Network Working Group
Internet-Draft
Intended status: Experimental
Expires: January 28, 2010

C. Latze
U. Ultes-Nitsche
University of Fribourg
F. Baumgartner
Swisscom Schweiz AG
July 27, 2009

Extensible Authentication Protocol Method for Trusted Computing Groups (TCG) Trusted Platform Modules
draft-latze-emu-eap-tpm-01

Status of this Memo

Copyright Notice

Abstract

This document describes an Extensible Authentication Protocol (EAP) [RFC3748] method for identity distribution, authentication and session key distribution using the Trusted Computing Group's (TCG) Trusted Platform Module (TPM). The TPM has been defined by the TCG in order to establish a root of trust and measurements in (consumer) computers. It provides several cryptographic functions and a secure storage for keys and hashes. There is also a TPM specification for mobile devices called Mobile Trusted Module (MTM), which can also be used for EAP-TPM. This new EAP method allows network authentication, which also supports user anonymity, the usage of different user identities for the authentication with different network operators, result indication, and a fast re-authentication.

Table of Contents

1. Introduction

This document specifies a new Extensible Authentication Protocol (EAP) [RFC3748] method based on Trusted Platform Modules (TPMs). TPMs are hardware chips attached to the motherboard of the majority of newly shipped computers. They provide small secure storage, cryptographic functions and a root of trust and measurement. In addition to TPMs there are also Mobile Trusted Modules (MTMs) that provide a subset of the TPMs functionality and are meant to be built into mobile handsets like mobile phones.

TPMs/MTMs can be identified uniquely all over the world and may obtain an identity certificate that proofs that it comes from a genuine TPM/MTM. Therefore, TPMs/MTMs are perfectly suited for certificate based authentication schemes.

EAP-TPM provides support for different user identities which allows the user to hide its original identity at the authenticator as requested in [RFC4017]. Furthermore, it provides a zero-configuration mode where the user does not need to request any identity before authenticating to an EAP-TPM secured authenticator.

EAP-TPM should be understood as a more comfortable but not less secure EAP-TLS.

2. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.

3. Terms and Abbreviations

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 [RFC2119].

Furthermore, this document uses the following terms and abbreviations:

Authenticator

The EAP authentication server, that does the real authentication of the peer.

Client

EAP-Peer - the one who wants to be authenticated.

MTM

Mobile Trusted Module as specified in [MTMSpec]

TPM

Trusted Platform Module as specified in [TCGMainSpec]

User

EAP-Peer - the one who wants to be authenticated.

4. Motivation

EAP-TPM has the goal to make a secure authentication protocol like EAP-TLS more userfriendly without weakening its security. EAP-TLS [RFC5216] as it is provides mutual authentication but requires the client to request its own valid X.509 certificates. Two problems arise with that approach: First of all, the user has to be capable to request a certificate, which is a non-trivial task; second the user has to know the acceptable Certificate Authorities (CAs) in advance which is a strong constraint for a real world setup. Therefore, the motivation for EAP-TPM was to develop an EAP method that is as secure, scalable, and automatable as EAP-TLS and comfortable in its usage for a naive "normal" user.

5. Client Certificates

The process of retrieving TPM certificates starts with retrieving an identity certificate as specified in the TCG Main Specification [TCGMainSpec]:

1. The TPM has to create a new identity certificate request using TPM_MakeIdentity, which generates a new identity key that has to be signed by a Privacy CA.

2. The Trusted Subsystem (TSS) has to collect all the information needed by the Privacy CA to certify the formerly created identity key. According to the TPM Main Specification, that function is called TSS_CollateIdentityRequest.

3. This request will be sent to the Privacy CA, which verifies all the data and certifies the key and replies with an identity certificate.

4. The TPM will now activate the new identity using TPM_ActivateIdentity.

5. Finally, the TSS has to retrieve a plain text copy of the new identity certificate using TSS_RecoverTPMIdentity.

The detailed process is described in [TCGMainSpec] and not part of this document.

According to [TCGMainSpec] those certificates are special purpose certificate, restricted to SHA-1 signing and MUST have the CA:false constraint. As TLS only uses standards signature from version 1.2 on [RFC5246], EAP-TPM MUST be used with TLS 1.2.

6. Authentication

Authentication in EAP-TPM can be divided into authentication without zero-configuration, where the user MUST request his certificate(s) before connecting to an EAP-TPM authenticator, and authentication with zero-configuration, where the user will get a certificate during the authentication process. The first scenario described in Section 6.1 is more suitable for an operator controlled setup with accounting as it allows to register certificates with users, whereas the second scenario described in Section 6.2 is more suitable for corporate environments or environment without accounting in general.

6.1. Authentication without Zero-Configuration

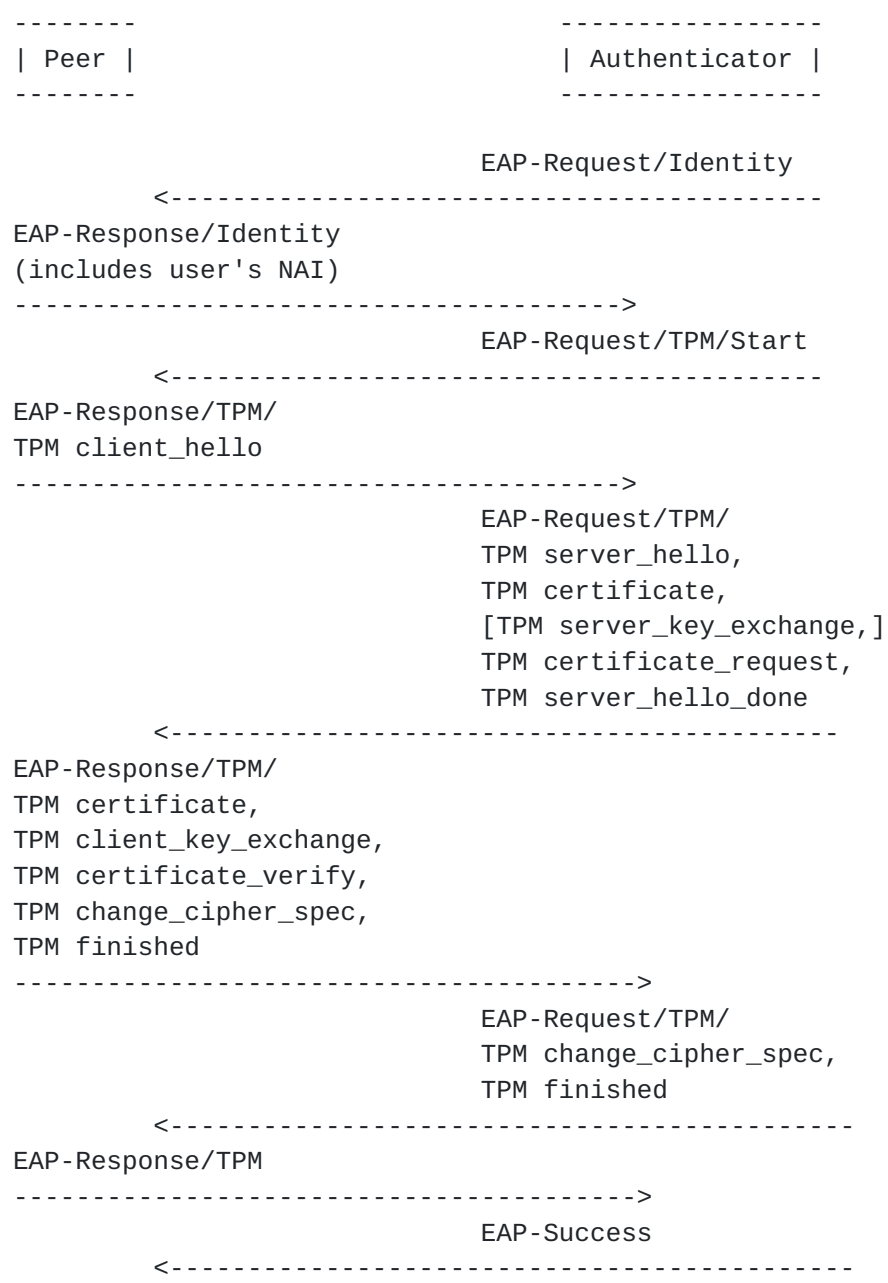


Figure 1: Successful EAP-TPM Authentication With Existing Certificates

Figure 1 shows the full authentication using EAP-TPM in case the peer could be authenticated successfully. The authentication with already existing certificates is very similar to EAP-TLS [RFC5216]:

The authentication will start with the authenticator asking the peer for its identity sending EAP-Request/Identity. The peer will answer with EAP-Response/Identity containing his Network Address Identifier (NAI) [RFC4282]. Afterwards the authenticator sends the EAP-Request/

TPM/Start message to indicate the real beginning of EAP-TPM, which will be followed by a normal TLS handshake.

Finally the EAP authentication is closed with EAP-Response/TPM sent by the client and EAP-Success sent by the authenticator.

6.2. Authentication with Zero-Configuration

Section 6.1 requires valid certificates on the client before starting the authentication. This section deals with authentication without existing certificates. The peer tries to authenticate to an authenticator starting as described in Section 6.1. During the authentication, the authenticator has to send a TPM certificate_request message in order to request the peer's certificate. TLS [RFC5246] allows to include acceptable certificate authority in this certificate_request message:

```
struct{
    CertificateType certificate_types<1..2^8-1>;
    DistinguishedName certificate_authorities<3..2^16-1>;
}certificate_request;
```

In EAP-TPM with zero configuration, the authenticator has to specify acceptable Privacy CAs (PCAs) within the certificate_authorities field in the certificate_request message. After receiving the EAP-Request/TPM, TPM server_hello, TPM certificate, [TPM server_key_exchange,] TPM certificate_request, TPM server_hello_done messages, the peer has to check whether it has a valid certificate from one of the PCAs specified in TPM certificate_request->certificate_authorities or not. In case it has such a certificate, the authentication goes on as shown in figure Figure 1. In case the peer does not possess a valid certificate from one of the acceptable PCAs, it has to answer with EAP-Response/TPM/ TPM no_such_certificate, TPM need_certificate, where no_such_certificate is an alert with level warning(1) and description no_certificate(41) [RFC5246]:

```
struct {
   AlertLevel level;
   AlertDescription description;
}no_such_certificate;
```

This message will be followed by TPM need_certificate, which specifies the PCA the peer wants to ask for a certificate:

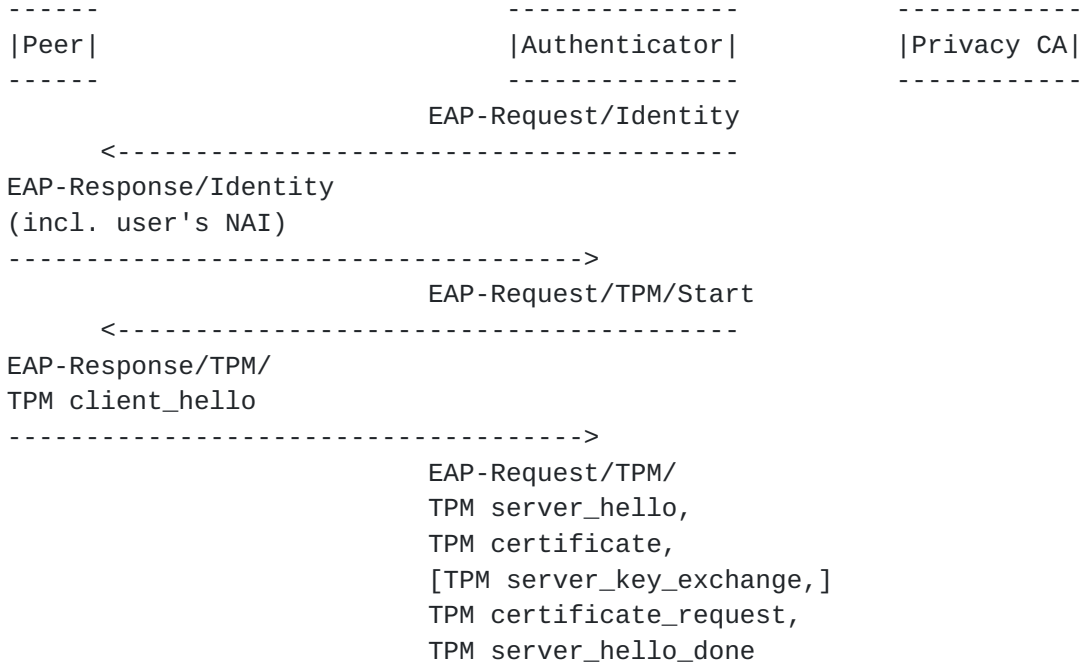
```
struct {
   opaque privacy_ca<1..2^16-1>;
}need_certificate;
```

The PCA specified in the need_certificate message MUST be one of the PCAs proposed in TPM certificate_request. The authenticator will now ask the peer for its certificate request message and request the certificate at the desired PCA on behalf of the peer. He starts the certificate request process with an EAP-Request/TPM/TPM request_certificate:

```
struct {
   ConnectionAllowed allowed;
   opaque privacy_ca<1..2^16-1>;
}request_certificate

where
enum {
   true(1), false(2)
}ConnectionAllowed
```

The peer MUST check the privacy_ca value. If it does not match the PCA specified in need_certificate, the peer MUST close the authentication immediately. If the privacy_ca value matches and allowed is set to true(1), the peer is allowed to request a new certificate as shown in figure Figure 2. It sends its certificate_request described in Section 5 inside EAP-Response/TPM/ TPM certificate_request to the authenticator, which will then request the peer's certificate at the PCA. The PCA sends the peer's certificate back to the authenticator, who will forward the certificate to the peer in the EAP-Request/TPM/TPM client_certificate message. Afterwards, the authentication goes on as shown in figure Figure 1.



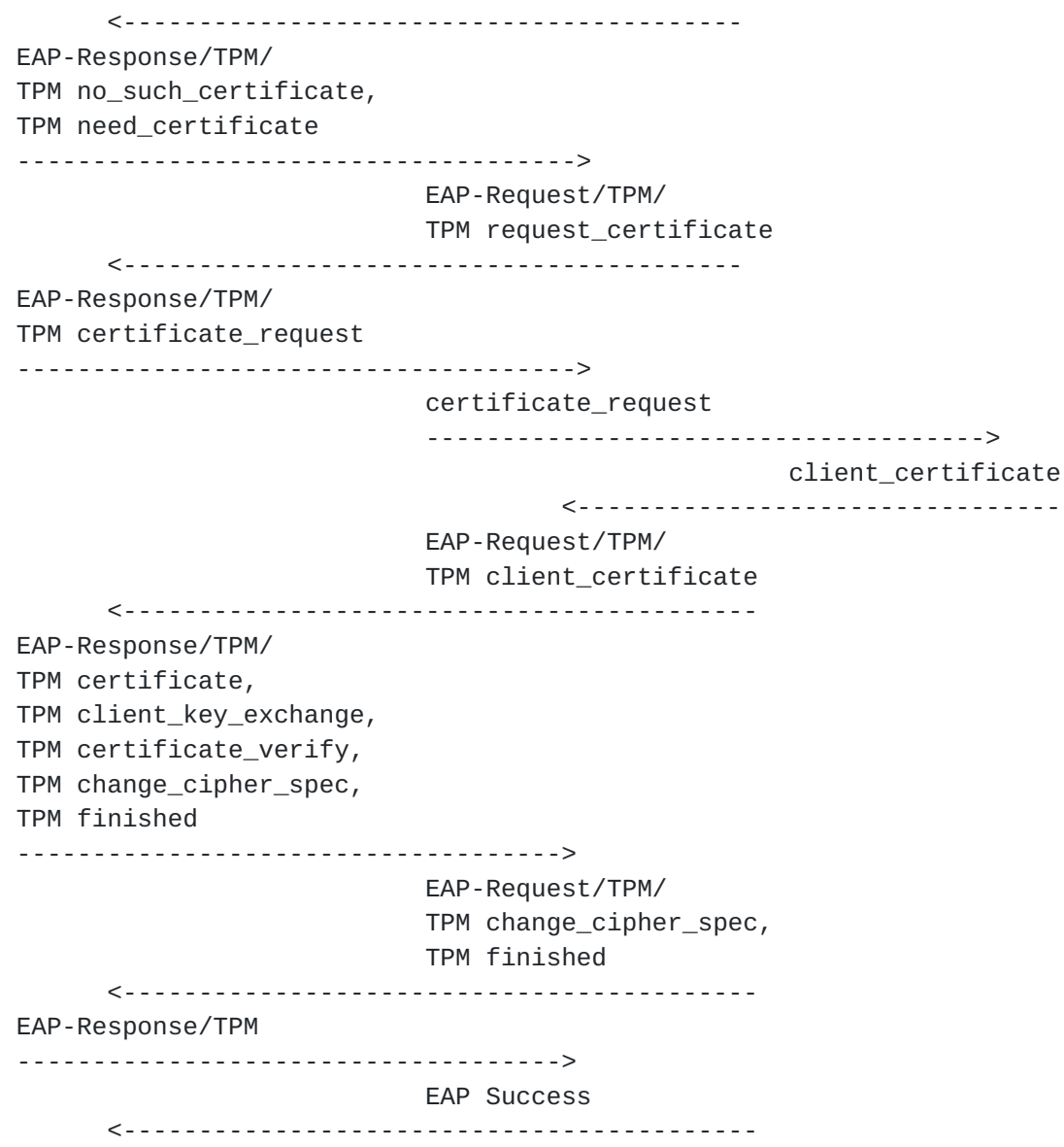


Figure 2: Successful EAP-TPM Authentication With Zero Configuration

7. Failure, Fragmentation and Fast Re-Authentication

Failed authentication requests, fragmentation and fast re-authentication are handled in exactly the same way as in EAP-TLS [RFC5216].

8. Key Derivation

Key derivation in EAP-TPM occurs similar to key derivation in EAP-TLS [RFC5216]. The encryption keys are calculated using a pseudo random

function (PRF) that takes the master secret obtained during the TLS handshake and a random number which is the concatenation out of the random value in client_hello and server_hello as argument. The initialisation vector (IV) which may be used for symmetric encryption will be calculated out of a PRF using an empty string and the random number mentioned above as argument.

9. Security Considerations

EAP-TPM as described in this document fulfills the mandatory and recommended requirements for wireless LANs specified in [RFC4017]. The mandatory criteria "Generation of symmetric keying material", "Key strength", "Shared state equivalence", "Resistance to dictionary attacks", "Protection against man-in-the-middle attacks" and "Protected cipher suite negotiation" are fulfilled like in EAP-TLS since there is no difference to EAP-TLS regarding those points. The "Mutual authentication support" is also fulfilled by definition since EAP-TPM is only made for mutual authentication. The recommended requirements of [RFC4017] are also fulfilled: "Fragmentation" is provided as in EAP-TLS and "End-user identity hiding" is provided by the fact that the user may use different identities for every authentication.

Furthermore, the request of an identity certificate MUST be acknowledged by the user in order to ensure that he is informed about the identities of his device.

As the certificate request mentioned Section 6.2 has to be relayed over the authentication server, it might be possible for a malicious server to tamper with that request. But as [TCGMainSpec] specifies some security measures for that certificate request (encrypted with the TPM key), the authentication server will not be able to modify the request without the peer or privacy CA noticing it.

10. Acknowledgements

The authors want to thank Bernhard Hoeneisen and Alan DeKok for their discussions regarding administrative matters and for the comments and Joe Salowey for suggestions about how to improve the specification.

11. Normative References

[MTMSpec] TCG, "TCG Mobile Trusted Module Specification Version 1.0", June 2008, <https://www.trustedcomputinggroup.org/specs/mobilephone/

Revision_6-tcg-mobile-trusted-module-1_0.pdf>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3748] Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowetz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.

[RFC4017] Stanley, D., Walker, J., and B. Aboba, "Extensible Authentication Protocol (EAP) Method Requirements for Wireless LANs", RFC 4017, March 2005.

[RFC4282] Aboba, B., Beadles, M., Arkko, J., and P. Eronen, "The Network Access Identifier", RFC 4282, December 2005.

[RFC5216] Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.

[RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", RFC 5246, August 2008.

[TCGMainSpec] TCG, "TCPA Main Specification Version 1.1b", February 2002, <https://www.trustedcomputinggroup.org/specs/TPM/TCPA_Main_TCG_Architecture_v1_1b.pdf>.

Authors' Addresses

Carolin Latze
University of Fribourg
Boulevard de Perolles 90
Fribourg, FR 1700
Switzerland

Email: carolin.latze@unifr.ch

Ulrich Ultes-Nitsche
University of Fribourg
Boulevard de Perolles 90
Fribourg, FR 1700
Switzerland

Email: uun@unifr.ch

Florian Baumgartner
Swisscom Schweiz AG
Ostermundigenstrasse 93
Bern, BE 3006
Switzerland

Email: florian.baumgartner@swisscom.com