Slide 1

Canonical XML Encoding Rules (CXER) for Secure Messages

An ASN.1 Schema for XML Markup

Copyright © 2002 Griffin Consulting, All Rights Reserved.
Griffin Consulting 1625 Glenwood Avenue Raleigh, North Carolina 27608 - 2319
+1 - 919 - 29 1- 0019 phil.griffin@ASN-1.com

1

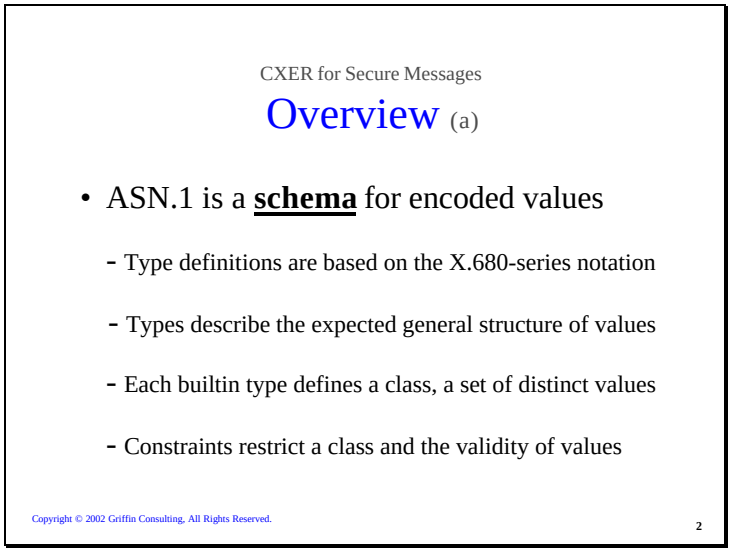
A new form of ASN.1 value notation was created in Geneva a year ago and is currently concluding ballot this Spring.

The format of this value notation is based on XML.

This work allows ASN.1 values to be transferred or displayed in a variety of textual of binary formats. (PER BER HTML XML plain text, etc.)

Users can leverage browsers for XML display and still have efficient binary transfer in BER.

Slide 2

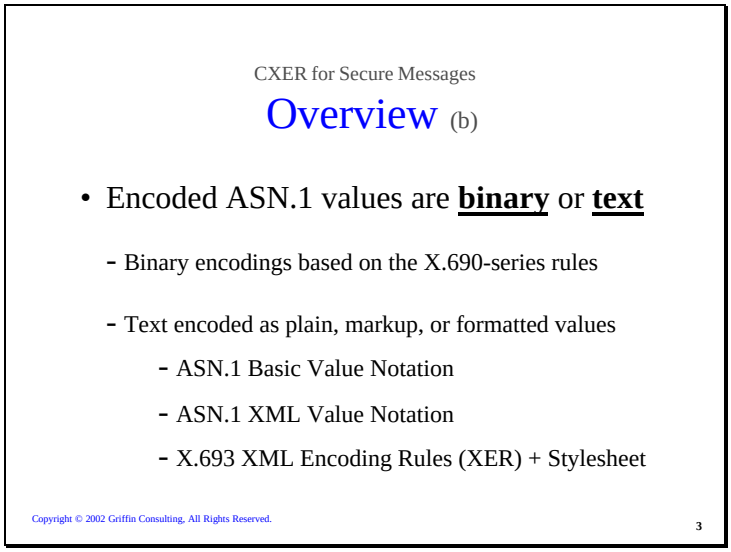


ASN.1 is a schema, which allows users to define the gross structure of messages. SEQUENCE, SET, CHOICE, etc.

ASN.1 data types (INTEGER, BIT STRING, etc.) allow users to choose distinct sets of values for message components.

ASN.1 constraints allow users to further restrict the sets of values in a given type to a valid subset of values. (Y or N) or (male or female) or (positive integers only)

Slide 3



The ASN.1 notation (X.680-series) is separate from the encoding rules (X.690-series).

Binary formats include BER, DER, PER and CER.
- Can be augmented by Encoding Control Notation (ECN)

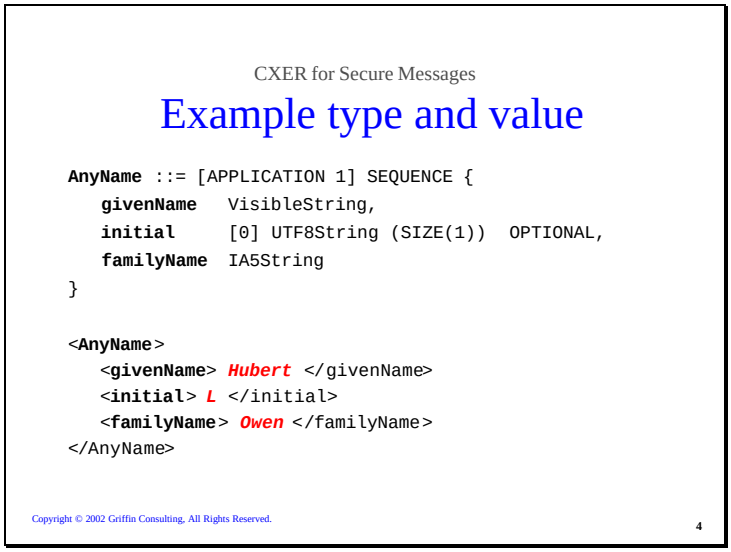
ASN.1 definitions do not depend on a particular set of encoding rules, unless specified by users.

They specify ABSTRACT values - the various encoding rules specify the TRANSFER syntax of the abstract values.

- X.509 and PKIX restrict their Certificate encodings to DER in a BER wrapper
- X9.73 & SMIME use BER mainly, with small pockets of DER
- 3GPP and many other wireless protocols use PER

NOTE - The XML Value Notation provides a means of representing ASN.1 values using XML. Thus, an ASN.1 type definition also specifies the structure and content of an XML element. This makes ASN.1 a simple schema language for XML.

Slide 4



Every ASN.1 type definition can be represented using markup.

Every ASN.1 type has a default markup notation. This default markup notation uses the ASN.1 defined type name and user specified identifiers shown highlighted here.

Given only the ASN.1 type definition of AnyName, tools on both ends of a wire can communicate values of type AnyName using the same default XML markup tags.

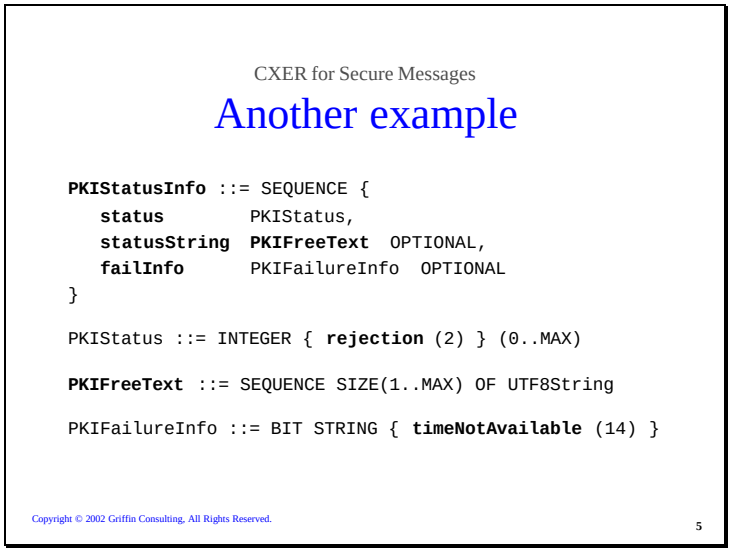
Alternatively, they can display the XML value <AnyName> on both ends of the wire, and transfer (or store) the value in BER

This value BER encodes in 19 bytes (14 PER).
Using XML markup requires 96 bytes - 85 bytes just for tags.

Notice that the markup does not contain:
- any hint of the tags in the ASN.1 definition
- any information on the data types used
- any hint that the middle initial is optional
- any restrictions on the size of the middle initial

All of this additional information must be carried in the schema in the transfer (as in XML) or understood one time only (as in ASN.1) and assumed for all subsequent transfers.

Slide 5



This is a more complex example using a common security type found in the timestamp protocols defined by the IETF, X9, and ISO/IEC JTC 1/SC 27 Security Techniques organizations.

Here again, the highlighted text in the schema definition will be used to form XML markup in a value of PKIStatusInfo.

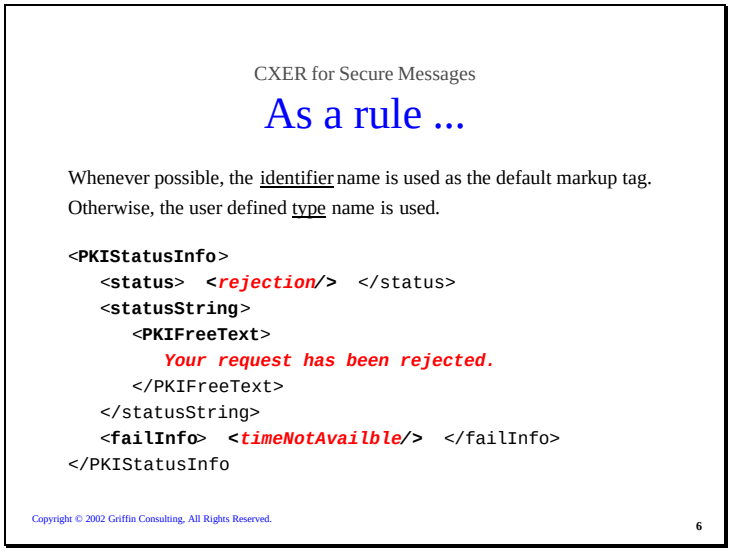
Here, PKIStatusInfo is an outer level type and its name appears in the markup. PKIFreeText is another outer level type used in the markup along with the identifier names status, statusString and failInfo.

The identifiers for the named integer value rejection and the named bit value timeNotAvailable are also used in the markup.

We can't tell from only looking at the markup value that the "<status>" must be an integer greater than or equal to zero.

We can not even tell in markup whether a value of PKIStatus is an integer or a character string. This information can be understood only from looking at the ASN.1 schema.

Slide 6



The names PKIStatusInfo, PKIFreeText, status, statusString, failInfo, rejection and timeNotAvailable from the ASN.1 type definitions are all used as markup tags.

The ASN.1 identifier "rejection" is a value of type PKIStatus, whose definition is based on the builtin ASN.1 type INTEGER.

The ASN.1 identifier " timeNotAvailable" is a value of type PKIFailuerInfo, whose definition is based on the built-in ASN.1 type BIT STRING.

These identifiers, rejection and timeNotAvailable, carry more semantic information to a human reader than their integer and bit values.

So as a rule, ASN.1 identifiers are used in the default markup whenever possible. When no identifier exists, as with the outer level type "PKIFreeText", then the user defined type name is used as the default markup tag.

TO RECAP:

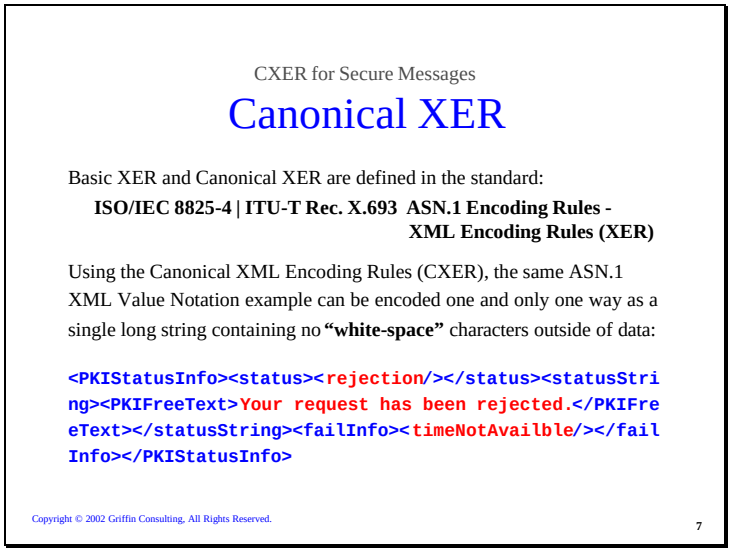
ASN.1 notation is a schema for the structure/types of values.
- No need to use XML DTDs or Schemas to leverage browsers since ASN.1 coder tools enforce the ASN.1 schema. There is no need to validate the XML markup or the values it carries against another XML-based schema.

Every ASN.1 value can be transferred in both text or binary.
- One definition supports both wireles/smartcard and WWW.
  No need for multiple, different schemas which may hinder interoperable solutions.

Every ASN.1 value can be represented using markup.
- This can be processed to create HTML, various file formats and ordinary text, too.

Every ASN.1 type definition has a default markup format.
- Identifier names are used when possible, then type names

Slide 7



X.693 states:

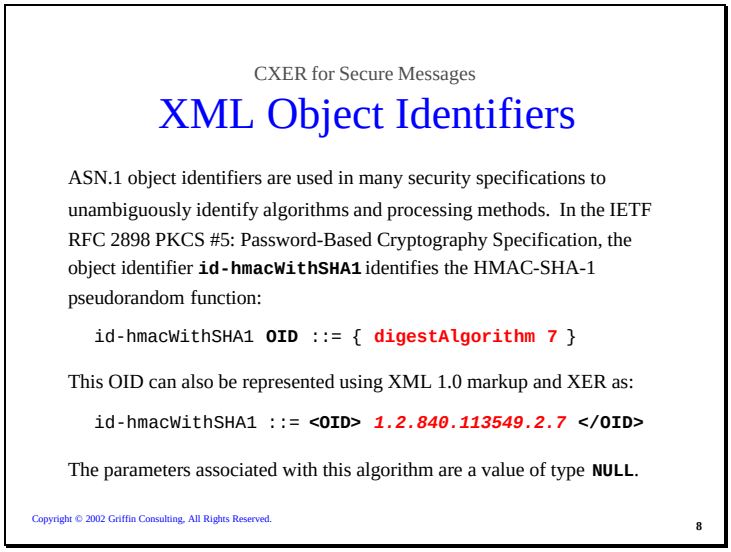
"There is more than one set of encoding rules that can be applied to values of ASN.1 types. This Recommendation | International Standard defines two sets of encoding rules that use the Extensible Markup Language (XML). These are called the XML Encoding Rules (XER) for ASN.1, and both produce an XML document compliant to W3C XML 1.0. The first set is called the Basic XML Encoding Rules. The second set is called the Canonical XML Encoding Rules because there is only one way of encoding an ASN.1 value using these encoding rules. (Canonical encoding rules are generally used for applications using security-related features such as digital signatures.)"

white-space means one or more of the following characters:
HORIZONTAL TABULATION (9),
LINE FEED (10),
CARRIAGE RETURN (13),
SPACE (32).

The numbers in parentheses are the decimal value of the ISO/IEC 10646-1 characters. The number and choice of characters that constitutes "white-space" is an encoder's option using Basic XER.

Clause 9 of X.693 specifies all of the restrictions on XER needed to produce Canonical XER.

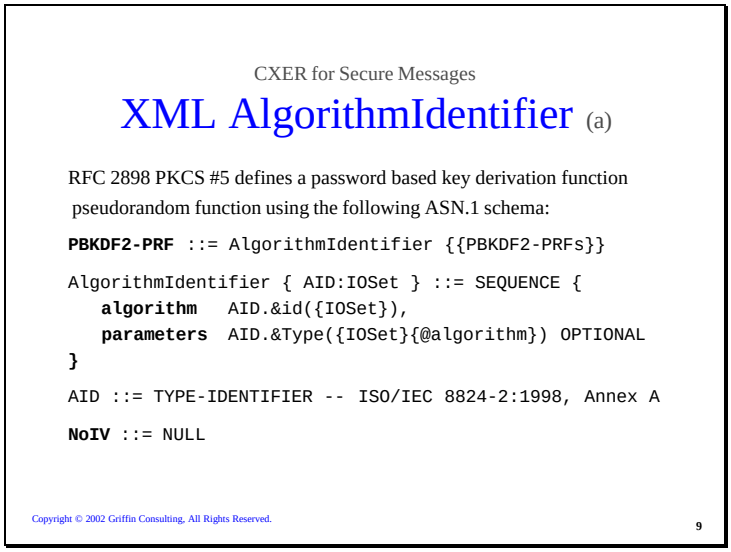
Slide 8



In this example, a defined type OID based on the built-in ASN.1 type OBJECT IDENTIFIER is assumed.

This dotted form of values of ASN.1 type OBJECT IDENTIFIER is the same string format used in the Lightweight Directory Access Protocol (LDAP) progressed by the IETF.

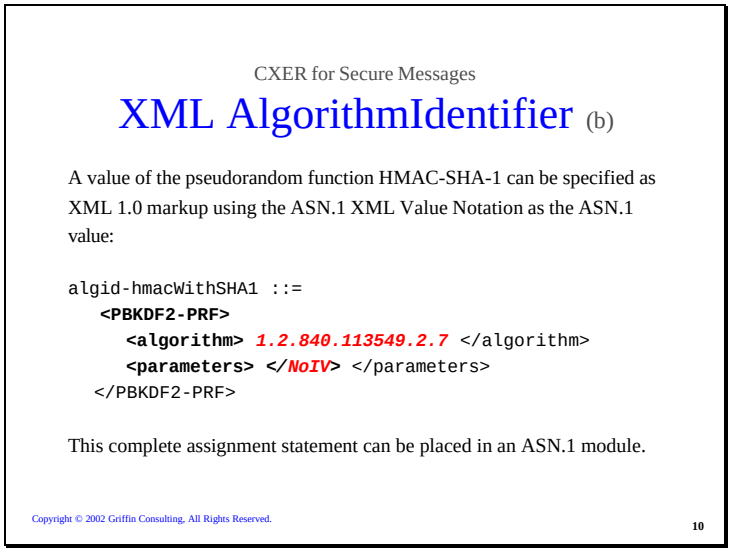
Slide 9



Type AlgorithmIdentifier has been used in many security specifications developed since the mid 1980s.

In this example, it is represented as a parameterized type, whose parameters are a set of valid algorithms. Each algorithm is composed of an algorithm identifier value and associated parameters, which are often a value of type NULL, used to indicate that there are no initialization vectors required by a given algorithm.

Slide 10

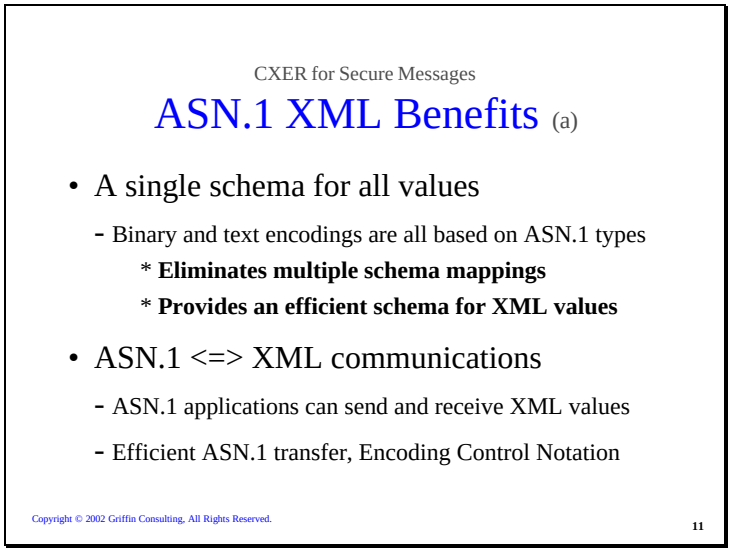


Here again, the highlighted XML markup indicate names that have been derived form the ASN.1 definition of type PBKDF2-PRF.

Notice that a value of type NoIV (NULL) has no content, and need not be represented with a start and end tag pair.

This ASN.1 value can appear in an ASN.1 module. So in some sense, XML 1.0 markup has become valid ASN.1 notation.

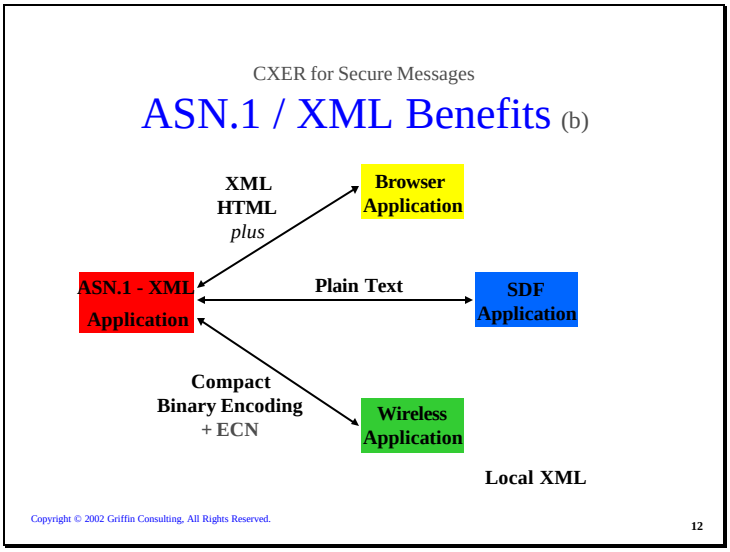
Slide 11



ASN.1 as a simple schema for XML values provides a means of specifying XML values abstractly, independent of their encodings as transfer items.

This means that ASN.1 abstract values can be transferred in a verbose XML markup format, or in a compact binary format.

Slide 12



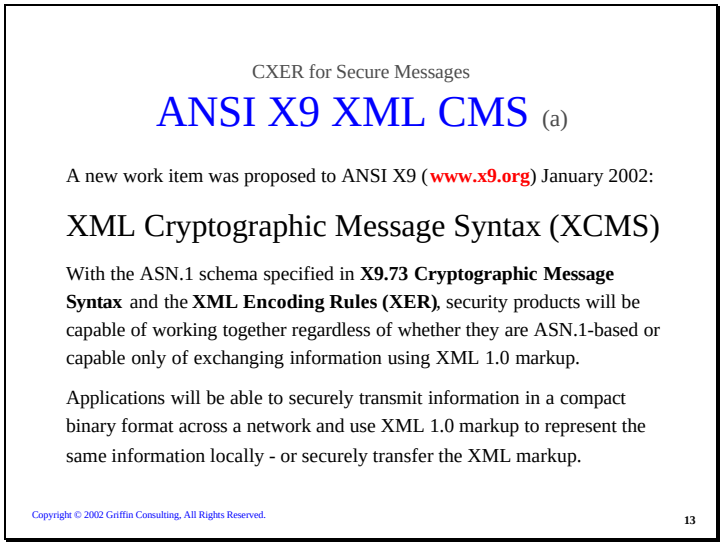
The Browser Application may receive either XML markup, HTML, or compact binary encoded data. For the later, the Browser Application may need to pass XML on to another process, perhaps with one or more style sheets to control formatting for different devices.

The Wireless Application may need to receive compact binary encodings (perhaps PER colored with ECN). But it may need to decode some values into markup for display to a human user.

The SDF Application may only be capable of receiving plain text, such as table input for a spreadsheet or SQL data base. This simple format can be rendered from generated XML markup and a style sheet, perhaps to form a plain text log record or data base query, or to assist in programming development and debugging.

XSL may also be used for ASN.1 message conversion from XML markup into many popular file formats. This can be of benefit in creating documentation for standards writers.

Slide 13



The identical abstract values specified using a binary format in X9.73 and the X9 standards it references, will be defined in a human readable XML format, but afforded the same levels of security provided by X9.73.

The proposed standard will:

- Provide an XML 1.0 markup representation for the message payloads defined in X9.73 including AuthenticatedData, Data, SignedData, EnvelopedData, DigestedData, and EncryptedData

- Provide an XML 1.0 markup representation for the types defined in other X9 standards and referenced in X9.73 including X9.84 BiometricSyntax, X9.68 DomainCertificate, and X9.69 Constructive Key Management extensions

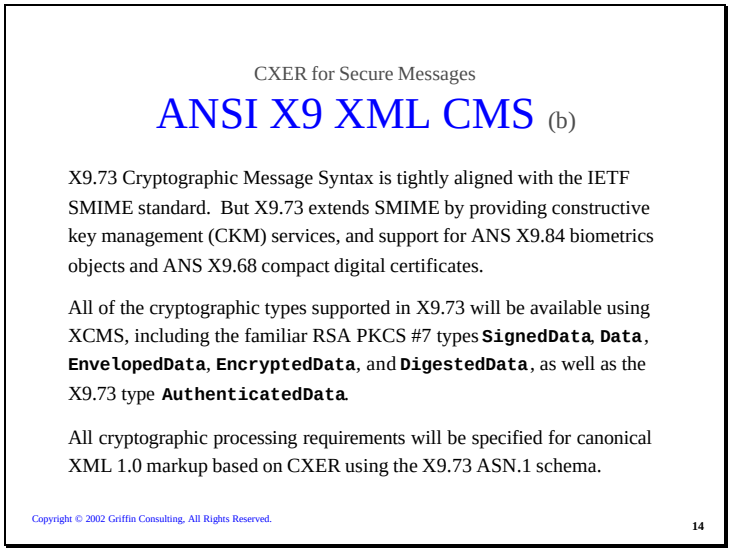
- Provide confidentiality, integrity, origin authentication, and non-repudiation support for XML aware applications which need message encryption, digital signature, MAC, and key management services

- Provide standard tags and structures that can be located without extensive searching of an XML document, and which support multiple signatures, per-signer authenticated attributes, and countersignatures

- Allow selective field protection to be implemented by applications which combine multiple protected messages into a composite message

- Restrict cryptographic operations to only those cryptographic algorithms approved for use by the financial services industry

Slide 14

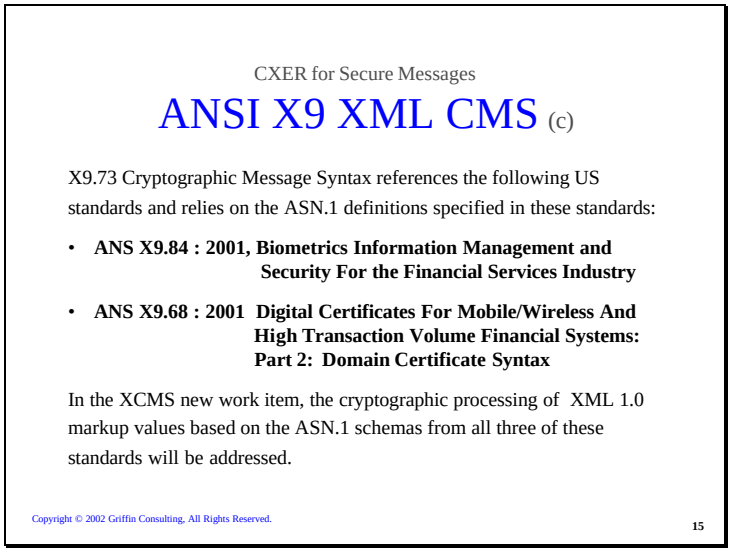


RSA PKCS #7 --> IETF SMIME --> X9.73 CMS

X9.73 references X9.68 and X9.84 and CKM as defined in ANS X9.69-1999: Framework for Key Management Extensions

Using the same ASN.1 definitions for XCMS as used in X9.73 means that the same abstract values are used for both of the canonical ASN.1 encodings. Both senders and receivers will be able to construct identical encodings of values needed to support cryptographically enhanced messages.

Slide 15

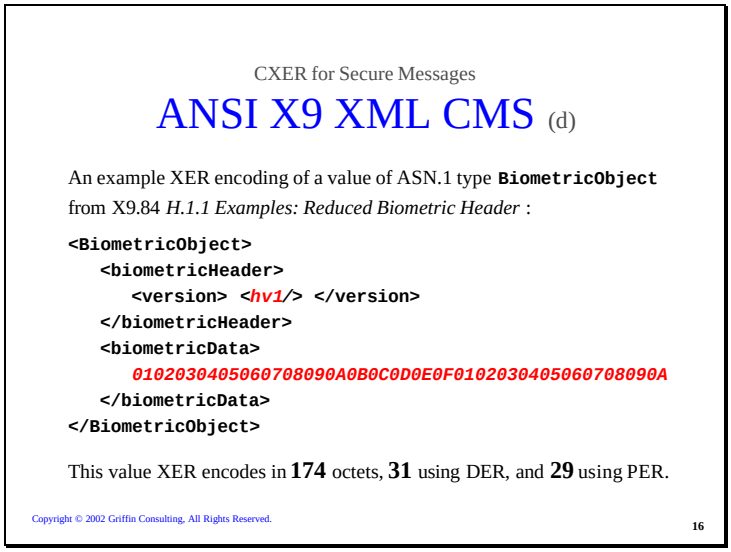


Note that SignedData and the other cryptographic types used in X9.84 are the same types defined in X9.73. These were duplicated in X9.84 so that X9.73 and X9.84 could be progressed as separate standards in different X9F working groups.

X9.84 is the first US national biometrics security standard. Its BiometricObject is a "patron format" in the NIST/ITL Common Biometrics Exchange File Format (CBEFF), a proposed US standard that is expected to be progressed upon passage as an international standard under ISO/IEC JTC 1.

Examples in X9.68 show that this compact binary format can produce digital certificates that support all of the required X.509 extensions specified in ISO 15782-2, a profile of X.509 for the Financial Services industry, in less than 150 bytes. This makes the X9.68 format ideal for resource constrained environments.

Slide 16

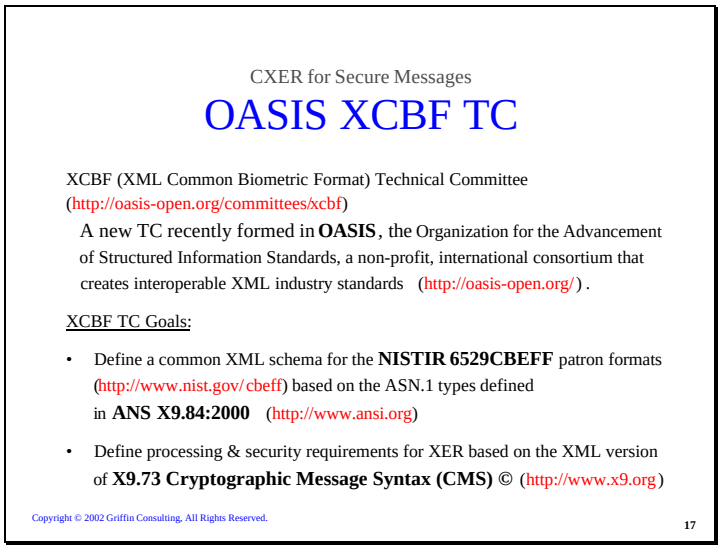


Griffin Consulting will soon announce a tool kit that implements the CBEFF formats and provides an XML enoding for the values of type X9.84. A first version will provide support in Java for raw objects, and subsequent releases of the code will add support for the cryptographic types in X9.84.

This value may be encoded using the Packed Encoding Rules of ASN.1 in 29 octets and stored on a secure card in this compact binary format.

Once handed to an application, the canonical PER encoding can be converted to the XER representation shown here. This format might be processed with one style sheet to create HTML for a display, another style sheet to create a data base query, and still another to create a log record for an audit trail archive.

Slide 17



A new OASIS Technical Committee, XML Common Biometric Format (XCBF) has recently been formed to define a single XML schema to represent the patron formats in the NIST CBEFF standard.

OASIS, the Organization for the Advancement of Structured Information Standards, is a non-profit, international consortium that creates interoperable industry specifications based on public standards such as XML and SGML, as well as others that are related to structured information processing, such as ASN.1.

This Technical Committee will define a common set of XML 1.0 encodings for the patron formats defined in CBEFF, the Common Biometric Exchange File Format (NISTIR 6529, available at http://www.nist.gov/NISTIR-6529-CBEFF). These formats will be specified in accordance with the ASN.1 schema definitions published in ANS X9.84:2000 Biometrics Information Management and Security For The Financial Services Industry© (available from the ANSI electronic bookstore at http://www.ansi.org).

The Common Biometric Exchange File Format defines a common set of data elements necessary to support multiple biometric technologies and to promote interoperability of biometric-based application programs and systems by allowing for biometric data exchange. CBEFF describes a set of "required" and "optional" data fields, a "domain of use", and "CBEFF Patron" formats that utilize some combination of these standard elements.

Patron formats specify encoding of the data elements and any additional (non-common) data elements. The two defined CBEFF Patron formats are the BioAPI Biometric Identification Record (BIR) format specified in the BioAPI Consortium BioAPI

Specification Version 1.0 and the X9.84:2000 BiometricSyntax type. But all of the encoding formats defined in CBEFF and X9.84 are binary encodings, making their use in XML systems and applications limited or difficult.

A common XML schema that can carry the values in all of the CBEFF standard elements will promote the ability to exchange biometric information, between users of binary patron formats, and with XML-aware applications and systems. As new patron formats are added to CBEFF in future revisions, a common XML schema based on the X9.84 definitions should make it possible to exchange data with applications implemented using earlier versions of CBEFF.

Common, underlying type definitions will allow information to be validated and exchanged without ambiguity. The exact same values specified in binary encodings will be used in XML representations of these values. This feature of the work will serve to promote interoperable solutions.

The canonical variant of the XML Encoding Rules (CXER) will produce inputs suitable for cryptographic enhancement of the XML representations of X9.84 biometric objects, but which in that standard are expected to be processed solely in binary. All processing and security requirements used by this TC will be harmonized with standardization of the XML formats of CMS messages undertaken by ANSI X9F working groups.

Proposed list of deliverables and projected dates

* Published document defining the ASN.1/XML CBEFF schema, an introduction and overview of canonical DER, PER, and XER, and the processing and security requirements needed for the creation and verification of all cryptographic types defined in X9.84, in the form of XML encoded objects

* Published working module including ASN.1 XML Markup Value Notation examples of X9.84 biometric types defined in the CBEFF standard, along with the underlying ASN.1 schema

* Published example DER, PER and XER encodings of values of X9.84 biometric types and equivalent BIR encodings

* Published documentation of new CBEFF patron formats that may become standard during the period of this work, in particular an anticipated smart card patron format

* Initial drafts of all deliverables: May 30 2002
* Draft version of working module: November 2002
* Draft version of example binary encodings of XML values: March 2003
* Draft final versions of all remaining deliverables: May 2003
* Final versions of any new CBEFF formats: July 2003
* Final versions of all deliverables: November 2003

In order to achieve maximum interoperability between the efforts of various standards organizations, these XCBF deliverables may depend on work being done by ANSI X9F working groups and the new INCITS M1 Biometrics TC, so the completion of these deliverables may be affected by delays (if any) in the work of these groups. Though the schedule extends into November 2003, it is hoped that this work can be concluded no later than May 2003.

Language in which the TC will conduct business: English. Speakers of other languages are encouraged to participate.

Date, time, and place of first meeting

Monday March 18 - Conference call at 7:00 pm. All interested UBL attendees will meet for the call in the Hotel Barcelona Universal, Barcelona, Spain

Proposed meeting schedule for the first year

* Monday March 18, 2002, 7:00 pm, Barcelona, Spain
* Thursday, May 16, 2002, Somerset, NJ USA.
* Thursday, August 22, 2002, Net Meeting/conference call
* Tuesday, November 19, 2002, Geneva, Switzerland
  (just prior to the ITU-T SG17 meeting)
* Further dates for 2003 to be announced

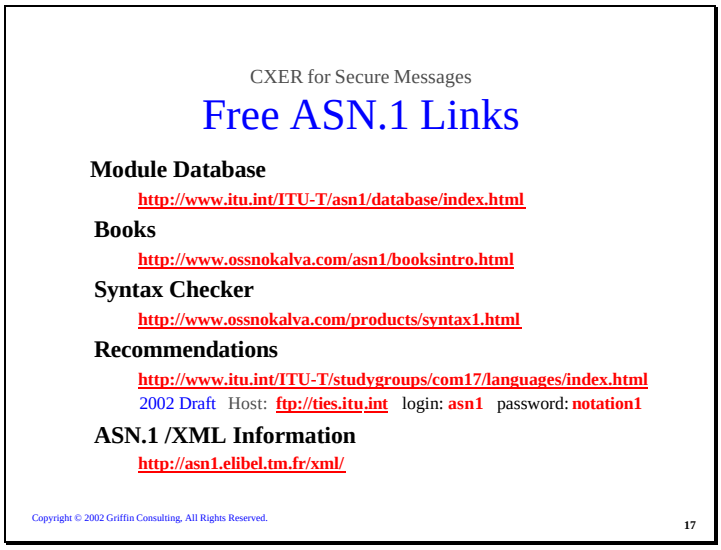
Name of chair

Phillip H. Griffin, Griffin Consulting, <phil.griffin@ASN.1.com>

Name of meeting sponsors (for both phone and face-to-face meetings)

OSS Nokalva and other participants in conjunction with OASIS.

Slide 18



The current ASN.1 standards are available at the link shown in this slide. The 2002 version of the standards are now being progressed. When they complete the balloting process, they too will be made available at this link. The 2002 edition will contain all of the amendments used to specify ASN.1 XML Value Notation and the XML Encoding Rules (XER).

Slide 19

Fin

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

p: **+1 919 291 0019**
f: **+1 919 832 7390**
e: **phil.griffin@ASN-1.com**
w: **http::/ASN-1.com**