TOC

| | |
|---|---|
| EMU Working Group | K. Hoeper |
| Internet-Draft | NIST |
| Intended status: Informational | S. Hanna |
| Expires: May 5, 2009 | Juniper Networks |
| | H. Zhou |
| | J. Salowey, Ed. |
| | Cisco Systems, Inc. |
| | November 01, 2008 |

# Requirements for an Tunnel Based EAP Method
draft-ietf-emu-eaptunnel-req-01.txt

## Status of this Memo

By submitting this Internet-Draft, each author represents that any applicable patent or other IPR claims of which he or she is aware have been or will be disclosed, and any of which he or she becomes aware will be disclosed, in accordance with Section 6 of BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as “work in progress.”

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt.

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html.

This Internet-Draft will expire on May 5, 2009.

## Abstract

This memo defines the requirements for a tunnel-based Extensible Authentication Protocol (EAP) Method. This method will use Transport Layer Security (TLS) to establish a secure tunnel. The tunnel will provide support for password authentication, EAP authentication and the transport of additional data for other purposes.

**Table of Contents**

---

## 1. Introduction

TOC

Running EAP methods within a TLS protected tunnel has been deployed in several different solutions. EAP methods supporting this include PEAP, TTLS (Funk, P. and S. Blake-Wilson, “Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0),” August 2008.) [RFC5281] and EAP-FAST (Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, “The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST),” May 2007.) [RFC4851]. There have been various reasons for employing a protected tunnel for EAP processes. They include protecting weak authentication exchanges, such as username and password. In addition a protected tunnel can provide means to provide peer identity protection and EAP method chaining. Finally, systems have found it useful to transport additional types of data within the protected tunnel.
This document describes the requirements for an EAP tunnel method as well as for a password protocol supporting legacy password verification within the tunnel method.

---

TOC

## 2. Conventions Used In This Document

Because this specification is an informational specification (not able to directly use [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.)), the following capitalized words are used to express requirements language used in this specification. Use of each capitalized word within a sentence or phrase carries the following meaning during the EMU WG's method selection process:

> MUST - indicates an absolute requirement
>
> MUST NOT - indicates something absolutely prohibited
>
> SHOULD - indicates a strong recommendation of a desired result
>
> SHOULD NOT - indicates a strong recommendation against a result
>
> MAY - indicates a willingness to allow an optional outcome

Lower case uses of "MUST", "MUST NOT", "SHOULD", "SHOULD NOT" and "MAY" carry their normal meaning and are not subject to these definitions.

---

## 3. Use Cases

TOC

To motivate and explain the requirements in this document, a representative set of use cases for the EAP tunnel method are supplied here. The candidate tunnel method is expected to meet all of the use cases marked as MUST.

---

### 3.1. Password Authentication

TOC

Many legacy systems only support user authentication with passwords. Some of these systems require transport of the actual username and password to the authentication server. This is true for systems where the authentication server does not have access to the cleartext password or a consistent transform of the cleartext password. Example of such systems are one time password (OTP) systems and other systems where the username and password are submitted to an external party for validation. The tunnel method MUST meet this use case. However, it MUST

NOT expose the username and password to untrusted parties and it MUST provide protection against man-in-the-middle and dictionary attacks. Since EAP authentication occurs before network access is granted the tunnel method SHOULD enable an inner exchange to provide support for minimal password management tasks including password change, "new PIN mode", and "next token mode" required by some systems.

---

### 3.2. Protect Weak EAP Methods

TOC

Some existing EAP methods have vulnerabilities that could be eliminated or reduced by running them inside a protected tunnel. For example, a method such as EAP-MD5 does not provide mutual authentication or protection from dictionary attacks. In addition, tunneled EAP methods are subject to a specific form of man-in-the-middle attack described in [TUNNEL-MITM] (Asokan, N., Niemi, V., and K. Nyberg, “Man-in-the-Middle in Tunnelled Authentication Protocols,” November 2002.).
The tunnel method MUST support protection of weak inner methods and protect against man-in-the-middle attacks associated with tunneled authentication.

---

### 3.3. Chained EAP Methods

TOC

Several circumstances are best addressed by using chained EAP methods. For example, it may be desirable to authenticate the user and also authenticate the device that he or she is using. However, chained EAP methods from different conversations can be re-directed into the same conversation by an attacker giving the authenticator the impression that both conversations terminate at the same end-point. Cryptographic binding can be used to bind the results of key generating methods together or to an encompassing tunnel.
The tunnel method MUST support chained EAP methods while including strong protection against attacks on the method chaining.

---

### 3.4. Identity Protection

TOC

When performing an EAP authentication, the peer may want to protect its identity, only disclosing its identity to a trusted backend authentication server. This helps to maintain the privacy of the peer's identity.
The tunnel method MUST support identity protection, ensuring that peer identity is not disclosed to the authenticator and any other

intermediaries before the server that terminates the tunnel method. Note that the peer may need to expose the realm portion of the EAP outer identity in the NAI [RFC4282] (Aboba, B., Beadles, M., Arkko, J., and P. Eronen, “The Network Access Identifier,” December 2005.) in a roaming scenario in order to reach the appropriate authentication server.

---

### 3.5. Emergency Services Authentication

TOC

When wireless VOIP service is provided, some regulations require any user to be able to gain access to the network to make an emergency telephone call. To avoid eavesdropping on this call, it's best to negotiate link layer security as with any other authentication. Therefore, the tunnel method SHOULD allow anonymous peers or server-only authentication, but still derive keys that can be used for link layer security. The tunnel method MAY also allow for the bypass of server authentication processing on the client. Forgoing authentication increases the chance of man-in-the-middle and other types of attacks that can compromise the derived keys used for link layer security.

---

### 3.6. Network Endpoint Assessment

TOC

The Network Endpoint Assessment (NEA) protocols and reference model described in [RFC5209] (Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, “Network Endpoint Assessment (NEA): Overview and Requirements,” June 2008.) provide a standard way to check the health ("posture") of a device at or after the time it connects to a network. If the device does not comply with the network's requirements, it can be denied access to the network or granted limited access to remediate itself. EAP is a convenient place for conducting an NEA exchange. The tunnel method SHOULD support carrying NEA protocols such as PB-TNC [I-D.ietf-nea-pb-tnc] (Sahita, R., Hanna, S., and K. Narayan, “PB-TNC: A Posture Broker Protocol (PB) Compatible with TNC,” October 2009.). Depending on the specifics of the tunnel method, these protocols may be required to be carried in an EAP method.

---

### 3.7. Resource Constrained Environments

TOC

A growing number of "resource constrained" devices (e.g. printers and phones) are connecting to IP networks and those networks increasingly require EAP authentication to gain access. Therefore, it is natural to expect that new EAP methods be designed to work as well as possible with these devices.

For the purposes of this document, the phrase "resource constrained" means any combination of the following constraints: little processing power, small amounts of memory (both ROM and RAM), small amounts of long-term mutable storage (e.g. flash or hard drive) or none at all, and constrained power usage (perhaps due to small battery).

The tunnel method SHOULD be designed so it can be configured to work with "resource constrained" devices, when possible.

---

### 3.8. Client Authentication During Tunnel Establishment

TOC

In cases where client authentication can be performed as part of the tunnel establishment it is efficient for the tunnel method to allow this. The tunnel MUST be capable of providing client side authentication during tunnel establishment.

---

### 3.9. Extensibility

TOC

The tunnel method MUST provide extensibility so that additional types of data can be carried inside the tunnel in the future. This removes the need to develop new tunneling methods for specific purposes.

One example of a application for extensibility is credential provisioning. When a peer has authenticated with EAP, this is a convenient time to distribute credentials to that peer that may be used for later authentication exchanges. For example, the authentication server can provide a private key or shared key to the peer that can be used by the peer to perform rapid re-authentication or roaming. In addition there have been proposals to perform enrollment within EAP, such as [I-D.mahy-eap-enrollment] (Mahy, R., “An Extensible Authentication Protocol (EAP) Enrollment Method,” March 2006.).

---

## 4. Requirements

TOC

---

### 4.1. General Requirements

TOC

---

### 4.1.1. RFC Compliance

TOC

The tunnel method MUST include a Security Claims section with all security claims specified in Section 7.2 in RFC 3748 (Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” June 2004.) [RFC3748]. In addition, it MUST meet the requirement in Sections 2.1 and 7.4 of RFC 3748 that tunnel methods MUST support protection against man-in-the-middle attacks. Furthermore, all tunnel methods MUST support identity protection as specified in Section 7.3 in RFC 3748.

The tunnel method MUST be unconditionally compliant with RFC 4017 (Stanley, D., Walker, J., and B. Aboba, “Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs,” March 2005.) [RFC4017] (using the definition of "unconditionally compliant" contained in section 1.1 of RFC 4017). This means that the method MUST satisfy all the MUST, MUST NOT, SHOULD, and SHOULD NOT requirements in RFC 4017.

The tunnel method MUST meet all the EAP method requirements contained in the EAP Key Management Framework [RFC5247] (Aboba, B., Simon, D., and P. Eronen, “Extensible Authentication Protocol (EAP) Key Management Framework,” August 2008.) or its successor. The tunnel method MUST include MSK and EMSK generation. This will enable the tunnel method to properly fit into the EAP key management framework, maintaining all of the security properties and guarantees of that framework.

The tunnel method MUST NOT be tied to any single cryptographic algorithm. Instead, it MUST support run-time negotiation to select among an extensible set of cryptographic algorithms. This "cryptographic algorithm agility" provides several advantages. Most important, when a weakness in an algorithm is discovered or increased processing power overtakes an algorithm, users can easily transition to a new algorithm. Also, users can choose the algorithm that best meets their needs.

The tunnel method MUST meet requirements pertinent to EAP method contained in Section 3 of RFC 4962 (Housley, R. and B. Aboba, “Guidance for Authentication, Authorization, and Accounting (AAA) Key Management,” July 2007.) [RFC4962]. This includes: cryptographic algorithm independence; strong, fresh session keys; replay detection; keying material confidentiality and integrity; confirm cipher suite selection; and uniquely named keys.

### 4.1.2. Draw from Existing Work

TOC

Several existing tunnel methods are already in widespread usage: EAP-FAST [RFC4851] (Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, “The Flexible Authentication via Secure Tunneling Extensible

Authentication Protocol Method (EAP-FAST),” May 2007.), EAP-TTLS [RFC5281] (Funk, P. and S. Blake-Wilson, “Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0),” August 2008.), and PEAP. Considerable experience has been gained from various deployments with these methods. This experience SHOULD be considered when evaluating tunnel methods. If one of these existing tunnel methods can meet the requirements contained in this specification then that method SHOULD be preferred over a new method.

Even if minor modifications or extensions to an existing tunnel method are needed, this method SHOULD be preferred over a completely new method so that the advantage of accumulated deployment experience and security analysis can be gained.

### 4.2. Tunnel Requirements

TOC

Existing tunnel methods make use of TLS [RFC5246] (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008.) to provide the protected tunnel. In general this has worked well so there is consensus to continue to use TLS as the basis for a tunnel method.

### 4.2.1. TLS Requirements

TOC

The tunnel based method MUST support TLS version 1.2 [RFC5246] (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” August 2008.) and SHOULD support TLS version 1.0 [RFC2246] (Dierks, T. and C. Allen, “The TLS Protocol Version 1.0,” January 1999.) and version 1.1 [RFC4346] (Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.1,” April 2006.) to enable the possibility of backwards compatibility with existing deployments. The following section discusses requirements for TLS Tunnel Establishment.

### 4.2.1.1. Cipher Suites

TOC

TOC

#### 4.2.1.1.1. Cipher Suite Negotiation

Cipher suite negotiations always suffer from downgrading attacks when they are not secured by any kind of integrity protection. A common practice is a post integrity check in which, as soon as available, the established keys (here the tunnel key) are used to derive integrity keys. These integrity keys are then used by peer and authentication server to verify whether the cipher suite negotiation has been maliciously altered by another party.
Integrity checks prevent downgrading attacks only if the derived integrity keys and the employed integrity algorithms cannot be broken in real-time. See Section 6.1 (Cipher Suite Selection) or [KHLC07] (Hoeper, K. and L. Chen, “Where EAP Security Claims Fail,” August 2007.) for more information on this. Hence, the tunnel method MUST provide integrity protected cipher suite negotiation with secure integrity algorithms and integrity keys.
All versions of TLS meet these requirements as long as the cipher suites used provide strong authentication, key establishment and data integrity protection.

---

#### 4.2.1.1.2. Tunnel Data Protection Algorithms

TOC

In order to prevent attacks on the cryptographic algorithms employed by inner authentication methods, a tunnel protocol's protection needs to provide a basic level of algorithm strength. The tunnel method MUST provide at least one mandatory to implement cipher suite that provides the equivalent security of 128-bit AES for encryption and message authentication. See [NIST SP 800-57] (Barker, E., Barker, W., Burr, W., Polk, W., and M. Smid, “Recommendation for Key Management - Part 1: General (Revised),” March 2007.) for a discussion of the relative strengths of common algorithms.

---

#### 4.2.1.1.3. Tunnel Authentication and Key Establishment

TOC

A tunnel method MUST provide unidirectional authentication from authentication server to EAP peer or mutual authentication between authentication server and EAP peer. The tunnel method MUST provide at least one mandatory to implement cipher suite that provides certificate based authentication of the server and provides optional certificate based authentication of the client. Other types of authentication MAY be supported.
At least one mandatory to implement cipher suite MUST meet the following requirements for secure key establishment along with the

previous requirements for authentication and data protection algorithms:

- One-way key derivation, i.e., a compromised key leads to the compromise of all descendant keys but does not affect the security of any precedent key in the same branch of the key hierarchy.
- Cryptographically separated keys, i.e., a compromised key in one branch of the key hierarchy does not affect the security of keys in other branches.
- Cryptographically separated entities, i.e., keys held by different entities are cryptographically separate. As a result, the compromise of a single peer does not compromise keying material held by any other peer within the system, including session keys and long-term keys.
- Identity binding, i.e., each derived key is bound to the EAP peer and authentication server by including their identifiers as input to the key derivation.
- Context binding, i.e., each derived key is bound to its context by including appropriate key labels in the input of the key derivation.
- Key lifetime, i.e., each key has a lifetime assigned that does not exceed the lifetime of any key higher in the key hierarchy.
- Mutual implicit key authentication, i.e., the keying material derived upon a successful key establishment execution is only known to the EAP peer and authentication server and is kept confidential.
- Key freshness, i.e. EAP peer and EAP server are assured that the derived keys are fresh and the re-use of expired key material is prevented. The freshness property is typically achieved by using one or more of the following techniques: nonces, sequence numbers, timestamps.

The mandatory to implement cipher suites MUST NOT include "export strength" cipher suites, cipher suites providing mutually anonymous authentication or static Diffie-Hellman cipher suites. NIST publication [NIST SP 800-52] (Chernick, C., Edington III, C., Fanto, M., and R. Rosenthal, “Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations,” June 2005.) can be consulted for a list of acceptable TLS v1.0 cipher suites and [NIST SP 800-108] (Chen, L., “Recommendation for Key Derivation Using Pseudorandom Functions,” April 2008.) for additional information on secure key derivation. In addition a tunnel method SHOULD provide cipher suites to meet the following additional recommendations for good key establishment algorithms:

- Key control , i.e., EAP peer and authentication server each contribute to the key computation of the tunnel key. This property prevents that a single protocol participant controls the value of an established key. In that way, protocol participants can ensure that generated keys are fresh and have good random properties.

- Key confirmation, i.e., one protocol participant is assured that another participant actually possesses a particular secret key. In the case of mutual key confirmation both the EAP peer and the authentication server are assured that they possess the same key. Key confirmation is commonly achieved by using one of the derived keys to generate a message authentication code. Mutual key confirmation combined with mutual implicit key authentication leads to mutual explicit key authentication.

- Forward secrecy (FS), i.e., if a long-term secret key is compromised, it does not compromise keys that have been established in previous EAP executions. This property is typically achieved by executing an ephemeral Diffie-Hellman key establishment.

---

### 4.2.1.2. Tunnel Replay Protection

TOC

In order to prevent replay attacks on a tunnel protocol, the message authentication MUST be generated using a time-variant input such as timestamps, sequence numbers, nonces, or a combination of these so that any re-use of the authentication data can be detected as invalid. TLS makes use of an 8 byte sequence number to protect against replay.

#### 4.2.1.3. TLS Extensions

TOC

In order to meet the requirements in this document TLS extensions MAY be used. For example, TLS extensions may be useful in providing certificate revocation information via the TLS OCSP extension (thus meeting the requirement in Section 4.5.1.3 (Server Credential Revocation Checking)).

#### 4.2.1.4. Peer Identity Privacy

TOC

A tunnel protocol MUST support peer privacy. This requires that the username and other attributes associated with the peer are not transmitted in the clear or to an unauthenticated, unauthorized party. If applicable, the peer certificate is sent confidentially (i.e. encrypted).

#### 4.2.1.5. Session Resumption

TOC

The tunnel method MUST support TLS session resumption as defined in [RFC5246] (Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2," August 2008.). The tunnel method MAY support other methods of session resumption such as those defined in [RFC5077] (Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State," January 2008.).

### 4.2.2. Fragmentation

TOC

Tunnel establishment sometimes requires the exchange of information that exceeds what can be carried in a single EAP message. In addition information carried within the tunnel may also exceed this limit. Therefore a tunnel method MUST support fragmentation and reassembly.

### 4.2.3. Protection of Data External to Tunnel

TOC

A tunnel method MUST provide protection of any data external to the TLS tunnel that can cause a problem if it is modified by an attacker. This may include data such as type codes and version numbers

### 4.3. Tunnel Payload Requirements

TOC

This section describes the payload requirements inside the tunnel. These requirements frequently express features that a candidate protocol must be capable of offering so that a deployer can decide whether to make use of that feature. This section does not state requirements about what features of each protocol must be used during a deployment.

#### 4.3.1. Extensible Attribute Types

TOC

The payload MUST be extensible. Some standard payload attribute types will be defined to meet known requirements listed below, such as password authentication, inner EAP method, vendor specific attributes, and result indication. Additional payload attributes MAY be defined in the future to support additional features and data types.

#### 4.3.2. Request/Challenge Response Operation

TOC

The payload MUST support request and response type of half-duplex operation typical of EAP. Multiple attributes may be sent in a single payload. The payload MAY support carrying on multiple authentications in a single payload packet.

#### 4.3.3. Mandatory and Optional Attributes

TOC

The payload MUST support marking of mandatory and optional attributes, as well as an attribute used for rejecting mandatory attributes. Mandatory attributes are attributes sent by the requester that the responder is expected to understand and MUST respond to. If the responder does not understand or support one of the mandatory attributes in the request, it MUST ignore the rest of the attributes and send a NAK attribute to decline the request. The NAK attribute MUST support inclusion of which mandatory attribute is not supported. The optional attributes are attributes that are not mandatory to support and respond to. If the responder does not understand or support the optional attributes, it can ignore these attributes.

### 4.3.4. Vendor Specific Support

TOC

The payload MUST support communication of an extensible set of vendor-specific attributes. These attributes will be segmented into uniquely identified vendor specific name spaces. They can be used for experiments or vendor specific features.

### 4.3.5. Result Indication

TOC

The payload MUST support result indication and its acknowledgement, so both the EAP peer and server will end up with a synchronized state. The result indication is needed after each chained inner authentication method and at the end of the authentication, so separate result indication for intermediate and final result MUST be supported.

### 4.3.6. Internationalization of Display Strings

TOC

The payload MAY provide a standard attribute format that supports international strings. This attribute format MUST support encoding strings in UTF-8 [RFC3629] (Yergeau, F., “UTF-8, a transformation format of ISO 10646,” November 2003.) format. Any strings sent by the server intended for display to the user MUST be sent in UTF-8 format and SHOULD be able to be marked with language information and adapted to the user's language preference.

## 4.4. EAP Channel Binding Requirements

TOC

The tunnel method MUST be capable of meeting EAP channel binding requirements described in [I-D.clancy-emu-chbind] (Clancy, C. and K. Hoeper, “Channel Binding Support for EAP Methods,” November 2008.).

## 4.5. Requirements Associated with Carrying Username and Passwords

TOC

This section describes the requirements associated with tunneled password authentication. The password authentication mentioned here refers to user or machine authentication using a legacy password database or verifier, such as LDAP, OTP, etc. These typically require the password in its original text form in order to authenticate the peer, hence they require the peer to send the clear text user name and password to the EAP server.

#### 4.5.1.  Security

TOC

Due to the fact that the EAP peer needs to send clear text password to the EAP server to authenticate against the legacy user information, the security measures in the following sections MUST be met.

##### 4.5.1.1.  Confidentiality and Integrity

TOC

The clear text password exchange MUST be integrity and confidentiality protected. As long as the password exchange occurs inside an authenticated and encrypted tunnel, this requirement is met.

##### 4.5.1.2.  Authentication of Server

TOC

The EAP server MUST be authenticated before the peer can send the clear text password to the server.

##### 4.5.1.3.  Server Credential Revocation Checking

TOC

In some cases, the EAP peer needs to present its password to the server before it has network access to check the revocation status of the server's credentials. Therefore, the tunnel method MUST support mechanisms to check the revocation status of a credential. The tunnel method SHOULD make use of Online Certificate Status Protocol (OCSP) [RFC2560] (Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” June 1999.) or Server-based Certificate Validation Protocol (SCVP) [RFC5055] (Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, “Server-Based Certificate Validation Protocol (SCVP),” December 2007.) to obtain the revocation status of the EAP server certificate.

### 4.5.2. Internationalization

TOC

The password authentication exchange MUST support user names and passwords in international languages. It MUST support encoding of user name and password strings in UTF-8 [RFC3629] (Yergeau, F., “UTF-8, a transformation format of ISO 10646,” November 2003.) format. Any strings sent by the server during the password exchange and intended for display to the user MUST be sent in UTF-8 format and SHOULD be able to be marked with language information and adapted to the user's language preference.

### 4.5.3. Meta-data

TOC

The password authentication exchange MUST support additional associated meta-data which can be used to indicate whether the authentication is for a user or a machine. This allows the EAP server and peer to request and negotiate authentication specifically for a user or machine. This is useful in the case of multiple inner authentications where the user and machine both need to be authenticated.

### 4.5.4. Password Change

TOC

The password authentication exchange MUST support password change, as well as other multiple round trips exchanges like new pin mode and next token mode for OTP verifiers.

### 4.6. Requirements Associated with Carrying EAP Methods

TOC

The tunnel method MUST be able to carry inner EAP methods without modifying them.

### 4.6.1. Method Negotiation

TOC

The tunnel method MUST support the protected negotiation of the inner EAP method. It MUST NOT allow the inner EAP method negotiation to be downgraded or manipulated by intermediaries.

### 4.6.2. Chained Methods

TOC

The tunnel method MUST support the chaining of multiple EAP methods. The tunnel method MUST allow for the communication of intermediate result and verification of compound binding between executed inner methods when chained methods are employed.

---

### 4.6.3. Cryptographic Binding with TLS Tunnel

TOC

The tunnel method MUST provide a mechanism to bind the tunnel protocol and the inner EAP method. This property is referred to as cryptographic binding. Without such bindings attacks are feasible on tunnel methods [TUNNEL-MITM] (Asokan, N., Niemi, V., and K. Nyberg, “Man-in-the-Middle in Tunnelled Authentication Protocols,” November 2002.) and chained methods.

Cryptographic bindings are typically achieved by securely mixing the established keying material (say tunnel key TK) from the tunnel protocol with the established keying material (say method key MK) from the inner authentication method(s) in order to derive fresh keying material. If chained EAP methods are executed in the tunnel, all derived inner keys are combined to one method key MK. The keying material derived from mixing tunnel and method keys is also referred to as compound tunnel key (CTK). In particular, CTK is used to derive the EAP MSK, EMSK and other transient keys (TEK), such as transient encryption keys and integrity protection keys. The key hierarchy for tunnel methods executions that derive compound keys for the purpose of cryptographic binding is depicted in Figure 1.

```
      -----------
      | TK | MK |
      -----------
         |   |
         v   v
       --------
       | CTK  |
       --------
           |
           v
    ----------------
    |      |       |
    v      v       v
-------  ------  -------
| TEK | | MSK | | EMSK |
------- ------- --------
```

**Figure 1: Compound Keys**

For every key deriving inner EAP method that completes successfully within the tunnel cryptographic binding MUST be performed similar to the following:

- compute a compound key CTK using the keying material from tunnel protocol and all tunneled inner authentication method(s) as inputs

- use compound key CTK to derive transient keys for use in a cryptographic protocol that verifies the integrity of the tunnel and the inner authentication method.

Furthermore, the compound key CTK and all keys derived from it SHOULD be derived in accordance to the guidelines for key derivations and key hierarchies as specified in Section 4.2.1.1.3 (Tunnel Authentication and Key Establishment). In particular, all derived keys MUST have a lifetime assigned that does not exceed the lifetime of any key higher in the key hierarchy, and MUST prevent domino effects where a compromise in one part of the system leads to compromises in other parts of the system.

TOC

### 4.6.4. Peer Initiated

The tunnel method SHOULD allow for the peer to initiate an inner EAP authentication in order to meet its policy requirements for authenticating the server.

### 4.6.5. Method Meta-data

TOC

The tunnel method MUST allow for the communication of additional data associated with an EAP method. This can be used to indicate whether the authentication is for a user or a machine. This allows the EAP server and peer to request and negotiate authentication specifically for a user or machine. This is useful in the case of multiple inner EAP authentications where the user and machine both need to be authenticated.

## 5. IANA Considerations

TOC

This document has no IANA considerations.

## 6. Security Considerations

TOC

A tunnel method is often deployed to provide mutual authentication between EAP Peer and EAP Server and to generate strong key material for use in protecting lower layer protocols. In addition the tunnel is used to protect the communication of additional data, including peer identity between the EAP Peer and EAP Server from disclosure to or modification by an attacker. These sections cover considerations that affect the ability for a method to achieve these goals.

### 6.1. Cipher Suite Selection

TOC

TLS supports a wide variety of cipher suites providing a variety of security properties. The selection of strong cipher suites is critical to the security of the tunnel method. Selection of a cipher suite with weak or no authentication, such as an anonymous Diffie- Hellman based cipher suite will greatly increase the risk of system compromise. Since a tunnel method uses the TLS tunnel to transport data, the selection of a ciphersuite with weak data encryption and integrity algorithms will also increase the vulnerability of the method to attacks.

A tunnel protocol is prone to downgrading attacks if the tunnel protocol supports any key establishment algorithm that can be broken on-line. In a successful downgrading attack, an adversary breaks the selected "weak" key establishment algorithm and optionally the "weak" authentication algorithm without being detected. Here, "weak" refers to a key establishment algorithm that can be broken in real-time, and an authentication scheme that can be broken off-line, respectively. See [KHLC07] (Hoeper, K. and L. Chen, “Where EAP Security Claims Fail,” August 2007.) for more details. The requirements in this document disapprove the use of key establishment algorithms that can be broken on-line.

Mutually anonymous tunnel protocols are prone to man-in-the-middle attacks described in [KHLC07] (Hoeper, K. and L. Chen, “Where EAP Security Claims Fail,” August 2007.). During such an attack, an adversary establishes a tunnel with each the peer and the authentication server, while peer and server believe that they established a tunnel with each other. Once both tunnels have been established, the adversary can eavesdrop on all communications within the tunnels, i.e. the execution of the inner authentication method(s). Consequently, the adversary can eavesdrop on the identifiers that are exchanged as part of the EAP method and thus, the privacy of peer and/or authentication server is compromised along with any other data transmitted within the tunnels. This document requires server authentication to avoid the risks associated with anonymous cipher suites.

---

### 6.2. Tunneled Authentication

TOC

In many cases a tunnel method provides mutual authentication by authenticating the server during tunnel establishment and authenticating the peer within the tunnel using an EAP method. As described in [TUNNEL-MITM] (Asokan, N., Niemi, V., and K. Nyberg, “Man-in-the-Middle in Tunnelled Authentication Protocols,” November 2002.), this mode of operation can allow a man- in-the-middle to authenticate to the server as the peer by tunneling the inner EAP protocol messages to and from a peer executing the method outside a tunnel or with an untrustworthy server. Cryptographic binding between the established keying material from the inner authentication method(s) and the tunnel protocol verifies that the endpoints of the tunnel and the inner authentication method(s) are the same. This can thwart the attack if the inner method derived keys of sufficient strength that they cannot be broken in real-time.

In cases where the inner authentication method does not generate any or only weak key material care must be taken to ensure that the peer does not execute the inner method with the same credentials outside a protective tunnel or with an untrustworthy server.

### 6.3. Data External to Tunnel

TOC

The tunnel method will use data that is outside the TLS tunnel such as the EAP type code or version numbers. If an attacker can compromise the protocol by modifying these values the tunnel method MUST protect this data.

## 7. References

TOC

### 7.1. Normative References

TOC

| | |
|---|---|
| [I-D.clancy-emu-chbind] | Clancy, C. and K. Hoeper, “Channel Binding Support for EAP Methods,” draft-clancy-emu-chbind-04 (work in progress), November 2008 (TXT). |
| [RFC2119] | Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” BCP 14, RFC 2119, March 1997 (TXT, HTML, XML). |
| [RFC2560] | Myers, M., Ankney, R., Malpani, A., Galperin, S., and C. Adams, “X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP,” RFC 2560, June 1999 (TXT). |
| [RFC3629] | Yergeau, F., “UTF-8, a transformation format of ISO 10646,” STD 63, RFC 3629, November 2003 (TXT). |
| [RFC3748] | Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, “Extensible Authentication Protocol (EAP),” RFC 3748, June 2004 (TXT). |
| [RFC4017] | Stanley, D., Walker, J., and B. Aboba, “Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs,” RFC 4017, March 2005 (TXT). |
| [RFC4962] | Housley, R. and B. Aboba, “Guidance for Authentication, Authorization, and Accounting (AAA) Key Management,” BCP 132, RFC 4962, July 2007 (TXT). |
| [RFC5055] | Freeman, T., Housley, R., Malpani, A., Cooper, D., and W. Polk, “Server-Based Certificate Validation Protocol (SCVP),” RFC 5055, December 2007 (TXT). |
| [RFC5246] | Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.2,” RFC 5246, August 2008 (TXT). |

| | |
|---|---|
| [RFC5247] | Aboba, B., Simon, D., and P. Eronen, “Extensible Authentication Protocol (EAP) Key Management Framework,” RFC 5247, August 2008 (TXT). |

---

## 7.2. Informative References

TOC

| | |
|---|---|
| [I-D.ietf-nea-pb-tnc] | Sahita, R., Hanna, S., and K. Narayan, “PB-TNC: A Posture Broker Protocol (PB) Compatible with TNC,” draft-ietf-nea-pb-tnc-06 (work in progress), October 2009 (TXT). |
| [I-D.mahy-eap-enrollment] | Mahy, R., “An Extensible Authentication Protocol (EAP) Enrollment Method,” draft-mahy-eap-enrollment-01 (work in progress), March 2006 (TXT). |
| [KHLC07] | Hoeper, K. and L. Chen, “Where EAP Security Claims Fail,” ICST QShine , August 2007. |
| [NIST SP 800-108] | Chen, L., “Recommendation for Key Derivation Using Pseudorandom Functions,” Draft NIST Special Publication 800-108, April 2008. |
| [NIST SP 800-52] | Chernick, C., Edington III, C., Fanto, M., and R. Rosenthal, “Guidelines for the Selection and Use of Transport Layer Security (TLS) Implementations,” NIST Special Publication 800-52, June 2005. |
| [NIST SP 800-57] | Barker, E., Barker, W., Burr, W., Polk, W., and M. Smid, “Recommendation for Key Management - Part 1: General (Revised),” NIST Special Publication 800-57, March 2007. |
| [RFC2246] | Dierks, T. and C. Allen, “The TLS Protocol Version 1.0,” RFC 2246, January 1999 (TXT). |
| [RFC4282] | Aboba, B., Beadles, M., Arkko, J., and P. Eronen, “The Network Access Identifier,” RFC 4282, December 2005 (TXT). |
| [RFC4346] | Dierks, T. and E. Rescorla, “The Transport Layer Security (TLS) Protocol Version 1.1,” RFC 4346, April 2006 (TXT). |
| [RFC4851] | Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, “The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST),” RFC 4851, May 2007 (TXT). |
| [RFC5077] | Salowey, J., Zhou, H., Eronen, P., and H. Tschofenig, “Transport Layer Security (TLS) Session Resumption without Server-Side State,” RFC 5077, January 2008 (TXT). |
| [RFC5209] | Sangster, P., Khosravi, H., Mani, M., Narayan, K., and J. Tardo, “Network Endpoint Assessment (NEA): Overview and Requirements,” RFC 5209, June 2008 (TXT). |

| | |
|---|---|
| [RFC5281] | Funk, P. and S. Blake-Wilson, “Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0),” RFC 5281, August 2008 (TXT). |
| [TUNNEL-MITM] | Asokan, N., Niemi, V., and K. Nyberg, “Man-in-the-Middle in Tunnelled Authentication Protocols,” Cryptology ePrint Archive: Report 2002/163, November 2002. |

---

### Authors' Addresses

TOC

| | |
|---|---|
| | Katrin Hoeper |
| | NIST |
| | 100 Bureau Drive, MS: 8930 |
| | Gaithersburg, MD 20899 |
| | USA |
| Email: | khoeper@nist.gov |
| | |
| | Stephen Hanna |
| | Juniper Networks |
| | 3 Beverly Road |
| | Bedford, MA 01730 |
| | USA |
| Email: | shanna@juniper.net |
| | |
| | Hao Zhou |
| | Cisco Systems, Inc. |
| | 4125 Highlander Parkway |
| | Richfield, OH 44286 |
| | USA |
| Email: | hzhou@cisco.com |
| | |
| | Joseph Salowey (editor) |
| | Cisco Systems, Inc. |
| | 2901 3rd. Ave |
| | Seattle, WA 98121 |
| | USA |
| Email: | jsalowey@cisco.com |

---

### Full Copyright Statement

TOC

Copyright © The IETF Trust (2008).

This document is subject to the rights, licenses and restrictions contained in BCP 78, and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an “AS IS” basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

**Intellectual Property**

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in BCP 78 and BCP 79.

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at http://www.ietf.org/ipr.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at ietf-ipr@ietf.org.