XML Security Derived Keys

W3C Working Draft 26 February 2009

This version:
http://www.w3.org/TR/2009/WD-xmlsec-derivedkeys-20090226/
Latest version:
http://www.w3.org/TR/xmlsec-derivedkeys/
Editor:
Magnus Nyström, RSA, The Security Division of EMC

Abstract

Key derivation is a well-established mechanism for generating new cryptographic keys material from some existing, original ("master") key material and potentially other information. This document augments XML Signature and XML Encryption by defining XML types and elements necessary to enable use of derived keys in XML security applications.

Status of this Document

This section describes the status of this document at the time of its publication. Other documents may supersede this document. A list of current W3C publications and the latest revision of this technical report can be found in the W3C technical reports index at http://www.w3.org/TR/.

This is a First Public Working Draft of "XML Security Derived Keys."

This specification defines an XML Syntax and processing rules for use of derived keys within both the XML Signature and XML Encryption frameworks. This specification does not define any new cryptographic algorithms, and does not include any additional mandatory to implement algorithms.

This document was developed by the XML Security Working Group. The Working Group expects to advance this Working Draft to Recommendation Status.

Please send comments about this document to public-xmlsec-comments@w3.org (with public archive).

Publication as a Working Draft does not imply endorsement by the W3C Membership. This is a draft document and may be updated, replaced or obsoleted by other documents at any time. It

is inappropriate to cite this document as other than work in progress.

This document was produced by a group operating under the 5 February 2004 W3C Patent Policy. W3C maintains a public list of any patent disclosures made in connection with the deliverables of the group; that page also includes instructions for disclosing a patent. An individual who has actual knowledge of a patent which the individual believes contains Essential Claim(s) must disclose the information in accordance with section 6 of the W3C Patent Policy.

Table of Contents

---

1 Introduction

In cryptographic applications, it is common to make use of derived cryptographic key material. In these applications, derived keys are used for a variety of purposes including data encryption and message authentication. The reason for doing key derivation itself is typically a combination of a desire to expand a given, but limited, set of original key material and prudent security practices of limiting use (exposure) of such key material. Key separation (such as avoiding use of the same key material for multiple purposes) is an example of such practices. Key derivation has wide use. The key derivation process may be based on passphrases agreed upon or remembered by users, or it can be based on some master keys (and be intended to reduce exposure of such master keys), etc. Derived keys themselves may be used in XML Signature and XML Encryption as any other keys; in particular, they may be used to compute message authentication codes (e.g. digital signatures using symmetric keys) or for encryption/decryption purposes. This specification defines a derived key XML type and associated elements; conformance requirements are specified by way of schema definitions and prose.

This document specifies an XML syntax and processing rules for creating and referencing cryptographic keys derived from some initial, originating key material and possibly additional

information. The document augments XML Signature [XMLDSIG2nd] and XML Encryption [XMLENC].

This document does not normatively specify how derived keys are to be used; rather it focuses on the basis for interoperability, namely the fundamental data types required for usage of derived keys in XML-based security applications and the meaning of those data types.

1.1 Editorial

The key words "MUST" and "OPTIONAL" in this specification are to be interpreted as described in RFC2119 [RFC2119]:

" they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm "

Consequently, these capitalized keywords are used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. These key words are not used (capitalized) to describe XML grammar; schema definitions unambiguously describe such requirements. For instance, an XML attribute might be described as being "optional."

2 Versions, Namespaces and Identifiers

No provision is made for an explicit version number in this syntax. If a future version is needed, it will use a different namespace. The XML namespace [XML-NS] URI that MUST be used by implementations of this (dated) specification is:

```
xmlns dkey="http://www.w3.org/2009/01/xmlsec-derivedkey"
```

> ***Warning: this namespace URI is currently subject to change.***

This namespace is also used as the prefix for identifiers defined by this specification. While applications MUST support XML and XML namespaces, the use of internal entities or our `dkey` XML namespace prefix and defaulting/scoping conventions are OPTIONAL; we use these facilities to provide compact and readable examples.

This specification uses Uniform Resource Identifiers [RFC2396] to identify resources, algorithms, and semantics. The URI in the namespace declaration above is also used as a prefix for URIs under the control of this specification.

This document does not change the URI associated with XML Signature or XML Encryption itself.

3 Usage Scenarios

3.1 Introduction

These scenarios are for illustrative purposes only and is not to be seen as inclusive. Clearly there are many more usages for derived keys than described in these scenarios.

3.2 Passphrase-based Derived Keys

In this scenario, two users, Alice and Bob, need to exchange encrypted data by use of XML Encryption. They agree on a password that they both can remember. Alice then derives a cryptographic key from this password, encrypts the data using the derived key, and provides information about the used password and the key derivation technique in the XML instance containing the encrypted data. Bob, recognizing which password has been used (possibly through a password hint conveyed in the `MasterKeyName` element, see below), provides the password to his local application which then decrypts the data for him.

3.3 Digital Signatures Using Derived Keys

An application has access to a long-term secret (cryptographic key) but in order to reduce exposure of this secret it periodically derives short-lived keying material from the secret, e.g. by hashing the long-term secret with the current time. The short-lived keying material is then used to digitally sign messages (e.g. to integrity-protect an event log). In this case, the application signs the messages using XML Signature and indicates in the instance document the master keying material as well as the method used to derive the actual signing key (along with any other information required to verify the signature such as the time used).

3.4 Key Separation

An application has access to a long-term secret but needs to make use of this secret for multiple purposes. For key separation purposes, it therefore derives multiple keys (one derived key for each usage) from the long-term secret and use the derived keys instead of the secret when performing the cryptographic operations.

4 Requirements

The XML Security Requirements document [XMLREQ] defines requirements on the design presented herein.

5 Design

This section summarizes schema definitions and processing rules for derived keys.

5.1 The `DerivedKey` Element

Identifier

```
Type="http://www.w3.org/2009/01/xmlsec-derived-key#DerivedKey
```

(This can be used within a `ds:RetrievalMethod` element to identify the referent's type.)

The `DerivedKey` element is used to transport information about a derived key from the originator to recipient(s). It may be used as a stand-alone XML document, be placed within an application document, or appear inside an `EncryptedData` or `Signature` element as a child of a `ds:KeyInfo` element. The key value itself is never sent by the originator. Rather, the originator provides information to the recipient(s) by which the recipient(s) can derive the same key value. When the key has been derived the resulting octets are made available to the `EncryptionMethod` or `SignatureMethod` algorithm without any additional processing.

5.1 Schema Definition

```
<element name="DerivedKey" type="dkey:DerivedKeyType"/>
   <complexType name="DerivedKeyType">
  <sequence>
    <element ref="dkey:KeyDerivationMethod" minOccurs="0"/>
    <element ref="xenc:ReferenceList" minOccurs="0"/>
    <element name="MasterKeyName" type="string" minOccurs="0"/>
  </sequence>
  <attribute name="Id" type="ID" use="optional"/>
  <attribute name="Type" type="anyURI" use="optional"/>
</complexType>

<element name="KeyDerivationMethod" type="dkey:KeyDerivationMethodType"/>
<complexType name="KeyDerivationMethodType">
  <sequence>
    <any namespace="##other" minOccurs="0" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
```

`KeyDerivationMethod` is an optional element that describes the key derivation algorithm applied to the master (underlying) key material. If the element is absent, the key derivation algorithm must be known by the recipient or the recipient's key derivation will fail.

`ReferenceList` is an optional element containing pointers to data and keys encrypted using this key. The reference list may contain multiple references to `EncryptedKey` or `EncryptedData` elements. This is done using `KeyReference` and `DataReference` elements from XML Encryption.

`MasterKeyName` is an optional element for associating a user readable name with the master key (or secret) value. This element may also be used to reference the key using the `ds:KeyName` element within `ds:KeyInfo`. The same `MasterKeyName` label, unlike an ID type, may occur multiple times within a single document. The value of the master key is to be the same in all `DerivedKey` elements identified with the same `MasterKeyName` label within a single XML document. Note that because whitespace is significant in the value of the `ds:KeyName` element, whitespace is also significant in the value of the `MasterKeyName` element. If no `MasterKeyName` is provided, the master key material must be known by the recipient or key derivation will fail.

The optional `Id` attribute provides for the standard method of assigning a string id to the element within the document context.

The `Type` attribute can be used to further specify the type of the derived key if the `KeyDerivationMethod` algorithm does not define an unambiguous encoding/representation.

6 Key Derivation Algorithms

This document does not define any key derivation algorithms. However, key derivation algorithms are defined elsewhere, e.g. in PKCS #5 v2.0 Amendment 1 [PKCS5].

7 Examples

7.1 Passphrase-based Data Encryption

In this (fictitious, but syntactically correct) example, which builds on PKCS #5 v2.0 Amendment 1, a derived key element is illustrated. The underlying passphrase is identified by the name "Our shared secret".

Example: `DerivedKey` Element

```
<dkey:DerivedKey
   xmlns:dkey="http://www.w3.org/2009/01/xmlsec-derived-key"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xmlns:xenc="http://www.w3.org/2001/04/xmlenc#"
   xmlns:pkcs-5="http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5v2-0#">
   <dkey:KeyDerivationMethod
       Algorithm="http://www.rsasecurity.com/rsalabs/pkcs/schemas/pkcs-5#pbkdf2">
     <pkcs-5:PBKDF2-params>
       <Salt>
         <Specified>Df3dRAhjGh8=</Specified>
       </Salt>
       <IterationCount>2000</IterationCount>
       <KeyLength>16</KeyLength>
       <PRF/>
     </pkcs-5:PBKDF2-params>
   </dkey:KeyDerivationMethod>
   <xenc:ReferenceList>
     <xenc:DataReference URI="#ED"/>
   </xenc:ReferenceList>
   <MasterKeyName>Our shared secret</MasterKeyName>
</dkey:DerivedKey>
```

8 Security Considerations

8.1 General

As derived keys are to be used with XML Signature and XML Encryption, the security considerations of those specifications applies.

8.2 Use of Passwords

The usual warnings and recommendations apply when basing security schemes on passwords to be remembered by users. In particular, bad choices for the `MasterKeyName` may provide an attacker with information which can simplify brute-force attacks. As an example, using "My Spouse's Name" as the value of the `MasterKeyName` element would not be a good choice (if the value of the master key (password) is the user's spouse's name, that is).

9 Conformance

An implementation is conformant to this specification if it successfully generates syntax according to the schema definitions and satisfies any and all MUST/REQUIRED/SHALL requirements.

10 Acknowledgments

TBD

11 References

PKCS5

PKCS #5 Version 2.0 Amendment 1: XML Schema for Password-Based Cryptography, RSA Laboratories. 29 March 2007, http://www.rsa.com/rsalabs/pkcs/

RFC2119

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

RFC2396

Uniform Resource Identifiers (URI): Generic Syntax, T. Berners-Lee, R. Fielding, L. Masinter. IETF RFC 2396, August 1998, http://ietf.org/rfc/rfc2396.txt.

XML-NS

Namespaces in XML, T. Bray, D. Hollander, A. Layman. W3C Recommendation, January 1999, http://www.w3.org/TR/1999/REC-xml-names-19990114.

XMLDSIG2nd

XML Signature Syntax and Processing (Second Edition), W3C Recommendation 10 June 2008, http://www.w3.org/TR/2008/REC-xmldsig-core-20080610/

XMLENC

XML Encryption Syntax and Processing , D. Eastlake, J. Reagle, W3C Recommendation 10 December 2002, http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

XMLREQ

XMLSecurity Use Cases and Requirements, F. Hirsch, T. Roessler. W3C Working Draft, 26 February 2009, http://www.w3.org/TR/2009/WD-xmlsec-reqs-20090226/