Key Management Interoperability Protocol HTTPS Profile Version 1.0

Working Draft 04

27 June 2012

Technical Committee:
OASIS Key Management Interoperability Protocol (KMIP) TC

Chairs:
Robert Griffin (robert.griffin@rsa.com), EMC Corporation
Subhash Sankuratripati (Subhash.Sankuratripati@netapp.com), NetApp

Editors:
Tim Hudson (tjh@cryptsoft.com), Cryptsoft

Related work:

This specification is related to:

- *Key Management Interoperability Protocol Profiles Version 1.0*. 01 October 2010. OASIS Standard. http://docs.oasis-open.org/kmip/profiles/v1.0/os/kmip-profiles-1.0-os.html.
- *Key Management Interoperability Protocol Specification Version 1.1*. Latest version. http://docs.oasis-open.org/kmip/spec/v1.1/kmip-spec-v1.1.html
- *Key Management Interoperability Protocol Use Cases Version 1.1*. Latest version. http://docs.oasis-open.org/kmip/usecases/v1.1/kmip-usecases-v1.1.html
- *Key Management Interoperability Protocol Usage Guide Version 1.1*. Latest version. http://docs.oasis-open.org/kmip/ug/v1.1/kmip-ug-v1.1.html

Abstract:

Describes an HTTP encoding alternative to the raw TTLV encoding.

Status:

This Working Draft (WD) has been produced by one or more TC Members; it has not yet been voted on by the TC or approved as a Committee Draft (Committee Specification Draft or a Committee Note Draft). The OASIS document Approval Process begins officially with a TC vote to approve a WD as a Committee Draft. A TC may approve a Working Draft, revise it, and re-approve it any number of times as a Committee Draft.

Table of Contents

1 Introduction

For normative definition of the elements of KMIP see the KMIP Specification ([KMIP-SPEC-1_0 and KMIP-SPEC-1_1]) and the KMIP Profiles ([KMIP-PROF]).

Illustrative guidance for the implementation of KMIP clients and servers is provided in the KMIP Usage Guide ([KMIP-UG]) and KMIP Use Cases ([KMIP_UC]).

This profile defines the necessary encoding rules for the transport of KMIP TTLV messages via the Hypertext Transfer Protocol ([RFC2616]) over TLS as specified in HTTP over TLS ([RFC2818]).

1.1 Terminology

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].

1.2 Normative References

[RFC1945] T. Berners-Lee, R. Fielding, H. Frystyk, *Hypertext Transfer Protocol -- HTTP/1.0*, http://www.ietf.org/rfc/rfc1945.txt, IETF RFC 1945, May 1996.

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

[RFC2246] T. Dierks and C. Allen, *The TLS Protocol, Version 1.0*, IETF RFC 2246, Jan 1999, http://www.ietf.org/rfc/rfc2246.txt

[RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol -- HTTP/1.1*, http://www.ietf.org/rfc/rfc2616.txt, IETF RFC 2616, June 1999.

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

[KMIP-SPEC-1_0] OASIS Standard, Key Management Interoperability Protocol Specification Version 1.0, October 2010, http://docs.oasis-open.org/kmip/spec/v1.0/os/kmip-spec-1.0-os.doc

[KMIP-SPEC-1_1] *Key Management Interoperability Protocol Specification Version 1.1.* http://docs.oasis-open.org/kmip/spec/v1.1/csd01/kmip-spec-v1.1-csd01.doc Committee Specification Draft 01.1 December 2011.

[KMIP-PROF] *Key Management Interoperability Protocol Usage Guide Version 1.1*. 01 December 2011. OASIS Standard. http://docs.oasis-open.org/kmip/profiles/v1.1/cd01/kmip-profiles-1.1-cd-01.doc

1.3 Non-Normative References

[KMIP-UG] *Key Management Interoperability Protocol Usage Guide Version 1.1.* http://docs.oasis-open.org/kmip/ug/v1.1/kmip-ug-v1.1-cnd01.doc Committee Note Draft, 1 December 2011,

[KMIP-TC] *Key Management Interoperability Protocol Test Cases Version 1.1.* http://docs.oasis-open.org/kmip/usecases/v1.1/kmip-usecases-v1.1-cnd01.doc, Committee Note Draft, 1 December 2011.

2 HTTPS Profile

The Hypertext Transfer Protocol over Transport Layer Security (HTTPS) is simply the use of HTTP over TLS in the same manner that HTTP is used over TCP.

KMIP over HTTPS is simply the use of KMIP messages over HTTPS in the same manner than KMIP is used over TLS.

2.1 Authentication Suite

Implementations conformant to this profile SHALL support one or more of the Authentication Suites defined within section 3 of [KMIP-PROF]. The establishment of the trust relationship between the KMIP client and the KMIP server is same as the defined base profiles.

2.2 KMIP Port Number

KMIP servers conformant to this profile MAY use TCP port number 5696, as assigned by IANA, to receive and send KMIP messages provided that both HTTP and non-HTTP encoded messages are supported.

KMIP clients SHALL enable end user configuration of the TCP port number used, as a KMIP server may specify a different TCP port number.

2.3 Request URI

KMIP servers conformant to this profile SHOULD support the value /kmip as the target URI.

KMIP clients SHALL enable end user configuration of the target URI used as a KMIP server may specify a different target URI.

2.4 HTTP Encoding

KMIP client implementations conformant to this profile SHALL:

1. Support HTTP/1.0 and/or HTTP/1.1 over TLS conformant to [RFC2818]
2. Use the POST request method
3. Specify a Content-Type of “application/octet-stream”
4. Specify a Content-Length
5. Specify a Cache-Control of “no-cache”
6. Send KMIP TTLV message in binary format as the body of the HTTP request

KMIP server implementations conformant to this profile SHALL:

1. Support HTTP/1.0 and HTTP/1.1 over TLS conformant to [RFC2818]
2. Return HTTP response code 200 if a KMIP response is available
3. Specify a Content-Type of “application/octet-stream”
4. Specify a Content-Length
5. Specify a Cache-Control of “no-cache”
6. Send KMIP TTLV message in binary format as the body of the HTTP request

KMIP servers that support server to client operations shall behave as an HTTPS client. KMIP clients that support responding to server to client operations shall behave as a HTTPS server.

3 HTTPS Profile Test Cases

This section contains a test case that demonstrates the HTTPS profile encoding using test case 12.1 from [KMIP-TC] using protocol version 1.0 which exercises the Query operation and the Maximum Response Size header field.

3.1 Query, Maximum Response Size

3.1.1 Test Case: Query, Maximum Response Size

Perform a Query operation, querying the Operations and Objects supported by the server, with a restriction on the Maximum Response Size set in the request header. Since the resulting Query response is too big, an error is returned. Increase the Maximum Response Size, resubmit the Query request, and get a successful response.

Time	Request/Response messages

0	Query (operations, objects)
	In (header): maximumResponseSize='256'
	In: queryFunctions={ '00000001', '00000002' }
	
	Tag: REQUEST_MESSAGE (0x420078), Type: STRUCTURE (0x01), Data:
	  Tag: REQUEST_HEADER (0x420077), Type: STRUCTURE (0x01), Data:
	  Tag: PROTOCOL_VERSION (0x420069), Type: STRUCTURE (0x01), Data:
	  Tag: PROTOCOL_VERSION_MAJOR (0x42006a), Type: INTEGER (0x02), Data: 0x00000001
	  Tag: PROTOCOL_VERSION_MINOR (0x42006b), Type: INTEGER (0x02), Data: 0x00000000
	  Tag: MAXIMUM_RESPONSE_SIZE (0x420050), Type: INTEGER (0x02), Data: 0x00000100
	  Tag: BATCH_COUNT (0x42000d), Type: INTEGER (0x02), Data: 0x00000001
	  Tag: BATCH_ITEM (0x42000f), Type: STRUCTURE (0x01), Data:
	  Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000018 (QUERY)
	  Tag: REQUEST_PAYLOAD (0x420079), Type: STRUCTURE (0x01), Data:
	  Tag: QUERY_FUNCTION (0x420074), Type: ENUMERATION (0x05), Data: 0x00000001 (QUERY_OPERATIONS)
	  Tag: QUERY_FUNCTION (0x420074), Type: ENUMERATION (0x05), Data: 0x00000002 (QUERY_OBJECTS)
	
	42007801000000904200770100000048420069010000002042006a020000000400000001000000004200 6b020000000400000000000000004200500200000004000001000000000042000d0200000004 0000000100000000042000f01000000384200 5c050000000400000018000000004200790100000020 4200740500000004000000010000000042007405000000040000000200000000
	
	0000 - 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f POST /kmip HTTP/
	0010 - 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d 1.0..Pragma: no-
	0020 - 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e cache..Cache-Con
	0030 - 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a trol: no-cache..
	0040 - 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70 Connection: keep
	0050 - 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d -alive..Content-
	0060 - 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f Type: applicatio
	0070 - 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a n/octet-stream..
	0080 - 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20 Content-Length:
	0090 - 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00 152 ....B.
	00a0 - 78 01 00 00 00 90 42 00-77 01 00 00 00 48 42 00 x.....B.w....HB.
	00b0 - 69 01 00 00 00 20 42 00-6a 02 00 00 00 04 00 00 i.... B.j.......
	00c0 - 00 01 00 00 00 00 42 00-6b 02 00 00 00 04 00 00 ......B.k.......
	00d0 - 00 00 00 00 00 00 42 00-50 02 00 00 00 04 00 00 ......B.P.......
	00e0 - 01 00 00 00 00 00 42 00-0d 02 00 00 00 04 00 00 ......B.........
	00f0 - 00 01 00 00 00 00 42 00-0f 01 00 00 00 38 42 00 ......B......8B.
	0100 - 5c 05 00 00 00 04 00 00-00 18 00 00 00 00 42 00 \.............B.
	0110 - 79 01 00 00 00 20 42 00-74 05 00 00 00 04 00 00 y.... B.t.......
	0120 - 00 01 00 00 00 00 42 00-74 05 00 00 00 04 00 00 ......B.t.......
	0130 - 00 02 00 00 00 00

Out: Operation Failed, Response Too Large

```
Tag: RESPONSE_MESSAGE (0x42007b), Type: STRUCTURE (0x01), Data:
  Tag: RESPONSE_HEADER (0x42007a), Type: STRUCTURE (0x01), Data:
    Tag: PROTOCOL_VERSION (0x420069), Type: STRUCTURE (0x01), Data:
      Tag: PROTOCOL_VERSION_MAJOR (0x42006a), Type: INTEGER (0x02), Data:
0x00000001
      Tag: PROTOCOL_VERSION_MINOR (0x42006b), Type: INTEGER (0x02), Data:
0x00000000
    Tag: TIME_STAMP (0x420092), Type: DATE_TIME (0x09), Data: 0x000000004feafb9f
Wed Jun 27 22:25:03 2012
    Tag: BATCH_COUNT (0x42000d), Type: INTEGER (0x02), Data: 0x00000001
  Tag: BATCH_ITEM (0x42000f), Type: STRUCTURE (0x01), Data:
    Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000018 (QUERY)
    Tag: RESULT_STATUS (0x42007f), Type: ENUMERATION (0x05), Data: 0x00000001
(OPERATION_FAILED)
    Tag: RESULT_REASON (0x42007e), Type: ENUMERATION (0x05), Data: 0x00000002
(RESPONSE_TOO_LARGE)
    Tag: RESULT_MESSAGE (0x42007d), Type: TEXT_STRING (0x07), Data: TOO_LARGE

42007B01000000C842007A0100000048420069010000002042006A020000000400000001000000004
2006B020000000400000001000000004200920900000008000000004F9A556B42000D020000000400
0001000000004200 0F01000000704200 7F0500000004000000010000000042007E0500000004000000
020000000042007D07000000435265737 06F6E73652073697A653A203634382C204D6178696D756D20
526573706F6E73652053697A6520696E6469636174656420696E20726571756573743A203235360000
000000

0000 - 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d   HTTP/1.1 200 OK.
0010 - 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61   .Content-Type: a
0020 - 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74   pplication/octet
0030 - 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74   -stream..Content
0040 - 2d 4c 65 6e 67 74 68 3a-20 31 36 38 0d 0a 0d 0a   -Length: 168....
0050 - 42 00 7b 01 00 00 00 a0-42 00 7a 01 00 00 00 48   B.{.....B.z....H
0060 - 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04   B.i.... B.j.....
0070 - 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04   ........B.k.....
0080 - 00 00 00 00 00 00 00 00-42 00 92 09 00 00 00 08   ........B.......
0090 - 00 00 00 00 4f ea fb 4b-42 00 0d 02 00 00 00 04   ....O..KB.......
00a0 - 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 00 48   ........B......H
00b0 - 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00   B.\.............
00c0 - 42 00 7f 05 00 00 00 04-00 00 00 01 00 00 00 00   B...............
00d0 - 42 00 7e 05 00 00 00 04-00 00 00 02 00 00 00 00   B.~.............
00e0 - 42 00 7d 07 00 00 00 09-54 4f 4f 5f 4c 41 52 47   B.}.....TOO_LARG
00f0 - 45 00 00 00 00 00 00 00-                          E.......
```

1

Query (operations, objects)
In (header): maximumResponseSize=’2048’
In: queryFunctions={ ‘00000001’, ‘00000002’}

```
Tag: REQUEST_MESSAGE (0x420078), Type: STRUCTURE (0x01), Data:
  Tag: REQUEST_HEADER (0x420077), Type: STRUCTURE (0x01), Data:
    Tag: PROTOCOL_VERSION (0x420069), Type: STRUCTURE (0x01), Data:
      Tag: PROTOCOL_VERSION_MAJOR (0x42006a), Type: INTEGER (0x02), Data:
0x00000001
      Tag: PROTOCOL_VERSION_MINOR (0x42006b), Type: INTEGER (0x02), Data:
0x00000000
    Tag: MAXIMUM_RESPONSE_SIZE (0x420050), Type: INTEGER (0x02), Data: 0x00000800
    Tag: BATCH_COUNT (0x42000d), Type: INTEGER (0x02), Data: 0x00000001
  Tag: BATCH_ITEM (0x42000f), Type: STRUCTURE (0x01), Data:
    Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000018 (QUERY)
    Tag: REQUEST_PAYLOAD (0x420079), Type: STRUCTURE (0x01), Data:
      Tag: QUERY_FUNCTION (0x420074), Type: ENUMERATION (0x05), Data: 0x00000001
(QUERY_OPERATIONS)
      Tag: QUERY_FUNCTION (0x420074), Type: ENUMERATION (0x05), Data: 0x00000002
(QUERY_OBJECTS)

42007801000000904200770100000048420069010000002042006a020000000400000001000000000
42006b020000000400000000000000004200500200000004000008000000000042000d0200000004
```

```
0000000100000000042000f010000003842005c050000000400000018000000004200790100000020
420074050000000400000001000000004200740500000004000000200000000

0000 - 50 4f 53 54 20 2f 6b 6d-69 70 20 48 54 54 50 2f   POST /kmip HTTP/
0010 - 31 2e 30 0d 0a 50 72 61-67 6d 61 3a 20 6e 6f 2d   1.0..Pragma: no-
0020 - 63 61 63 68 65 0d 0a 43-61 63 68 65 2d 43 6f 6e   cache..Cache-Con
0030 - 74 72 6f 6c 3a 20 6e 6f-2d 63 61 63 68 65 0d 0a   trol: no-cache..
0040 - 43 6f 6e 6e 65 63 74 69-6f 6e 3a 20 6b 65 65 70   Connection: keep
0050 - 2d 61 6c 69 76 65 0d 0a-43 6f 6e 74 65 6e 74 2d   -alive..Content-
0060 - 54 79 70 65 3a 20 61 70-70 6c 69 63 61 74 69 6f   Type: applicatio
0070 - 6e 2f 6f 63 74 65 74 2d-73 74 72 65 61 6d 0d 0a   n/octet-stream..
0080 - 43 6f 6e 74 65 6e 74 2d-4c 65 6e 67 74 68 3a 20   Content-Length:
0090 - 31 35 32 20 20 20 20 20-20 20 0d 0a 0d 0a 42 00   152       ....B.
00a0 - 78 01 00 00 00 90 42 00-77 01 00 00 00 48 42 00   x.....B.w....HB.
00b0 - 69 01 00 00 00 20 42 00-6a 02 00 00 00 04 00 00   i.... B.j.......
00c0 - 00 01 00 00 00 00 42 00-6b 02 00 00 00 04 00 00   ......B.k.......
00d0 - 00 00 00 00 00 00 42 00-50 02 00 00 00 04 00 00   ......B.P.......
00e0 - 08 00 00 00 00 00 42 00-0d 02 00 00 00 04 00 00   ......B.........
00f0 - 00 01 00 00 00 00 42 00-0f 01 00 00 00 38 42 00   ......B......8B.
0100 - 5c 05 00 00 00 04 00 00-00 18 00 00 00 00 42 00   \.............B.
0110 - 79 01 00 00 00 20 42 00-74 05 00 00 00 04 00 00   y.... B.t.......
0120 - 00 01 00 00 00 00 42 00-74 05 00 00 00 04 00 00   ......B.t.......
0130 - 00 02 00 00 00 00                                 ......
```

Out: operations, objects, serverInformation

```
Tag: RESPONSE_MESSAGE (0x42007b), Type: STRUCTURE (0x01), Data:
  Tag: RESPONSE_HEADER (0x42007a), Type: STRUCTURE (0x01), Data:
    Tag: PROTOCOL_VERSION (0x420069), Type: STRUCTURE (0x01), Data:
      Tag: PROTOCOL_VERSION_MAJOR (0x42006a), Type: INTEGER (0x02), Data:
0x00000001
      Tag: PROTOCOL_VERSION_MINOR (0x42006b), Type: INTEGER (0x02), Data:
0x00000000
    Tag: TIME_STAMP (0x420092), Type: DATE_TIME (0x09), Data: 0x000000004feafb9f
Wed Jun 27 22:25:03 2012
    Tag: BATCH_COUNT (0x42000d), Type: INTEGER (0x02), Data: 0x00000001
  Tag: BATCH_ITEM (0x42000f), Type: STRUCTURE (0x01), Data:
    Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000018 (QUERY)
    Tag: RESULT_STATUS (0x42007f), Type: ENUMERATION (0x05), Data: 0x00000000
(SUCCESS)
    Tag: RESPONSE_PAYLOAD (0x42007c), Type: STRUCTURE (0x01), Data:
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000018
(QUERY)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000008
(LOCATE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000014
(DESTROY)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000000a (GET)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000001
(CREATE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000003
(REGISTER)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000000b
(GET_ATTRIBUTES)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000000c
(GET_ATTRIBUTE_LIST)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000000d
(ADD_ATTRIBUTE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000000e
(MODIFY_ATTRIBUTE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000000f
(DELETE_ATTRIBUTE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000012
(ACTIVATE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000013
(REVOKE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000001a (POLL)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000019
(CANCEL)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000009
(CHECK)
```

```
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000011
(GET_USAGE_ALLOCATION)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000002
(CREATE_KEY_PAIR)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000004
(RE_KEY)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000015
(ARCHIVE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000016
(RECOVER)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000010
(OBTAIN_LEASE)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x0000001d
(RE_KEY_KEY_PAIR)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000006
(CERTIFY)
      Tag: OPERATION (0x42005c), Type: ENUMERATION (0x05), Data: 0x00000007
(RE_CERTIFY)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000001
(CERTIFICATE)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000002
(SYMMETRIC_KEY)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000007
(SECRET_DATA)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000003
(PUBLIC_KEY)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000004
(PRIVATE_KEY)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000006
(TEMPLATE)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000008
(OPAQUE_OBJECT)
      Tag: OBJECT_TYPE (0x420057), Type: ENUMERATION (0x05), Data: 0x00000005
(SPLIT_KEY)

42007b010000029042007a0100000048420069010000002042006a020000000400000001000000000
42006b020000000400000000000000004200920900000008000000004feafb9f42000d0200000004
0000000100000000042000f01000023842005c050000000400000018000000042007f0500000004
0000000000000000042007c01000021042005c050000000400000018000000042005c0500000004
0000000800000000042005c050000000400000014000000042005c050000000400000000a00000000
42005c050000000400000001000000042005c050000000400000003000000042005c0500000004
0000000b000000042005c050000000400000000c000000042005c050000000400000000d00000000
42005c050000000400000000e000000042005c050000000400000000f000000042005c0500000004
0000001200000000042005c050000000400000013000000042005c050000000400000001a00000000
42005c050000000400000019000000042005c050000000400000009000000042005c0500000004
0000001100000000042005c050000000400000002000000042005c050000000400000004000000000
42005c050000000400000015000000042005c050000000400000016000000042005c0500000004
0000001000000000042005c050000000400000001d000000042005c050000000400000006000000000
42005c050000000400000007000000042005705000000040000000100000000420057050000000400
0000002000000004200570500000000400000007000000042005705000000040000003000000000
42005705000000040000000400000000420057050000000400000006000000004200570500000004
0000000800000000420057050000000400000005000000000
```

```
0000 - 48 54 54 50 2f 31 2e 31-20 32 30 30 20 4f 4b 0d   HTTP/1.1 200 OK.
0010 - 0a 43 6f 6e 74 65 6e 74-2d 54 79 70 65 3a 20 61   .Content-Type: a
0020 - 70 70 6c 69 63 61 74 69-6f 6e 2f 6f 63 74 65 74   pplication/octet
0030 - 2d 73 74 72 65 61 6d 0d-0a 43 6f 6e 74 65 6e 74   -stream..Content
0040 - 2d 4c 65 6e 67 74 68 3a-20 36 36 34 0d 0a 0d 0a   -Length: 664....
0050 - 42 00 7b 01 00 00 02 90-42 00 7a 01 00 00 00 48   B.{.....B.z....H
0060 - 42 00 69 01 00 00 00 20-42 00 6a 02 00 00 00 04   B.i.... B.j.....
0070 - 00 00 00 01 00 00 00 00-42 00 6b 02 00 00 00 04   ........B.k.....
0080 - 00 00 00 00 00 00 00 00-42 00 92 09 00 00 00 08   ........B.......
0090 - 00 00 00 00 4f ea fb 9f-42 00 0d 02 00 00 00 04   ....O...B.......
00a0 - 00 00 00 01 00 00 00 00-42 00 0f 01 00 00 02 38   ........B......8
00b0 - 42 00 5c 05 00 00 00 04-00 00 00 18 00 00 00 00   B.\.............
00c0 - 42 00 7f 05 00 00 00 04-00 00 00 00 00 00 00 00   B...............
00d0 - 42 00 7c 01 00 00 02 10-42 00 5c 05 00 00 00 04   B.|.....B.\.....
00e0 - 00 00 00 18 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
00f0 - 00 00 00 08 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0100 - 00 00 00 14 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0110 - 00 00 00 0a 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0120 - 00 00 00 01 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
```

```
0130 - 00 00 00 03 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0140 - 00 00 00 0b 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0150 - 00 00 00 0c 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0160 - 00 00 00 0d 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0170 - 00 00 00 0e 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0180 - 00 00 00 0f 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0190 - 00 00 00 12 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
01a0 - 00 00 00 13 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
01b0 - 00 00 00 1a 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
01c0 - 00 00 00 19 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
01d0 - 00 00 00 09 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
01e0 - 00 00 00 11 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
01f0 - 00 00 00 02 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0200 - 00 00 00 04 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0210 - 00 00 00 15 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0220 - 00 00 00 16 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0230 - 00 00 00 10 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0240 - 00 00 00 1d 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0250 - 00 00 00 06 00 00 00 00-42 00 5c 05 00 00 00 04   ........B.\.....
0260 - 00 00 00 07 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
0270 - 00 00 00 01 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
0280 - 00 00 00 02 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
0290 - 00 00 00 07 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
02a0 - 00 00 00 03 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
02b0 - 00 00 00 04 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
02c0 - 00 00 00 06 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
02d0 - 00 00 00 08 00 00 00 00-42 00 57 05 00 00 00 04   ........B.W.....
02e0 - 00 00 00 05 00 00 00 00-                          ........
```

Appendix A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Original HTTPS Profile Proposal:
Alan Frindell, SafeNet, Inc.

Original HTTPS Profile Contributors:
Mathias Björkqvist, IBM