Canonical XML Encoding Rules (cXER) for Secure Messages

ASN.1 Schema for Secure XML Markup

Griffin Consulting   1625 Glenwood Avenue   Raleigh, North Carolina 27608 - 2319
+1 - 919 - 29 1- 0019   phil.griffin@ASN-1.com

Overview (a)

- ASN.1 is a **schema** for encoded values
  - Types describe general structure of abstract values
  - Each builtin type defines a class, a set of distinct values
  - Constraints restrict a class and the validity of values
  - Encoding rules define how abstract values are transferred

Overview (b)

- Encoded ASN.1 values are **binary** or **text**

  - Binary and XML Canonical Forms

    - Distinguished Encoding Rules => **DER**
    - Canonical XML Encoding Rules => **cXER**

  - Each DER encoding maps to a cXER value

Example type and value

```
AnyName ::= [APPLICATION 1] SEQUENCE {
   givenName   VisibleString,
   initial     [0] UTF8String (SIZE(1))  OPTIONAL,
   familyName  IA5String
}
```

A value of type **AnyName** ***encoded as XML markup***

```
<AnyName>
   <givenName> Hubert </givenName>
   <initial> L </initial>
   <familyName> Owen </familyName>
</AnyName>
```

Another example

```
PKIStatusInfo ::= SEQUENCE {
   status         PKIStatus,
   statusString   PKIFreeText  OPTIONAL,
   failInfo       PKIFailureInfo  OPTIONAL
}

PKIStatus ::= INTEGER { rejection (2) } (0..MAX)

PKIFreeText ::= SEQUENCE SIZE(1..MAX) OF UTF8String

PKIFailureInfo ::= BIT STRING { timeNotAvailable (14) }
```

As a rule ...

Whenever possible, the identifier name is used as the default markup tag. Otherwise, the user defined type name is used.

```
<PKIStatusInfo>
   <status>  <rejection/>  </status>
   <statusString>
      <PKIFreeText>
         Your request has been rejected.
      </PKIFreeText>
   </statusString>
   <failInfo>  <timeNotAvailble/>  </failInfo>
</PKIStatusInfo
```

Canonical XER

The Canonical XML Encoding Rules (cXER) are defined in:
ISO/IEC 8825-4 | ITU-T X.693 ASN.1 XML Encoding Rules (XER)

The same ASN.1 value is **cXER** encoded in *one and only one way* as a single long string containing no "white-space" characters outside of data:

```
<PKIStatusInfo><status><rejection/></status><statusStri
ng><PKIFreeText>Your request has been rejected.</PKIFre
eText></statusString><failInfo><timeNotAvailble/></fail
Info></PKIStatusInfo>
```

XML Object Identifiers

Object identifiers are used in security specifications to unambiguously identify algorithms, parameters, processing methods, and biometric types.

In the RSA PKCS #5: *Password-Based Cryptography Specification* **`id-hmacWithSHA1`** identifies the HMAC-SHA-1 pseudorandom function:

```
id-hmacWithSHA1 OID ::= { digestAlgorithm 7 }
```

Using cXER, this OID is represented as:

```
id-hmacWithSHA1 ::= <OID> 1.2.840.113549.2.7 </OID>
```

The associated algorithm parameters are a value of type **`NULL`**.

XML AlgorithmIdentifier (a)

RFC 2898 PKCS #5 defines a password based key derivation function pseudorandom function using the following ASN.1 schema:

```
PBKDF2-PRF ::= AlgorithmIdentifier {{PBKDF2-PRFs}}

AlgorithmIdentifier { AID:IOSet } ::= SEQUENCE {
   algorithm    AID.&id({IOSet}),
   parameters   AID.&Type({IOSet}{@algorithm}) OPTIONAL
}

AID ::= TYPE-IDENTIFIER -- ISO/IEC 8824-2:1998, Annex A

NoIV ::= NULL
```

XML AlgorithmIdentifier (b)

A value of the pseudorandom function HMAC-SHA-1 can be specified as XML 1.0 markup using the ASN.1 XML Encoding Rules as the value:

```
 <PBKDF2-PRF>
    <algorithm> 1.2.840.113549.2.7 </algorithm>
    <parameters> </NoIV> </parameters>
</PBKDF2-PRF>
```

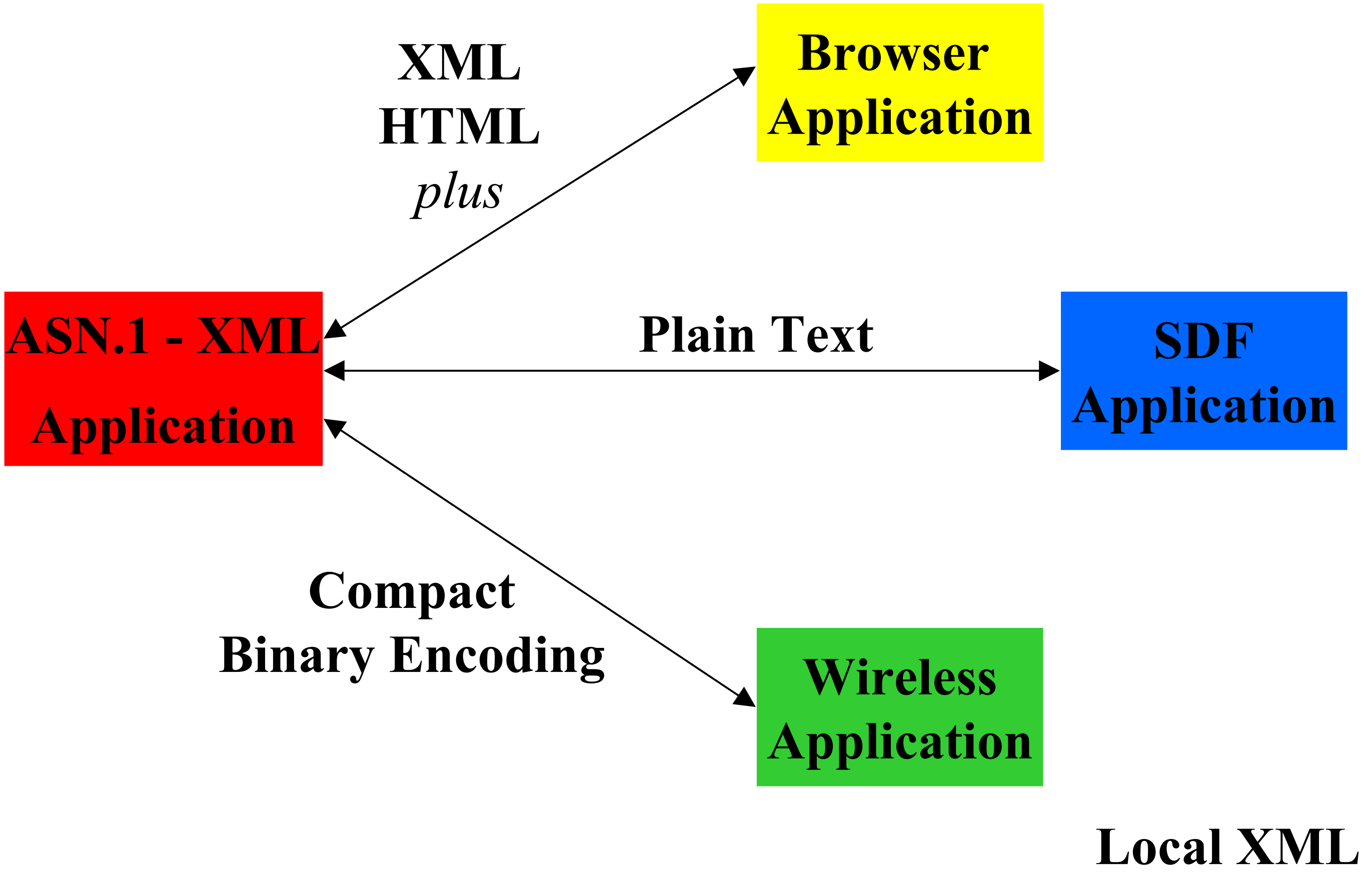
Notice that the new dotted form of OID is used.

Notice that the NULL value has no start and end tags.

ASN.1 XML Benefits (a)

- A single schema for all values
  - Binary and text encodings are all based on ASN.1 types
    - * **Eliminates multiple schema mappings**
    - * **Provides an efficient schema for XML values**
- ASN.1 <=> XML communications
  - ASN.1 applications can send and receive XML values
  - Efficient transfer, simple signature processing

ASN.1 / XML Benefits (b)



X9.96 XML CMS - **XCMS** (a)

XML Cryptographic Message Syntax (XCMS)

Schema: **X9.73 Cryptographic Message Syntax** (CMS) => **DER**

Formats: **X.693 Canonical XML Encoding Rules** => **cXER**

X9.96 XML Cryptographic Message Syntax:

- Same abstract values in X9.73 are secured using XML
- Same level of security in X9.73
- Same cryptographic processing in X9.73 - encode, digest, sign.

CMS/XCMS applications can have it both ways:

- transfer compact, binary DER and use XML markup locally
- or transfer exactly the same information using XML

X9.96 XML CMS - XCMS (b)

The X9.96 XML CMS schema is identical to the X9.73 CMS schema.

X9.73 CMS is tightly aligned with the IETF S/MIME CMS schema.

But X9.73 and X9.96 extend S/MIME by providing support for

- X9.69 Constructive Key Management (CKM)
- X9.84 Biometric Information Management for Security
- X9.68 Domain Certificates

X9.96 XCMS includes the familiar RSA PKCS #7 types `SignedData`, `Data`, `EnvelopedData`, `EncryptedData`, and `DigestedData`.

DER and cXER => one simple, fast signature processing method

X9.96 XML CMS - XCMS (c)

X9.96 XML Cryptographic Message Syntax generalizes X9.84 and XCBF:

– **ANS X9.84 : 2001, Biometrics Information Management and Security For the Financial Services Industry ©**

– **XCBF - OASIS XML Common Biometric Format**

XCBF uses the X9.84 ASN.1 definitions as its schema

X9.84 uses the same markup tags as XCBF

X9.73, X9.96, X9.84 and XCBF use the same signature processing:
cXER - for canonical XML markup
DER - for compact, canonical binary

X9.96 XML CMS - **XCMS** (d)

An example XER encoding of a value of ASN.1 type **`BiometricObject`** from OASIS XCBF and X9.84 *H.1.1 Examples: Reduced Biometric Header*

```
<BiometricObject>
   <biometricHeader>
      <version> <hv1/> </version>
   </biometricHeader>
   <biometricData>
      0102030405060708090A0B0C0D0E0F0102030405060708090A
   </biometricData>
</BiometricObject>
```

This value XER encodes in **174** octets, **31** using DER

XML Common Biometric Format

XML Common Biometric Format Technical Committee (OASIS XCBF TC)
http://oasis-open.org/committees/xcbf/

A security TC in **OASIS**, the Organization for the Advancement of Structured Information Standards, a non-profit, international consortium that creates interoperable XML industry standards (**http://oasis-open.org/**) .

XCBF TC Goals:

- Define a common XML schema for the **NIST 6529 CBEFF** patron formats (**http://www.nist.gov/cbeff**) based on the **X9.84:2002** ASN.1 schema

- Define simple XML signature and encryption methods based on cXER

cXER for Secure Messages

Links?

BioloJava Security Tool Kit - ASN.1/XML Biometrics

http://asn-1.com/biolojava.htm Example programs, XML encodings

XCBF - XML Common Biometric Format

http://oasis-open.org/committees/xcbf/ XML biometric security standard

X9.84 - Biometric Information Management for Security

http://asn-1.com/x984.htm XML schema for XCBF

XML Encoding Rules

http://www.itu.int/ITU-T/studygroups/com17/languages/index.html

2002 Draft Host: **ftp://ties.itu.int** login: **asn1** password: **notation1**

Questions?

Griffin Consulting
1625 Glenwood Avenue
Hayes Barton at Five Points
Raleigh, North Carolina 27608 - 2319   USA

p: **+1 919 291 0019**
f: **+1 919 856 1132**
e: **phil.griffin@asn-1.com**
w: **http::/asn-1.com**