

Network Working Group
Internet-Draft
Expires: June 16, 2007

E. Rescorla
Network Resonance
M. Salter
National Security Agency
December 13, 2006

Opaque PRF Inputs for TLS
draft-rescorla-tls-opaque-prf-input-00.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/lid-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 June 16, 2007.

Copyright Notice

Copyright (C) The Internet Society (2006).

Abstract

This document describes a mechanism for using opaque PRF inputs with Transport Layer Security (TLS) and Datagram TLS (DTLS).

Table of Contents

1.	Introduction	3
2.	Conventions Used In This Document	3
3.	The OpaquePRFInput Extension	3
3.1.	Negotiating the OpaquePRFInput Extension	4
3.2.	PRF Modifications	4
4.	Security Considerations	5
4.1.	Threats to TLS	5
4.2.	New Security Issues	5
4.3.	Scope of Randomness	5
5.	IANA Considerations	5
6.	Acknowledgements	6
7.	References	6
7.1.	Normative References	6
7.2.	Informative References	6
	Authors' Addresses	7
	Intellectual Property and Copyright Statements	8

1. Introduction

TLS [[RFC4346](#)] and DTLS [[RFC4347](#)] use a 32-byte "Random" value consisting of a 32-bit time value time and 28 randomly generated bytes:

```
struct {  
    uint32 gmt_unix_time;  
    opaque random_bytes[28];  
} Random;
```

The client and server each contribute a Random value which is then mixed with secret keying material to produce the final per-association keying material.

In a number of United States Government applications, it is desirable to have some material with the following properties:

1. It is contributed both by client and server.
2. It is arbitrary-length.
3. It is mixed into the eventual keying material.
4. It is structured and decodable by the receiving party.

These requirements are incompatible with the current Random mechanism, which supports a short, fixed-length value. This document describes a mechanism called "Opaque PRF Inputs for TLS" that meets 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.](#) The OpaquePRFInput Extension

The OpaquePRFInput is carried in a new TLS extension called "OpaquePRFInput".

```
struct {  
    opaque opaque_prf_input_value<0..2^16-1>;  
} OpaquePRFInput;
```

The opaque_prf_input_value is an opaque byte-string which is generated in an implementation-dependent fashion. It MAY be generated by and/or made available to the TLS/DTLS-using application.

[3.1.](#) Negotiating the OpaquePRFInput Extension

The client requests support for the opaque PRF input feature by sending an "opaque_prf_input" extension in its ClientHello. The "extension_data" field contains an OpaquePRFInput value.

When a server which does not recognize the "opaque_prf_input" extension receives one, it will ignore it as required by [\[RFC4366\]](#). A server which recognizes the extension MAY choose to ignore it, in which case it SHOULD continue with the exchange as if it had not received the extension.

If the server wishes to use the opaque PRF input feature, it MUST send its own "opaque_prf_input" extension with an opaque_prf_input_value equal in length to the client's opaque_prf_input_value. Clients SHOULD check the length of the server's opaque_prf_input_value and generate a fatal "illegal_parameter" error if it is present but does not match the length that was transmitted in the ClientHello.

Because [RFC 4366](#) does not permit servers to request extensions which the client did not offer, the client may not offer the "opaque_prf_input" extension even if the server requires it. In this case, the server should generate a fatal "handshake_failure" alert.

Because there is no way to mark extensions as critical, the server may ignore the "opaque_prf_input" extension even though the client requires it. If a client requires the opaque PRF input feature but the server does not negotiate it, the client SHOULD generate a fatal

"handshake_failure" alert.

[3.2.](#) PRF Modifications

When the opaque PRF input feature is in use, the opaque PRF input values **MUST** be mixed into the PRF along with the client and server random values during the PMS->MS conversion. Thus, the PRF becomes:

```
master_secret = PRF(pre_master_secret, "master secret",
                    ClientHello.random +
                    ClientHello.opaque_prf_input_value +
                    ServerHello.random +
                    ServerHello.opaque_prf_input_value)[0..47];
```

Because new extensions may not be introduced in resumed handshakes, mixing in the opaque PRF inputs during the MS->keying material conversion would simply involve mixing in the same material twice. Therefore, the opaque PRF inputs are only used when the PMS is converted into the MS.

[4.](#) Security Considerations

[4.1.](#) Threats to TLS

When this extension is in use it increases the amount of data that an attacker can inject into the PRF. This potentially would allow an attacker who had partially compromised the PRF greater scope for influencing the output. Hash-based PRFs like the one in TLS are designed to be fairly indifferent to the input size (the input is already greater than the block size of most hash functions), however there is currently no proof that a larger input space would not make attacks easier.

Another concern is that bad implementations might generate low entropy opaque PRF input values. TLS is designed to function correctly even when fed low-entropy random values because they are primarily used to generate distinct keying material for each connection.

[4.2.](#) New Security Issues

As noted in [Section 3](#) it is anticipated that applications may want to

have access to the opaque PRF input values and that they may contain data that is meaningful at a higher layer. Because the values are covered by the TLS Finished message, they are integrity-protected by TLS. However, the application must independently provide any confidentiality necessary for those values.

[4.3.](#) Scope of Randomness

[RFC 4366](#) specifies that when a session is resumed the extensions from the original connection are used:

If, on the other hand, the older session is resumed, then the server **MUST** ignore the extensions and send a server hello containing none of the extension types. In this case, the functionality of these extensions negotiated during the original session initiation is applied to the resumed session.

This motivates why the the opaque PRF input does not get mixed into the PRF when generating the keying material from the master secret. Because the same opaque PRF inputs would be used for every connection in a session, they would not provide any differentiation in the keying material between the connections.

[5.](#) IANA Considerations

Rescorla & Salter

Expires June 16, 2007

[Page 5]

Internet-Draft

TLS Opaque PRF Inputs

December 2006

This document defines an extension to TLS, in accordance with [\[RFC4366\]](#):

```
enum { opaque_prf_input (??) } ExtensionType;
```

[[NOTE: These values need to be assigned by IANA]]

[6.](#) Acknowledgements

This work was supported by the US Department of Defense.

[7.](#) References

[7.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC4366] Blake-Wilson, S., Nystrom, M., Hopwood, D., Mikkelsen, J., and T. Wright, "Transport Layer Security (TLS) Extensions", [RFC 4366](#), April 2006.
- [RFC4346] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.1", [RFC 4346](#), April 2006.
- [RFC4347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security", [RFC 4347](#), April 2006.
- [I-D.ietf-tls-rfc4346-bis]
Dierks, T. and E. Rescorla, "The TLS Protocol Version 1.2", [draft-ietf-tls-rfc4346-bis-02](#) (work in progress), October 2006.

[7.2.](#) Informative References

Rescorla & Salter	Expires June 16, 2007	[Page 6]
-------------------	-----------------------	----------

Internet-Draft	TLS Opaque PRF Inputs	December 2006
----------------	-----------------------	---------------

Authors' Addresses

Eric Rescorla
Network Resonance
2483 E. Bayshore #212
Palo Alto, CA 94303
USA

Email: ekr@networkresonance.com

Margaret Salter
National Security Agency
9800 Savage Rd.
Fort Meade 20755-6709
USA

Email: msalter@restarea.ncsc.mil

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.

Disclaimer of Validity

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 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.

Copyright Statement

Copyright (C) The Internet Society (2006). 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.

Acknowledgment

Funding for the RFC Editor function is currently provided by the Internet Society.