Converting Digital Signature Formats for XML Documents

Hiroki Itoh, Tsuyoshi Abe, and Kenji Takahashi
NTT Information Sharing Platform Laboratories, NTT Corporation
3-9-11 Midori-cho, Musashino-shi, Tokyo 180-8585, Japan
{itoh.hiroki, abe.tsuyoshi, takahashi.kenji} @lab.ntt.co.jp

Abstract

We propose a method of format conversion of digital signatures for XML documents from the PKCS#7 (RFC2315[1]) format into the XML Signature (RFC3275[2]) format, which preserves the validity of the signatures through the conversion. This method enables existing devices, e.g. mobile phones and smart cards, designed for PKCS#7 format to calculate XML Signatures for emerging applications, such as secure XML document management and Web services. Based on the format conversion method, we have implemented a SAML[3] Identity Provider (IdP) capability as a Java application on a mobile phone. The IdP issues SAML Assertions, one of the XML documents with XML Signature messages converted from PKCS#7 messages originally created by the mobile phone.

1. Introduction

With the growth of the emerging secure XML application and Web service markets, the adoption of XML Signature is expected to expand. In fact, XML Signature has been widely adopted as a digital signature in technical standards, such as ebXML[4], Liberty Alliance[5] ID-WSF and SAML.

There is another digital signature standard, PKCS#7, that has been used before XML Signature. PKCS#7 is already widely used with tamper resistant devices such as smart cards. However, the PKCS#7 format is not suitable for Web Services because the format is binary.

We propose a method in which an XML Signature message for an XML document will be created with a PKCS#7 message. This method enables existing devices designed for PKCS#7 to calculate XML Signature messages for emerging applications.

2. Problems

The main problem to discuss in this paper is converting the data format between two digital signature formats. For the conversion, we define the following two requirements: (1) Validity - the converted XML Signature messages should be valid and be able to be verified by conventional XML Signature implementations, and (2) Implementability - the method should be able to be implemented in existing device environments with few or no modifications.

The merits and demerits of XML Signature and PKCS#7 are shown in Table 1. XML Signature and PKCS#7 are complementary to each other. XML Signature is mainly used for XML documents and PKCS#7 is used for other documents and data.

XSS4J[6] ASN.1/XML Translator[7] has been proposed by T. Iwamura *et al.*, as a method in which a PKCS#7 digital signature or encrypted messages are converted into an XML document. A translated XML document describes each values of PKCS#7 format, but has no compatibility with XML Signature. Therefore, to make a Web Service adopt ASN.1/XML Translator, the Web Service has to be extended from original standards such as ebXML, SAML, and Liberty Alliance, for example.

However, if an XML Signature message of an XML document could be calculated with a PKCS#7 message, the Web Service needs no extension or needs just a little. Our format conversion method will enable existing devices, which are designed for PKCS#7, to calculate XML Signature messages. That will be the easy adoption of XML Signature. XML Signature will be used for such devices.

Table 1: Merits and demerits table of XML Signature and PKCS#7

	Merits	Demerits
XML Signature (RFC3275)	• Ready for legibility • Combinations with other XML documents can be facilitated	• Large overhead and inefficient method because of c14n, signing, and verification, for example.
PKCS#7 (RFC2315)	• Small overhead and efficient method	• No legibility • Low capability with Web Services

3. Solution

In this chapter, we describe 1) basic idea of format conversion, 2) how to calculate an XML Signature message of an XML document, and 3) a prototype based on this method.

3.1. Analysis of two formats

In this section, we describe i) comparison of two digital signature formats and ii) the original value to be signed by PKCS#7. The schema of XML Signature and the data structure of a PKCS#7 message are shown in List 1 and Table

2.

List 1: Schema of XML Signature

```
<OriginalDocument ID="xxxxxxxx">
  <ChildElement01>......</ChildElement01>
  <ChildElement02>......</ChildElement02>
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm="xml-exc-c14n"/>
      <ds:SignatureMethod Algorithm="rsa-sha1"/>
      <ds:Reference URI="xxxxxxxx">
        <ds:Transforms>
          <ds:Transform Algorithm="enveloped-signature"/>
          <ds:Transform Algorithm="xml-exc-c14n"/>
        </ds:Transforms>
        <ds:DigestMethod Algorithm="sha1"/>
        <ds:DigestValue>digest value</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>signature value</ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>X.509 Public Key Certificate</ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
</OriginalDocument>
```

Table 2: PKCS#7 Digital Signature Message of SignedData type

version			e.g., v1
digestAlgorithms			e.g., sha1
contentInfo			Original value
certificates			
	certificate		X.509 public key certificate
signerInfos			
	signerInfo		
		digestAlgorithm	e.g., sha1
		digestEncryptionAlgorithm	e.g., rsa
		encryptedDigest	signature value

The comparison of three items, i) Original value, ii) Signature value, and iii) X.509 public key certificate, which are the most important entities for a digital signature, are shown in Table 3.
Referring to Table 3, “X.509 public key certificate” is a public key certificate the digital signature signer has and is commonly used between two digital signature methods. In addition, <ds:SignatureValue> which is a “Signature value” of an XML Signature, is an encrypted message digest or hash of a canonicalized <ds:SignedInfo>. Therefore, to calculate the same “Signature value” in PKCS#7, “contentInfo” which is “Original value” of PKCS#7 format should be equal to one of XML Signature. Once “encryptedDigest” of PKCS#7 format is calculated, it will be used as <ds:SignatureValue> of XML Signature.
An XML document can be trusted if there is no falsification of i) Original value or ii) X.509 public key certificate in its XML Signature message.

Table 3: Comparison of data structure of XML Signature and PKCS#7

	XML Signature	PKCS#7
Original value	c14n-ed <ds:SignerInfo>	contentInfo
Signature value	<ds:SignatureValue>	encryptedDigest
X.509 public key certificate	<ds:X509Certificate>	certifiacte

3.2. Algorithm

The conversion procedure of this method is shown below.

1. The signer canonicalizes an XML document to be signed and calculates a message digest or a hash of the canonicalized XML document. Hence, the signer gets the <ds:SignedInfo> element of the XML document. The <ds:SignedInfo> element should be canonicalized as a matter of course. (Figure 1)
2. The signer signs the <ds:SignedInfo> element with a PKCS#7. (Figure 1)
3. The signer calculates a <ds:Signature> element with the <ds:SignedInfo> element, <ds:SignedValue> element whose value is “encryptedDigest” of the PKCS#7 format, and the <ds:KeyInfo> element, which is the signer’s X.509 public key certificate. (Figure 2)

4. The signer inserts the <ds:Signature> element into the XML document to be signed, and the signer gets the XML document using the XML Signature. (Figure 2)

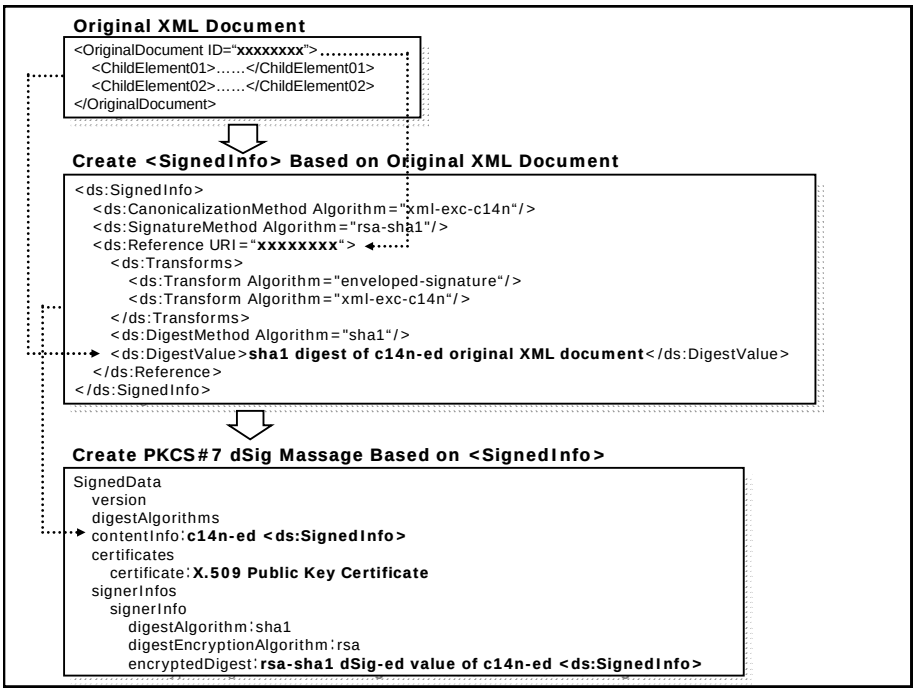

Figure 1: Conversion Procedure of This Method [1/2]

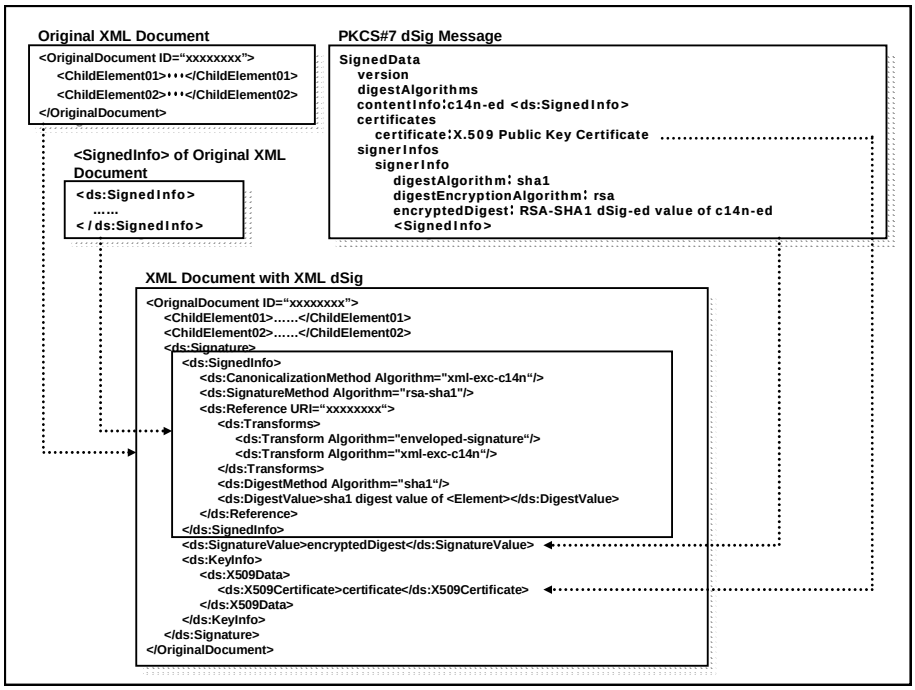

Figure 2: Conversion Procedure of This Method [2/2]

3.3. Prototype

We have implemented a prototype[8] of the method that uses a PKCS#7 capability of a mobile phone. The prototype serves as a SAML 2.0 Identity Provider (IdP) which issues a SAML Assertion with XML Signature upon requests from Service Providers (SP). Figure 3 illustrates an overview of the prototype architecture. The prototype consists of a Java application on a mobile phone and a relay server (RS).

We use NTT DoCoMo's F903i phone as the mobile phone, which has FirstPass[9] capabilities. FirstPass is a PKI platform of NTT DoCoMo. The USIM (Universal Subscriber Identity Module) that is used for FirstPass is tamper resistant, and it stores and handles five private/public key pairs and creates digital signatures in PKCS#7 format. Therefore, FirstPass enables the mobile phone client-side SSL, digital signature, and encryption, for example.

A Java application on a mobile phone authenticates the user, creates SAML Assertion, creates PKCS#7 messages by

using FirstPass.
The RS is introduced to convert digital signature messages so that the mobile phone can skip the conversion and thus significantly decrease its computing load. The mobile phone sends a SAML Assertion, a PKCS#7 message to the RS. Then the RS convert the PKCS#7 message into a XML Signature message and inserts the XML Signature message into the SAML Assertion and sends the SAML Assertion to the SP
In this way, we have developed a prototype that satisfies Validaty and Implementability requirements. The validaty is assured by the algorithm described in the previous section (Section 3.2) and the implementability is demonstrated in this section with commercial mobile phones and a server.

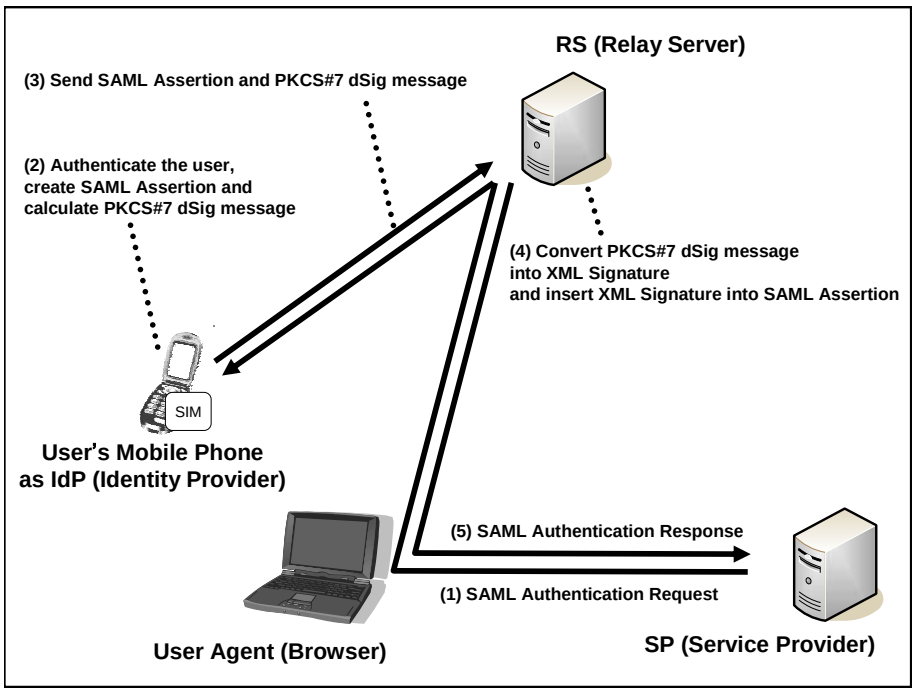

Figure 3: Overview of the prototype

4. Summary and Future Works

In this paper we proposed 1) a format conversion method of digital signatures for XML documents, 2) a prototype of this method. Such an XML document can be trusted if it can be proven that there is no falsification of the signature value of the XML Signature. This method will be suitable for some platforms designed for PKCS#7.
Through security analysis of the method and the implementation remains to be carried out. Also the standardization of the method should be explored to expand the deployment of the method, which, in turn we believe, accelerates the adoption of XML Signature.

REFERENCES

[1] http://rfc.net/rfc2315.html
[2] http://rfc.net/rfc3275.html
[3] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security
[4] http://www.ebxml.org/
[5] http://www.projectliberty.org/
[6] XML Security Suite, http://www.trl.ibm.com/projects/xml/xss4j/
[7] T. Iwamura, H. Maruyama, Mapping between ASN.1 and XML, http://www.trl.ibm.com/projects/xml/xss4j/docs/RT0362.pdf
[8] T. Abe, H. Itoh, K. Takahashi, Implementing an Identity Provider on a Mobile Phone, ACM CCS2007 Workshop on Digital Identity Management 2007, to appear
[9] http://www.nttdocomo.co.jp/service/other/firstpass/ [in Japanese only]