##### Proposal for Supporting Client System Measurement within KMIP

Created by: Kelley Burgin, NSA

July 11, 2012

Version 0.1

***Reason for this proposal***

In some scenarios the KMIP server may want assurance of the integrity of the client’s system before returning the requested Managed Object (e.g. TCG integrity reports). Additionally, the server may want a guarantee of the freshness of the integrity measurement. Note: think of “measurement” as a hash of some portion of the client system OS, and the “assurance” is there’s no malware on the client’s system.

Generally, the process takes four passes:

1. The client sends a message to the server that it wants to request a Managed Object
2. The server returns a random nonce to the client
3. The client sends a request for a Managed Object to the server which includes a measurement of the client’s system, and the measurement contains the nonce from the server
4. If the measurement checks out, the server returns the Managed Object

***Proposed Solution***

I recognize up front that there are many ways to solve this problem. I am just putting one out there as a starting point, hoping to get feedback/input from others in the TC. One solution, as described in this proposal, is two define a new object, a new credential, and two new Client to Server operations.

Some points that are not addressed in this proposal, but will need to be eventually, are how the server will know to require a measurement before returning the Managed Object (does the Object have an Attribute that says so?) and what types of errors to return (what if the client issues a standard *Get* request for the Object? What if the measurement doesn’t check out?).

#### New Base Object (Section 2.1) Challenge

To provide the guarantee of freshness, the server needs to supply the client with a random nonce to be included with the request for a Managed Object. This proposal defines a *Challenge Object* which is returned by a server to a client in the response to a *Get Challenge* request described in the next paragraph. A *Challenge Object* contains a random nonce viewed as a byte string that the client will include in afuture request to guarantee the freshness of the information in the request. The server will also need to be able to track the time when a *Challenge Object* was sent to a client, with the expectation that it will be returned by the Client in a request within some small amount of time since the point is to provide a guarantee of freshness of the measurement. It is assumed that the Time Stamp in the message header will serve this purpose. If not, something like Lease Time could be used.

|  |  |  |
| --- | --- | --- |
| Object | Encoding | REQUIRED |
| Challenge | Structure |  |
| Nonce | Byte String | Yes |

Table : Challenge Object Structure

#### New Client to Server Operation (Section 4) Get Challenge

This proposal also defines two new Client to Server Operations, the *Get Challenge* request and the *Get with Measurement* request. The *Get Challenge* requests the server send a *Challenge Object* to the client to be included in a future *Get with Measurement* request. To tie the two requests together, optional fields for *Unique Identifier* and *Attribute Name* are included in the *Get Challenge* request to specifiy which key will be requested with a *Get with Measurement* request.

|  |
| --- |
| Request Payload |
| Object | REQUIRED | Description  |
| Unique Identifier, see 3.1 | No | Determines the object that will be requested with the challenge, to tie together client requests. |
| Attribute Name, see 2.1.1 | No |  |

Table 2: Get Challenge Request Payload

|  |
| --- |
| Response Payload |
| Object | REQUIRED | Description  |
| Object Type, see 3.3 | Yes | Type of object.  |
| Challenge Object, see 2.1 | Yes | The Challenge Object being returned. |

Table 3: Get Challenge Response Payload

#### New Credential (Section 2.1.2) Measurement

The Client will include either a measurement of its system or an Assertion from a third party in the Measurement Credential Object. The measurement data (e.g. TCG integrity report) will include a recently obtained Challenge Object. An Assertion may be an “Attestation Assertion” from a trusted appraisor and will include the Challenge Object when passed back to the server. Instead of specifying a particular format the Measurement data and Assertion should have, it is specified only as a byte string for the server to interpret however it wants to allow for different uses of the Measurement Credential Object.

|  |  |  |
| --- | --- | --- |
| Object | Encoding | REQUIRED |
| Credential Value | Structure |  |
| Measurement | Byte String | No |
| Assertion | Byte String | No |

Table 4: Credential Value Structure for the Measurement Credential

#### New Client to Server Operation (Section 4) Get with Measurement

The *Get with Measurement* request is essentially a *Get* request with an additional required field that contains the measurement data or assertion encapsulated in a Meaurement Credential Object.

|  |
| --- |
| Request Payload |
| Object | REQUIRED | Description  |
| Unique Identifier, see 3.1 | No | Determines the object being requested. If omitted, then the ID Placeholder value is used by the server as the Unique Identifier. |
| Key Format Type, see 9.1.3.2.3 | No | Determines the key format type to be returned. |
| Key Compression Type, see 9.1.3.2.2 | No | Determines the compression method for elliptic curve public keys. |
| Key Wrapping Specification, see 2.1.6 | No | Specifies keys and other information for wrapping the returned object. This field SHALL NOT be specified if the requested object is a Template. |
| Credential, see 2.1.2 | Yes | Measurement or Assertion data which includes a recently obtained Challenge Object. |

Table 5: Get With Measurement Request Payload

|  |
| --- |
| Response Payload |
| Object | REQUIRED | Description  |
| Object Type, see 3.3 | Yes | Type of object.  |
| Unique Identifier, see 3.1 | Yes | The Unique Identifier of the object. |
| Certificate, Symmetric Key, Private Key, Public Key, Split Key, Template, Secret Data, or Opaque Object, see 2.2 | Yes | The cryptographic object being returned. |

Table 6: Get With Measurement Response Payload