ~~If~~ When EAP/SIM or EAP/AKA are run inside PEAP, the user identity disclosed in the initial EAP exchange (prior to completion of PEAP part 1) must only identify the realm of the home network and not contain any user-specific information. For example, if the IMSI is 234150999999999 (MCC = 234, MNC = 15), the UE must report its user identity as something like **234150000000000@15.234.WLAN.3gppnetwork.org.** Once the PEAP channel is established, a second protected identity request is made, and the UE at that point should respond with the true IMSI identification (for example, **234150999999999@15.234.WLAN.3gppnetwork.org**).

Annex A

A1. PEAP Protocol Overview

Protected EAP (PEAP) is built atop two standards: Extensible Authentication Protocol (EAP), and Transport Layer Security (TLS). EAP is described in RFC2284, and TLS is described in RFC2246. The user authentication scheme is governed by EAP, which allows for many different authentication methods (SIM-based, username/password, etc). EAP is a well-accepted industry standard that allows for arbitrary authentication methods.

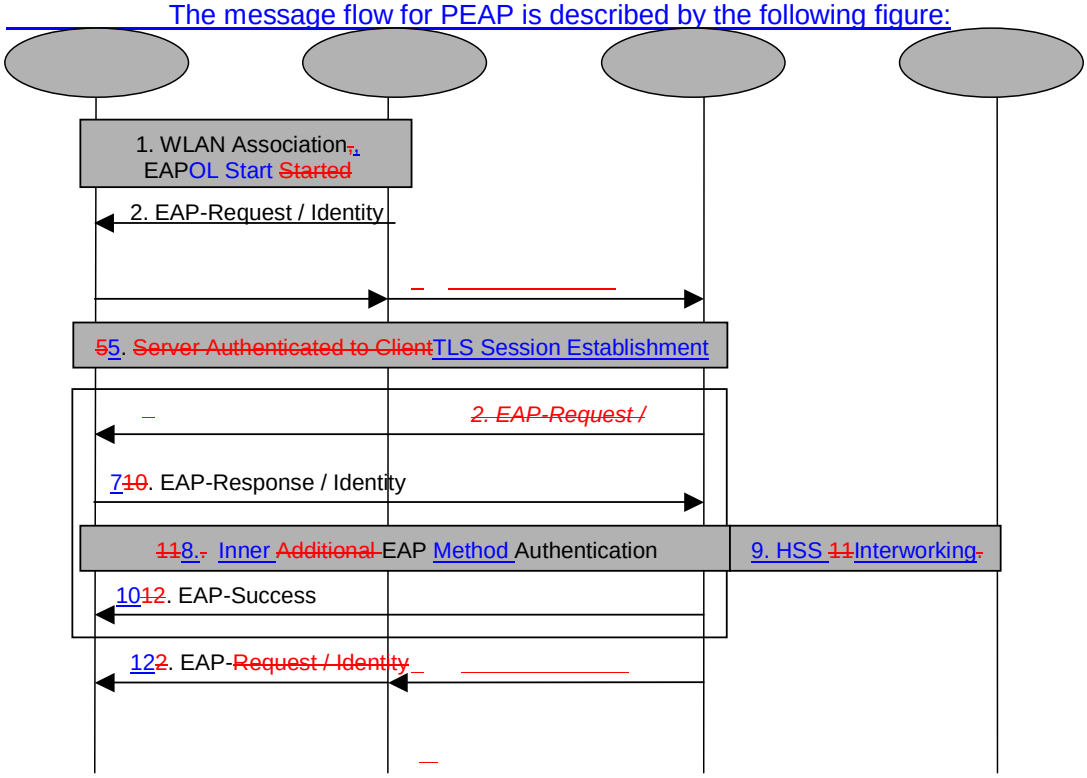
The secure connection is provided by TLS, which is very similar to but not interoperable with SSL 3.0. Another thing PEAP provides through TLS is a well-reviewed method for key derivation, which can provide keying material for a wide range of link layer cipher suites in the form of the master secret [2]. This keying material can be used for ciphering in 802.11 applications including WEP, TKIP or 802.11i.

A requirement for PEAP is a way for a client to authenticate the AAA server. To do this securely, the AAA server must have a server certificate issued by a Certificate Authority (CA). The server can then present this certificate to the client, which it can verify with the CA. In order to verify this certificate, the WLAN client must have a root certificate issued by the same CA that issued the server certificate.

One of the goals of PEAP is to provide a secure channel in which authentication can take place, and therefore obviate the need for an authentication method that is secure in isolation. The secure channel runs end-to-end, which makes PEAP suitable for Wireless LAN (WLAN) authentication, where there can be no assumed security at the physical layer. PEAP is described by an internet-draft [2].

A2. PEAP Protocol Details

The message flow for PEAP is described by the following figure:



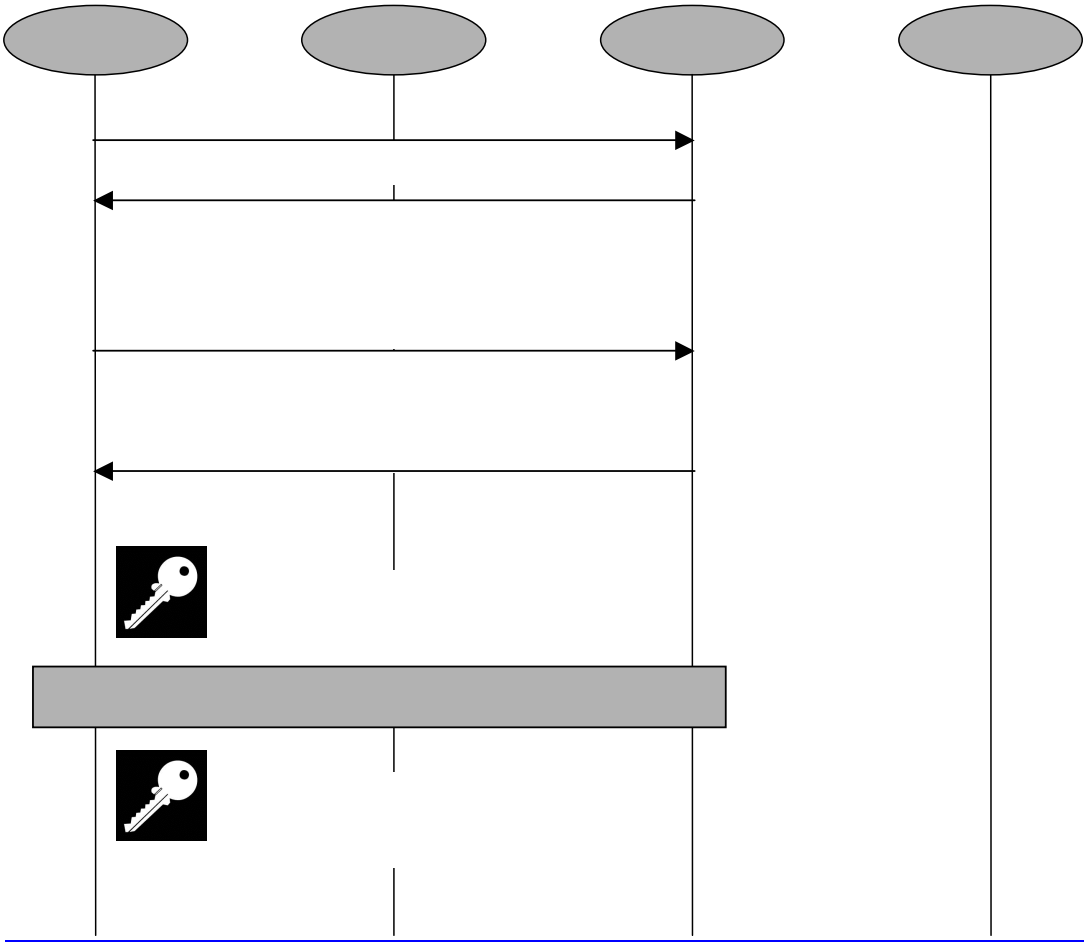
1. After establishing a WLAN connection (out of scope of 3GPP), the Extensible Authentication Protocol (EAP) can begin.
2. The WLAN sends an EAP Request / Identity to the UE. EAP packets are transmitted within a WLAN technology specific protocol, which is also out of scope of 3GPP.
3. The UE responds with an EAP Response / Identity message. This message contains the UE's identity, which must comply with the Network Access Identifier (NAI) format specified in RFC 2486. This identity is sent in the clear, without any security measures, so care should be taken to only provide anonymous user

identity information along with complete realm information for routing the subscriber to the correct 3GPP AAA Server.

4. This EAP/Identity response is translated to a RADIUS Access Request from the WLAN to the 3GPP AAA Server along with the NAI.
5. The 3GPP AAA Server sends its certificate to the UE as part of the TLS negotiation and initiates establishment of a TLS Session between the UE and the 3GPP AAA Server (See next Section A3 for details).
6. With a secure TLS channel established, the 3GPP AAA Server sends another EAP Request / Identity in the secure session for initiating the UE preferred authentication method (inner method).
7. The UE responds with an EAP Response / Identity, passing once again the NAI compliant user identity which signals a preferred authentication method like EAP/SIM, EAP/AKA or any other method..
8. The inner method results in several EAP Request/Responses depending on the particular method triggered. The inner EAP method terminates on the 3GPP AAA Server.
9. The figure doesn't detail the interworking aspects with the HSS, but appropriate interworking messages are assumed.
10. Once the inner method authentication is complete, an EAP-Success message will be sent to the UE. This message is protected by the TLS channel
11. A RADIUS Accept message is sent by the 3GPP AAA Server to the WLAN
12. A follow on EAP-Success message is sent to the UE in the clear. This message can be ignored or made optional.

A3. TLS Session Details

The process of negotiating a TLS session (step 5 above) is detailed in the diagram below. Each of these steps are TLS messages, and would be encapsulated within EAP messages when negotiating a PEAP session.



1. The Client Hello message is sent from the client to the AAA server ("server"). This message contains the client's TLS version, a session id, a random number, and the set of TLS ciphersuites supported by the client.

2. As a response to the Client Hello message, the Server Hello message is sent to the Client. The Server Hello message contains a TLS version number, another random number, a session id, and a specified TLS cipher suite. If the session id sent by the client was null or unrecognized by the server, the server must choose the session id to establish a new session; otherwise the server simply returns the same session id that was passed by the client. In this case, the session is said to be resumed, and message flow proceeds directly ahead to the "finished" message (step 7). The server also selects a TLS cipher suite from the list sent in the Client Hello message.
3. The EAP server must include a TLS certificate if the client is not resuming a previously established session. The certificate message contains a public key certificate for either a key exchange public key or a signature public key.
4. If the TLS certificate sent above contained a signature public key, a Server Key Exchange handshake message must be included. The Server Key Exchange message conveys the cryptographic information necessary to communicate the premaster secret to the server, and is necessary to avoid using the same key for encryption and signing. This information may either be an RSA public key with which to encrypt the premaster secret or a Diffie-Hellman public key with which the client can complete a key exchange for the premaster secret.
5. A Client Key Exchange message is sent to the server, which communicates the premaster secret to the server. The premaster secret is an encrypted package which includes the client's newest supported TLS protocol version and a 46-byte random number. If RSA is used for key agreement and authentication, the premaster secret is 48 bytes long.
6. Now that a ciphering method is agreed upon and enough keying information has been transmitted, the client sends the Change Cipher Suite message. This message contains no information, rather signalling that the cipher suite previously agreed upon can now be used.
7. The server replies with a Change Cipher Suite message, communicating to the client that the server agrees to the new ciphering method.
8. The client and server now both have the premaster secret and two random numbers: one generated by the client, and one generated by the server. These random numbers were respectively transmitted in the Client Hello and Server Hello messages (steps 1 and 2 above). This allows for the master secret to be computed in the following way:

    Master Secret = PRF(premaster secret, "master secret", client random, server random).

The master secret is always exactly 48 bytes in length. Once the master secret is computed, the premaster secret should be purged from memory.
9. The master secret should not be used for ciphering, and instead ciphering keys should be generated from the master secret. This can be done using a similar method to the one in which the master secret was generated from the premaster secret:

    Key = PRF(master secret, "client EAP encryption", client random, server random)

This method allows for ciphering keys that may be changed without requiring a renegotiation of the TLS session. These keys can be used for any purposes that require keying material, including WEP key generation at the access point.

A4. Session Resumption

The session id within the TLS protocol allows for the client to resume sessions when the client repeatedly and quickly tries to authenticate to an EAP server. Session resumption is "particularly useful for support of wireless roaming" [2].

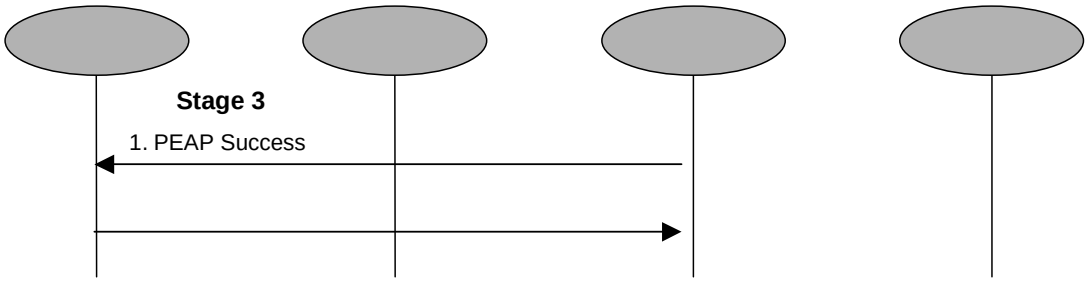
The peer decides whether to attempt to resume a session, which shortens the PEAP authentication message flow (section 3). This decision by the peer is based on the time elapsed since the last authentication attempt or also based on data volume. The server will then decide whether to allow resumption or choose a new TLS session, as described in section 4 step 2.

A5. Security Considerations

If the peer does not support PEAP, a negative acknowledgement (NAK) is sent to negotiate another authentication protocol. These packets can be spoofed, fooling the client into negotiating a weaker form of authentication than it would otherwise choose. For this reason, once again PEAP must always be used in cases where the extra security is required. Should a NAK be received by the server, it should respond with an EAP/Failure message and deny authentication.

PEAP provides a reasonable level of security if carefully applied. Its chief advantage is that it runs end-to-end and protects even insecure authentication protocols. Problems arise if PEAP is used with the same credential that is applied elsewhere on the network without the protection provided by PEAP. In this case, the underlying credential is exposed to attack in its unprotected form, and the tunnelled authentication provided by PEAP is vulnerable to Man in the Middle attacks [5].

This problem can also be addressed for EAP methods that generate keying material by implementing a Compound MAC [4] and also generating Compound Keys. This scheme adds to PEAP an optional third stage, responsible for verifying the authentication based on keying material unavailable to the man in the middle. The third stage fits between steps 8 and 10 in section A2 above and works as follows:



The PEAP Success and Success/ACK messages contains a Compound MAC, calculated from the keying material generated by both TLS and the EAP method from PEAP stage 2. Additionally Compound Keys derived can be used for link layer cipher key generation requirements. This proposed solution is submitted to IETF and is documented in [4].

A6. Glossary/Abbreviations:

CA: Certificate Authority.
EAP: The PPP Extensible Authentication Protocol, described in <RFC2284>.
Master Secret: Secure secret data used for generating encryption keys, MAC secrets, and IVs. [TLS]
PEAP: Protected Extensible Authentication Protocol, described in <draft-josefsson-ppext-eap-tls-eap-03.txt>,
Premaster Secret: A hashed random number generated by the client in TLS negotiation. The premaster secret is used along with the server and client random numbers to generate the Master Secret.
PRF: Pseudo Random Function.
TLS: Transport Layer Security, described in RFC2246.
UE: User Equipment.
WEP: Wired Equivalent Privacy, a method for encrypting 802.11 packets.

A7. References

[1] H. Haverinen, J. Salowey, "EAP SIM Authentication", draft-haverinen-pppext-eap-sim-07.txt, work-in-progress, ~~October~~ November 2002.

[2] H. Andersson, S. Josefsson, G. Zorn, D. Simon, A. Palekar, "Protected EAP Protocol (PEAP)", draft-josefsson-pppext-eap-tls-eap-05.txt, work-in-progress, September 2002. ~~(INFORMATIVE)~~

[3] J. Arkko, H. Haverinen, "EAP AKA Authentication", draft-arkko-pppext-eap-aka-05.txt, work-in-progress, November 2002.

[4] Puthenkulam, J., Lortz, V., Palekar, A., Simon, D., Aboba, B., “The Compound Authentication Binding Problem”, <draft-puthenkulam-eap-binding-01.txt>, Oct. 2002. (Work in Progress)

[5] Asokan, N., Valtteri, N., Nyberg, K.,“Man-in-the-Middle in Tunnelled Authentication”, http://www.saunalahti.fi/~asokan/research/mitm.html>, Oct 2002. (Work in Progress)

[6] L. Blunk, J. Vollbrecht, B. Aboba, "Extensible Authentication Protocol (EAP)", draft-ietf-pppext-rfc2284bis-07.txt, work-in-progress, October 2002. ~~(NORMATIVE)~~

[7] P. Calhoun, J. Loughney, E. Gutmann, G. Zorn, J. Arkko, “Diameter Base Protocol”, draft-ietf-aaa-diameter-15.txt, October 2002. Section 2.2 and 13.2 (Work in Progress)