For further information, contact:
xbulk@research.baltimore.com

XML Key Management Specification Bulk Operation (X-BULK)

Baltimore Technologies
Gemplus
Oberthur Card Systems
Schlumberger

Draft Version 1.1 draft 4: July 6 2001

Table Of Contents

Executive Summary

This document extends the [XKMS] protocol to encompass the bulk registration operations necessary for interfacing with such systems as smart card management systems.

X-BULK is defined in terms of structures expressed in the XML Schema Language and web services description language [WSDL].

1 Introduction

1.1 Overview

XKMS currently addresses one-by-one registration (X-KRSS) and key information and validation services (X-KISS). However, we feel that a standard must also address bulk issuance cases and are proposing that an X-BULK specification, built on the basis of X-KRSS be included in scope of the work.

The use cases where X-BULK is required include:

- Smart card factories for enterprise, wireless and cable-modem applications
- Device factories in general (e.g. TCPA-like TPM modules)
- To handle functionality analogous to separated RAs and CAs from the X.509 world

Key differences between X-KRSS and X-BULK include:

- X-BULK is required to correlate batches of requests and responses.
- X-KRSS doesn't support some legacy key registration formats (e.g. PKCS#10), which are important for existing hardware modules.
- Authentication and response profiling should be at the level of the batch, not the individual request.
- Batch status is not the same as key status.
- X-BULK addresses interfacing with card administration and deployment back-end servers (a.k.a. card management systems).

X-BULK does however reuse element definitions from the current X-KRSS specification.

Separating bulk from one-by-one registration has the benefit that the separately defined messages required are simpler than if a single message format handling both one-by-one and bulk cases were to be defined. It is also better not to burden a client for one-by-one operation with the additional complexity required in batch operation.

Demand for this functionality is shown by the emergence of a number of proprietary solutions in this space.

Design criteria include:

- Consistency with and reuse of [XKMS]
- To handle batches of register requests and responses
- To handle legacy cryptographic formats prevalent in current hardware environments, in particular, [PKCS#10] and [PKCS#1] as well as KeyInfo from [XMLDSIG]
- To handle both client and server side key generation
- Simplicity and flexibility

1.2 Namepaces

For clarity, some examples of XML are not complete documents and namespace declarations may be omitted from XML fragments. In this document, certain namespace prefixes represent certain namespaces. References to XML schema defined herein use the prefix "xbulk" and are in the namespace:

xmlns:xbulk=http://www.xmltrustcenter.org/xkms/docs/x-bulk-2001-07-04.xsd

This specification uses the elements already defined in the XKMS and XML Signature namespaces. The XKMS namespace is represented by the prefix `s0` and is declared as xmlns:s0=http://www.xmltrustcenter.org/xml/schema/2000-11-12-XKMS.sdl.

The XML Signature namespace is represented by the prefix `ds` and is declared as xmlns:ds=http://www.w3.org/2000/09/xmldsig#. The XML Signature schema is defined in http://www.w3.org/2000/09/xmldsig, and the `<ds:KeyInfo>` element (and all of its contents) are found at http://www.w3.org/2000/09/xmldsig#sec-KeyInfo. The corresponding XML Schema definition can be found in xmldsig-core-schema.xsd.

1.3 Bulk Operations Service Specification

X-BULK defines a batch element that can contain registration requests, responses and status requests. The basic idea is that a single batch can contain a number of independently referencable requests or responses. Batches are produced both from the requestor and responder. A responder will process an entire batch and produce a single batch of responses after processing.

All batches MUST be authenticated; that is, the originator of a batch must be able to protect the batch. Implementations MUST support the RSA algorithm for digitally signing batches. Implementations MUST be capable of using the XKMS Register operation for the registration of the keys used to authenticate batches.

Implementations MUST include an X509Data element in the Signature, which is used together with the BatchID to ensure batch uniqueness. All such Signature elements MUST be enveloped signatures, that is, they MUST authenticate the entire X-BULK message.

The basic mode of operation is that a batch of requests is submitted. The responder processes the batch and produces a response batch that contains one response for each request in the batch. Other, more flexible modes of operation may be defined later (e.g. allowing responses to be spread over multiple batches). This mode of "full batch processing" is sufficient for most use cases and is considerably simpler than supporting "selective batch processing."

Each batch has a header that identifies the batch (and can contain additional unprocessed information).

In order to allow the requestor to track the progress of batch processing implementations MAY support status requests. A status request is a request to determine the status of processing of the referenced batch. The response gives a simple indication of the numbers of requests from the batch that are in the various possible states (processed, failed, etc.).

A batch response contains one response for each request, not necessarily in the same order as in the request batch. That is, requestors MUST be able to handle responses that are not sorted in any particular way.

In many use cases, the requestor requires "additional information" to be "carried around" with a batch or request, but which is not intended for processing by the responder.

Responders MAY also add more additional information to the specific responses. Requestors MUST be able to handle such additional information.

The basic request and response structures are as in [XKMS] and can support the same level of functionality (registration of new keys, local/central key generation, revocation, etc.).

1.4 Structure of this document

The remainder of this document describes the X-BULK messages, first the "outside" elements, then the request specifics, the status messages and finally the response elements. Finally the complete schema for X-BULK and the WSDL definition are presented.

2 BatchHeader

The header is common to all the elements in this document; it will optionally contain the following information:

- A Batch ID
- A Batch Creation Date
- ProcessInfo contains additional processing information for the responder.

BatchID combined with the originating party X509Data (from the Signature) MUST be unique at a given moment in time.

Things like customer name, that are not actually required for the batch to be processed, MAY be included as <ProcessInfo>.

```
<BatchHeader>
      <BatchID>1234</BatchID>
      <BatchTime>22 May 2001</BatchTime>
       <ProcessInfo>
        <CustomerName>ACME, Inc.</CustomerName>
      </ProcessInfo>
</BatchHeader>
```

3 BulkRegister

The <BulkRegister> is an XML element that will consist of a <BatchHeader>, a response profile, a sequence of requests and <Signature> over the header and requests.

3.1 ResponseProfile - <Respond>

The format of the response to be sent for each request that does not contain an individual <Respond> element. The <Respond> can, for example, indicate that the requests are for:

- X509Cert (an X.509 certificate)
- RetrievalMethod (Certificate URLs (LDAP or HTTP))

The full set of allowed <Respond> values is defined in [XKMS]. Implementations MUST support: KeyName, KeyValue, X509Cert, X509Chain, RetrievalMethod and Private.

3.2 Requests

<Requests> consists of an unbounded number of <Request> elements. The number of <Request> elements in a <Requests> MUST be an attribute to <Requests>. A request MUST also contain the following:

- A <KeyID>, which must be unique within the batch (this can anything; examples include a smart card serial number , a MAC Address (cable modem), ICCID (used by S/WIM cards) or even simply 1,2,3…).

- <KeyInfo>

A <Request> may also contain <Respond>, <ClientInfo>, and <ProcessInfo> elements. <ClientInfo>, if present, MUST be ignored and treated as opaque data by the Responder. This element must be returned in the corresponding <RegisterResult> element. This can be used by the client for bookkeeping.

3.2.1 KeyInfo

The <KeyInfo> MUST contain a PKCS#10 request, or an X509SubjectName and key data (in which case the key data can either be a <ds:KeyValue> or a PKCS#1 public key), or an <X509SubjectName> without key data (server generated keys).

In addition, the following KeyInfo types from [XMLDSIG] SHOULD be supported: KeyName, KeyValue, RetrievalMethod, and X509Data.

3.3 Example

```
<BulkRegister>
   <SignedPart ID="top">

   <BatchHeader>
     <BatchID>1234</BatchID>
     <BatchTime>22 May 2001</BatchTime>

     <ProcessInfo>
       <CustomerName>ACME, Inc.</CustomerName>
     </ProcessInfo>
   </BatchHeader>

   <Respond>
     <string>RetrievalMethod</string>
   </Respond>

   <Requests number=2>
     <Request>
       <KeyID>1</KeyID>
       <ds:KeyInfo>
         <ds:X509SubjectName>
           cn=Niall,o=Baltimore,c=IE
         </ds:X509SubjectName>
         <PKCS1>MMAsDASFDAsfBarQW4SDGjSDfbgwEJRH=</PKCS1>
       </ds:KeyInfo>
       <ProcessInfo>
         <foo>bar</foo>
       </ProcessInfo>
       <ClientInfo>hello</ClientInfo>
     </Request>

     <Request>
       <KeyID>2</keyID>
       <ds:KeyInfo>
         <PKCS10>MMAsDASFDAsfBarQW4SDGjSDfbgwEJRH=</PKCS10>
```

```
          </ds:KeyInfo>
          <ProcessInfo>
            <foo>baz</foo>
          </ProcessInfo>
          <ClientInfo>goodbye</ClientInfo>
        </Request>
      </Requests>

      </SignedPart>

      <Signature>
        <!-- Stuff -->
         ...
         <Reference URI="#top">
         ...
      </Signature>
   </BulkRegister>
```

4 BulkStatus

4.1 BulkStatusRequest

The <BulkStatusRequest> is a request for the current status of a batch. The batch is identified by its header. The element consists of a <BatchHeader> and <Signature>.

The <BatchID> MUST be exactly the same as in the header of the corresponding <BulkRegister> element.

4.2 Example

```
   <BulkStatusRequest>
      <SignedPart ID="top">

       <BatchHeader>
         <BatchID>1234</BatchID>
         <BatchTime>22 May 2001</BatchTime>
         <ProcessInfo>
           <CustomerName>ACME, Inc.</CustomerName>
         </ProcessInfo>
       </BatchHeader>

      </SignedPart>
      <Signature>
        <!-- Stuff -->
         ...
         <Reference URI="#top">
         ...
      </Signature>
   </BulkStatusRequest>
```

4.3 BulkStatus

The <BulkStatus> is a very simple element that contains the <BatchHeader> and the <Status> of a batch. The <BatchHeader> MUST be exactly the same as the header in the <BulkRegister> element. This will contain the following:

- The number of requests pending
- The number that have completed successfully
- The number that have failed

One application might be to use a XML style sheet to present this information to the customer over a web page.

4.4 Example

```
<BulkStatus>
   <SignedPart ID="top">

    <BatchHeader>
      <BatchID>1234</BatchID>
      <BatchTime>22 May 2001</BatchTime>
      <ProcessInfo>
        <CustomerName>ACME, Inc.</CustomerName>
      </ProcessInfo>
    </BatchHeader>

    <Status>
      <Pending>856</Pending>
      <Successful>142</Successful>
      <Failed>2</Failed>
    </Status>

   </SignedPart>

   <Signature>
     <!-- Stuff -->
      ...
      <Reference URI="#top">
      ...
   </Signature>

</BulkStatus>
```

5 BulkRegisterResult

The <BulkRegisterResult> contains the header and a certificate response for each request that was in the batch. The element contains a <BatchHeader>, <RegisterResults>, and <Signature>.

5.1 RegisterResults

The <RegisterResults> element contains one (and only one) <RegisterResult> for each and every <Request> element in the corresponding <BulkRegister>. The number of <RegisterResult> elements MUST be an attribute to <RegisterResults>.

<RegisterResult> is defined in [XKMS]; its contents are dictated by the <Respond> elements in the <BulkRegister>.

5.2 Example 1

This is a sample response to the first example request:

```
<BulkRegisterResult>
   <SignedPart ID="top">

   <BatchHeader>
     <BatchID>1234</BatchID>
     <BatchTime>22 May 2001</BatchTime>
     <ProcessInfo>
       <CustomerName>ACME, Inc.</CustomerName>
     </ProcessInfo>
   </BatchHeader>

   <RegisterResults number=2>
     <RegisterResult>
        <Result>Success</Result>
        <Answer>
          <KeyBinding>
            <Status>Valid</Status>
            <KeyID>1</KeyID>
            <ds:KeyInfo>
               <ds:RetrievalMethod
  URI="ldap://ldap.baltimore.com/c=US,o=Baltimore,cn=Niall"
  Type="http://www.w3.org/2000/09/xmldsig#X509Data"/>
            </ds:KeyInfo>
            <ClientInfo>hello</ClientInfo>
          </KeyBinding>
        </Answer>
     </RegisterResult>

     <RegisterResult>
        <Result>Success</Result>
        <Answer>
          <KeyBinding>
            <Status>Valid</Status>
            <KeyID>2</KeyID>
            <ds:KeyInfo>
               <ds:RetrievalMethod
  URI=" ldap://ldap.baltimore.com/c=US,o=Baltimore,cn=Gianfranco"
  Type="http://www.w3.org/2000/09/xmldsig#X509Data"/>
            </ds:KeyInfo>
            <ClientInfo>goodbye</ClientInfo>
          </KeyBinding>
        </Answer>
     </RegisterResult>
   </RegisterResults>

   </SignedPart>

   <Signature>
```

```
        <!-- Stuff -->
         ...
         <Reference URI="#top">
         ...
      </Signature>

   </BulkRegisterResult>
```

5.3 Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema
targetNamespace="http://www.xmltrustcenter.org/xkms/docs/x-bulk-2001-07-04.xsd"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xkms="http://www.xkms.org/schema/xkms-2001-01-20"
xmlns:xbulk="http://www.xmltrustcenter.org/xkms/docs/x-bulk-2001-07-04.xsd"
xmlns="http://www.w3.org/2000/10/XMLSchema">

    <annotation>
        <documentation xml:lang="en">
                 XML Schema for X-BULK version 1.1 draft 4
        </documentation>
    </annotation>
    <!-- General Stuff -->
    <complexType name="BatchHeaderType">
        <choice minOccurs="0" maxOccurs="unbounded">
            <sequence>
                <element name="BatchID" type="string"/>
                <element name="BatchTime" type="dateTime"/>
                <element name="NumberOfRequests" type="positiveInteger"/>
                <element ref="xbulk:ProcessInfo" minOccurs="0"/>
            </sequence>
        </choice>
    </complexType>
    <element name="ClientInfo">
        <complexType>
            <sequence maxOccurs="unbounded">
                <any namespace="##other"/>
            </sequence>
        </complexType>
    </element>
    <!-- copied from XKMS, since not typed there -->
    <element name="KeyID" type="uriReference"/>
    <element name="ProcessInfo">
        <complexType>
            <sequence minOccurs="0" maxOccurs="unbounded">
                <any namespace="##other"/>
            </sequence>
        </complexType>
    </element>
    <element name="Respond">
        <complexType>
            <sequence>
                <element name="string" type="string"
                    minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </complexType>
    </element>
    <!-- Register Stuff -->
    <element name="BulkRegister">
        <complexType>
            <sequence>
                <element name="SignedPart">
                    <complexType>
                        <sequence>
                            <element name="BatchHeader" type="xbulk:BatchHeaderType"/>
                            <element ref="xbulk:Respond"/>
                            <element name="Requests">
                                <complexType>
                                    <sequence>
                                        <element ref="xbulk:Request"
                                            maxOccurs="unbounded"/>
                                    </sequence>
                                    <attribute name="number" use="required"/>
                                </complexType>
                            </element>
                        </sequence>
                        <attribute name="ID" type="ID" use="required"/>
                    </complexType>
                </element>
                <element ref="ds:Signature"/>
            </sequence>
        </complexType>
    </element>
    <element name="Request">
        <complexType>
            <sequence>
                <element ref="xbulk:KeyID"/>
                <element ref="ds:KeyInfo"/>
                <element ref="xbulk:Respond" minOccurs="0"/>
                <element ref="xbulk:ProcessInfo" minOccurs="0"/>
                <element ref="xbulk:ClientInfo" minOccurs="0"/>
            </sequence>
        </complexType>
    </element>
    <!-- new types of ds:KeyInfo -->
    <element name="PKCS1" type="binary"/>
    <element name="PKCS10" type="binary"/>
    <!-- Status Specific Stuff -->
    <element name="BulkStatusRequest">
        <complexType>
            <sequence>
                <element name="SignedPart">
                    <complexType>
                        <sequence>
                            <element name="BatchHeader" type="xbulk:BatchHeaderType"/>
                        </sequence>
                    <attribute name="ID" type="ID" use="required"/>
                    </complexType>
                </element>
                <element ref="ds:Signature"/>
            </sequence>
        </complexType>
    </element>
    <element name="BulkStatus">
        <complexType>
            <sequence>
                <element name="SignedPart">
                    <complexType>
                        <sequence>
                            <element name="BatchHeader" type="xbulk:BatchHeaderType"/>
                            <element ref="xbulk:Status"/>
                        </sequence>
                        <attribute name="ID" type="ID" use="required"/>
                    </complexType>
                </element>
                <element ref="ds:Signature"/>
            </sequence>
        </complexType>
    </element>
    <element name="Status">
        <complexType>
            <sequence>
                <element name="Pending" type="positiveInteger"/>
                <element name="Successful" type="positiveInteger"/>
                <element name="Failed" type="positiveInteger"/>
            </sequence>
        </complexType>
    </element>
    <!-- Result Specific Stuff -->
    <element name="BulkRegisterResult">
        <complexType>
            <sequence>
                <element name="SignedPart">
                    <complexType>
                        <sequence>
                            <element name="BatchHeader" type="xbulk:BatchHeaderType"/>
                            <element name="RegisterResults">
                                <complexType>
                                    <sequence>
                                        <element ref="xkms:RegisterResult"
                                            maxOccurs="unbounded"/>
                                    </sequence>
                                    <attribute name="number" use="required"/>
                                </complexType>
                            </element>
                        </sequence>
                        <attribute name="ID" type="ID" use="required"/>
                    </complexType>
                </element>
                <element ref="ds:Signature"/>
            </sequence>
        </complexType>
    </element>
</schema>
```

6 WSDL

```
<?xml version="1.0" encoding="UTF-8"?>
    <definitions xmlns:s="http://www.w3.org/2000/10/XMLSchema."
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
xmlns:xkms="http://www.xkms.org/schema/xkms-2001-01-20"
xmlns:xbulk="http://www.xmltrustcenter.org/xkms/docs/x-bulk-2001-07-04.xsd"
targetNamespace="http://www.xmltrustcenter.org/xkms/docs/x-bulk-2001-07-04.xsd"
xmlns="http://schemas.xmlsoap.org/wsdl/">

    <types>
        <s:schema targetNamespace="http://www.xmltrustcenter.org/xkms/docs/x-bulk-
2001-07-04.xsd"
            attributeFormDefault="unqualified" elementFormDefault="qualified">
            <!-- General Stuff -->
            <s:complexType name="BatchHeaderType">
                <s:choice minOccurs="0" maxOccurs="unbounded">
                    <s:sequence>
                        <s:element name="BatchID" type="string"/>
                        <s:element name="BatchTime" type="dateTime"/>
                        <s:element name="NumberOfRequests" type="positiveInteger"/>
                        <s:element ref="xbulk:ProcessInfo" minOccurs="0"/>
                    </s:sequence>
                </s:choice>
            </s:complexType>
            <s:element name="ClientInfo">
                <s:complexType>
                    <s:sequence maxOccurs="unbounded">
                        <s:any namespace="##other"/>
                    </s:sequence>
                </s:complexType>
            </s:element>
            <!-- copied from XKMS, since not typed there -->
            <s:element name="KeyID" type="uriReference"/>
            <s:element name="ProcessInfo">
                <s:complexType>
                    <s:sequence minOccurs="0" maxOccurs="unbounded">
                        <s:any namespace="##other"/>
                    </s:sequence>
                </s:complexType>
            </s:element>
            <s:element name="Respond">
                <s:complexType>
                    <s:sequence>
                        <s:element name="string" type="string"
                            minOccurs="0" maxOccurs="unbounded"/>
                    </s:sequence>
                </s:complexType>
            </s:element>
            <!-- Register Stuff -->
            <s:element name="BulkRegister">
                <s:complexType>
                    <s:sequence>
                        <s:element name="SignedPart">
                            <s:complexType>
                                <s:sequence>
                            <s:element name="BatchHeader"
                                    type="xbulk:BatchHeaderType"/>
                                        <s:element ref="xbulk:Respond"/>
                                        <s:element name="Requests">
                                            <s:complexType>
                                                <s:sequence>
                                                    <s:element ref="xbulk:Request"
                                                        maxOccurs="unbounded"/>
                                                </s:sequence>
                                                <s:attribute name="number" use="required"/>
                                            </s:complexType>
                                        </s:element>
                                </s:sequence>
                                <s:attribute name="ID" type="ID" use="required"/>
                            </s:complexType>
                        </s:element>
                        <s:element ref="ds:Signature"/>
                    </s:sequence>
                </s:complexType>
```

```
          </s:element>
          <s:element name="Request">
              <s:complexType>
                  <s:sequence>
                      <s:element ref="xbulk:KeyID"/>
                      <s:element ref="ds:KeyInfo"/>
                      <s:element ref="xbulk:Respond" minOccurs="0"/>
                      <s:element ref="xbulk:ProcessInfo" minOccurs="0"/>
                      <s:element ref="xbulk:ClientInfo" minOccurs="0"/>
                  </s:sequence>
              </s:complexType>
          </s:element>
          <!-- new types of ds:KeyInfo -->
          <s:element name="PKCS1" type="binary"/>
          <s:element name="PKCS10" type="binary"/>
          <!-- Status Specific Stuff -->
          <s:element name="BulkStatusRequest">
              <s:complexType>
                  <s:sequence>
                      <s:element name="SignedPart">
                          <s:complexType>
                              <s:sequence>
                                  <s:element name="BatchHeader"
                                      type="xbulk:BatchHeaderType"/>
                              </s:sequence>
                              <s:attribute name="ID" type="ID" use="required"/>
                          </s:complexType>
                      </s:element>
                      <s:element ref="ds:Signature"/>
                  </s:sequence>
              </s:complexType>
          </s:element>
          <s:element name="BulkStatus">
              <s:complexType>
                  <s:sequence>
                      <s:element name="SignedPart">
                          <s:complexType>
                              <s:sequence>
                                  <s:element name="BatchHeader"
                                      type="xbulk:BatchHeaderType"/>
                                  <s:element ref="xbulk:Status"/>
                              </s:sequence>
                              <s:attribute name="ID" type="ID" use="required"/>
                          </s:complexType>
                      </s:element>
                      <s:element ref="ds:Signature"/>
                  </s:sequence>
              </s:complexType>
          </s:element>
          <s:element name="Status">
              <s:complexType>
                  <s:sequence>
                      <s:element name="Pending" type="positiveInteger"/>
                      <s:element name="Successful" type="positiveInteger"/>
                      <s:element name="Failed" type="positiveInteger"/>
                  </s:sequence>
              </s:complexType>
          </s:element>
          <!-- Result Specific Stuff -->
          <s:element name="BulkRegisterResult">
              <s:complexType>
                  <s:sequence>
                      <s:element name="SignedPart">
                          <s:complexType>
                              <s:sequence>
                                  <s:element name="BatchHeader"
                                      type="xbulk:BatchHeaderType"/>
                                  <s:element name="RegisterResults">
                                      <s:complexType>
                                          <s:sequence>
                                              <s:element ref="xkms:RegisterResult"
                                                  maxOccurs="unbounded"/>
                                          </s:sequence>
                                          <s:attribute name="number" use="required"/>
                                      </s:complexType>
                                  </s:element>
                              </s:sequence>
                              <s:attribute name="ID" type="ID" use="required"/>
                          </s:complexType>
                      </s:element>
```

```
                        <s:element ref="ds:Signature"/>
                      </s:sequence>
                    </s:complexType>
                  </s:element>
                </s:schema>
              </types>
              <message name="BulkRegisterSoapIn">
                <part name="parameters" element="xbulk:BulkRegister"/>
              </message>
              <message name="BulkRegisterSoapOut">
                <part name="parameters" element="xbulk:BulkRegisterResult"/>
              </message>
              <message name="BulkStatusSoapIn">
                <part name="parameters" element="xbulk:BulkStatusRequest"/>
              </message>
              <message name="BulkStatusSoapOut">
                <part name="parameters" element="xbulk:BulkStatus"/>
              </message>
              <portType name="XBulkSoap">
                <operation name="BulkRegister">
                  <input message="BulkRegisterSoapIn"/>
                  <output message="BulkRegisterSoapOut"/>
                </operation>
                <operation name="BulkStatus">
                  <input message="BulkStatusSoapIn"/>
                  <output message="BulkStatusSoapOut"/>
                </operation>
              </portType>
              <portType name="XBulkHttpPost"/>
              <portType name="XBulkHttpGet"/>
              <binding name="XBulkSoap" type="xbulk:XBulkSoap">
                <soap:binding transport="http://schemas.xmlsoap.org/soap/http"
style="document"/>
                <operation name="BulkRegister">
                  <soap:operation soapAction="http://id.baltimore.com/BulkRegister"
style="document"/>
                  <input>
                    <soap:body use="literal"/>
                  </input>
                  <output>
                    <soap:body use="literal"/>
                  </output>
                </operation>
                <operation name="BulkStatus">
                  <soap:operation soapAction="http://id.baltimore.com/BulkStatus"
style="document"/>
                  <input>
                    <soap:body use="literal"/>
                  </input>
                  <output>
                    <soap:body use="literal"/>
                  </output>
                </operation>
              </binding>
              <binding name="XBulkHttpPost" type="xbulk:XBulkHttpPost">
                <http:binding verb="POST"/>
              </binding>
              <binding name="XBulkHttpGet" type="xbulk:XBulkHttpGet">
                <http:binding verb="GET"/>
              </binding>
              <service name="XBulk">
                <port name="XBulkSoap" binding="xbulk:XBulkSoap">
                  <soap:address location="http://id.baltimore.com/Test/XBulk.asmx"/>
                </port>
                <port name="XBulkHttpPost" binding="xbulk:XBulkHttpPost">
                  <soap:address location="http://id.baltimore.com/Test/XBulk.asmx"/>
                </port>
                <port name="XBulkHttpGet" binding="xbulk:XBulkHttpGet">
                  <soap:address location="http://id.baltimore.com/Test/XBulk.asmx"/>
                </port>
              </service>
            </definitions>
```

7 Authors

Stephen Farrell, Baltimore Technologies, stephen.farrell@baltimore.ie
Merlin Hughes, Baltimore Technologies, merlin@baltimore.ie
Niall O'Sullivan, Baltimore Technologies, nosullivan@baltimore.com
Patrick George, Gemplus, patrick.george@gemplus.com
François Rajchenbach, Oberthur Card Systems, f.rajchenbach@oberthurcs.com
Jean-Claude Perrin, Schlumberger, perrin@montrouge.tt.slb.com

Thanks also to Pierre Heuze (Cardbase) and James Webster (Cards etc) for commenting on earlier versions of the specification.

The authors would also like to acknowledge the authors of, and contributors to, the base XKMS specification, in particular Warwick Ford and Phill Hallam-Baker of Verisign.

Appendix A References

[PKCS#1]	Kaliski, B., "PKCS #1: RSA Encryption, Version 1.5.", RFC 2313, March 1998.
[PKCS#10]	Kaliski, B., Nystrom, M., "PKCS #10: Certification Request Syntax Specification Version 1.7" RFC 2986, November 2000.
[WSDL]	E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, Web Services Description Language (WSDL) 1.0 September 25, 2000, http://www.w3.org/TR/wsdl
[XKMS]	P. Hallam-Baker et al, "XML Key Management Specification", World Wide Web Consortium, http://www.w3.org/TR/xkms
[XMLDSIG]	D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, World Wide Web Consortium. http://www.w3.org/TR/xmldsig-core/

Appendix B Legal Notices