Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: September 1, 2009

E. Rescorla
RTFM, Inc.

February 28, 2009

Keying Material Exporters for Transport Layer Security (TLS) draft-ietf-tls-extractor-04.txt

Status of this Memo

Copyright Notice

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

Note: The mechanism described in this document was previously known as "TLS Extractors" but was changed to avoid a name conflict with the use of the term "Extractor" in the cryptographic community.

A number of protocols wish to leverage Transport Layer Security (TLS) [RFC5246] or Datagram TLS (DTLS) [RFC4347] 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], which uses DTLS to perform a key exchange and negotiate the SRTP [RFC3711] protection suite and then uses the DTLS master_secret to generate the SRTP keys.

These applications imply a need to be able to export 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 exporting the keying material has the following requirements:

- Both client and server need to be able to export 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 export 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

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].

3. Binding to Application Contexts

In addition to exporting 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 exporter 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 exported 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 exporter 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 exported 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. Exporter Definition

An exporter takes as input three values:

- A disambiguating label string
- A per-association context value provided by the exporter 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]. Note that exporter labels have the potential to collide with existing PRF labels. In order to prevent this, labels SHOULD begin with "EXPORTER". 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 exporter to mix its own data with the TLS PRF for the exporter output. The context value length is encoded as an unsigned 16-bit quantity (uint16) representing the length of the context value.

5. Security Considerations

Because an exporter 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 exporters 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

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

Value	Reference
client finished	[RFC5246]
server finished	[RFC5246]
master secret	[RFC5246]
key expansion	[RFC5246]
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

Thanks to Pasi Eronen for valuable comments and the contents of the IANA section and Section 3.

8. References

8.1. Normative References

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

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434,

October 1998.

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

8.2. Informational References

[RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", RFC 4347, April 2006.

[RFC3711] Baugher, M., McGrew, D., Naslund, M., Carrara, E., and K. Norrman, "The Secure Real-time Transport Protocol (SRTP)", RFC 3711, March 2004.

[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-06 (work in progress), October 2008.

Author's Address

Eric Rescorla
RTFM, Inc.
2064 Edgewood Drive
Palo Alto, CA 94303
USA

Email: ekr@rtfm.com