

NETWORK WORKING GROUP  
Internet-Draft  
Intended status: Standards Track  
Expires: October 1, 2010

J. Altman  
Secure Endpoints  
N. Williams  
Oracle  
L. Zhu  
Microsoft Corporation  
March 30, 2010

Channel Bindings for TLS  
draft-altman-tls-channel-bindings-10.txt

## Abstract

This document defines three channel binding types for Transport Layer Security (TLS), `tls-unique`, `tls-server-end-point`, and `tls-unique-for-telnet`, in accordance with [RFC 5056](#) (On Channel Binding).

## Status of this Memo

This Internet-Draft is submitted to IETF in full conformance with the provisions of [BCP 78](#) and [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 October 1, 2010.

## Copyright Notice

Copyright (c) 2010 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents

---

Internet-Draft

TLS Channel Bindings

March 2010

(<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the BSD License.

This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

## Table of Contents

<a href="#">1.</a>	Conventions used in this document . . . . .	<a href="#">3</a>
<a href="#">2.</a>	Introduction . . . . .	<a href="#">4</a>
<a href="#">3.</a>	The 'tls-unique' Channel Binding Type . . . . .	<a href="#">5</a>
<a href="#">3.1.</a>	Description . . . . .	<a href="#">5</a>
<a href="#">3.2.</a>	Registration . . . . .	<a href="#">6</a>
<a href="#">4.</a>	The 'tls-server-end-point' Channel Binding Type . . . . .	<a href="#">7</a>
<a href="#">4.1.</a>	Description . . . . .	<a href="#">7</a>
<a href="#">4.2.</a>	Registration . . . . .	<a href="#">7</a>
<a href="#">5.</a>	The 'tls-unique-for-telnet' Channel Binding Type . . . . .	<a href="#">9</a>
<a href="#">5.1.</a>	Description . . . . .	<a href="#">9</a>
<a href="#">5.2.</a>	Registration . . . . .	<a href="#">9</a>
<a href="#">6.</a>	Applicability of TLS Channel Binding Types . . . . .	<a href="#">11</a>
<a href="#">7.</a>	Required Application Programming Interfaces . . . . .	<a href="#">14</a>
<a href="#">8.</a>	Description of backwards-incompatible changes made herein to 'tls-unique' . . . . .	<a href="#">15</a>
<a href="#">9.</a>	IANA Considerations . . . . .	<a href="#">16</a>
<a href="#">10.</a>	Security Considerations . . . . .	<a href="#">17</a>
<a href="#">10.1.</a>	Cryptographic Algorithm Agility . . . . .	<a href="#">17</a>
<a href="#">10.2.</a>	On Disclosure of Channel Bindings Data by Authentication Mechanisms . . . . .	<a href="#">18</a>
<a href="#">11.</a>	References . . . . .	<a href="#">20</a>
<a href="#">11.1.</a>	Normative References . . . . .	<a href="#">20</a>
<a href="#">11.2.</a>	Normative References for 'tls-server-end-point' . . . . .	<a href="#">20</a>
<a href="#">11.3.</a>	Informative References . . . . .	<a href="#">20</a>
	Authors' Addresses . . . . .	<a href="#">22</a>

## 1. 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](#)].

## [2.](#) Introduction

Subsequent to the publication of "On Channel Bindings" [[RFC5246](#)], three channel binding types for Transport Layer Security (TLS) were proposed, reviewed and added to the IANA channel binding type registry, all in accordance with [[RFC5246](#)]. Those channel binding types are: 'tls-unique', 'tls-server-end-point', and 'tls-unique-for-telnet'. It has become desirable to have these channel binding types re-registered through an RFC so as to make it easier to reference them, and to correct them to describe actual implementations. This document does just that. The authors of those three channel binding types have, or have indicated that they will, transferred "ownership" of those channel binding types to the IESG.

We also provide some advice on the applicability of these channel binding types, as well as advice on when to use which. And we provide an abstract API that TLS implementors should provide, by which to obtain channel bindings data for a TLS connection.

WARNING: it turns out that the first implementor implemented and deployed something rather different than what was described in the IANA registration for 'tls-unique'. Subsequently it was decided that

we should adopt that form of 'tls-unique'. This means that this document makes a backwards-incompatible change to 'tls-unique'. See [Section 8](#) for more details.

### [3.](#) The 'tls-unique' Channel Binding Type

IANA is hereby directed to update the registration of the 'tls-unique' channel binding type to match the following. There are material and substantial changes from the original registration, both in the description as well as registration meta-data (such as registration ownership).

#### [3.1.](#) Description

Description: The first TLS Finished message sent (note: the Finished struct) in the most recent TLS handshake of the TLS connection being bound to (note: TLS connection, not session, so that the channel binding is specific to each connection regardless of whether session resumption is used). If TLS re-negotiation takes place before the

channel binding operation, then the first TLS Finished message sent of the latest/inner-most TLS connection is used. Note that for full TLS handshakes the first Finished message is sent by the client, while for abbreviated TLS handshakes (session resumption) the first Finished message is sent by the server.

**WARNING:** The definition, security and interoperability considerations of this channel binding type have changed since the original registration. Implementors should read the document that last updated this registration for more information.

Interoperability note:

This definition of `tls-unique` means that the channel's bindings data may change over time, which in turn creates a synchronization problem should the channel's bindings data change between the time that the client initiates authentication with channel binding and the time that the server begins to process the client's first authentication message. If that happens the authentication will fail spuriously.

This synchronization problem can be avoided by clients and servers as follows, based on the fact that while servers may request TLS re-negotiation, only clients may initiate it. Server applications **MUST NOT** request TLS re-negotiation during phases of the application protocol during which application layer authentication occurs. Client applications **SHOULD NOT** initiate TLS re-negotiation between the start and completion of authentication.

The rationale for making the server behavior a requirement while the client behavior is only a recommendation is that there typically exist TLS APIs for requesting re-negotiation on the server side of a TLS connection, while many client TLS stacks do

not provide fine-grained control over when TLS re-negotiation occurs.

Application protocols should be designed in such a way that a server would never need to request TLS re-negotiation immediately before or during application-layer authentication.

### [3.2.](#) Registration

- o Channel binding unique prefix: tls-unique
- o Channel binding type: unique
- o Channel type: TLS [[RFC5246](#)]
- o Published specification: <this document>
- o Channel binding is secret: no
- o Description: <See specification>
- o Intended usage: COMMON
- o Person and email address to contact for further information: Larry Zhu (lzhu@microsoft.com), Nicolas Williams (Nicolas.Williams@sun.com).
- o Owner/Change controller name and email address: IESG.
- o Expert reviewer name and contact information: IETF TLS WG (tls@ietf.org, failing that, ietf@ietf.org)
- o Note: see the published specification for advice on the applicability of this channel binding type.

#### [4.](#) The 'tls-server-end-point' Channel Binding Type



IANA is hereby directed to update the registration of the 'tls-server-end-point' channel binding type to match the following. Note that the only material changes from the original registration should be: the "owner" (now the IESG), the contacts, the published specification, and a note indicating that the published specification should be consulted for applicability advice. References were added to the description. All other fields of the registration are copied here for the convenience of readers.

#### [4.1.](#) Description

Description: The hash of the TLS server's certificate [[RFC5280](#)] as it appears, octet for octet, in the server's Certificate message (note that the Certificate message contains a certificate\_list, the first element of which is the server's certificate).

The hash function is to be selected as follows:

- o if the certificate's signatureAlgorithm uses a single hash function, and that hash function is either MD5 [[RFC1321](#)] or SHA-1 [[RFC3174](#)] then use SHA-256 [[FIPS-180-2](#)];
- o if the certificate's signatureAlgorithm uses a single hash function and that hash function neither MD5 nor SHA-1, then use the hash function associated with the certificate's signatureAlgorithm;
- o if the certificate's signatureAlgorithm uses no hash functions or multiple hash functions, then this channel binding type's channel bindings are undefined at this time (updates to this channel binding type may occur to address this issue if it ever arises).

The reason for using a hash of the certificate is that some implementations need to track the channel binding of a TLS session in kernel-mode memory, which is often at a premium.

#### [4.2.](#) Registration

- o Channel binding unique prefix: tls-server-end-point
- o Channel binding type: end-point
- o Channel type: TLS [[RFC5246](#)]
- o Published specification: <this document>

- o Channel binding is secret: no
- o Description: <See specification>
- o Intended usage: COMMON
- o Person and email address to contact for further information: Larry Zhu (lzhu@microsoft.com), Nicolas Williams (Nicolas.Williams@sun.com).
- o Owner/Change controller name and email address: IESG.
- o Expert reviewer name and contact information: IETF TLS WG (tls@ietf.org, failing that, ietf@ietf.org)
- o Note: see the published specification for advice on the applicability of this channel binding type.

## [5.](#) The 'tls-unique-for-telnet' Channel Binding Type

IANA is hereby directed to update the registration of the 'tls-unique-for-telnet' channel binding type to match the following. Note that the only material changes from the original registration should be: the "owner" (now the IESG), the contacts, the published specification, and a note indicating that the published specification should be consulted for applicability advice. The description is also clarified. We also moved security considerations notes to the security considerations section of this document. All other fields of the registration are copied here for the convenience of readers.

### [5.1.](#) Description

Description: There is a proposal for adding a "StartTLS" extension to TELNET, and a channel binding extension for the various TELNET AUTH mechanisms whereby each side sends the other a "checksum" (MAC) of their view of the channel's bindings. The client uses the TLS Finished messages (note: the Finished struct) sent by the client and server, each concatenated in that order and in their clear text form, of the first TLS handshake of the connection being bound to. The server does the same but in the opposite concatenation order (server, then client).

### [5.2.](#) Registration

- o Channel binding unique prefix: tls-unique-for-telnet
- o Channel binding type: unique
- o Channel type: TLS [[RFC5246](#)]
- o Published specification: <this document>
- o Channel binding is secret: no
- o Description: <See specification>
- o Intended usage: COMMON

- o Person and email address to contact for further information: Jeff Altman (jaltman@secure-endpoints.com), Nicolas Williams (Nicolas.Williams@sun.com).
- o Owner/Change controller name and email address: IESG.
- o Expert reviewer name and contact information: IETF TLS WG (tls@ietf.org, failing that, ietf@ietf.org)

Altman, et al.

Expires October 1, 2010

[Page 10]

---

Internet-Draft

TLS Channel Bindings

March 2010

- o Note: see the published specification for advice on the applicability of this channel binding type.

## 6. Applicability of TLS Channel Binding Types

The 'tls-unique-for-telnet' channel binding type is only applicable to TELNET [[RFC0854](#)], and is available for all TLS connections.

The 'tls-unique' channel binding type is available for all TLS connections, while 'tls-server-end-point' is only available when TLS cipher suites with server certificates are used, specifically: cipher suites that use the Certificate handshake message, which typically involve the use of PKIX [[RFC5280](#)]. For example, 'tls-server-end-point' is available when using TLS ciphers suites such as (this is not an exhaustive list):

- o TLS\_DHE\_DSS\_WITH\_\*
- o TLS\_DHE\_RSA\_WITH\_\*
- o TLS\_DH\_DSS\_WITH\_\*
- o TLS\_DH\_RSA\_WITH\_\*
- o TLS\_ECDHE\_ECDSA\_WITH\_\*
- o TLS\_ECDHE\_RSA\_WITH\_\*

- o TLS\_ECDH\_ECDSA\_WITH\_\*
- o TLS\_ECDH\_RSA\_WITH\_\*
- o TLS\_RSA\_PSK\_WITH\_\*
- o TLS\_RSA\_WITH\_\*
- o TLS\_SRP\_SHA\_DSS\_WITH\_\*
- o TLS\_SRP\_SHA\_RSA\_WITH\_\*

but is not available when using TLS cipher suites such as (this is not an exhaustive list):

- o TLS\_DHE\_PSK\_WITH\_\*
- o TLS\_DH\_anon\_WITH\_\*
- o TLS\_ECDHE\_PSK\_WITH\_\*
- o TLS\_ECDH\_anon\_WITH\_\*

- o TLS\_KRB5\_WITH\_\*
- o TLS\_PSK\_WITH\_\*
- o TLS\_SRP\_SHA\_WITH\_\*

Nor is 'tls-server-end-point' applicable for use with OpenPGP server certificates [[RFC5081](#)] [[RFC4880](#)] (since these don't use the Certificate handshake message).

Therefore 'tls-unique' is generally better than 'tls-server-end-point'. However, 'tls-server-end-point' may be used with existing TLS server-side proxies ("concentrators") without modification to the proxies, whereas 'tls-unique' may require firmware or software updates to server-side proxies. Therefore there may be cases where 'tls-server-end-point' may interoperate but where 'tls-unique' may not.

Also, authentication mechanisms may arise which depend on channel bindings to contribute entropy, in which case unique channel bindings would have to always be used in preference to end-point channel bindings. At this time there are no such mechanisms, though one such SASL mechanism has been proposed. Whether such mechanisms should be allowed is out of scope for this document.

In other words, for many applications there may be two potentially applicable TLS channel binding types. Channel binding is all or nothing for the GSS-API [[RFC2743](#)], and likely other frameworks. Therefore agreement on the use of channel binding, and a particular channel binding type is necessary. Such agreement can be obtained a priori, by convention, or negotiated.

The specifics of whether and how to negotiate channel binding types are beyond the scope of this document. However, it is RECOMMENDED that application protocols making use of TLS channel bindings, use 'tls-unique' exclusively, except, perhaps, where server-side proxies are common in deployments of an application protocol. In the latter case an application protocol MAY specify that 'tls-server-end-point' channel bindings must be used when available, with 'tls-unique' being used when 'tls-server-end-point' channel bindings are not available. Alternatively, the application may negotiate which channel binding type to use, or may make the choice of channel binding type configurable.

Specifically, application protocol specifications MUST indicate at least one mandatory to implement channel binding type, MAY specify a negotiation protocol, MAY allow for out-of-band negotiation or configuration, and SHOULD have a preference for 'tls-unique' over

'tls-server-end-point'.

## [7.](#) Required Application Programming Interfaces

TLS implementations supporting the use of 'tls-unique' and/or 'tls-unique-for-telnet' channel binding types, MUST provide application



programming interfaces by which applications (clients and servers both) may obtain the channel bindings for a TLS connection. Such interfaces may be expressed in terms of extracting the channel bindings data for a given connection and channel binding type. Alternatively the implementor may provide interfaces by which to obtain the initial client Finished message, the initial server Finished message and/or the server certificate (in a form that matches the description of the 'tls-server-end-point' channel binding type). In the latter case the application has to have knowledge of the channel binding type descriptions from this document. This document takes no position on which form these application programming interfaces must take.

TLS implementations supporting TLS re-negotiation SHOULD provide APIs that allow for application control over when re-negotiation can take place. For example, a TLS client implementation may provide a "callback" interface to indicate that the server requested re-negotiation, but may not start re-negotiation until the application calls a function to indicate that now is a good time to re-negotiate.

[8.](#) Description of backwards-incompatible changes made herein to 'tls-unique'

The original description of 'tls-unique' read as follows:

```
|OLD| Description: The client's TLS Finished message (note: the  
|OLD| Finished struct) from the first handshake of the connection  
|OLD| (note: connection, not session, so that the channel binding  
|OLD| is specific to each connection regardless of whether session  
|OLD| resumption is used).
```

Original 'tls-unique' description

In other words: the client's Finished message from the first handshake of a connection, regardless of whether that handshake was a full or abbreviated handshake, and regardless of how many subsequent handshakes (re-negotiations) might have followed.

As explained in [Section 2](#) this is no longer the description of 'tls-unique', and the new description is not backwards compatible with the original except in the case of TLS connections where: a) only one handshake has taken place before application-layer authentication, and b) that one handshake was a full handshake.

This change has a number of implications:

- o Backwards-incompatibility. It is possible that some implementations of the original 'tls-unique' channel binding type may have been deployed. We know of at least one TLS implementation that exports 'tls-unique' channel bindings with the original semantics, but we know of no deployed application using the same. Implementations of the original and new 'tls-unique' channel binding type will only interoperate when: a) full TLS handshakes are used, b) TLS re-negotiation is not used.
- o Security considerations -- see [Section 10](#).
- o Interoperability considerations. As described in [Section 3](#) the new definition of the 'tls-unique' channel binding type has an interoperability problem that may result in spurious authentication failures unless the application implements one or both of the techniques described in that section.

## [9.](#) IANA Considerations

The IANA is hereby directed to update three existing channel binding type registrations. See the rest of this document.

## 10. Security Considerations

The Security Considerations sections of [[RFC5056](#)], [[RFC5246](#)] and [[RFC5746](#)] apply to this document.

The TLS Finished messages (see [section 7.4.9 of \[RFC5246\]](#)) are known to both endpoints of a TLS connection, and are cryptographically bound to it. For implementations of TLS that correctly handle re-negotiation [[RFC5746](#)] each handshake on a TLS connection is bound to the preceding handshake, if any. Therefore the TLS Finished messages can be safely used as a channel binding provided that the authentication mechanism doing the channel binding conforms to the requirements in [[RFC5056](#)]. Applications utilizing 'tls-unique' channel binding with TLS implementations without support for secure re-negotiation [[RFC5746](#)] MUST ensure that that ChangeCipherSpec has been used in any and all re-negotiations prior to application-layer authentication, and MUST discard any knowledge learned from the server prior to the completion of application-layer authentication.

The server certificate, when present, is also cryptographically bound to the TLS connection through its use in key transport and/or authentication of the server (either by dint of its use in key transport, by its use in signing key agreement, or by its use in key agreement). Therefore the server certificate is suitable as an end-point channel binding as described in [[RFC5056](#)].

### 10.1. Cryptographic Algorithm Agility

The 'tls-unique' and 'tls-unique-for-telnet' channel binding types do not add any use of cryptography beyond that used by TLS itself. Therefore these two channel binding types add no considerations with respect to cryptographic algorithm agility.

The 'tls-server-end-point' channel binding type consist of a hash of a server certificate. The reason for this is to produce manageably small channel binding data, as some implementations will be using kernel-mode memory (which is typically scarce) to store these. This use of a hash algorithm is above and beyond TLS's use of cryptography, therefore the 'tls-server-end-point' channel binding type has a security consideration with respect to hash algorithm agility. The algorithm to be used, however, is derived from the server certificate's signature algorithm as described in [Section 4.1](#); to recap: use SHA-256 if the certificate signature algorithm uses MD5 or SHA-1, else use whatever hash function the certificate uses (unless the signature algorithm uses no hash functions or more than one hash function, in which case 'tls-server-end-point' is undefined). This construction automatically makes 'tls-server-end-point' hash algorithm agile, with a dependency on PKIX and TLS for

hash agility.

Current proposals for randomized signatures algorithms [[I-D.irtf-cfrg-rhash](#)] [[NIST-SP.800-106.2009](#)] use hash functions in their construction -- a single hash function in each algorithm. Therefore the 'tls-server-end-point' channel binding type should be available even in cases where new signatures algorithms are used that are based on current randomized hashing proposals (but we cannot guarantee this, of course).

#### [10.2.](#) On Disclosure of Channel Bindings Data by Authentication Mechanisms

When these channel binding types were first considered, one issue that some commenters were concerned about was the possible impact on the security of the TLS channel, of disclosure of the channel bindings data by authentication mechanisms. This can happen, for example, when an authentication mechanism transports the channel bindings data, with no confidentiality protection, over other transports (for example, in communicating with a trusted third party), or when the TLS channel provides no confidentiality protection and the authentication mechanism does not protect the confidentiality of the channel bindings data. This section considers that concern.

When the TLS connection uses a cipher suite that does not provide

confidentiality protection, the TLS Finished messages will be visible to eavesdroppers, regardless of what the authentication mechanism does. The same is true of the server certificate which, in any case, is generally visible to eavesdroppers. Therefore we must consider our choices of TLS channel bindings here to be safe to disclose by definition -- if that were not the case then TLS with cipher suites that don't provide confidentiality protection would be unsafe. Furthermore, the TLS Finished message construction depends on the security of the TLS PRF, which in turn needs to be resistant to key recovery attacks, and we think that it is, as it is based on HMAC, and the master secret is, well, secret (and the result of key exchange).

Note too that in the case of an attempted active man-in-the-middle attack, the attacker will already possess knowledge of the TLS finished messages for both inbound and outbound TLS channels (which will differ, given that the attacker cannot force them to be the same). No additional information is obtained by the attacker from the authentication mechanism's disclosure of channel bindings data -- the attacker already has it, even when cipher suites providing confidentiality protection are provided.

None of the channel binding types defined herein produce channel bindings data that must be kept secret. Moreover, none of the channel binding types defined herein can be expected to be private (known only to the end-points of the channel), except that the unique TLS channel binding types can be expected to be private when a cipher suite that provides confidentiality protection is used to protect the Finished message exchanges and the application data records containing application-layer authentication messages.

## [11.](#) References

### [11.1.](#) Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC5056] Williams, N., "On the Use of Channel Bindings to Secure Channels", [RFC 5056](#), November 2007.
- [RFC5246] Dierks, T. and E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.2", [RFC 5246](#), August 2008.

- [RFC5746] Rescorla, E., Ray, M., Dispensa, S., and N. Oskov, "Transport Layer Security (TLS) Renegotiation Indication Extension", [RFC 5746](#), February 2010.

### [11.2.](#) Normative References for 'tls-server-end-point'

- [FIPS-180-2]  
United States of America, National Institute of Standards and Technology, "Secure Hash Standard (Federal Information Processing Standard (FIPS) 180-2".

### [11.3.](#) Informative References

- [I-D.irtf-cfrg-rhash]  
Halevi, S. and H. Krawczyk, "Strengthening Digital Signatures via Randomized Hashing", [draft-irtf-cfrg-rhash-01](#) (work in progress), October 2007.
- [NIST-SP.800-106.2009]  
National Institute of Standards and Technology, "NIST Special Publication 800-106: Randomized Hashing for Digital Signatures", February 2009.
- [RFC0854] Postel, J. and J. Reynolds, "Telnet Protocol Specification", STD 8, [RFC 854](#), May 1983.
- [RFC1321] Rivest, R., "The MD5 Message-Digest Algorithm", [RFC 1321](#), April 1992.
- [RFC2743] Linn, J., "Generic Security Service Application Program Interface Version 2, Update 1", [RFC 2743](#), January 2000.
- [RFC3174] Eastlake, D. and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", [RFC 3174](#), September 2001.

- [RFC4880] Callas, J., Donnerhacke, L., Finney, H., Shaw, D., and R. Thayer, "OpenPGP Message Format", [RFC 4880](#), November 2007.
- [RFC5081] Mavrogiannopoulos, N., "Using OpenPGP Keys for Transport Layer Security (TLS) Authentication", [RFC 5081](#),



November 2007.

- [RFC5280] Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", [RFC 5280](#), May 2008.

## Authors' Addresses

Jeff Altman  
Secure Endpoints  
255 W 94TH ST PHB  
New York, NY 10025  
US

Email: [jaltman@secure-endpoints.com](mailto:jaltman@secure-endpoints.com)

Nicolas Williams  
Oracle  
5300 Riata Trace Ct  
Austin, TX 78727  
US

Email: [Nicolas.Williams@oracle.com](mailto:Nicolas.Williams@oracle.com)

Larry Zhu  
Microsoft Corporation  
One Microsoft Way  
Redmond, WA 98052  
US

Email: [lzhu@microsoft.com](mailto:lzhu@microsoft.com)

