TCG Trusted Network Communications
TNC IF-T: Protocol Bindings for Tunneled EAP Methods

Specification Version 2.0
Revision 5
8 May 2014
Published

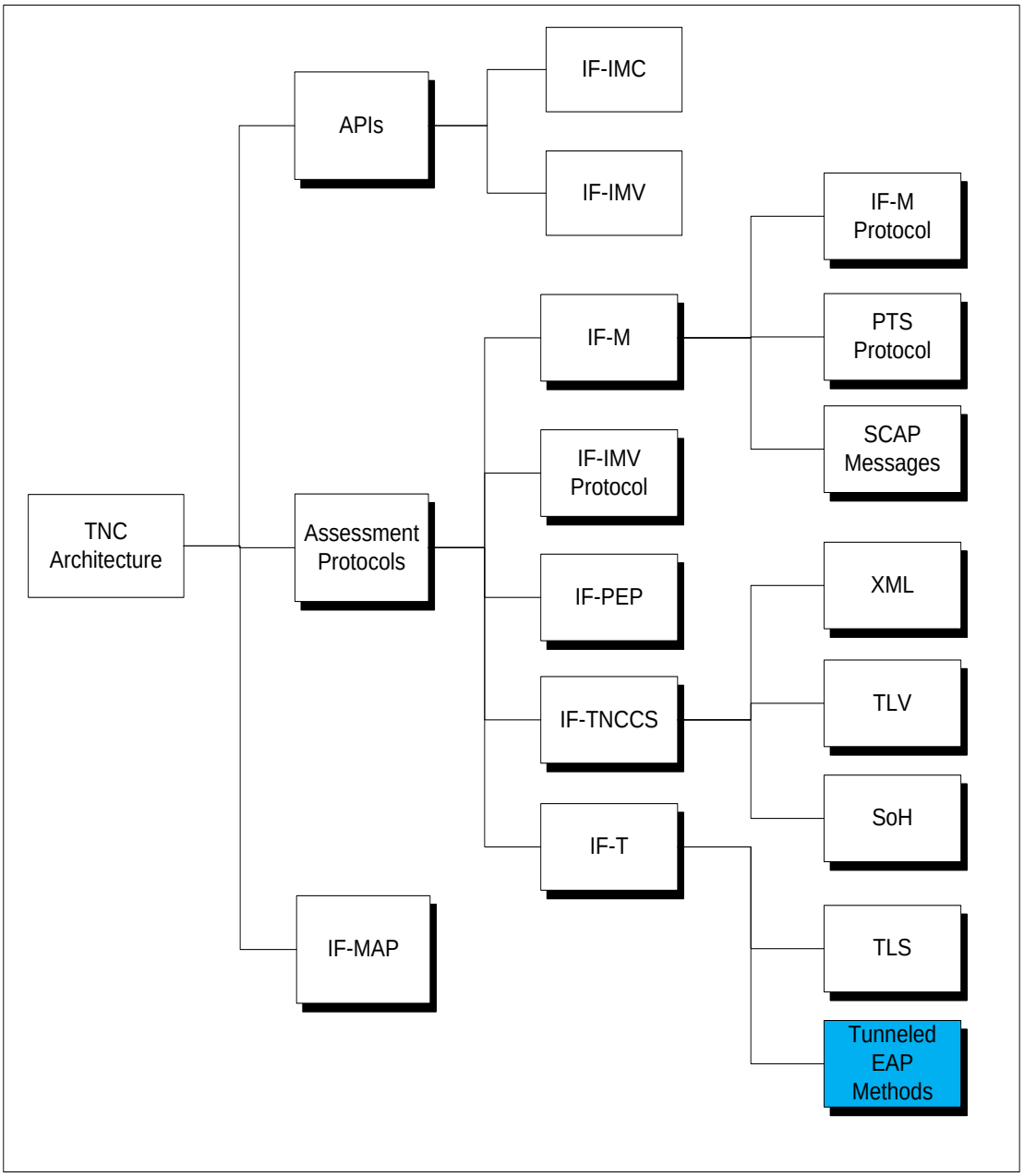
Contact:
admin@trustedcomputinggroup.org

TCG

TCG Published

Disclaimers, Notices, and License Terms

TNC Document Roadmap



Acknowledgements

The TCG wishes to thank all those who contributed to this specification. This document builds on work done in several other working groups in the TCG.

Special thanks to the active and previously active members of the TNC who contributed to the discussions and/or wordings used in this document:

Aman Garg	**3Com**
Bipin Mistry	**3Com**
Scott Kelly	**Aruba Networks**
Amit Agarwal	**Avaya**
Mahalingam Mani	**Avaya**
Steven Venema	**The Boeing Company**
Nancy Cam-Winget	**Cisco Systems**
Scott Pope	**Cisco Systems**
Max Pritikin	**Cisco Systems**
Allan Thomson	**Cisco Systems**
Aaron Woland	**Cisco Systems**
Henk Birkholz	**Fraunhofer SIT**
Hidenobu Ito	**Fujitsu Limited**
Sung Lee	**Fujitsu Limited**
Kazuaki Nimura	**Fujitsu Limited**
Boris Balacheff	**Hewlett-Packard**
Mauricio Sanchez	**Hewlett-Packard**
Ira McDonald	**High North**
Diana Arroyo	**IBM**
Seiji Munetoh	**IBM**
Lee Terrell	**IBM**
Chris Hessing	**Identity Engines**
Morteza Ansari	**Infoblox**
Stuart Bailey	**Infoblox**
Ivan Pulleyn	**Infoblox**
James Tan	**Infoblox**
Uri Blumenthal	**Intel Corporation**
David Grawrock	**Intel Corporation**
Ravi Sahita	**Intel Corporation**
Ned Smith	**Intel Corporation**
Chris Trytten	**iPass**
Barbara Nelson	**iPass**
Steve Hanna (Editor)	**Juniper Networks**
Clifford Kahn	**Juniper Networks**
PJ Kirner	**Juniper Networks**
Lisa Lorenzin (TNC-WG Co-Chair)	**Juniper Networks**
Dean Sheffield	**Juniper Networks**
John Jerrim	**Lancope**
Kent Landfield	**McAfee**
Gene Chang	**Meetinghouse Data Communications**
Alex Romanyuk	**Meetinghouse Data Communications**
John Vollbrecht	**Meetinghouse Data Communications**
Ryan Hurst	**Microsoft Corporation**
Atul Shah (TNC-WG Co-Chair)	**Microsoft Corporation**
Beth Abramowicz	**The MITRE Corporation**
Charles Schmidt	**The MITRE Corporation**
Sandilya Garimella	**Motorola**

Rainer Enders	**NCP**
Joseph Tardo	**Nevis Networks**
Pasi Eronen	**Nokia Corporation**
Meenakshi Kaushik	**Nortel Networks**
Ron Pon	**Nortel Networks**
Dick Wilkins	**Phoenix Technologies**
Bryan Kingsford	**Symantec Corporation**
Paul Sangster	**Symantec Corporation**
Curtis Simonson	**University of New Hampshire InterOperability Labs**
Jessica Fitzgerald-McKay	**U.S. Government**
Chris Salter	**U.S. Government**
Jeff Six	**U.S. Government**
Richard Struse	**U.S. Government**
David Waltermire	**U.S. Government**
Rod Murchison	**Vernier Networks**
Michelle Sommerstad	**Vernier Networks**
Scott Cochrane	**Wave Systems**
Thomas Hardjono	**Wave Systems**
Greg Kazmierczak	**Wave Systems**

Table of Contents

1 Scope and Audience

Trusted Network Communications (TNC) is a working group within the Trusted Computing Group (TCG). TNC is defining an open solution architecture that enables network operators to enforce policies regarding endpoint integrity when granting access to a network infrastructure. Part of the TNC architecture is IF-T, a standard for mapping the communications between TNC Clients and TNC Servers onto existing protocols. Because TNC enables assessment to occur during the process of joining a network and after the endpoint has been placed on the network, several bindings of IF-T exist to address these different scenarios.

This document defines and specifies the IF-T protocol used when the endpoint has not yet joined the network. In this circumstance, the assessment is carried as EAP messages over 802.1X or IKE. This document is equivalent to IETF's PT-EAP specification and does not add any requirements to PT-EAP. Rather, it simply clarifies where PT-EAP fits in the TNC architecture. Readers interested in the use of IF-T when the endpoint has an IP address should refer to the TNC IF-T: Binding for TLS specification [6].

IF-T is integral to the TNC reference architecture. The relationship of IF-T to other components of the basic TNC reference architecture is shown below in Figure 1.

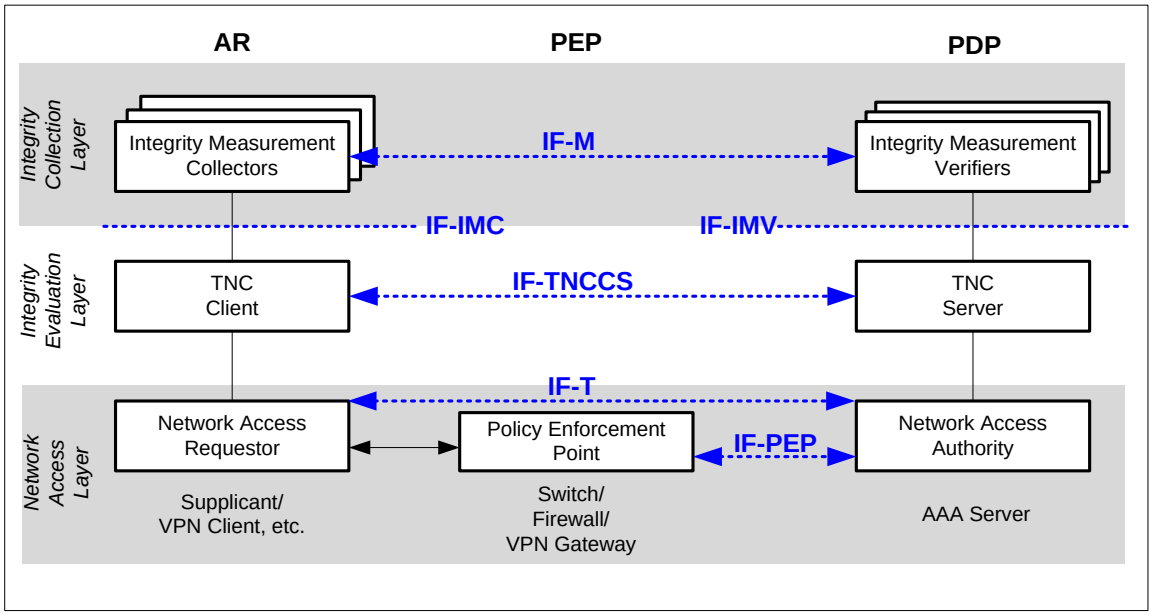


Figure 1. Basic TNC Architecture

Architects, designers, developers, and technologists interested in the development, deployment, and interoperation of trusted systems will find this document necessary in providing specific mechanisms for transporting integrity information.

Before reading this document any further, the reader should review and understand the TNC architecture as described in [3].

1.1 Interoperable with IETF PT-EAP

One of the goals of the Trusted Network Communications WG is to maximize interoperability using open standards. As part of fulfilling this goal, the TCG chose to take the TCG standard IF-T Binding to Tunneled EAP Methods protocol to the IETF for standardization. The IETF standardization process has now been completed, allowing both the TCG and IETF to publish interoperable standards at approximately the same time. This specification defines a new version 2.0 of the IF-T Binding to Tunneled EAP Methods protocol that is interoperable with the IETF's equivalent protocol PT-EAP [2]. The TCG intends to keep the IF-T Binding for Tunneled EAP Methods protocol and the IETF's PT-EAP protocol interoperable for the future.

1.2 IETF Terminology Mapping to TNC

In case readers of this specification are also looking at the IETF Network Endpoint Assessment (NEA)'s PT-EAP specification, this section provides some guidance on how the terminology aligns between the IETF and NEA specifications.

PA-TNC - IETF NEA name for the application layer protocol [19] that is interoperable with IF-M [5]. "PA" is short for "Posture Attribute" protocol and "-TNC" highlights that the protocol is based upon work originally submitted by the TNC and is interoperable with this specification.

PB-TNC - IETF NEA name for the protocol between the NEA client to NEA server that is interoperable with the TNC's IF-TNCCS 2.0. As with PA-TNC, the PB-TNC [20] protocol is based upon work originally submitted by the TNC and is interoperable with IF-TNCCS 2.0 thus carries the "-TNC" suffix.

PT-EAP - IETF NEA name for the transport protocol equivalent to this document. The PT-EAP specification was largely based upon the TCG predecessor specification and the current versions of these documents are fully interoperable.

PT-TLS - IETF NEA name for the transport protocol equivalent to the IF-T Binding for TLS specification from TCG. The PT-TLS specification was largely based upon the TCG predecessor specification and the current versions of these documents are fully interoperable.

Posture – IETF NEA term for "measurement information" or "integrity measurement" used by TNC. The posture is returned from the NEA client (typically from its Posture Collectors) as part of an assessment. This is synonymous with the measurement information returned by the TNC client's IMCs.

Posture Collector – IETF NEA term synonymous with TNC's Integrity Measurement Collector (IMC)

Posture Validator – IETF NEA term synonymous with TNC's Integrity Measurement Validator (IMV)

2 Background

2.1 Purpose of IF-T and EAP Protocol Bindings

As shown in Figure 1, IF-T is the transport that carries IF-TNCCS messages between Network Access Requestor (NAR) and the Network Access Authenticator (NAA). IF-TNCCS [4] is a TNC specified protocol for carrying IF-M [5] protocol messages between Integrity Measurement Collectors (IMCs) and Integrity Measurement Verifiers (IMVs). This specification is for a protocol mapping of IF-T to a set of Tunneled EAP protocol methods that can be used to provide IF-T transport during access request dialogs.

The TNC usage of IF-T enables assessments of endpoints as they are joining the network or after the endpoints are on the network. For scenarios when the endpoint is in the process of joining the network, the TNC assessment needs to be carried within the protocol used during the joining process. This protocol could be a layer two (link level) protocol, which needs to leverage an existing protocol such as 802.1 X that allows for the exchange of EAP messages. This network join-time usage is the subject of this specification (IF-T Bindings for Tunneled EAP Methods). In contrast, the TNC IF-T Bindings for TLS specification focuses on the IF-T usage model where the endpoint is already present on the network and thus has an IP address assigned, so is reachable using TCP/IP by other systems.

This document describes and specifies a mapping of IF-T to tunneled Extensible Authentication Protocol (EAP) methods. A tunneled EAP method is one that provides a cryptographically protected wrapper within which other protocol elements can be exchanged. Suitable tunneled EAP methods for IF-T are those able to carry nested EAP exchanges as protected protocol elements. This document further specifies an EAP wrapper for IF-TNCCS, enabling IF-TNCCS to be carried as a nested EAP method within a suitable tunneled EAP method.

For interoperability, the protocol bindings specified in this document MUST be implemented in any product claiming TNC compliance and providing IF-T using tunneled EAP methods. These tunneled EAP bindings make it possible to implement IF-T over a number of existing access protocols that use EAP at the access level. Some examples of such access protocols include 802.1X for wired and wireless, and IKEv2 for establishing VPNs over IP networks.

2.2 Requirements

Here are the requirements for IF-T Protocol Bindings for Tunneled EAP Methods.

- Meets the needs of the TNC architecture

  The IF-T Binding for Tunneled EAP Methods must support all the use cases described in the TNC architecture as they apply to transporting IF-TNCCS messages between the TNCC and TNCS.

- Provide security

The integrity and confidentiality of communications between IMCs and IMVs must be protected. The IF-T Binding for Tunneled EAP Methods must specify how to provision secure communications between the TNCC and TNCS to transport IF-TNCCS messages. See the Security Considerations section.

- Be efficient

  The TNC architecture delays network access until certain endpoint integrity checks have been performed. To minimize user frustration, it is essential to minimize delays and make IF-T communications as rapid and efficient as possible. Efficiency is also important when you consider that some network endpoints are small and low powered.

- Provide a half duplex message protocol

  IF-T Binding for Tunneled EAP Methods guarantees delivery of messages in the order received, and provides reliable transmission of data, handling retransmission and fragmentation of messages if needed.

- Be extensible

  IF-T will need to be expanded over time as new features are added to the TNC architecture and new use cases identified.

2.3 Keywords

The key words “MUST”, “MUST NOT”, “REQUIRED”, “SHALL”, “SHALL NOT”, “SHOULD”, “SHOULD NOT”, “RECOMMENDED”, “MAY”, and “OPTIONAL” in this document are to be interpreted as described in RFC 2119 [1]. This specification does not distinguish blocks of informative comments and normative requirements. Therefore, for the sake of clarity, note that lower case instances of must, should, etc. do not indicate normative requirements.

2.4 Features Provided by IF-T

The TNC architecture does not specify that any particular protocol be used for IF-T, and in fact specifies that different protocols may be used. This document provides the specification of Protocol Bindings when a Tunneled EAP method is used as the method to carry IF-T. This protocol should be used when using a version of IF-T that uses a tunneled EAP method.

In particular this document describes the mapping of IF-TNCCS messages to a standard EAP method: PT-EAP. This document does not define the PT-EAP method. Instead, it refers implementers to the PT-EAP specification published by

IETF [2]. It also describes how to use four existing tunneled EAP methods to carry PT-EAP: EAP-FAST, EAP, EAP-TTLS, and PEAP.

The PT-EAP method is an EAP inner method which is compatible with the EAP framework defined by IETF [24]. PT-EAP should be used when TNC is used with tunneled EAP methods. PT-EAP carries the IF-TNCCS messages, and is itself carried as an inner method by one of the tunneled EAP methods.

3 Use of PT-EAP with Tunneled EAP Methods

3.1 Model

Figure 2 shows the protocol layers that combine to provide IF-T using PT-EAP over tunneled EAP methods. All of the highlighted layers have components that are part of the IF-T Protocol Binding for Tunneled EAP Methods.

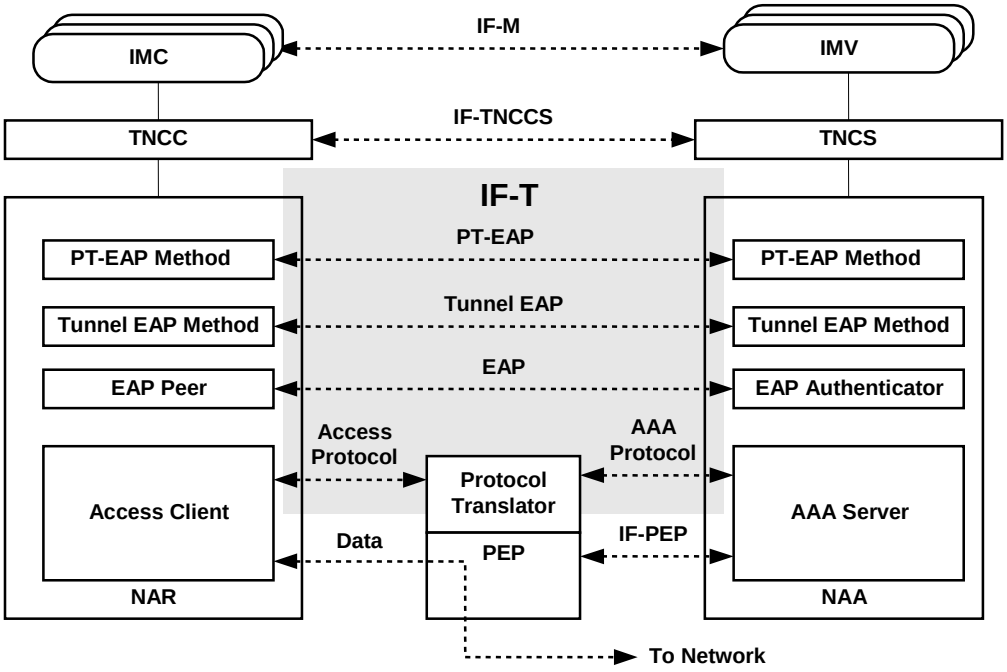


Figure 2. PT-EAP and EAP Protocol Layers

The Access Requestor consists of the NAR, the TNCC and the IMCs. The PDP consists of the NAA, the TNCS and the IMVs. In Figure 2, the NAR and NAA communicate via four protocol layers which combine to form the IF-T Protocol Binding for Tunneled EAP Methods. Each side must send protocol messages that interoperate at each of these layers.

Starting from the top, the PT-EAP method is a simple EAP method. This method is specified by IETF in the PT-EAP specification [2]. PT-EAP encapsulates IF-TNCCS messages so that they can be carried over tunneled EAP methods using standard techniques.

The tunneled EAP Method creates a cryptographically protected tunnel over EAP. It then carries a sequence of EAP frames over the tunnel it has created. The EAP Peer and Authenticator exchange EAP messages and manage EAP negotiation and protocol sequencing. EAP is described in [24], and the EAP state machine in [8].

The access client initiates the access control dialog with the protocol translator. 802.1X and IKEv2 are examples of access protocols. In this document, 802.1X

and IKEv2 are shown as example access use cases. However, other access protocols may be used as long as they support EAP authentication.

An AAA protocol is used to communicate between the Protocol Translator and the NAA. This AAA protocol carries EAP messages for IF-T as well as IF-PEP. RADIUS and Diameter are examples of AAA protocols.

3.1.1 Tunneling

IF-TNCCS messages are carried within the PT-EAP method. The details of the PT-EAP method are defined in the IETF's PT-EAP specification [2].

PT-EAP can be carried within any EAP tunneled method that supports inner EAP methods as an "inner EAP stream." This inner EAP stream is carried in different ways depending upon the particular tunneled EAP method. These differences are described below.

Interoperability requires that both sides use the same tunneled EAP method, and that peer and authenticator vendors implement to the same PT-EAP standard. This allows a client with tunneled method peer from one vendor can communicate with a tunneled method authenticator from a different vendor,

EAP messages are carried by an access protocol (e.g., 802.1X) from the NAR to the Protocol Translator, and by RADIUS (or other AAA protocol) from the Protocol Translator to the NAA. If vendors implement according to EAP, access protocol, and AAA protocol specifications, then a peer from one vendor can talk with an authenticator from another.

Finally the access protocol on the client must work with the protocol translator. This specification provides examples of 802.1X and IKEv2 mapping in Section 4. Additional access protocol mappings may be specified later.

3.1.2 Protocol Encapsulation

The following figures show protocol encapsulation of messages on the client and server side. In both cases messages are exchanged with the protocol translator (e.g. wireless access point, switch, or gateway). The protocol translator removes an EAP message from the access protocol (e.g., 802.1X or IKEv2) and forwards it over the AAA protocol (e.g., RADIUS). It also does the reverse. Figure 3 shows how protocols are encapsulated at the different layers, and how a message "on the wire" looks.

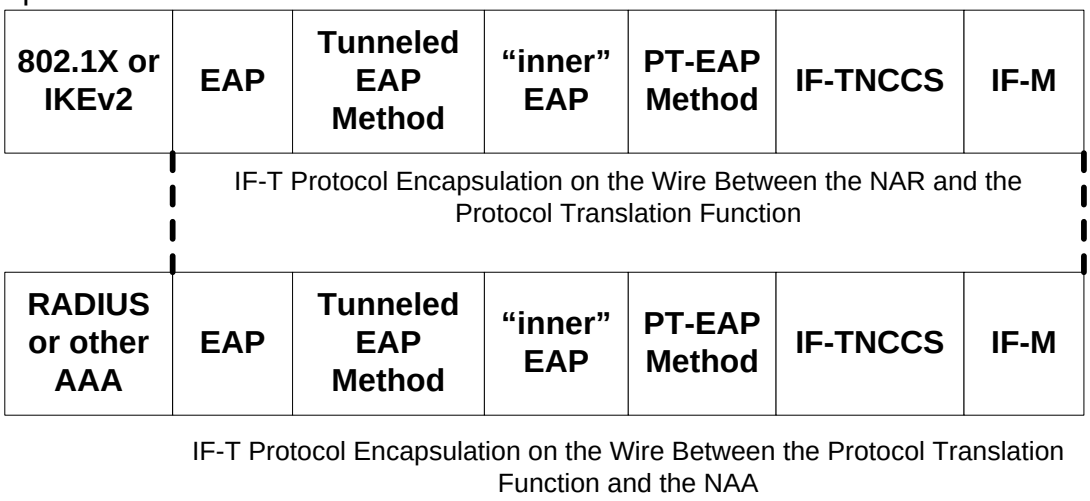


Figure 3. IF-T Protocol Encapsulation on the Wire

3.2 PT-EAP

PT-EAP is a very simple EAP method that MUST run over a tunneled EAP method. It is described in the IETF's PT-EAP specification [2]. Figure 4 below shows how PT-EAP carries an IF-TNCCS handshake.

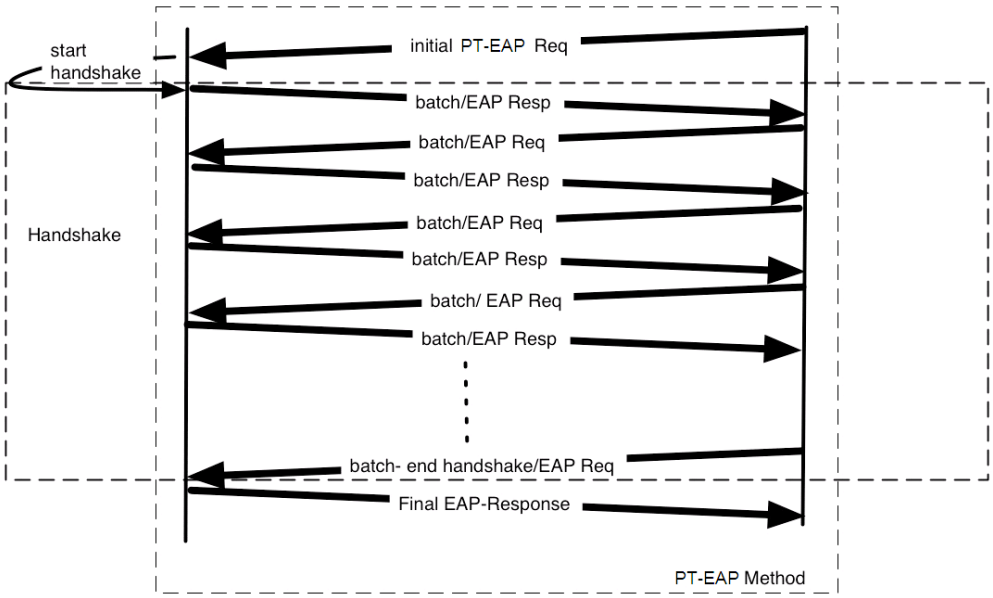


Figure 4. PT-EAP IF-TNCCS Handshake

PT-EAP is an “inner” method. EAP messages for the PT-EAP method are sent over a tunnel created by a tunneled EAP method. It is required that PT-EAP be carried on a TLS tunnel in order that the conversation between the client and the server be protected. TLS provides integrity and confidentiality between the client and the RADIUS Server.

3.3 Inner EAP Peer and Authenticator

Tunneled EAP methods make it possible to carry one or more “inner” EAP methods over a protected tunnel created in the first phase of the tunneled EAP method. Phase 1 is often called the “outer” method, and methods carried over the tunnel are called “inner” methods. In existing EAP methods a particular protocol element, either a Type-Length-Value (TLV) or Attribute-Value Pair (AVP) is defined for carrying “inner” EAP messages.

Inner EAP messages are sent end-to-end between inner EAP peer and authenticator. The inner peer and authenticator work just like the normal peer and authenticator, with a few exceptions noted below. Tunneled EAP methods that provide security for inner methods are allowed to carry sequences of EAP methods. In addition, most tunneled EAP methods allow the peer and authenticator pair to signal each other’s tunnel endpoint using special AVPs.

3.4 Tunneled EAP Methods

A number of Tunneled EAP methods have been implemented by different vendors. IETF is working on a standard Tunneled EAP Method to replace them. These methods all consist of two phases: the first phase creates a TLS tunnel over EAP; and the second phase carries other information protected using the TLS tunnel. A major intent of these methods has been to provide a secure path over which other authentication or authorization dialogs can be done. The outer tunneled EAP method secures the inner dialogs. This allows otherwise insecure EAP methods to be used securely as long as the outer EAP method meets the security requirements.

Because the outer EAP method provides protection against a wide variety of active attacks, the inner PT-EAP method largely focuses on reporting of integrity information. Hence, PT-EAP MUST NOT be used as a stand-alone method. PT-EAP MUST only be used as an inner method within a protected tunneled EAP created by an outer EAP method.

The currently defined PT-EAP inner method does not provide its own user or platform authentication mechanisms. PT-EAP message payloads may carry IF-M messages that include additional authentications, but PT-EAP does not depend upon and is not aware of such services occurring. PT-EAP SHOULD be used within a tunneled EAP method that provides authentication or can carry other authentication methods within the tunnel. If the authentication method is itself a tunneled EAP method, the tunneled EAP method must allow a sequence of EAP methods to be carried within it. It is assumed that PT-EAP may be run in addition to other authentication methods within the tunneled EAP method.

The following sections provide further guidance on using PT-EAP with specific tunneled EAP methods.

3.4.1 EAP-FAST, TEAP, and PEAPv2

EAP Flexible Authentication via Secure Tunneling (FAST) [13], TEAP [21], and Protected EAP Version 2 (PEAPv2) [17] are all tunneled EAP methods that define how to run multiple inner EAP methods. In these methods, PT-EAP is carried in an EAP payload TLV.

EAP FAST, TEAP, and PEAPv2 also provide an “over the tunnel” protocol with messages between tunnel endpoints at each end. This protocol manages messages sent over the TLS tunnel created in its initial phase. This “over the tunnel” protocol includes messages such as intermediate success/fail and crypto binding.

3.4.2 EAP-TTLS

In EAP-TTLS [15], PT-EAP is carried in an EAP AVP, as described in the EAP-TTLS specification. Sequences of inner EAP methods are supported by this specification.

3.4.3 PEAPv0/1

Neither PEAPv0 [18] nor PEAPv1 [16] define how to run multiple inner EAP methods. EAP sequences are not prohibited in PEAPv0/1 but are implementation dependent. Hence, in PEAPv0/1, PT-EAP would be carried directly on the tunnel.

When using PEAPv0/1 with PT-EAP and a traditional authentication method, the server is responsible for sending the sequence of inner EAP methods and checking results.

The following provides an example as to how this could be done. Note, one method of supporting inner EAP sequences that can easily be implemented is to use the same mechanism as defined for EAP-TTLS version 0: The authentication server starts the next EAP method by sending EAP-Request with the new method type once the previous method is completed but before sending inner result indication.

3.5 PT-EAP Sequencing

PT-EAP is one of a number of possible dialogs that can take place over the tunnel created in the first phase of tunneled EAP methods. TNC does not require any specific ordering of dialogs.

Possible scenarios include

1. PT-EAP is the only dialog that runs over the tunnel. In this case Phase 1 of the tunneled EAP method provides client and server authentication as needed.
2. PT-EAP is used in addition to one or more other inner EAP methods which might include a user authentication dialog all within the same EAP

outer tunnel. For example PT-EAP could run either before or after MD5 or MSCHAP allowing for an authenticated identity to be linked to the TNC integrity exchange.

It should also be noted that efficiency should be considered when using and ordering multiple EAP dialogs.

4 Access Protocol Bindings (Informative)

This section shows example protocol bindings that allow access protocols that use EAP authentication to support TNC capabilities using tunneled EAP methods. This section is non-normative.

Figure 5 shows the relationship of Access client and higher level protocols used to provide IF-T.

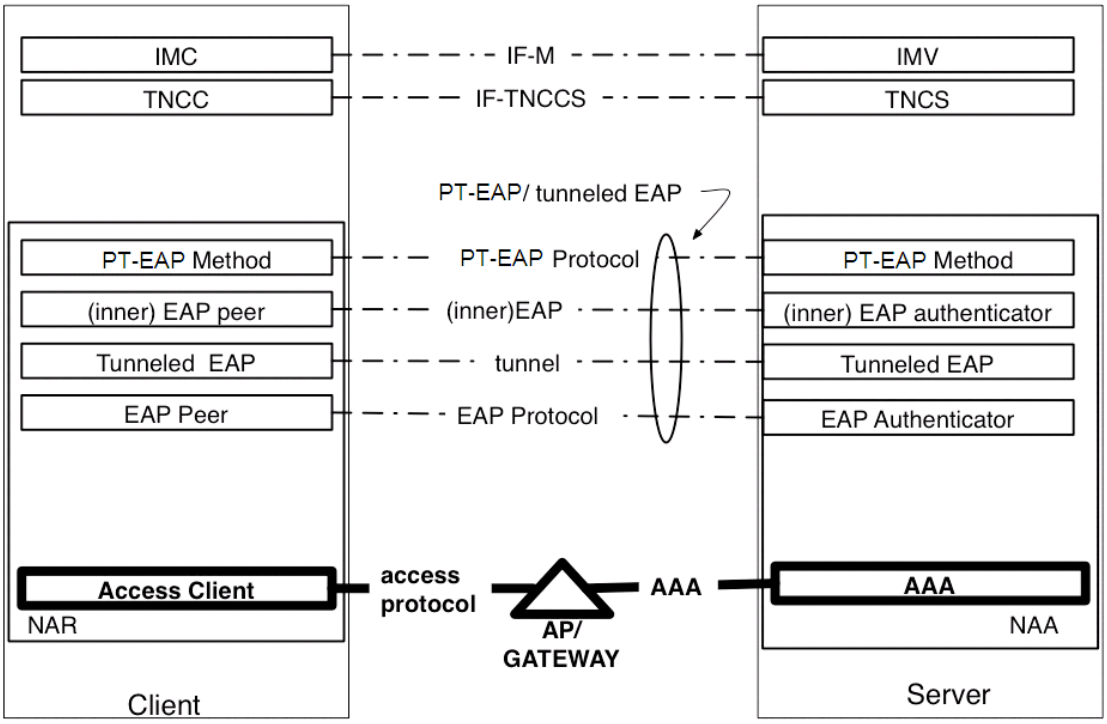


Figure 5. Access Client and Gateway Relationship

The examples in the following sections show how two different access protocols, 802.1X and IKEv2, interface with a tunnel EAP method to provide TNC functionality.

Access protocols are used by the NAR to establish network connectivity (e.g. at the link layer for 802.1X or the network layer for IKEv2 with IPsec.) Other access protocols exist that fit the TNC model including: PPP for Dial Access and 802.16e.

4.1 802.1X

Figure 6 below shows how 802.1X [7] can use PT-EAP to facilitate incorporating TNC capabilities into access decisions. 802.1X is used to control access to 802.3 (wired Ethernet switch) and 802.11 (wireless) networks.

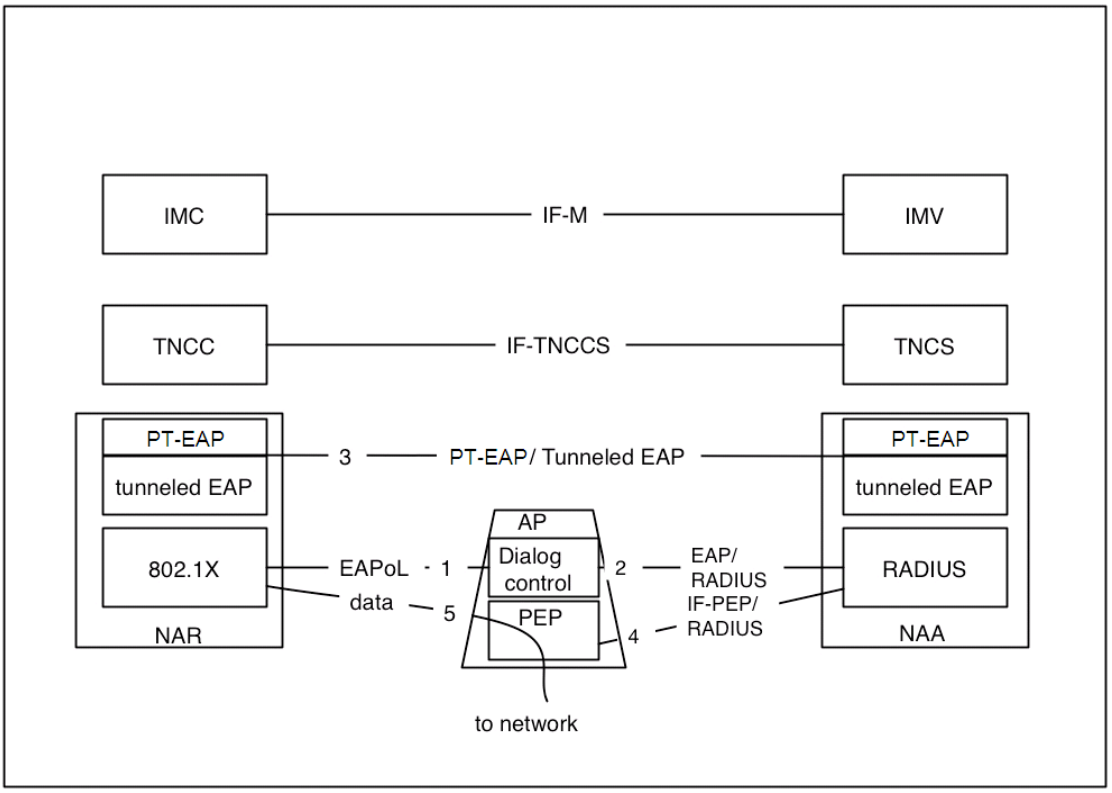


Figure 6. IF-T Over 802.X

This diagram explodes the low level protocol, showing EAPoL (EAP over LAN) being used between the 802.1X supplicant (client) and the 802.1X authenticator (wireless access point or fixed LAN switch), and RADIUS being used between the 802.1X authenticator and the 802.1X authentication server (RADIUS server). EAPoL and RADIUS carry EAP messages, and the authenticator creates an end-to-end EAP path between the client and server by moving EAP messages between the two protocols.

In addition to carrying EAP messages, both EAPoL and RADIUS have other functions and messages. For example, EAPoL includes EAPoL-Start, EAPoL-Logoff, and EAPoL-Key messages, all of which communicate only between the client and AP. On the server side, RADIUS messages typically may contain several attributes in addition to EAP messages.

Thus, at the bottom layer there are actually five dialogs, as shown in Figure 6.

1. The 802.1X dialog between the client and AP to control client access
2. The RADIUS dialog between AP and server to authorize access
3. The EAP dialog between the client and server to authenticate and validate the client

4. The RADIUS dialog between the server and the PEP on the AP. This dialog is actually a command that tells the AP how to respond to the access request
5. Data from the client to the AP which is controlled by the PEP.

4.2 IKEv2

IKEv2 is used to negotiate security settings and ultimately establish shared keys normally used with IPsec to protect subsequent packet exchanges. For example a client system may use IKEv2 to establish keys with a VPN gateway. These keys can then used to create an encrypted IPsec tunnel between the client and gateway. Figure 7 shows how TNC can be incorporated into this capability.

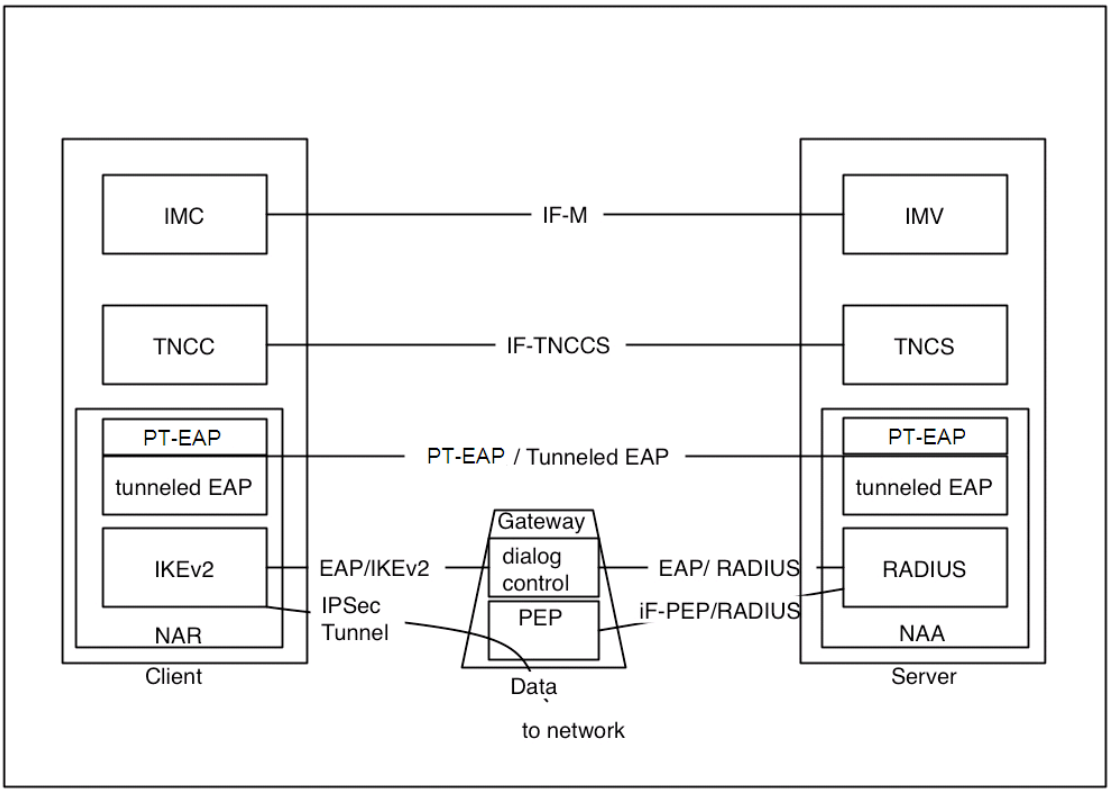


Figure 7. IF-T Over IKEv2

IKEv2 supports use of EAP as an authentication framework as one of the possible standard methods of providing authentication. When EAP is used with IKEv2, it allows the gateway to use RADIUS to request authorization from a remote server, just as an authenticator does for 802.1X. One reason for allowing EAP in IKEv2 is to permit use of legacy authentication mechanisms, such as passwords or OTP one-way schemes, in addition to pre-shared-secret or certificate-based mutual authentication mechanisms supported in IKEv1.

IKEv2 with TNC must use PT-EAP as an inner method of a tunneled EAP method. This is necessary because IKEv2 doesn't support the ability to make an

authorization decision based on a sequence of EAP methods. In the case where PT-EAP and another EAP dialog such as user authentication (e.g., EAP-MD5) run over the outer tunneled EAP method, the outer method provides aggregation of the result of the multiple inner methods.

Thus, at the bottom layer there are the equivalent five dialogs (as shown in Figure 6) using IKE in Figure 7.

1. The IKEv2 dialog between the client and AP to control client access
2. The RADIUS dialog between AP and server to authorize access
3. The EAP dialog (over IKEv2) between the client and server to authenticate and validate the client
4. The RADIUS dialog between the server and the PEP on the AP. This dialog is actually a command that tells the AP how to respond to the access request
5. Data from the client to the AP which is controlled by the PEP.

4.2.1 IKEv2 Dialog

The basic IKEv2 authentication sequence consists of a four-message handshake between the two IKE peers, referred to as the initiator and the responder. In this case the initiator is the endpoint system requesting access and the IPsec gateway might be the responder. The first two messages carry an ephemeral Diffie-Hellman exchange and cipher suite negotiation parameters. In the third message, the initiator proves its identity by sending a signed AUTH payload. In the fourth message, the responder proves its identity with a signed AUTH payload. The signature algorithm may be based on a pre-shared secret or on RSA X.509 certificates containing an RSA public key.

When using EAP for authentication, the remote access client omits the AUTH payload in the third message while simply declaring its identity. This signals the responder, such as an IKEv2 gateway, that EAP authentication is requested. If supported and configured, the responder returns a fourth message containing an EAP request. IKEv2 endpoints then carry EAP messages until the EAP authentication is complete. Note: IKEv2 authentication may be provided by the EAP method, and when doing TNC with IKEv2 it is required to use EAP as the IKEv2 Authentication method.

When doing TNC, the initial outer tunneled EAP-method creates its own shared key. That shared key is used by both the initiator and responder to generate AUTH payloads using the syntax for shared secrets specified in [14]. The shared key from EAP is the field from the EAP specification named Master Session Key (MSK).

5 PT-EAP Protocol Reference (Normative)

For a description of the PT-EAP method, see the PT-EAP specification [2]. All implementations of IF-T Bindings for Tunneled EAP Methods MUST implement this method as specified in the PT-EAP specification.

6 Security Considerations

6.1 Threat Model

The threat model for IF-T asserts that there are two parties interested in interoperating (client and server) and a third (attacker) interested in exploiting vulnerabilities. IF-T provides protection between the NAR and the NAA against attacks on the communication path between them. IF-T authenticates the NAR and NAA, and provides a secure channel for carrying IF-TNCCS messages. The security includes integrity protection against data modification and encryption to protect against eavesdropping.

6.1.1 Threats

The attacker's goals with regard to IF-T are assumed to be the following.

1. Exploit a vulnerability on the end system to defeat the protection provided by IF-T
2. Attack the IF-T authentication dialog to enable a spoofing attack
3. Mount a cryptographic attack on IF-T to expose TNC data
4. Mount a Man in the Middle Attack on the initial access attempt

6.2 IF-T Capabilities

6.2.1 Interaction with Platform Trust Services (PTS)

The PTS is a local service that can optionally be leveraged by the TNC architecture to measure and report upon the state of software present on the system which TNC relies upon for its security. The PTS can leverage the TPM and other trusted components on the system in such a way that it could provide for protected evidence and optionally prevention of malware from running on the system. TNC verifiers can request evidence from the client PTS in order to be more assured that the responses from the other IMCs are trustworthy and not subject to subversion by malware running on the system. Such information can be factored into the network connection and subsequent access decisions made by the verifier.

One major benefit of TNC participation in TCG is the ability to provide the client and server a facility to cryptographically verify the integrity of the TNC components of the peer system. IF-T may also be provided proof of cryptographic integrity of all or part of the peer system as a whole.

Cryptographic verification of client modules by PTS is done by either 1) requesting PTS to measure IF-T modules prior to an IF-T request, or 2) registering IF-T modules with PTS and automatically measuring them during the boot process of the platform. Cryptographic measurements of client side IF-T modules may be sent to the Server as part of data sent by PTS-IMC and checked by PTS-IMV. PTS and the IF-M protocol used by PTS IMC and IMV are described in a forthcoming specification. The local IF-PTS interface (used by PTS IMC) is described in the

IF-PTS Specification [11]. A set of XML-based schemas used by the PTS to report integrity information can be found at the TCG website [12].

In addition to measurement of modules, IF-T modules may interact directly with other aspects of the TCG Trusted Platform architecture to utilize cryptographic signing and encryption of messages sent between client and server.

6.2.2 Authentication Protection

For this specification of mapping IF-T to tunneled EAP methods, authentication protection is provided by the tunneled EAP method optionally augmented by use of other authenticating inner EAP methods. For this protocol binding, TNC recommends that the outer tunnel method be based on either a secure mutual authentication using symmetric keys or one way or mutual public key authentication.

6.2.3 Protection of TNC Data

TNC data is protected by the tunnel provided by the outer method of the tunneled EAP method. The tunnel is typically provided using TLS. The amount of data in a TNC exchange is likely to be limited to that required for an authorization dialog, which is typically small. See section 6.4.2 for a summary of the required security protections provided around the PT-EAP data messages by the outer tunneling methods.

6.3 Some Attack Scenarios

IF-T is concerned with the communication channel between the TNCC and TNCS. An attacker may use the following capabilities or techniques. The protocol binding specification protects against all of these types of attack:

1. Eavesdropping
    a. To learn client vulnerabilities
    b. To extract client identification to impersonate client in IF-TNCCS handshakes.
2. Modification
    a. To represent a client as in compliance when not, so the server does not remediate, permitting an exploit against the client
    b. To represent client as out of compliance, so that server isolates or blocks
    c. To misrepresent the TNCS recommendation to the client
    d. To deliver erroneous remediation instructions to the client
3. Impersonation
    a. Of server, to discover vulnerabilities
    b. Of client, to obtain or infer reference measurement data

These can all be mounted using a man in the middle (MITM) attack.

6.4 Philosophy of Protection

6.4.1 Scope of Protection

IF-T as mapped to tunneled EAP methods is a network protocol providing a protected transport to the TNCC and TNCS for message exchange. Because it is a protocol, the protections it affords are limited to the network communication channel and do not extend beyond the IF-T interface. Protocols layered on top of IF-T can assume the presence of the mandated security protections for IF-T described in section 6.4.2, but SHOULD provide security for higher layer protocol attacks (e.g. message falsification) that impact their ability to perform the higher layer function.

IF-T does not offer protection from local attacks. If malware has infected a system and is capable of interception, replacement or deletion of IMC or IMV messages before they receive IF-T's protections, IF-T will not be able to detect or prevent this from occurring. Use of proper host-based security protection is necessary to address such attacks and assure the proper operation of the IF-T mechanism. Other TNC architecture specifications such as IF-PTS SHOULD be used to address such attacks.

6.4.2 Minimum security Protection

In order for higher level protocols such as IF-TNCCS and IF-M to make assumptions as to the minimum level of protection that IF-T provides, this section describes the required security properties that any IF-T MUST meet. All IF-T bindings MUST include an explanation of how these properties will be achieved.

The security requirements described in this section MUST be implemented in any product claiming to be TNC compliant. The decision of whether a particular deployment chooses to use these protections is a deployment issue. A customer may choose to avoid potential deployment issues or performance penalties associated with the use of cryptography when the required protection has been achieved through other mechanisms (e.g. physical isolation). If security mechanisms may be deactivated by policy, an implementation should offer an interface to query how a message will be (or was) protected by IF-T.

Compliant IF-T bindings and products implementing them using tunneled EAP methods MUST support:

1. Cryptographic authentication of the NAA to the NAR
2. NAR authentication and TNC dialog protected by at least a cryptographic transport
3. Encryption of the message stream tied to at least the transport authentication

4. Cryptographic integrity protection of the message tied to at least the transport authentication
5. Protection against replay attack

Having the NAR always authenticate the NAA provides assurance to the NAR that the NAA is authentic (not a rogue or MITM) prior to disclosing secret or potentially privacy sensitive information about what is running or configured on the system. However the NAA's policy may allow for the delay of the authentication of the NAR until a suitable protected channel has been established allowing for non-cryptographic NAR credentials (e.g. username/password) to be used. Whether the communication channel is established with both or one party performing a cryptographic authentication, the resulting channel needs to provide strong integrity and confidentiality protection to its contents. These protections are to be bound to at least the authentication of the NAA, so the session is cryptographically bound to a particular authentication event.

6.4.3 Tunneled EAP Minimum Protections

This section discusses how PT-EAP used within the tunneled EAP methods described in section 3.4 meets the IF-T requirements from section 6.4.2 above.

EAP-FAST [13], PEAPv0/v1/v2 [18] [16] [17], TEAP [21], and EAP-TTLS [15] all make use of TLS [9] to protect the transport of information between the NAR and NAA. Each of these has two phases, and in the first phase a TLS tunnel is established between NAR and NAA, and in the second phase the tunnel is used to pass other information. IF-T requires that establishing this tunnel include authentication of the NAA by the NAR.

The phase two dialog may include authentication of the user by doing other EAP methods or in the case of EAP-TTLS by using non-EAP authentication dialogs. PT-EAP is also carried by the phase 2 tunnel. The phase 2 TLS tunnel provides support for requirements 2-5 above.

With all these methods, a cryptographic key is derived from the authentication that may be used to secure later transmissions. For these methods this means that server side certificates are required. Within each tunneled EAP method will exist a set of inner EAP methods (or an equivalent using TLVs if inner authentication methods are directly supported.) These inner methods may perform additional security handshakes including more granular authentications or exchanges of integrity information (such as PT-EAP.) At some point after the conclusion of these inner methods, some of the methods will export the established secret keys to the outer tunneling method. It's expected that the outer method will cryptographically mix these keys into any keys it is currently using to protect the session and perform a final operation to determine whether both parties have arrived at the same mixed key. This is essential for detection of a number of nested method attacks (see 5.4.5 below for one such attack.)

6.4.4 Recommended Security Practices

In order to enable a strongly protected use of the TNC architecture for endpoint compliance, the following measures are necessary.

1. The NAA SHOULD authenticate the platform integrity of the trusted components on the TNC client to prevent falsification of integrity measurement reports about the current state of the platform. This would involve use of other TCG and TNC components (e.g. TPM and PTS) via IF-PTS.
2. The NAA and NAR SHOULD protect authentication tokens such as: private keys, trust anchor public keys/certificates, and traffic encryption keys from unauthorized access. Theft of cryptographic material can be catastrophic to the security of the system since the party could impersonate a party in the session. Countermeasures to this type of attack also may involve use of the platform's TPM or a secure key storage device.
3. To ensure that the endpoint checked with IF-T is the same one used for network access, either the tls-unique channel binding capability included in PT-EAP or suitable other protections against this attack SHOULD be employed.
4. When the use of PTS for verification of endpoint integrity is combined with user or platform authentication, the authentication SHOULD be bound to the verification, either by including the tls-unique channel binding in the TPM Quote or by authenticating with credentials tied to the TPM (like SKAE) and verifying that the credentials are associated with the same TPM as the one used for the PTS exchange. See Section 6.4.5 for a discussion of the attack and how this countermeasure operates.

6.4.5 Protecting against MiTM attacks against PT-EAP

The IF-T binding for tunneled EAP methods works on the premise that the tunneling method is capable of carrying (and protecting) various inner methods that perform additional security operations to establish the authenticity and integrity of the NAR. While this model is very flexible since it allows for the variety of existing EAP methods to be leveraged within the tunnel, it may introduce vulnerabilities. One such vulnerability is an attack described in "Man-in-the-Middle Attack against Tunneled Authentication Protocols" described in the 2003 Security Protocols Workshop paper by Asokan, Niemi, and Nyberg [9] and in the NEA Asokan Attack Analysis [22].

6.4.5.1 Example Attack Against EAP Nested Tunnels

This section describes a TNC oriented example of the Asokan, Niemi and Nyberg attack against an environment where Trusted Platforms are required to join the network. Trusted Platforms include an enabled TPM and a TBB measuring each

software component as it is loaded so the network can assess what software is running on the system and detect malware.

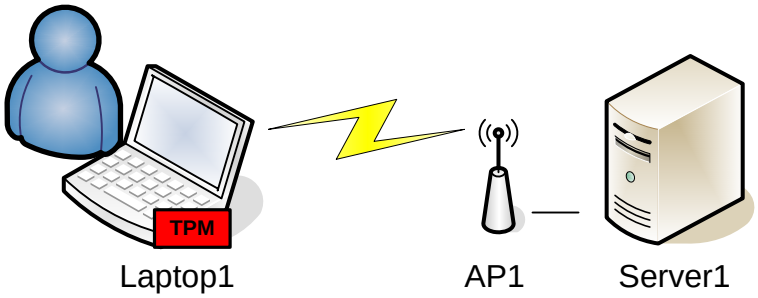


Figure 8. Clean Laptop Accessing Wireless Network

In this example, the NAR is called Laptop1 and the NAA is referred to as Server1 (see figure 8.) Laptop1 contains a Trusted Platform with an operating TPM and TBB and is wishing to gain access to the company intranet via an Access Point (AP1.) Server1 requires user authentication using PEAP, and verification of a PTS generated Integrity Report describing the running software on the laptop. Therefore, if Laptop1 gets compromised via a software malware attack (e.g. rootkit), it cannot get network access since the quoted PCR values in the Integrity Report will not match the expected values, and Server1 will not allow access.

Next, let's assume that Laptop1 does get compromised and can be controlled remotely by the attacker (maybe over the laptop's LTE card independent of the WLAN NIC used in this example.) So now sometime after the CRTM and early RTMs have performed their early platform measurements the attacker's malware is loaded and can communicate with the attacker. Now the attacker sets up his own equipment, Laptop2, AP2, and Server2 (see figure 9) on a stub network to aid his attack. Laptop2 is configured to match the "good" configuration that would be accepted by Server1 and even includes an enabled TPM and TBB. However, Server2 contains malware to aid the attacker obtain the Integrity Report from Laptop2.

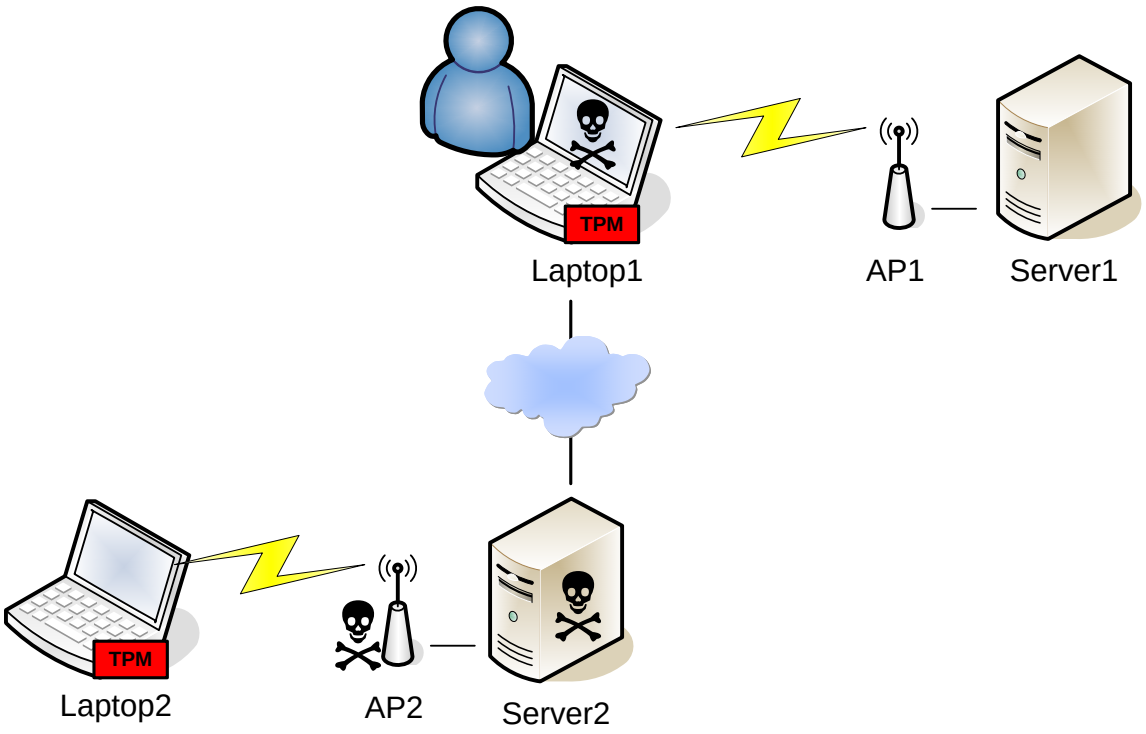


Figure 9. Attacker MiTM Network for Accessing Company Network

Next, the honest (unaware of the malware) user of Laptop1 tries to connect to the network, see figure 10 for flow diagram. Eventually, PEAP is started and the user is authenticated (steps 1-2). At the same time, the attacker uses Laptop2 and starts PEAP with his own server, and does user authentication on the stub network (steps 3-4).

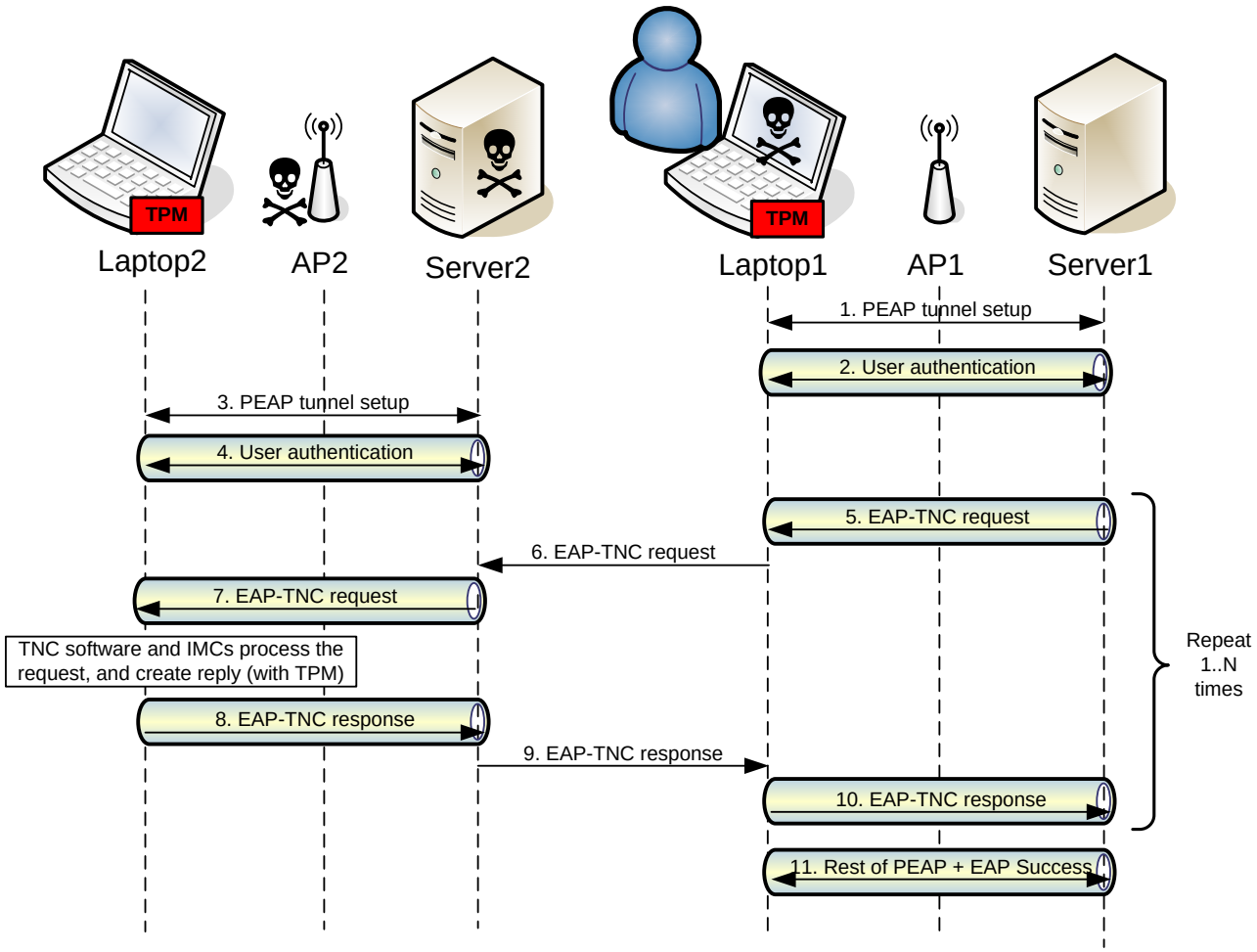


Figure 10. MiTM Attack Flows

Next, the TNC protocols start within the authenticated PEAP tunnel. At this point Server2 begins to interact with the compromised Laptop1. Server1 sends a PT-EAP request to Laptop1, which being compromised, forwards it to Server2 and eventually Laptop2. Laptop2 creates a response including an Integrity Report containing a TPM quote of the PCRs of Laptop2. The Integrity Report is sent to Server2 which forwards it to Laptop1 who presents it to Server1 as a description of the contents of Laptop1. Steps 5-10 can be repeated as often as is required by Server1.

Eventually the exchanges will succeed because Server1 is unable to tell that the Integrity Report it is receiving do not describe the same system that participated in the authenticated PEAP session, thus compromised Laptop1 gets access to the network despite the requirement for TPM rooted measurements.

6.4.5.2 Countermeasures using the Trusted Platform and TPM

Protection against this form of attack involves providing Server1 with a strong linkage between the party that performed the authentication and outer method with the party providing integrity information via the PT-EAP inner method. There are several ways this linkage might be established based upon leveraging the secret keys stored within the TPM of the valid (healthy) platform in such a way that the attacker is unable to successfully replay the responses to join the network. Version 1.0 of this specification defined an approach involving the use of certificates identified as containing a public key bound to a TPM resident private key. For backward compatibility with 1.0, the 2.0 version of this specification leaves this countermeasure which may be used and adds a second, preferred approach. Implementations of IF-T for Tunneled EAP Methods 2.0 SHOULD support the second countermeasure based on the tls-unique channel binding if they expect to be used on a system containing a TPM. The Diffie-Hellman pre-negotiation included in version 1.1 of this specification is not supported in version 2.0.

6.4.5.2.1 SKAE Certificates

The original countermeasure (present in version 1.0 of this specification and still permitted with this version of the specification) uses a non-migratable (or CMK) private key stored within a TPM and paired with a public key present in an X.509 certificate to perform the user or platform authentication of the outer tunneling method (e.g. using TLS). Because a single signed certificate can both identify the authenticated user (or platform) and bind that authentication to a particular TPM, this provides the strong linkage between the authenticating party and the TPM-based Integrity Report required to address this attack. The enrollment description and ASN.1 encoding for certifying that a private key is held within a TPM within an X.509 certificate is described in the SKAE [10] specification.

To understand how an SKAE is beneficial, we need to review how one is created during certificate enrollment. First the client system creates an AIK and obtains an AIK Credential from the Privacy CA. Next the client creates a “client identity authentication” key pair within the TPM and uses the TPM to “certify” that the key is only present within the TPM using the AIK. The TPM will produce evidence of this binding in a signed TPM_CERTIFY_INFO or TPM_CERTIFY_INFO2 structure. This structure is included in the proposed SKAE that is sent to the CA during enrollment for inclusion when creating the client identity authentication public key certificate. Now we have an X.509 identity certificate which contains signed evidence that the associated private key is housed in a TPM (thus a binding between a TPM and an identity.)

With SKAE, the verifier is expected to process the certificate as usual but also perform a validation of the SKAE’s evidence using the signing AIK public key. If the client can perform cryptographic operations using the identity private key, the verifier can be trust that it is the same platform with the described TPM and AIK. Similarly if the client system can perform cryptography using the AIK private key,

the verifier can trust that is the same user (or platform) as identified by the client identity certificate.

Assuming that Laptop1 or Laptop2 were created with an identity certificate leveraging a TPM resident private key, neither party would be able to sign information as the other party because they wouldn't have access to the private key. Clean laptop2 (created by the attacker) would likely have a difficult time obtaining an identity certificate in the first place for a legitimate user using TPM resident keys on Laptop2. The attacker would be unable to steal the private key for the identity certificate (with SKAE) on Laptop1 so would be unable to spoof that identity to Server1. Therefore if Laptop1 performed an authentication with this identity certificate to Server1 that understood SKAE (e.g. using a TLS tunneling EAP method) and later a quote came signed using an AIK from a different TPM, the verifier could detect this disparity.

6.4.5.2.2 tls-unique Channel Binding

PT-EAP introduces a new way of protecting against Asokan attacks when a TPM is present: the tls-unique channel binding [23]. This is a value unique to the TLS channel between the NAR and the NAA, derived from the TLS session key used in the tunneled EAP method and known only to the two endpoints of the TLS session.

When this protection technique is used, the PTS-IMC MUST obtain the tls-unique value from the TNCC and includes it in the quote command to the TPM. When the PTS-IMV receives the quote, it MUST compare the tls-unique value provided by the TNCC and signed by the TPM with the tls-unique value obtained from the TNCS. If these values match, either the TPM used for the quote is on the same platform that terminated the TLS session or security-critical software on the TPM's machine (e.g. NAR, TNCC, or IMC) is compromised. The TPM quote should reveal compromise of such security-critical software so a clean quote with the right tls-unique value indicates that the machine that terminated the TLS session is clean. The keying material exported by the tunneled EAP method will be known only to the clean machine and the NAA. This keying material MUST be used to derive cryptographic keys that ensure that only the clean machine can gain access to the protected network.

If an active MiTM (Laptop1) terminates the PT-EAP method, it will need to terminate the tunneled EAP method as well and therefore establish a tls-unique value shared with Server1 whereas Laptop2 will have a different tls-unique value shared with Server2 and use that value in Laptop2's attested integrity report. Laptop1 cannot use its tls-unique value to obtain a quote from the TPM on Laptop1 because that would reveal malware on Laptop1. Laptop1 could try to change the tls-unique value in the Integrity Report coming from Laptop2 but this would invalidate the signature that Laptop2's TPM applied to the Integrity Report.

If a MiTM (Laptop1) just forwarded the tunnel EAP method protocol to Server2 so that clean Laptop2 and Server1 were generating the tls-unique value, Laptop1 would be unable to determine the keying material exported by the tunnel EAP method.

In order for the MiTM protection to continue during the subsequent communications on the network, the communications SHOULD protect the data exchanges using keys based on the final tunneled EAP method keys. Early versions of 802.1X only used these keys for wireless networks, leaving wired 802.1X networks unprotected. However 802.1X-2010 [7] resolves this issue by enabling the use of MACsec to block access from unauthenticated devices on a wired network.

Although a MiTM attack against PT-EAP could be employed against systems that do not include a TPM, there would be no point in mounting such a sophisticated attack. A compromised endpoint could simply send false measurements (Laptop1 could include an IMC that just sends information it knows would comply with policy even if that information didn't reflect the true state of the system.) Similarly if Laptop1 was not using its TBB to measure boot, the attacker might be able to falsify measurements in the TPM without the need for the stub network.

7 References

The references are divided as to normative and non-normative. Normative references are those that are required to implement IF-T Protocol Bindings for Tunneled EAP Methods. Non-normative references are helpful in understanding use cases covered in this specification, but are not required to implement IF-T protocol bindings for tunneled EAP methods.

7.1 Normative References

[1] Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels”, Internet Engineering Task Force RFC 2119, March 1997

[2] Cam-Winget, N. and P. Sangster, “PT-EAP: Posture Transport (PT) Protocol For Extensible Authentication Protocol (EAP) Tunnel Methods”, RFC 7171, May 2014

7.2 Non-Normative References

[3] Trusted Computing Group, *TNC Architecture for Interoperability*, Specification Version 1.5, May 2012, http://www.trustedcomputinggroup.org/resources/tnc_architecture_for_interoperability_specification

[4] Trusted Computing Group, *TNC IF-TNCCS*, Specification Version 2.0, May 2014, http://www.trustedcomputinggroup.org/resources/tnc_iftnccs_specification

[5] Trusted Computing Group, *TNC IF-M*, Specification Version 1.0, May 2014, http://www.trustedcomputinggroup.org/resources/tnc_ifm_tlv_binding_specification

[6] Trusted Computing Group, *TNC IF-T Binding to TLS*, Specification Version 2.0, February 2013, http://www.trustedcomputinggroup.org/resources/tnc_ift_binding_to_tls

[7] LAN/MAN Standards Committee of the IEEE Computer Society, Standard for Local and Metropolitan Area Networks – Port Based Network Access Control, IEEE Std. 802.1X-2010, February 2010

[8] J. Vollbrecht, P. Eronen, N. Petroni, Y. Ohba, “State Machines for Extensible Authentication Protocol (EAP) Peer and Authenticator”, Internet Engineering Task Force RFC 4137, August, 2005

[9] N. Asokan, Valtteri Niemi, Kaisa Nyberg, “Man in the Middle Attacks in Tunneled Authentication Protocols”, Nokia Research Center, Finland, Nov. 11, 2002, http://eprint.iacr.org/2002/163.pdf

[10] Trusted Computing Group, *Subject Key Attestation Evidence Extension,* Specification version 1, revision 7, https://www.trustedcomputinggroup.org/specs/IWG/IWG_SKAE_Extension_1-00.pdf June 16, 2005.

[11] Greg Kazmierczak, Ned Smith, "TCG IWG Platform Trust Services Interface Specification (IF-PTS)", https://www.trustedcomputinggroup.org/resources/infrastructure_work_group_platform_trust_services_interface_specification_version_10, November, 2006.

[12] TCG Infrastructure WG, Miscellaneous schema documents, https://www.trustedcomputinggroup.org/developers/infrastructure, November 2012.

[13] Cam-Winget, N., McGrew, D., Salowey, J., and H. Zhou, "The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)" , IETF RFC 4851, May 2007

[14] C. Kaufman, P. Hoffman, Y. Nir, P. Eronen "Internet Key Exchange Protocol Version 2 (IKEv2)," Internet Engineering Task Force RFC 5996, September 2010

[15] Funk, P. and S. Blake-Wilson, "Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)", IETF RFC 5281, August 2008

[16] H. Andersson, S. Josefsson, Glen Zorn, Dan Simon, Ashwin Palekar IETF draft-josefsson-pppext-eap-tls-eap-05.txt Protected EAP Protocol (PEAP), September, 2002

[17] Ashwin Palekar, Dan Simon, Joe Salowey, Hao Zhou, Glen Zorn, S. Josefsson draft-josefsson-pppext-eap-tls-eap-10 Protected EAP Protocol (PEAP) Version 2, October 2004

[18] Vivek Kamath, Ashwin Palekar, Mark Wodrich draft-kamath-pppext-peapv0-00.txt Microsoft's PEAP version 0 (Implementation in Windows XP SP1, October, 2002

[19] Sangster, P. and K. Narayan, "PA-TNC: A Posture Attribute Protocol (PA) Compatible with TNC", IETF RFC 5792, March 2010

[20] Sahita, R., Hanna, S., Hurst, R., and K. Narayan, "PB-TNC: A Posture Broker Protocol (PB) Compatible with TNC", IETF RFC 5793, March 2010

[21] Zhou, H., Cam-Winget, N., Salowey, J., and S. Hanna, "Tunnel Extensible Authentication Protocol (TEAP) Version 1", RFC 7170, May 2014

[22] Salowey, J., and S. Hanna, "NEA Asokan Attack Analysis", IETF RFC 6813, December 2012

[23] Altman, J., Williams, N., and L. Zhu, "Channel Bindings for TLS", IETF RFC 5929, July 2010

[24] B. Aboba, L. Blunk, J. Vollbrecht, J. Carlson, H. Levkowetz, Ed., “Extensible Authentication Protocol (EAP)”, Internet Engineering Task Force RFC 3748, June 2004