Network Working Group	E. Rescorla	
Internet-Draft	Network Resonance	
Intended status: Standards Track	September 11, 2008	
Expires: March 15, 2009		

Keying Material Extractors for Transport Layer Security (TLS)
draft-ietf-tls-extractor-02.txt

Status of this Memo

Abstract

A number of protocols wish to leverage Transport Layer Security (TLS) to perform key establishment but then use some of the keying material for their own purposes. This document describes a general mechanism for allowing that.

Table of Contents

---

1. Introduction

TOC

A number of protocols wish to leverage Transport Layer Security (TLS) [RFC4346] (Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," April 2006.) or Datagram TLS (DTLS) [RFC4347] (Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security," April 2006.) to perform key establishment but then use some of the keying material for their own purposes. A typical example is DTLS-SRTP [I-D.ietf-avt-dtls-srtp] (McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)," February 2009.), which uses DTLS to perform a key exchange and negotiate the SRTP [RFC3711] (Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," March 2004.) protection suite and then uses the DTLS master_secret to generate the SRTP keys.
These applications imply a need to be able to extract keying material (later called Exported Keying Material or EKM) from TLS/DTLS, and securely agree on the upper-layer context where the keying material will be used. The mechanism for extracting the keying material has the following requirements:

- Both client and server need to be able to extract the same EKM value.
- EKM values should be indistinguishable from random by attackers who don't know the master_secret.
- It should be possible to extract multiple EKM values from the same TLS/DTLS association.
- Knowing one EKM value should not reveal any information about the master_secret or about other EKM values.

The mechanism described in this document is intended to fill these requirements.

---

2. Conventions Used In This Document

TOC

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 [RFC2119] (Bradner, S., “Key words for use in RFCs to Indicate Requirement Levels,” March 1997.).

---

3. Binding to Application Contexts

TOC

In addition to extracting keying material, an application using the keying material has to securely establish the upper-layer layer context where the keying material will be used. The details of this context depend on the application, but it could include things such as algorithms and parameters that will be used with the keys, identifier(s) for the endpoint(s) who will use the keys, identifier(s) for the session(s) where the keys will be used, and the lifetime(s) for the context and/or keys. At minimum, there should be some mechanism for signalling that an extractor will be used.
This specification does not mandate a single mechanism for agreeing on such context; instead, there are several possibilities that can be used (and can complement each other). For example:

- One important part of the context -- which application will use the extracted keys -- is given by the disambiguating label string (see Section 4).

- Information about the upper-layer context can be included in the optional data after the extractor label (see Section 4).

- Information about the upper-layer context can be exchanged in TLS extensions included in the ClientHello and ServerHello messages. This approach is used in [DTLS-SRTP]. The handshake messages are protected by the Finished messages, so once the handshake completes, the peers will have the same view of the information. Extensions also allow a limited form of negotiation: for example, the TLS client could propose several alternatives for some context parameters, and TLS server could select one of them.

- The upper-layer protocol can include its own handshake which can be protected using the keys extracted from TLS.

It is important to note that just embedding TLS messages in the upper-layer protocol may not automatically secure all the important context information, since the upper-layer messages are not covered by TLS Finished messages.

4. Extractor Definition

TOC

An extractor takes as input three values:

- *A disambiguating label string
- *A per-association context value provided by the extractor using application
- *A length value

It then computes:

```
PRF(master_secret, label,
    SecurityParameters.client_random +
    SecurityParameters.server_random +
    context_value_length + context_value
    )[length]
```

The output is a pseudorandom bit string of length bytes generated from the master_secret.
Label values beginning with "EXPERIMENTAL" MAY be used for private use without registration. All other label values MUST be registered via Specification Required as described by RFC 2434 [RFC2434] (Narten, T. and H. Alvestrand, “Guidelines for Writing an IANA Considerations Section in RFCs,” October 1998.). Note that extractor labels have the potential to collide with existing PRF labels. In order to prevent this, labels SHOULD begin with "EXTRACTOR". This is not a MUST because there are existing uses which have labels which do not begin with this prefix.
The context value allows the application using the extractor to mix its own data with the TLS PRF for the extractor output. The context value length is encoded as an unsigned 16-bit quantity (uint16) representing the length of the context value.

5. Security Considerations

TOC

Because an extractor produces the same value if applied twice with the same label to the same master_secret, it is critical that two EKM values generated with the same label be used for two different purposes--hence the requirement for IANA registration. However, because extractors depend on the TLS PRF, it is not a threat to the use of an

EKM value generated from one label to reveal an EKM value generated from another label.

6. IANA Considerations

TOC

IANA is requested to create (has created) a TLS Extractor Label registry for this purpose. The initial contents of the registry are given below:

```
Value                          Reference
-----                          ------------
client finished                [RFC4346]
server finished                [RFC4346]
master secret                  [RFC4346]
key expansion                  [RFC4346]
client EAP encryption          [RFC2716]
ttls keying material           [draft-funk-eap-ttls-v0-01]
```

Future values are allocated via RFC2434 Specification Required policy. The label is a string consisting of printable ASCII characters. IANA MUST also verify that one label is not a prefix of any other label. For example, labels "key" or "master secretary" are forbidden.

7. Acknowledgments

TOC

Thanks to Pasi Eronen for valuable comments and the contents of the IANA section and Section 3 (Binding to Application Contexts).

8. References

TOC

8.1. Normative References

TOC

[RFC2119]	Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels," BCP 14, RFC 2119, March 1997 (TXT, HTML, XML).
[RFC2434]	Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs," BCP 26, RFC 2434, October 1998 (TXT, HTML, XML).
[RFC4346]	

	Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1," RFC 4346, April 2006 (TXT).

8.2. Informational References

TOC

[RFC4347]	Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security," RFC 4347, April 2006 (TXT).
[RFC3711]	Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)," RFC 3711, March 2004 (TXT).
[I-D.ietf-avt-dtls-srtp]	McGrew, D. and E. Rescorla, "Datagram Transport Layer Security (DTLS) Extension to Establish Keys for Secure Real-time Transport Protocol (SRTP)," draft-ietf-avt-dtls-srtp-07 (work in progress), February 2009 (TXT).

Author's Address

TOC

	Eric Rescorla
	Network Resonance
	2064 Edgewood Drive
	Palo Alto, CA 94303
	USA
Email:	ekr@networkresonance.com

Full Copyright Statement

TOC

Intellectual Property