Object	Encoding
Signature Data	Byte String

Table 31: Signature Data Structure

2.1.13 MAC Data

The *MAC Data* is used in requests and responses in cryptographic operations that pass MAC data between the client and the server.

Object	Encoding
MAC Data	Byte String

Table 32: MAC Data Structure

2.1.14 Nonce

A *Nonce* object is a structure (see Table 33) used by the server to send a random value to the client. The Nonce Identifier is assigned by the server and used to identify the Nonce object. The Nonce Value consists of the random data created by the server.

Object	Encoding	REQUIRED
Nonce	Structure	
Nonce ID	Byte String	Yes
Nonce Value	Byte String	Yes

Table 33: Nonce Structure

2.1.15 Correlation Value

The *Correlation Value* is used in requests and responses in cryptographic operations that support multi-part (streaming) operations. This is returned in the first response to an operation that is being performed across multiple requests. Note: the server decides which operations are supported for multi-part usage. A server-generated correlation value SHALL be specified in any subsequent cryptographic operations that pertain to the original operation.

Object	Encoding
Correlation Value	Byte String

Table 34: Correlation Value Structure

2.1.16 Init Indicator

The *Init Indicator* is used in requests and responses in cryptographic operations that support multi-part (streaming) operations. This is provided in the first request with a value of True to an operation that is being performed across multiple requests.

Object	Encoding
Init Indicator	Boolean

Table 35: Init Indicator Structure

2.1.17 Final Indicator

The *Final Indicator* is used in requests and responses in cryptographic operations that support multi-part (streaming) operations. This is provided in the final (last) request with a value of True to an operation that is being performed across multiple requests.

Object	Encoding
Final Indicator	Boolean

Table 36: Final Indicator Structure

Formatte

2.2 Managed Objects

Managed Objects are objects that are the subjects of key management operations, which are described in Sections 4 and 5. *Managed Cryptographic Objects* are the subset of Managed Objects that contain cryptographic material (e.g., certificates, keys, and secret data).

2.2.1 Certificate

A Managed Cryptographic Object that is a digital certificate. It is a DER-encoded X.509 public key certificate. The PGP certificate type is deprecated as of version 1.2 of this specification and MAY be removed from subsequent versions of the specification. The PGP Key object (see section 2.2.9) SHOULD be used instead.

Object	Encoding	REQUIRED
Certificate	Structure	
Certificate Type	Enumeration, see 9.1.3.2.6	Yes
Certificate Value	Byte String	Yes

Table 37: Certificate Object Structure

2.2.2 Symmetric Key

A Managed Cryptographic Object that is a symmetric key.

Object	Encoding	REQUIRED
Symmetric Key	Structure	
Key Block	Structure, see 2.1.3	Yes

Table 38: Symmetric Key Object Structure

2.2.3 Public Key

A Managed Cryptographic Object that is the public portion of an asymmetric key pair. This is only a public key, not a certificate.

Object	Encoding	REQUIRED
Public Key	Structure	
Key Block	Structure, see 2.1.3	Yes

Table 39: Public Key Object Structure

4 Client-to-Server Operations

The following subsections describe the operations that MAY be requested by a key management client. Not all clients have to be capable of issuing all operation requests; however any client that issues a specific request SHALL be capable of understanding the response to the request. All Object Management operations are issued in requests from clients to servers, and results obtained in responses from servers to clients. Multiple operations MAY be combined within a batch, resulting in a single request/response message pair.

A number of the operations whose descriptions follow are affected by a mechanism referred to as the *ID Placeholder.*

The key management server SHALL implement a temporary variable called the ID Placeholder. This value consists of a single Unique Identifier. It is a variable stored inside the server that is only valid and preserved during the execution of a batch of operations. Once the batch of operations has been completed, the ID Placeholder value SHALL be discarded and/or invalidated by the server, so that subsequent requests do not find this previous ID Placeholder available.

The ID Placeholder is obtained from the Unique Identifier returned in response to the Create, Create Pair, Register, Derive Key, Re-key, Re-key Key Pair, Certify, Re-Certify, Locate, and Recover operations. If any of these operations successfully completes and returns a Unique Identifier, then the server SHALL copy this Unique Identifier into the ID Placeholder variable, where it is held until the completion of the operations remaining in the batched request or until a subsequent operation in the batch causes the ID Placeholder to be replaced. If the Batch Error Continuation Option is set to Stop and the Batch Order Option is set to true, then subsequent operations in the batched request MAY make use of the ID Placeholder by omitting the Unique Identifier field from the request payloads for these operations.

Requests MAY contain attribute values to be assigned to the object. This information is specified with a Template-Attribute (see Section 2.1.8) that contains zero or more template names and zero or more individual attributes. If more than one template name is specified, and there is a conflict between the single-instance attributes in the templates, then the value in the last of the conflicting templates takes precedence. If there is a conflict between the single-instance attributes in the request and the single-instance attributes in a specified template, then the attribute values in the request take precedence. For multi-instance attributes, the union of attribute values is used when the attributes are specified more than once.

Responses MAY contain attribute values that were not specified in the request, but have been implicitly set by the server. This information is specified with a Template-Attribute that contains one or more individual attributes.

For any operations that operate on Managed Objects already stored on the server, any archived object SHALL first be made available by a Recover operation (see Section 4.23) before they MAY be specified (i.e., as on-line objects).

Multi-part cryptographic operations (operations where a stream of data is provided across multiple requests from a client to a server) are optionally supported by those cryptographic operations that include the Correlation Value (see section 2.1.15), Init Indicator (see section 2.1.16) and Final Indicator (see section 2.1.17) request parameters.

For multi-part cryptographic operations the following sequence is performed

1. On the first request
    a. Provide an Init Indicator with a value of True
    b. Provide any other required parameters
    c. Preserve the Correlation Value returned in the response for use in subsequent requests
    d. Use the Data output (if any) from the response
2. On subsequent requests

Formatte

a. Provide the Correlation Value from the response to the first request
b. Provide any other required parameters
c. Use the next block of Data output (if any) from the response

3. On the final request
a. Provide the Correlation Value from the response to the first reply
b. Provide a Final Indicator with a value of True
c. Use the final block of Data output (if any) from the response

Single-part cryptographic operations (operations where a single input is provided and a single response returned) the following sequence is performed:

1. On each request
a. Do not provide an Init Indicator, Final Indicator or Correlation Value
b. Provide any other required parameters
c. Use the Data output from the response

4.1 Create

This operation requests the server to generate a new symmetric key as a Managed Cryptographic Object. This operation is not used to create a Template object (see Register operation, Section 4.3).

The request contains information about the type of object being created, and some of the attributes to be assigned to the object (e.g., Cryptographic Algorithm, Cryptographic Length, etc.). This information MAY be specified by the names of Template objects that already exist.

The response contains the Unique Identifier of the created object. The server SHALL copy the Unique Identifier returned by this operation into the ID Placeholder variable.

Request Payload		
Object	**REQUIRED**	**Description**
Object Type, see 3.3	Yes	Determines the type of object to be created.
Template-Attribute, see 2.1.8	Yes	Specifies desired attributes using to be associated with the new object templates and/or individual attributes.

Table 139: Create Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Object Type, see 3.3	Yes	Type of object created.
Unique Identifier, see 3.1	Yes	The Unique Identifier of the newly created object.
Template-Attribute, see 2.1.8	No	An OPTIONAL list of object attributes with values that were not specified in the request, but have been implicitly set by the key management server.

Table 140: Create Response Payload

Table 141 indicates which attributes SHALL be included in the Create request using the Template-Attribute object.

- *Failed* – The pending operation completed with a failure before the cancellation operation was able to cancel it.
- *Unavailable* – The specified correlation value did not match any recently pending or completed asynchronous operations.

The response to this operation is not able to be asynchronous.

Request Payload		
Object	**REQUIRED**	**Description**
Asynchronous Correlation Value, see 6.8	Yes	Specifies the request being canceled.

Table 202: Cancel Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Asynchronous Correlation Value, see 6.8	Yes	Specified in the request.
Cancellation Result, see 9.1.3.2.25	Yes	Enumeration indicating the result of the cancellation.

Table 203: Cancel Response Payload

4.28 Poll

This operation is used to poll the server in order to obtain the status of an outstanding asynchronous operation. The correlation value (see Section 6.8) of the original operation SHALL be specified in the request. The response to this operation SHALL NOT be asynchronous.

Request Payload		
Object	**REQUIRED**	**Description**
Asynchronous Correlation Value, see 6.8	Yes	Specifies the request being polled.

Table 204: Poll Request Payload

The server SHALL reply with one of two responses:

If the operation has not completed, the response SHALL contain no payload and a Result Status of Pending.

If the operation has completed, the response SHALL contain the appropriate payload for the operation. This response SHALL be identical to the response that would have been sent if the operation had completed synchronously.

4.29 Encrypt

This operation requests the server to perform an encryption operation on the provided data using a Managed Cryptographic Object as the key for the encryption operation.

The request contains information about the cryptographic parameters (mode and padding method), the data to be encrypted, and the IV/Counter/Nonce to use. The cryptographic parameters MAY be omitted from the request as they can be specified as associated attributes of the Managed Cryptographic Object. The IV/Counter/Nonce MAY also be omitted from the request if the cryptographic parameters indicate that the server shall generate a Random IV on behalf of the client or the encryption algorithm does not need an IV/Counter/Nonce. The server does not store or otherwise manage the IV/Counter/Nonce.

If the Managed Cryptographic Object referenced has a Usage Limits attribute then the server SHALL obtain an allocation from the current Usage Limits value prior to performing the encryption operation. If the allocation is unable to be obtained the operation SHALL return with a result status of Operation Failed and result reason of Permission Denied.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the encryption operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the encryption operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see 3.6	No	The Cryptographic Parameters (Block Cipher Mode, Padding Method, RandomIV) corresponding to the particular encryption method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed.
Data	Yes	The data to be encrypted (as a Byte String).
IV/Counter/Nonce	No	The initialization vector, counter or nonce to be used (where appropriate).
Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

Table 205: Encrypt Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that was the key used for the encryption operation.
Data	Yes	The encrypted data (as a Byte String).
IV/Counter/Nonce	No	The value used if the Cryptographic Parameters specified Random IV and the IV/Counter/Nonce value was not provided in the request and the algorithm requires the provision of an IV/Counter/Nonce.
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 206: Encrypt Response Payload

4.30 Decrypt

This operation requests the server to perform a decryption operation on the provided data using a Managed Cryptographic Object as the key for the decryption operation.

The request contains information about the cryptographic parameters (mode and padding method), the data to be decrypted, and the IV/Counter/Nonce to use. The cryptographic parameters MAY be omitted from the request as they can be specified as associated attributes of the Managed Cryptographic Object. The initialization vector/counter/nonce MAY also be omitted from the request if the algorithm does not use an IV/Counter/Nonce.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the decryption operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the decryption operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see 3.6	No	The Cryptographic Parameters (Block Cipher Mode, Padding Method) corresponding to the particular decryption method requested. If omitted then the Cryptographic

		Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed.
Data	Yes	The data to be decrypted (as a Byte String).
IV/Counter/Nonce	No	The initialization vector, counter or nonce to be used (where appropriate).
Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

Table 207: Decrypt Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the decryption operation.
Data	Yes	The decrypted data (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 208: Decrypt Response Payload

4.31 Sign

This operation requests the server to perform a signature operation on the provided data using a Managed Cryptographic Object as the key for the signature operation.

The request contains information about the cryptographic parameters (digital signature algorithm or cryptographic algorithm and hash algorithm) and the data to be signed. The cryptographic parameters MAY be omitted from the request as they can be specified as associated attributes of the Managed Cryptographic Object.

If the Managed Cryptographic Object referenced has a Usage Limits attribute then the server SHALL obtain an allocation from the current Usage Limits value prior to performing the signing operation. If the allocation is unable to be obtained the operation SHALL return with a result status of Operation Failed and result reason of Permission Denied.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the signature operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the signature operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see 3.6	No	The Cryptographic Parameters (Digital Signature Algorithm or Cryptographic Algorithm and Hashing Algorithm) corresponding to the particular signature generation method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed.
Data	Yes	The data to be signed (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

Table 209: Sign Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the signature operation.
Signature Data	Yes	The signed data (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 210: Sign Response Payload

4.32 Signature Verify

This operation requests the server to perform a signature verify operation on the provided data using a Managed Cryptographic Object as the key for the signature verification operation.

The request contains information about the cryptographic parameters (digital signature algorithm or cryptographic algorithm and hash algorithm) and the signature to be verified and MAY contain the data that was passed to the signing operation (for those algorithms which need the original data to verify a signature).

The cryptographic parameters MAY be omitted from the request as they can be specified as associated attributes of the Managed Cryptographic Object.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the OPTIONAL data recovered from the signature (for those signature algorithms where data recovery from the signature is supported). The validity of the signature is indicated by the Validity Indicator field.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the signature verify operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see 3.6	No	The Cryptographic Parameters (Digital Signature Algorithm or Cryptographic Algorithm and Hashing Algorithm) corresponding to the particular signature verification method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used.

		If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed.
Data	No	The data that was signed (as a Byte String).
Signature Data	Yes	The signature to be verified (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

Table 211: Signature Verify Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the verification operation.
Validity Indicator, see 9.1.3.2.23	Yes	An Enumeration object indicating whether the signature is valid, invalid, or unknown.
Data	No	The OPTIONAL recovered data (as a Byte String) for those signature algorithms where data recovery from the signature is supported.
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 212: Signature Verify Response Payload

4.33 MAC

This operation requests the server to perform message authentication code (MAC) operation on the provided data using a Managed Cryptographic Object as the key for the MAC operation.

1952 The request contains information about the cryptographic parameters (cryptographic algorithm) and the
1953 data to be MACed. The cryptographic parameters MAY be omitted from the request as they can be
1954 specified as associated attributes of the Managed Cryptographic Object.

1955 The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and
1956 the result of the MAC operation.

1957 The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason)
1958 in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the MAC operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see 3.6	No	The Cryptographic Parameters (Cryptographic Algorithm) corresponding to the particular MAC method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed.
Data	Yes	The data to be MACed (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

1959 *Table 213: MAC Request Payload*

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key

		used for the MAC operation.
MAC Data	Yes	The data MACed (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 214: MAC Response Payload

4.34 MAC Verify

This operation requests the server to perform message authentication code (MAC) verify operation on the provided data using a Managed Cryptographic Object as the key for the MAC verify operation.

The request contains information about the cryptographic parameters (cryptographic algorithm) and the data to be MAC verified and MAY contain the data that was passed to the MAC operation (for those algorithms which need the original data to verify a MAC). The cryptographic parameters MAY be omitted from the request as they can be specified as associated attributes of the Managed Cryptographic Object.

The response contains the Unique Identifier of the Managed Cryptographic Object used as the key and the result of the MAC verify operation. The validity of the MAC is indicated by the Validity Indicator field.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	No	The Unique Identifier of the Managed Cryptographic Object that is the key to use for the MAC verify operation. If omitted, then the ID Placeholder value SHALL be used by the server as the Unique Identifier.
Cryptographic Parameters, see 3.6	No	The Cryptographic Parameters (Cryptographic Algorithm) corresponding to the particular MAC method requested. If omitted then the Cryptographic Parameters associated with the Managed Cryptographic Object with the lowest Attribute Index SHALL be used. If there are no Cryptographic Parameters associated with the Managed Cryptographic Object and the algorithm requires parameters then the operation SHALL return with a Result Status of Operation Failed.
Data	No	The data that was MACed (as a Byte String).

MAC Data	Yes	The data to be MAC verified (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

Table 215: MAC Verify Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Unique Identifier, see 3.1	Yes	The Unique Identifier of the Managed Cryptographic Object that is the key used for the verification operation.
Validity Indicator, see 9.1.3.2.23	Yes	An Enumeration object indicating whether the MAC is valid, invalid, or unknown.
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 216: MAC Verify Response Payload

4.35 RNG Retrieve

This operation requests the server to return output from a Random Number Generator (RNG).

The request contains the quantity of output requested.

The response contains the RNG output.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Data Length	Yes	The amount of random number generator output to be returned (in bytes).

Table 217: RNG Retrieve Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Data	Yes	The random number generator output.

Table 218: RNG Retrieve Response Payload

4.36 RNG Seed

This operation requests the server to seed a Random Number Generator.

The request contains the seeding material.

The response contains the amount of seed data used.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

The server MAY elect to ignore the information provided by the client (i.e. not accept the seeding material) and MAY indicate this to the client by returning zero as the value in the Data Length response. A client SHALL NOT consider a response from a server which does not use the provided data as an error.

Request Payload		
Object	**REQUIRED**	**Description**
Data	Yes	The data to be provided as a seed to the random number generator.

Table 219: RNG Seed Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Data Length	Yes	The amount of seed data used (in bytes).

Table 220: RNG Seed Response Payload

4.37 Hash

This operation requests the server to perform a hash operation on the data provided.

The request contains information about the cryptographic parameters (hash algorithm) and the data to be hashed.

The response contains the result of the hash operation.

The success or failure of the operation is indicated by the Result Status (and if failure the Result Reason) in the response header.

Request Payload		
Object	**REQUIRED**	**Description**
Cryptographic Parameters, see 3.6	Yes	The Cryptographic Parameters (Hashing Algorithm) corresponding to the particular hash method requested.
Data	Yes	The data to be hashed (as a Byte String).

Correlation Value, see 2.1.15	No	Specifies the existing stream or by-parts cryptographic operation (as returned from a previous call to this operation).
Init Indicator, see 2.1.16	No	Initial operation as Boolean (reset state)
Final Indicator, see 2.1.17	No	Final operation as Boolean (reset state)

Table 221: MAC Request Payload

Response Payload		
Object	**REQUIRED**	**Description**
Data	Yes	The hashed data (as a Byte String).
Correlation Value, see 2.1.15	No	Specifies the stream or by-parts value to be provided in subsequent calls to this operation for performing cryptographic operations.

Table 222: HASH Response Payload

4.38 Create Split Key

This operation requests the server to generate a new split key and register all the splits as individual new Managed Cryptographic Objects.

The request contains attributes to be assigned to the objects (e.g., Split Key Parts, Split Key Threshold, Split Key Method, Cryptographic Algorithm, Cryptographic Length, etc.). The request MAY contain the Unique Identifier of an existing cryptographic object that the client requests be split by the server. If the attributes supplied in the request do not match those of the key supplied, the attributes of the key take precedence.

The response contains the Unique Identifiers of all created objects. The ID Placeholder value SHALL be set to the Unique Identifier of the split whose Key Part Identifier is 1.

Request Payload		
Object	**REQUIRED**	**Description**
Object Type, see 3.3	Yes	Determines the type of object to be created.
Unique Identifier, see 3.1	No	The Unique Identifier of the key to be split (if applicable).
Split Key Parts	Yes	The total number of parts.
Split Key Threshold	Yes	The minimum number of parts needed to reconstruct the entire key.
Split Key Method	Yes	

Tag	
Object	**Tag Value**
Attestation Measurement	4200CB
Attestation Assertion	4200CC
IV Length	4200CD
Tag Length	4200CE
Fixed Field Length	4200CF
Counter Length	4200D0
Initial Counter Value	4200D1
Invocation Field Length	4200D2
Correlation Value	4200D5
Init Indicator	4200D6
Final Indicator	4200D7
(Reserved)	~~4200D3~~ 4200D8 – 42FFFF
(Unused)	430000 – 53FFFF
Extensions	540000 – 54FFFF
(Unused)	550000 - FFFFFF

Table 254: Tag Values

9.1.3.2 Enumerations

The following tables define the values for enumerated lists. Values not listed (outside the range 80000000 to 8FFFFFFF) are reserved for future KMIP versions.

9.1.3.2.1 Credential Type Enumeration

Credential Type	
Name	**Value**
Username and Password	00000001
Device	00000002
Attestation	00000003
Extensions	8XXXXXXX

Table 255: Credential Type Enumeration