[MS-HCEP]:
Health Certificate Enrollment Protocol Specification

Revision Summary

Date	Revision History	Revision Class	Comments
03/14/2007	1.0		Version 1.0 release
04/10/2007	1.1		Version 1.1 release
05/18/2007	1.2		Version 1.2 release
06/08/2007	2.0	Major	Updated and revised the technical content.
07/10/2007	2.1	Minor	Updated the technical content.

Date	Revision History	Revision Class	Comments
08/17/2007	3.0	Major	Revised content based on Trustee feedback.
09/21/2007	4.0	Major	Updated and revised the technical content.
10/26/2007	5.0	Major	Updated and revised the technical content.
01/25/2008	5.0.1	Editorial	Revised and edited the technical content.

Table of Contents

1 Introduction

This document specifies the Health Certificate Enrollment Protocol. The Health Certificate Enrollment Protocol is a Microsoft proprietary remote procedure call (RPC) interface that allows a network endpoint to obtain digital certificates. These certificates are conditionally issued based on the compliance of that endpoint with security policy defined for the network.

1.1 Glossary

The following terms are defined in [MS-GLOS]:

Abstract Syntax Notation One (ASN.1)
Active Directory (AD)
Active Directory Domain
Base64
Basic Encoding Rules (BER)
Certificate Chain
Certification
Certificate Authority (CA)
Cryptographic Service Provider (CSP)
Directory
Distinguished Encoding Rules (DER)
Domain
Enroll/Enrollment
Extended Key Usage (EKU)
Fully Qualified Domain Name (FQDN)
Health Certificate
Health Certificate Enrollment Agent (HCEA)
Health Policy Server
Health Registration Authority (HRA)
Health State
HTTP Internal Server Error
HTTP OK
Hypertext Transfer Protocol (HTTP)
Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)
Internet Protocol Security (IPsec)
Object Identifier (OID)
Public Key Cryptography Standards (PKCS)
Registration Authority (RA)
Remote Access Dial-In User Service (RADIUS)
Self-Signed Certificate
Statement of Health (SoH)
Statement of Health Response (SoHR)
System Health Entity
Trusted Platform Module (TPM)
Uniform Resource Locator (URL)
User Agent

The following terms are specific to this document:

Cryptographic Application Programming Interface (CAPI): Also known as Windows **Cryptographic Application Programming Interface**, CryptoAPI, and Microsoft Cryptography API. An application programming interface (API) that allows developers using the Microsoft Windows operating system to secure Windows-based applications.

MAY, SHOULD, MUST, SHOULD NOT, MUST NOT: These terms (in all caps) are used as specified in [RFC2119]. All statements of optional behavior use either MAY, SHOULD, or SHOULD NOT.

1.2 References

1.2.1 Normative References

We conduct frequent surveys of the normative references to assure their continued availability. If you have any issue with finding a normative reference, please contact dochelp@microsoft.com. We will assist you in finding the relevant information. Please check the archive site, http://msdn2.microsoft.com/en-us/library/E4BD6494-06AD-4aed-9823-445E921C9624, as an additional source.

[ITUX680] ITU-T, "Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", Recommendation X.680, July 2002, http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-0207.pdf

[MS-GLOS] Microsoft Corporation, "Windows Protocols Master Glossary", March 2007.

[MS-SOH] Microsoft Corporation, "Statement of Health for Network Access Protection (NAP) Protocol Specification", July 2006.

[MS-WCCE] Microsoft Corporation, "Windows Client Certificate Enrollment Protocol Specification", July 2006.

[RFC20] Cerf, V., "ASCII Format for Network Interchange", RFC 20, October 1969, http://www.ietf.org/rfc/rfc20.txt

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2315] Kaliski, B., "PKCS #7: Cryptographic Message Syntax Version 1.5", RFC 2315, March 1998, http://www.ietf.org/rfc/rfc2315.txt

[RFC2409] Harkins, D. and Carrel, D., "The Internet Key Exchange (IKE)", RFC 2409, November 1998, http://www.ietf.org/rfc/rfc2409.txt

[RFC2446] Silverberg, S., Mansour, S., Dawson, F., and Hopson, R., "iCalendar Transport-Independent Interoperability Protocol (iTIP) Scheduling Events, BusyTime, To-Dos, and Journal Entries", RFC 2446, November 1998, http://www.ietf.org/rfc/rfc2446.txt

[RFC2616] Fielding, R., et al., "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2616, June 1999, http://www.ietf.org/rfc/rfc2616.txt

[RFC2818] Rescorla, E., "HTTP Over TLS", RFC 2818, May 2000, http://www.ietf.org/rfc/rfc2818.txt

[RFC2986] Nystrom, M. and Kaliski, B., "PKCS#10: Certificate Request Syntax Specification", RFC 2986, November 2000, http://www.ietf.org/rfc/rfc2986.txt

[RFC3174] Eastlake III, D. and Jones, P., "US Secure Hash Algorithm 1 (SHA1)", RFC 3174, September 2001, http://www.ietf.org/rfc/rfc3174.txt

[RFC3280] Housley, R., Polk, W., Ford, W., and Solo, D., "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 3280, April 2002, http://www.ietf.org/rfc/rfc3280.txt

[RFC3548] Josefsson, S., Ed., "The Base16, Base32, and Base64 Data Encodings", RFC 3548, July 2003, http://www.ietf.org/rfc/rfc3548.txt

[RFC4559] Jaganathan, K., Zhu, L., and Brezak, J., "SPNEGO-based Kerberos and NTLM HTTP Authentication in Microsoft Windows", RFC 4559, June 2006, http://www.ietf.org/rfc/rfc4559.txt

1.2.2 Informative References

[IANA-ENT] Internet Assigned Numbers Authority, "Private Enterprise Numbers", January 2007, http://www.iana.org/assignments/enterprise-numbers

[MS-GPOL] Microsoft Corporation, "Group Policy: Core Protocol Specification", July 2006.

[MS-RNAP] Microsoft Corporation, "Vendor-Specific RADIUS Attributes for Network Access Protection (NAP) Data Structure", July 2006.

[MSDN-CAPI] Microsoft Corporation, "Cryptography", http://msdn2.microsoft.com/en-us/library/aa380255.aspx

[MSDN-CSP] Microsoft Corporation, "Cryptographic Provider Names", http://msdn2.microsoft.com/en-us/library/aa380243.aspx

[MSDN-NAP] Microsoft Corporation, "Network Access Protection", http://www.microsoft.com/technet/network/nap/default.mspx

[MSFT-IPSEC] Microsoft Corporation, "IPsec", http://www.microsoft.com/technet/network/ipsec/default.mspx

[RFC2865] Rigney, C., Willens, S., Rubens, A., and Simpson, W., "Remote Authentication Dial In User Service (RADIUS)", RFC 2865, June 2000, http://www.ietf.org/rfc/rfc2865.txt

[RFC3447] Jonsson, J. and Kaliski, B., "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1", RFC 3447, February 2003, http://www.ietf.org/rfc/rfc3447.txt

[TPM] Trusted Computing Group, "TPM Work Group", https://www.trustedcomputinggroup.org/groups/tpm/

[X509] ITU-T, "Information Technology - Open Systems Interconnection - The Directory: Public-Key and Attribute Certificate Frameworks", Recommendation X.509, August 2005, http://www.itu.int/rec/T-REC-X.509/en

Note There is a charge to download the specification.

1.3 Protocol Overview (Synopsis)

Many network administrators maintaining a secure network require that clients accessing their networks comply with policies established for the network. For example, an administrator might require that every client accessing the network have an active firewall. One of the ways that administrators can ensure compliance of network endpoints is to require that clients **enroll** for a **health certificate**. Health certificates encapsulate the client's compliance with policy in a way that can be presented to interested parties without requiring those parties to perform the validation themselves.

The Health Certificate Enrollment Protocol is designed to accomplish health certificate enrollment. The client sends a Health Certificate Enrollment Protocol request to a **health registration authority (HRA)**. The Health Certificate Enrollment Protocol request includes a certificate request

and a report of the client's current **health state**. The HRA communicates with **health policy servers** and **certification authorities** to form and send a Health Certificate Enrollment Protocol response to the client. If the client is compliant, the response contains an issued certificate and the results of the validation of the client's health state against policy.

The Health Certificate Enrollment Protocol has authenticated and unauthenticated modes. In the authenticated mode, the Health Certificate Enrollment Protocol supports authentication of the server, client, or both. Authentication of the server in the Health Certificate Enrollment Protocol is achieved by using the **Hypertext Transfer Protocol (HTTP)** over Transport Layer Security (TLS), as specified in [RFC2818]. During the establishment of the TLS channel, as specified in [RFC2446], the server is authenticated by the client. Authentication of the client in the Health Certificate Enrollment Protocol is achieved by using SPNEGO-based Kerberos and NTLM HTTP authentication, as specified in [RFC4559].

The Health Certificate Enrollment Protocol is typically deployed in an environment such as the one in the following figure.

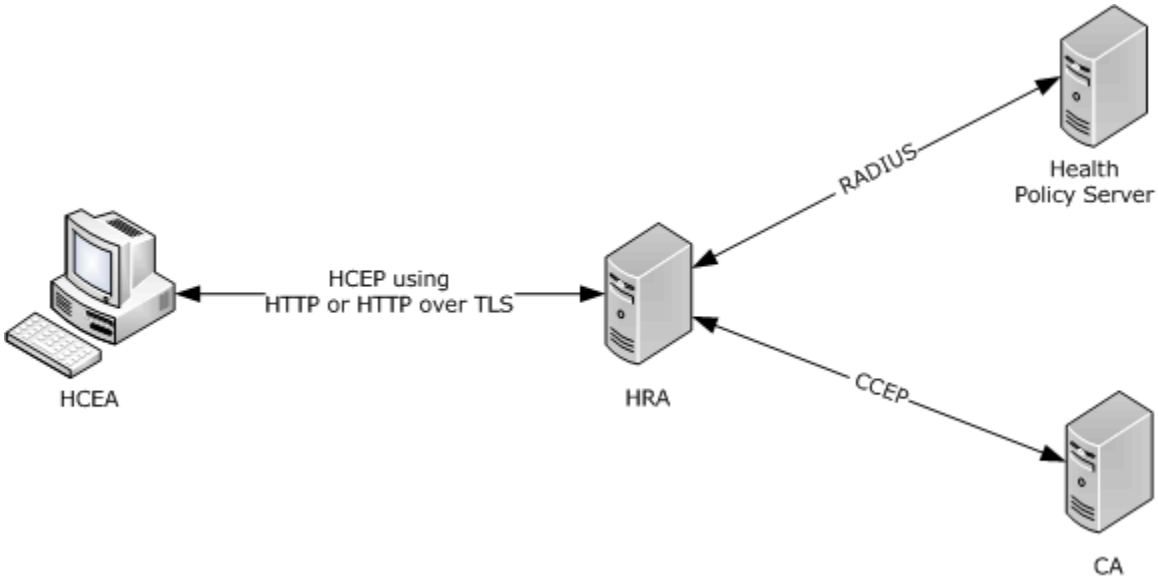

Figure 1: Deployment environment for Health Certificate Enrollment Protocol

In this example, the flow is as follows:

1. The **health certificate enrollment agent (HCEA)** sends a Health Certificate Enrollment Protocol request to the HRA. The HRA is identified by the **Uniform Resource Locator (URL)** with which the HCEA is provisioned. The protocol specified by the provisioned URL ("http" indicates HTTP and "https", or **Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)**, indicates HTTP over TLS) determines whether the HRA is authenticated using TLS.

   The payload of a Health Certificate Enrollment Protocol request message, sent by the HCEA, contains a **Public Key Cryptography Standards (PKCS)** #10 certificate request (as specified in [RFC2986]). The PKCS #10 request contains a **Statement of Health (SoH)** message, as specified in [MS-SOH] section 2.2.5.

2. The HRA sends the SoH to a health policy server for evaluation as per network policies over **Remote Access Dial-In User Service (RADIUS)** (for more information, see [RFC2865]) using Microsoft RADIUS Attributes for Network Access Protection (for more information, see [MS-

RNAP]). The health policy server determines the health state of the client and informs the HRA of this.

3. If the client's health state is compliant, the HRA requests a certificate authority (CA) to issue a certificate. The Microsoft implementation of the HRA uses the Windows Client Certificate Enrollment Protocol, as specified in [MS-WCCE], to request and receive the certificate.

4. The HRA sends a Health Certificate Enrollment Protocol response.

   If the client's health state is compliant, the payload of the response contains a **Statement of Health Response (SoHR)** (as specified in [MS-SOH]) and a PKCS #7 message (as specified in [RFC2315]) with an X.509 (for more information, see [X509]) certificate, as specified in [RFC3280].

   If the client's health state is not compliant, the payload of the response contains only an SoHR without a PKCS #7 message containing the certificate.

1.4 Relationship to Other Protocols

The Health Certificate Enrollment Protocol uses HTTP (as specified in [RFC2616]) or HTTP over TLS (as specified in [RFC2818]) as the transport for its messages. The payload of a Health Certificate Enrollment Protocol request message, sent by the HCEA, contains a PKCS #10 certificate request (as specified in [RFC2986]), which contains an SoH message (as specified in [MS-SOH]) section 2.2.5. The HRA sends an Health Certificate Enrollment Protocol response, the payload of which contains an SoHR, as specified in [MS-SOH] section 2.2.6, and if the client is compliant with health policies, it also includes a PKCS #7 message (as specified in [RFC2315]) with possibly an X.509 certificate, as specified in [RFC3280].

1.5 Prerequisites/Preconditions

For Health Certificate Enrollment Protocol communication to begin, the prerequisite configuration for HCEA is as follows:

1. The HCEA MUST be configured with the URL of the HRA via an implementation-dependent<1> method.

   The protocol specified by the provisioned URL ("http" indicates HTTP and "https" indicates HTTP over TLS) determines if the HRA is authenticated using TLS.

2. The HCEA MUST be configured with the required security parameters to construct a certificate request that is sent in the Health Certificate Enrollment Protocol request. These include, but are not limited to:

   - The algorithm and key length of the public-private key pair associated with the certificate. The HCEA MUST support the RSA algorithm with a key length of 2,048 bits. Other key lengths MAY<2> be supported. For more information, see [RFC3447].
   - The signature algorithm used to sign the certificate request. The HCEA MUST support the Secure Hash Algorithm 1 (SHA1), as specified in [RFC3174].

   The HCEA and HRA MUST agree on the algorithm that HCEP uses. However, the way these algorithms are configured on the HCEA and HRA is out of the scope of this document.<3>

3. The HCEA and HRA implementation need to agree on the same **object identifier (OID)** values for the fields in section 2.2.1.4.<4>

The prerequisite configuration for the HRA is the following:

- If the HRA is configured to authenticate the client, the settings required, as specified in [RFC4559] section 4.1, MUST be configured on the HRA.

1.6 Applicability Statement

The Health Certificate Enrollment Protocol allows a client machine to obtain an X.509 certificate, as specified in [RFC3280], that represents its compliance to policy. Because the Health Certificate Enrollment Protocol relies on the client to make accurate reports of its current state, the protocol is not applicable by itself in environments where the client's compliance must be absolutely guaranteed. However, the Health Certificate Enrollment Protocol MAY<5> be used in such environments if supplemented by the use of hardware credentials or other suitable security mechanisms (for more information, see [TPM]) that can improve the reliability of the client reports.

Applicable uses of such a X.509 certificate include, but are not limited to, certificate-based **Internet Protocol security (IPsec)**, as specified in [RFC2409]. In an IPsec scenario, network administrators can require clients to comply with the network security policies before accessing resources on the network. For instance, the administrators can configure IPsec policies to require a client to present a health certificate to the resource (as an indication of the client's compliance with network security policies) before the client can have access to the resource.

1.7 Versioning and Capability Negotiation

The Health Certificate Enrollment Protocol does not perform any version detection or capability negotiation by itself since this is the first version.

Although this version of the Health Certificate Enrollment Protocol does not support the concept of versioning, there is a version number field in Health Certificate Enrollment Protocol messages. This field is intended for future use. In the current version of the Health Certificate Enrollment Protocol, implementations MUST set the version field value to "1.0". The fixed value version field (see sections 2.2.1.2 and 2.2.2.2) is intended to enable future versions of the protocol to negotiate a version that is commonly supported by both HCEA and HRA.

1.8 Vendor-Extensible Fields

There are no vendor-extensible fields for this protocol.

1.9 Standards Assignments

Microsoft has been assigned the object identifier (OID) 1.3.6.1.4.1.311 by IANA as the Microsoft Private Enterprise IANA code. (For more information, see [IANA-ENT].) The Windows implementation of the Health Certificate Enrollment Protocol uses the extension 1.3.6.1.4.1.311.47.1.1 under this OID for identifying the health extension for certificates and certificate requests, as specified in section 2.2.1.4.

2 Messages

The following sections specify how Health Certificate Enrollment Protocol messages are encapsulated on the wire and common Health Certificate Enrollment Protocol data types.

2.1 Transport

Health Certificate Enrollment Protocol messages MUST be transmitted over HTTP, as specified in [RFC2616], or HTTP over TLS, as specified in [RFC2818]. HCEP MUST be encapsulated within these protocols as a POST method (as specified in [RFC2616] section 9.5) to a specific URL. (The existence of the URL is a precondition of the protocol, as specified in section 1.5). The POST MUST have specific HTTP headers and an entity body (as specified in [RFC2616]) formatted as specified later in section 2.2.1.

The Health Certificate Enrollment Protocol is encapsulated within and depends on HTTP or HTTP over TLS for delivery of messages. HCEP does not impose any message retransmissions or other requirements on those transports. The choice of the transport is based entirely on the URL present at the HCEA. For more information on the configuration of the HCEA, see section 1.5.

2.2 Message Syntax

All strings that follow are ASCII strings, as specified in [RFC20], and MUST conform to the general HTTP rules on string values in headers, as specified in [RFC2616] section 4.2.

2.2.1 HCEP Request

This is an HTTP POST request made to a provisioned URL.

The HCEP request encapsulated in HTTP includes a number of fields in the HTTP message header. Some of them are standard fields (as specified in [RFC2616] sections 4.5, 5.3, and 7.1) that must take on specific values, while others are new fields defined by HCEP. These fields MUST follow the rules as specified in [RFC2616] section 4.2.

2.2.1.1 Standard HTTP Message Header Fields

The HTTP message headers in the HCEP request MUST include the following fields with these specified values.

Tokens

Pragma: MUST be "no-cache". This is an HTTP header, as specified in [RFC2616] section 4.5.

Content-Type: MUST be "application/healthcertificate-request". This is an HTTP header field, as specified in [RFC2616] section 7.1.

Content-Length: MUST be the size in bytes of HTTP message body (as specified in section 2.2.1.3). This is an HTTP header field, as specified in [RFC2616] section 7.1.

The HTTP message header MAY<6> contain other fields, as specified in [RFC2616]. These fields MAY<7> be ignored by the HRA during the processing of the HTTP header. An example of such a field is the **User-Agent** field, as specified in [RFC2616] section 14.43:

- User Agent: "NAP IPSec Enforcement v1.0".

2.2.1.2 HTTP Message Header Fields Introduced by HCEP

The HTTP message headers in the HCEP request MUST include the following fields.

Tokens

HCEP-Version: MUST be a string with value "1.0".

HCEP-Correlation-Id: This MUST be a **base64**-encoded (as specified in [RFC3548] section 3) encoded Correlation Id. A Correlation Id (as specified in [MS-SOH] section 2.2.4.6) is a 24-byte value that uniquely identifies an HCEP transaction originating from the HCEA. It SHOULD<8> be the same as the Correlation Id that is present in SoH, as specified in [MS-SOH] section 2.2.4.

2.2.1.3 HTTP Message Body Used in HCEP Request

The HCEP request MUST contain an HTTP message body, which follows the rules specified in [RFC2616] section 4.3. The value of this message body MUST be an **ASN.1 Distinguished Encoding Rules (DER)**-encoded health certificate request. The format of the health certificate request MUST be as specified in section 2.2.1.4.

2.2.1.4 Health_Certificate_Request

The health certificate request is a PKCS #10 request (as specified in [RFC2986]) encoded in ASN.1, Distinguished Encoding Rules (DER) format, as specified in [ITUX680]. This MUST be present in an HCEP request. The HRA processing of the health certificate request is specified in section 3.2.3. It MUST contain the following parts, as specified in [RFC2986] section 4.

Tokens

subject: SHOULD be a zero-length string if the Boolean flag specified in section 3.1.1 is true; otherwise, it SHOULD be the predefined string "Anonymous System Health Authentication", as specified in [RFC3280] section 4.1.2.6.<9>

subjectPublicKeyInfo: MUST be a public key for the X.509 certificate, as specified in [RFC3280] section 4.1.

PKCS #10 Attributes: The following X.509 Certificate Extensions are encapsulated as PKCS #10 attributes in the health certificate request. PKCS #10 attributes are as specified in [RFC2986] section 4.1.

Extended Key Usage: A health certificate request MUST contain an **Extended Key Usage (EKU)** extension (as specified in [RFC3280] section 4.2.1.1) with the OID values specified as follows:

- MUST have an OID to indicate that the certificate request is a health certificate request.<10>
- SHOULD have the OID value that indicates id-kp-clientAuth, as specified in [RFC3280] section 4.2.1.13, if the Boolean flag specified in section 3.1.1 is TRUE.<11>

Subject Alternative Name: MUST be the **fully qualified domain name (FQDN)** of the client if the Boolean flag specified in section 3.1.1 is true. This extension MUST be as specified in [RFC3280] section 4.2.1.7.

Statement of Health Certificate Extension: The certificate request MUST have a certificate extension, as specified in [RFC3280]. This certificate extension MUST contain the ASN.1 DER

(as specified in [ITUX680]) encoded Statement of Health (SoH) data (as specified in [MS-SOH]).

Cryptographic Service Provider Certificate Extension: The certificate request MUST have a certificate extension to specify the name of the **Cryptographic Service Provider (CSP)** used to generate the key pair on the HCEA. This attribute contains the name of the CSP used to generate the key pair on the HCEA. The extension and the OID value MUST be as specified in [MS-WCCE] section 2.2.1.5.<12>

The certificate request MUST be signed to prevent tampering using one of the pre-configured signature algorithms specified in section 1.5. The signature over the certificate request MUST be included in the PKCS #10 message, as specified in [RFC2986].

2.2.2 HCEP Response

The HCEP response MUST be an **HTTP OK** response (status-code 200), indicating that there are no errors. Other HTTP response status-codes, as specified in [RFC2616] section 6.1.1, MUST be returned by the server in case of error events, as specified in section 3.1.6. The client-side handling of errors is specified in section 3.1.6.

The HCEP response encapsulated in HTTP includes a number of fields in the HTTP message header. Some are standard fields (as specified in [RFC2616] sections 4.5, 6.2, and 7.1) that must take on specific values, while others are new fields defined by HCEP. These fields MUST follow the rules as specified in [RFC2616] section 4.2.

2.2.2.1 Standard HTTP Message Header Fields

The HTTP message headers in the HCEP response MUST include the following fields.

Tokens

Cache-Control: MUST be "no-cache, must-revalidate" or "must-revalidate, no-cache". This is an HTTP header, as specified in [RFC2616] section 4.5.

Content-Type: MUST be "application/healthcertificate-response". This is an HTTP header field, as specified in [RFC2616] section 7.1.

Content-Length: MUST be the size of ASN.1 DER encoded health certificate response.

2.2.2.2 HTTP Message Header Fields Introduced by HCEP

The HTTP message headers in the HCEP response MUST include the following Tokens.

Tokens

HCEP-Version: MUST be a string with value "1.0".

HCEP-Correlation-Id: MUST be the base64-encoded correlation ID.

This is a 24-byte value that identifies an HCEP transaction. This value MUST be identical to the HCEP-Correlation-Id present in the corresponding HCEP request.

HCEP-SoHR: MUST be the Base64-encoded Statement of Health Response (SoHR), as specified in [MS-SOH] section 2.2.6.

The SoHR is the result of the validation of the SoH, as specified in [MS-SOH] section 2.2.5, received by the HRA from the HCEP request. It is an opaque sequence of bytes that is not interpreted or used directly by HCEP.

HCEP-AFW-Protection-Level: MUST be a string specified as follows:

This value MUST be sent back from the HRA in an HCEP response. Valid values for this field MUST be:

- "1" to indicate that the certificate payload in the HCEP response can be used for signing data.
- "2" to indicate that the certificate payload in the HCEP response can be used for signing and encrypting data.

HCEP-AFW-Zone: MUST be a string specified as follows:

This field MUST be sent back from the HRA in an HCEP response. Valid values for this field MUST be the ASCII (as specified in [RFC20]) representation of the decimal form of integers between 1 and 2^32 -1. This is used as a hint for dynamic selection of a preconfigured policy by the consumer of the health certificate on the client.<13>

The header MAY<14> contain other fields besides those listed here. All other fields SHOULD be ignored by the HCEP client.

2.2.2.3 HTTP Message Body Used in HCEP Response (HTTP OK Response)

The health certificate response MUST be present if the client is deemed to be compliant with policy as the HTTP message body, which follows the rules as specified in [RFC2616] section 4.3.<15> If the message body is present, the value of this message body MUST be an ASN.1 DER-encoded health certificate response. The format of the health certificate response MUST be as specified in section 2.2.2.4.

2.2.2.4 Health Certificate Response

The health certificate response MUST be an ASN.1 DER-encoded PKCS #7 message (as specified in [RFC2315]) containing the issued X.509 certificate (as specified in [RFC3280]) in a **certificate chain**. The format of this response MUST be as specified in [MS-WCCE] section 2.2.1.6.

The health certificate response MUST be present if the client is deemed to be compliant with policy.

The health certificate response SHOULD NOT be present if the client is deemed to be non-compliant with policy. In this case, the message body of the HCEP response will be of length 0.

3 Protocol Details

The Health Certificate Enrollment Protocol is a simple request-response protocol. The Health Certificate Enrollment Protocol allows a network endpoint to obtain digital certificates. These certificates are conditionally issued based on the compliance of that endpoint with security policy defined for the network.

The health certificate enrollment agent (HCEA) sends a Health Certificate Enrollment Protocol request and the HRA responds with a Health Certificate Enrollment Protocol response. The protocol is used by the HCEA to obtain a health certificate based on its compliance with the security policies defined for the network. The protocol is always a single Health Certificate Enrollment Protocol request followed by a single Health Certificate Enrollment Protocol response, as shown in the following diagram.

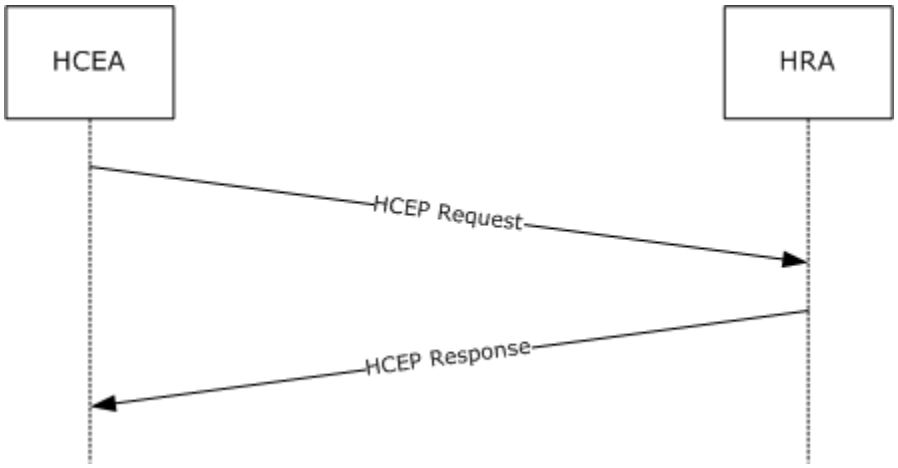

Figure 2: Health Certificate Enrollment Protocol single request and response

3.1 Client Details

3.1.1 Abstract Data Model

The HCEA MUST store the HCEP-Correlation-Id that it sent in its Health Certificate Enrollment Protocol request. It SHOULD<16> use this to verify that the corresponding Health Certificate Enrollment Protocol response is received.

The HCEA SHOULD<17> maintain a Boolean flag, which MUST specify if it expects the HRA to authenticate the client, as specified in [RFC4559]. The manner in which the HCEA determines the value of this flag is outside the scope of this document. This flag determines the values of the **Subject**, **Subject Alternative Name**, and **Extended Key Usage** fields in the health certificate request, as specified in section 2.2.1.4.

3.1.2 Timers

The Health Certificate Enrollment Protocol has no timers of its own.

3.1.3 Initialization

The HCEA constructs and sends a Health Certificate Enrollment Protocol request, as specified in section 2.2.1.1. If the transport specified by the provisioned URL is HTTP over TLS, as specified in

[RFC2818], then the client MUST be set up to trust the certificate authority issuing the server certificate before communication is started.

3.1.4 Higher-Layer Triggered Events

The following higher-layer events affect a Health Certificate Enrollment Protocol client's operation.

- Any state change that can affect the client's compliance to policy SHOULD trigger the HCEA to re-enroll for a health certificate from the configured HRA.<18>
- Any configuration state change that changes the HRA URL configuration SHOULD trigger the HCEA to re-enroll for a health certificate from the configured HRA.<19>

3.1.5 Message Processing Events and Sequencing Rules

3.1.5.1 Sending an HCEP Request

HCEA MUST create a Health Certificate Enrollment Protocol request as follows:

1. Create a Health Certificate Request:
    1. MUST set the subject, subjectPublicKeyInfo, EKU, and Subject Alternative Name in the PKCS #10 (as specified in [RFC2986]) part of the request, as specified in section 2.2.1.4.
    2. MUST obtain an SoH, as specified in [MS-SOH], from the **system health entity**, and set the SoH certificate extension as specified in section 2.2.1.4. Details as to how this is done are implementation specific.<20>
2. MUST set the HTTP message body to an ASN.1 DER (as specified in [ITUX680]) health certificate request as specified in section 2.2.1.
3. Set the HTTP message headers as follows:
    1. MUST set the **Pragma**, **Content-Type**, **HCEP-Version** and the **Content-Length** header fields, as specified in section 2.2.1.
    2. MUST set the HCEP-Correlation-Id, as specified in section 2.2.1.2. It MUST be Base64-encoded, as specified in [RFC3548].
4. The HCEA MUST then send the request to the transport to send to the HRA.
5. If the transport returns an error in either transmitting or receiving the response, HCEA MUST handle them as specified in section 3.1.6.

3.1.5.2 Processing an HCEP Response

If the HCEA receives a HTTP response other than HTTP OK, it MUST perform the error handling as specified in section 3.1.6. On receiving an HTTP OK response, HCEA MUST perform the validation as follows:

1. HCEA MUST validate that all the HCEP response header fields are present in the response, as specified in section 2.2.1.2, and MUST discard the response if the validation fails.
2. HCEA MAY<21> validate the values of all the HCEP response header fields, as specified in sections 2.2.2.1 and 2.2.2.2 and MUST discard the response if the validation fails.

3. HCEA MUST validate the format of the health certificate response present in the HCEP response as specified in section 2.2.2.4. If the health certificate response is invalid, the HCEA MUST discard the HCEP response.

Once the validation checks are complete, the HCEA processes the HCEP response, and MUST pass the SoHR information, as specified in [MS-SOH], in the response to the system health entity that generated the corresponding SoH, as specified in [MS-SOH].

3.1.6 Client-Side Error Handling

If the HCEA encounters any error (including HTTP errors as specified in [RFC2616] section 10) after creating a Health Certificate Enrollment Protocol request, either during transmission of the Health Certificate Enrollment Protocol request or while waiting for a Health Certificate Enrollment Protocol response, then it MUST discard the Health Certificate Enrollment Protocol request. Any subsequent request by the client MUST have a new HCEP-Correlation-Id value.

3.1.7 Timer Events

There are no timer events for this protocol.

3.1.8 Other Local Events

If the IP address of the client machine changes, a new network might be reachable. The client MAY<22> choose to send a Health Certificate Enrollment Protocol request to a server with which a previous request was unsuccessful.

3.2 Server Details

3.2.1 Abstract Data Model

HRA MUST store the HCEP-Correlation-Id that it receives in a Health Certificate Enrollment Protocol request. It MUST use the same HCEP-Correlation-Id in the Health Certificate Enrollment Protocol response that it generates.

The following is a list of predefined settings that SHOULD<23> be present on the server.

- A list of strings used to restrict the requests based on the user-agent strings, as specified in section 2.2.2.1, present in a Health Certificate Enrollment Protocol request. If no string in the list is a substring of the user-agent string, the request MUST be discarded. If this list is empty, all **user agents** SHOULD<24> be allowed.
- A list of algorithms used to restrict the allowed public keys in the certificate request. If the list is empty, all algorithms SHOULD<25> be allowed.
- A list of algorithms used to restrict the allowed signatures in the certificate request. If the list is empty, all algorithms SHOULD<26> be allowed.
- A list of cryptographic service providers (CSPs) used to restrict the CSPs that created the certificate requests. If the list is empty, certificate requests from all CSPs SHOULD<27> be allowed.
- A setting specifying the maximum size, in kilobytes, of the Health Certificate Enrollment Protocol request, including the HTTP headers and the body. If the size of request is less than the maximum size allowed, then the Health Certificate Enrollment Protocol request SHOULD<28> be allowed.

3.2.2 Initialization

No special initialization steps are necessary.

3.2.3 Message Processing Events and Sequencing Rules

3.2.3.1 Validating an HCEP Request

When a Health Certificate Enrollment Protocol request arrives from the client on the HRA, the HRA MUST perform correctness checks on the request.

The HRA MUST perform the following checks:

1. All the required fields in a Health Certificate Enrollment Protocol request MUST be present in the received request, as specified in section 2.2.1.
2. If the HRA is configured to authenticate the client (as specified in section 1.5), then it MUST validate that the Subject Alternative Name (as specified in section 2.2.1.4) is the same as the FQDN of the authenticated client.
3. If the HRA is configured to not authenticate the client (as specified in section 1.5), then it MUST validate that the Subject Alternative Name (specified in section 2.2.1.4) is empty.
4. The health certificate request present in the Health Certificate Enrollment Protocol request MUST be conformant with the specification in section 2.2.1.4 for the rest of the fields not mentioned above.

If any of these checks fail, the HRA MUST respond with an **HTTP internal server error**, as specified in [RFC2616] section 6.1.1.

The HRA SHOULD<29> perform the following checks:

1. The size of the request is less than the maximum size, as specified in section 3.2.1.
2. The user agent present in the HCEP request contains one of the strings from the list of strings specified in section 3.2.1.
3. The algorithm used for signing the certificate request is present in the list of algorithms specified in section 3.2.1.
4. The public key in the certificate request is created using an algorithm present in the list of algorithms specified in section 3.2.1.
5. The CSP used to create the health certificate request is acceptable. This is accomplished by checking that the CSP certificate extension of the health certificate request (as specified in section 2.2.1.4) specifies a CSP that is present in the list of CSPs specified in section 3.2.1.
6. If the HRA is configured to authenticate the client (as specified in section 1.5), then it MAY validate the Subject and Extended Key Usage as specified in section 2.2.1.4.
7. If the HRA is configured to not authenticate the client (as specified in section 1.5), then it MAY validate the Subject as specified in section 2.2.1.4).

If any of the previous checks fails, then the HRA MUST respond with an HTTP Internal Server error, as specified in [RFC2616] section 6.1.1.

3.2.3.2 Processing an HCEP Request

Once the validation checks are complete, the HRA MUST process the Health Certificate Enrollment Protocol request as follows:

1. Extract the HCEP-Correlation-Id from the HCEP request, for use when creating the Health Certificate Enrollment Protocol response.

2. Extract the health certificate request (as specified in section 2.2.1.1) from the Health Certificate Enrollment Protocol request.

3. Extract the SoH (as specified in [MS-SOH]) from the health certificate request, and MUST validate the SoH with a health policy server over RADIUS (for more information, see [RFC2865]) using Microsoft RADIUS Attributes for Network Access Protection. For more information, see [MS-RNAP].

4. The Health Certificate Enrollment Protocol response MUST be created as follows:

   - Create the Health Certificate Enrollment Protocol response header as specified in section 2.2.2. The HCEP-Correlation-Id MUST be the same as the one received in the HCEP request.

   - MUST encode the SoHR (as specified in [MS-SOH]) using ASN.1 DER (as specified in [X509]). Set the HCEP-SoHR in the Health Certificate Enrollment Protocol response header.

   - Set the **HCEP-AFW-Zone** and the **HCEP-AFW-Protection-Level** fields in the Health Certificate Enrollment Protocol response header. The fields MUST be set based on the values received from the health policy server, as specified in section 2.2.2.

   - If the health state is compliant with the network polices, the HRA MUST use the health certificate request to request a health certificate from a CA, for example through the implementation of a Windows Client Certificate Enrollment Protocol request, as specified in [MS-WCCE]. Then the HRA MUST do the following:

     - Constructs the health certificate response by encoding the PKCS #7 message as ASN.1 DER (as specified in [RFC2315]) from the CA.

     - MUST set the Content-Length and the HTTP message body according to the health certificate response.

   - If the health state is not compliant with the network polices, the HRA MUST NOT request a health certificate for the PKCS #10 (as specified in [RFC2986]) request from a CA. The SoHR is sent back in the **HCEP-SoHR** field in the HCEP response header. Since there is no PKCS #7 to be sent back, then it does the following:

     - MUST set the Content-Length to 0 because there is no HTTP message body to be sent.

     The HCEP response MUST then be sent to the client.

3.2.4 Error Handling

Error cases and typical error handling on the server MUST be as follows:

- Malformed HCEP request: If the server finds that the request is malformed according to the rules in section 3.2.3, it MUST respond with an HTTP internal server error, as specified in [RFC2616] section 6.1.1.

- Failure to contact health policy server or certificate authority: If there is any error while trying to contact these servers, the HRA MUST respond with an HTTP internal server error, as specified in [RFC2616] section 6.1.1.

4 Protocol Examples

The client determines through implementation-specific procedures that a health certificate is required. In a common scenario, the client is connected to a network, but the client does not have a valid health certificate for communicating on that network. To access the network, the client needs to acquire a new health certificate. Once the HCEP enrollment process is invoked, the following sequence of events occur.

Compliant Client Example

1. The HCEA obtains the Statement of Health (SoH) from the system health entity.
2. The HCEA then generates a public-private key pair and constructs a health certificate request (see section 2.2.1.4).
3. The client creates the Health Certificate Enrollment Protocol request (see section 2.2.1) and sends it to a pre-configured HRA URL.
4. The HRA receives the Health Certificate Enrollment Protocol request, extracts the SoH from the certificate request, and passes the SoH on to a health policy server that evaluates the SoH in the request.
5. The health policy server responds with the Statement of Health Response (SoHR), which contains the client's health state compliance to network policies.
6. If the client is compliant with network policies, the HRA obtains a health certificate for the certificate request in the Health Certificate Enrollment Protocol request. This may be done using the Windows Client Certificate Enrollment Protocol, as specified in [MS-WCCE].
7. The server creates a Health Certificate Enrollment Protocol response (see section 2.2.2) and sends it to the client.
8. If a certificate was received in the Health Certificate Enrollment Protocol response, the HCEA extracts the certificate and deposits it in the certificate store.

Non-compliant Client Example

1. The HCEA obtains the SoH from the system health entity.
2. The HCEA generates a public-private key pair and constructs a health certificate request (see section 2.2.1.4).
3. The client creates the Health Certificate Enrollment Protocol request (see section 2.2.1) and sends it to a pre-configured HRA URL.
4. The HRA receives the Health Certificate Enrollment Protocol request, extracts the SoH from the certificate request and passes the SoH on to a health policy server that evaluates the SoH.
5. The health policy server responds with the SoHR and the non-compliant results of the evaluation of the client's health state compliance with network policies.
6. Because the client is non-compliant with the network policies, the HRA does not obtain a health certificate for the certificate request in the HCEP request.
7. The server creates a Health Certificate Enrollment Protocol response (see section 2.2.2) and sends it to the client.

8. Because no certificate was received in the Health Certificate Enrollment Protocol response, the SoHR is evaluated and no certificate is deposited in the certificate store.

5 Security

The following sections specify security considerations for implementers of the Health Certificate Enrollment Protocol.

5.1 Security Considerations for Implementers

The Health Certificate Enrollment Protocol does not ensure the authenticity of the Statement of Health (SoH) that is sent to the HRA. The implementation MUST use secure algorithms and methods to ensure the security of the SoH.

The health state of the machine sending the Health Certificate Enrollment Protocol request can contain sensitive information. Hence implementers SHOULD ensure that the choice of transport of Health Certificate Enrollment Protocol messages is appropriate. For example, using HTTP over TLS (as specified in [RFC2818]) to authenticate the server and to provide confidentiality and integrity would be better than using HTTP alone for HCEP messages. When the HCEA is authenticated using Kerberos-based HTTP authentication (as specified in [RFC4559]) it allows the HRA to validate data present in the certificate request. HRA should not impersonate the client and should only identify the client using this authentication method. Otherwise, if the HRA is compromised, it can potentially allow attackers to impersonate clients.

5.2 Index of Security Parameters

Security parameter	Section
Transport protocol	2.1
Health certificate request	2.2.1.4
HRA authentication	2.1
HCEA authentication	1.5

6 Appendix A: Windows Behavior

The information in this specification is applicable to the following versions of Windows:

- Windows Server 2008
- Windows Vista

Exceptions, if any, are noted below. Unless otherwise specified, any statement of optional behavior in this specification prescribed using the terms SHOULD or SHOULD NOT implies Windows behavior in accordance with the SHOULD or SHOULD NOT prescription. Unless otherwise specified, the term MAY implies that Windows does not follow the prescription.

Windows Server 2008 can also act as an "HCEP client" to use HCEP to communicate with Windows Server 2008.

<1> Section 1.5: The Windows implementation of the HCEA depends on the administrator to configure the URL of the HRA. The administrator normally configures this manually or by using Group Policy (as defined in [MS-GPOL])..

<2> Section 1.5: The Windows implementation of the HCEA supports the RSA algorithm with a minimum key length of 1,024 bits. Key lengths longer than 2,048 bits are supported if the supporting cryptographic service providers (CSPs) are added to the system. For more information about the default list of CSPs that are present on a Windows system, see [MSDN-CSP].

<3> Section 1.5: The Windows implementations of the HCEA and the HRA depend on the **Cryptographic Application Programming Interface (CAPI)** (for more information, see [MSDN-CAPI]) for the implementation of the algorithms specified above. The CAPI has a pluggable model for cryptographic service providers (CSPs) to be added, removed, or replaced by the administrator. Hence the HCEA and the HRA support all the algorithms that are provided by CAPI through the pluggable CSPs. For more information on the default list of CSPs that are present on a Windows system, see [MSDN-CSP].

<4> Section 1.5: The Windows implementations of the HCEA and the HRA use the OID "1.3.6.1.4.1.311.47.1.1" (as specified in section 1.9) for the **Statement of Health** field in section 2.2.1.4, and use the OID "1.3.6.1.4.1.311.13.2.2".

<5> Section 1.6: The Windows HCEA depends on general platform security. Strong security is possible using a secure execution environment, but that is outside of the scope of this protocol.

<6> Section 2.2.1.1: The Windows implementation of the HCEA specifies the value of the **User-Agent** field as "NAP IPSec Enforcement v1.0".

<7> Section 2.2.1.1: The Windows implementation of the HRA ignores the user-agent field by default, but can be configured to accept HCEP requests containing only one of a predefined set of user agents.

<8> Section 2.2.1.2: The Windows implementation uses the same Correlation-Id value as provided in SoH, as specified in [MS-SOH] section 2.2.4.

<9> Section 2.2.1.4: The Windows implementation of the HCEA follows the behavior recommended in this section when assigning a value to the subject token.

<10> Section 2.2.1.4: The Windows implementation uses the EKU OID value of "1.3.6.1.4.1.311.47.1.1".

<11> Section 2.2.1.4: The Windows implementation uses EKU OID value of "id-kp-clientAuth" as specified in [RFC3280] section 4.2.1.13, if the Boolean flag as specified in section 3.1.1 is set to TRUE.

<12> Section 2.2.1.4: The Windows implementation uses OID "1.3.6.1.4.1.311.13.2.2" as a certificate extension in the certificate request.

<13> Section 2.2.2.2: The values of the **HCEP-AFW-Protection-Level** and **HCEP-AFW-Zone** fields are set as metadata on the health certificates received from the HRA. Windows Vista has a component that behaves as an IPsec (for more information, see [MSFT-IPSEC]) policy source. The hint provided by the value of **HCEP-AFW-Zone** is used to select one of the preconfigured IPsec policies on the computer. The value of the **HCEP-AFW-Zone** field is ignored if it is not "1", "2", or "3".

<14> Section 2.2.2.2: The Windows implementation of the HRA does not implement any response header fields other than the fields mentioned in this document.

<15> Section 2.2.2.3: Regarding the HTTP message body in the HCEP response in the Windows implementation: If the HCEP client is compliant with health policies, HRA received an SoHR from the health policy server, and HRA receives a certificate response (PKCS #7) from CA, then HRA returns the certificate response in the HTTP message body. Otherwise, if the HCEP client is not compliant with health policies, then no message body is present in the HCEP response.

<16> Section 3.1.1: The Windows implementation of the HCEA does not discard the response if the HCEP-Correlation-Id in the HCEP response mismatches the one sent in the corresponding Health Certificate Enrollment Protocol request.

<17> Section 3.1.1: The Windows implementation of the HCEA determines the value of this Boolean flag to be TRUE if the client computer is a part of an **Active Directory domain**; otherwise, the value of this flag is FALSE.

<18> Section 3.1.4: The Windows implementation of the HCEA re-enrolls for a health certificate when it determines a configuration change has occurred with respect to HRA URLs.

<19> Section 3.1.4: The Windows implementation has an empty list by default.

<20> Section 3.1.5.1: The Windows implementation of the HCEA obtains the SoH from the Windows implementation of the system health entity and sets the SoH certificate extension. For more information, see [MSDN-NAP].

<21> Section 3.1.5.2: The Windows implementation of the HCEA does not validate the values of the HCEP header fields.

<22> Section 3.1.8: The Windows implementation of the HCEA sends a Health Certificate Enrollment Protocol request to the configured servers when an IP address change is detected on any of the computer's interfaces and when a previously unreachable request is made to the servers.

<23> Section 3.2.1: The Windows implementation allows any of these settings; however, it does not contain any predefined settings on the server by default.

<24> Section 3.2.1: The Windows implementation has an empty list by default. Example value: "NAP IPSec Enforcement v1.0".

<25> Section 3.2.1: The Windows implementation has an empty list by default. Only OIDs are configured in the list. The name is provided in the table below for ease of readability.

Name	OID
RSA	1.2.840.113549.1.1.1
DSA	1.2.840.10040.4.1
DH	1.2.840.10046.2.1
RSASSA-PSS	1.2.840.113549.1.1.10
DSA	1.3.14.3.2.12
DH	1.2.840.113549.1.3.1
RSA_KEYX	1.3.14.3.2.22
mosaicKMandUpdSig	2.16.840.1.101.2.1.1.20
ESDH	1.2.840.113549.1.9.16.3.5
NO_SIGN	1.3.6.1.5.5.7.6.2
ECC	1.2.840.10045.2.1
ECDSA_P256	1.2.840.10045.3.1.7
ECDSA_P384	1.3.132.0.34
ECDSA_P521	1.3.132.0.35
RSAES_OAEP	1.2.840.113549.1.1.7
ECDH_STD_SHA1_KDF	1.3.133.16.840.63.0.2

<26> Section 3.2.1: The Windows implementation has an empty list by default. Only OIDs are configured in the list. The name is provided in the table below for ease of readability.

Name	OID
sha1RSA	1.2.840.113549.1.1.5
md5RSA	1.2.840.113549.1.1.4
sha1DSA	1.2.840.10040.4.3
sha1RSA	1.3.14.3.2.29
shaRSA	1.3.14.3.2.15
md5RSA	1.3.14.3.2.3
md2RSA	1.2.840.113549.1.1.2
md4RSA	1.2.840.113549.1.1.3
md4RSA	1.3.14.3.2.2
md4RSA	1.3.14.3.2.4

Name	OID
md2RSA	1.3.14.7.2.3.1
sha1DSA	1.3.14.3.2.13
dsaSHA1	1.3.14.3.2.27
mosaicUpdatedSig	2.16.840.1.101.2.1.1.19
sha1NoSign	1.3.14.3.2.26
md5NoSign	1.2.840.113549.2.5
sha256NoSign	2.16.840.1.101.3.4.2.1
sha384NoSign	2.16.840.1.101.3.4.2.2
sha512NoSign	2.16.840.1.101.3.4.2.3
sha256RSA	1.2.840.113549.1.1.11
sha384RSA	1.2.840.113549.1.1.12
sha512RSA	1.2.840.113549.1.1.13
RSASSA-PSS	1.2.840.113549.1.1.10
sha1ECDSA	1.2.840.10045.4.1
sha256ECDSA	1.2.840.10045.4.3.2
sha384ECDSA	1.2.840.10045.4.3.3
sha512ECDSA	1.2.840.10045.4.3.4
specifiedECDSA	1.2.840.10045.4.3

<27> Section 3.2.1: The Windows implementation has an empty list by default.

Example CSPs:

- Microsoft Base Cryptographic Provider v1.0
- Microsoft Base DSS and Diffie-Hellman Cryptographic Provider
- Microsoft Base DSS Cryptographic Provider
- Microsoft Base Smart Card Crypto Provider
- Microsoft DH SChannel Cryptographic Provider
- Microsoft Enhanced Cryptographic Provider v1.0
- Microsoft Enhanced DSS and Diffie-Hellman Cryptographic Provider
- Microsoft Enhanced RSA and AES Cryptographic Provider
- Microsoft Exchange Cryptographic Provider v1.0

- Microsoft RSA SChannel Cryptographic Provider
- Microsoft Strong Cryptographic Provider

<28> Section 3.2.1: The Windows implementation restricts maximum size to 64 KB by default.

<29> Section 3.2.3.1: The Windows implementation of the HRA has the following behavior with respect to optional requirements listed in this section.

- Maximum size: 64 KB by default. This can be configured to a different value.
- Accept all user agents by default. This behavior can be restricted to a configured list of user agents by specifying a substring of the user-agent values. Wildcards are not allowed in the list of user agents.
- Accept all public key algorithms by default. This behavior can be restricted to a configured list of public key algorithms.
- Accept all signature algorithms by default. This behavior can be restricted to a configured list of signature algorithms.
- Accept all CSPs by default. This behavior can be restricted to a configured list of CSPs.

7 Index

T

V

W