

XML
Emerging Authorization and Authentication Standards

Nov. 15, 2001 Mohammad al- Kahtani 1

XML
(Extensible Markup Language)

Nov. 15, 2001 Mohammad al- Kahtani 4

About the speaker	
<ul style="list-style-type: none"> • Name: Mohammad al-Kahtani • Doctoral Candidate at GMU • Educational Background: <ul style="list-style-type: none"> • BS in Computer Science: Riyadh University, Saudi Arabia • Masters in Software Engineering: George Mason University • Contact info: malkahta@gmu.edu 	
<p>Nov. 15, 2001 Mohammad al- Kahtani 2</p>	

XML in brief	
<ul style="list-style-type: none"> • XML is subset of the Standard Generalized Markup Language (SGML) • It is designed to make it easy to interchange structured documents over the Internet. • Structured documents contain: <ol style="list-style-type: none"> 1. Content (words, pictures, etc.) and 2. Some indication of what role that content plays • A markup language is a mechanism to identify structures in a document. 	
<p>Nov. 15, 2001 Mohammad al- Kahtani 5</p>	

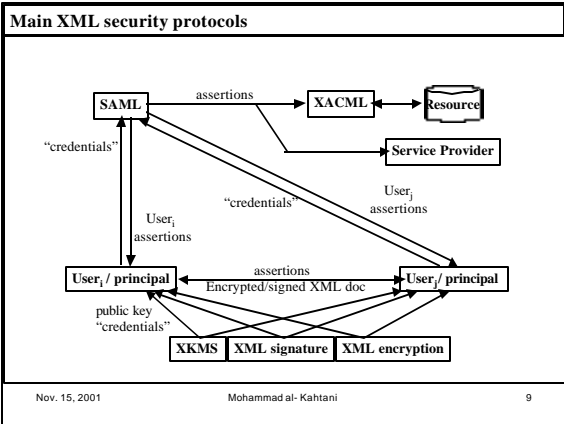
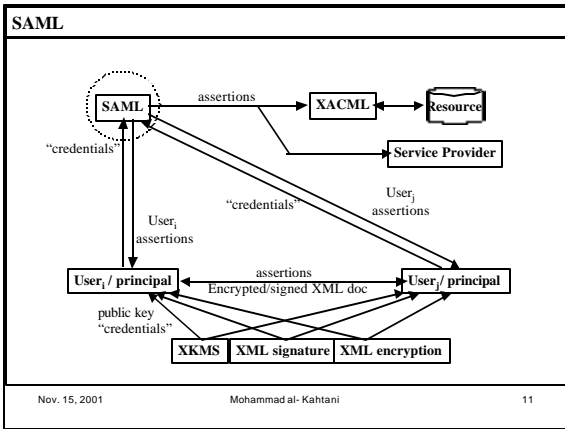
Topics of the Presentation	
<ol style="list-style-type: none"> 1. XML 2. Brief description for main emerging protocols <ol style="list-style-type: none"> a. SAML b. XACML c. XKMS d. XML signature 	
<p>Nov. 15, 2001 Mohammad al- Kahtani 3</p>	

XML and HTML	
<ul style="list-style-type: none"> • In HTML, both the tag semantics and the tag set are fixed: <ul style="list-style-type: none"> • New changes are confined by: <ol style="list-style-type: none"> 1. Browser implementations 2. Backward compatibility is paramount. • XML specifies neither semantics nor a tag set. • XML is really a meta-language for describing markup languages: <ul style="list-style-type: none"> • Provides a facility to define tags and the structural relationships between them. • No predefined tag set means there can't be any preconceived semantics. • All of the semantics of an XML document will either be defined by the applications that process them or by stylesheets. 	
<p>Nov. 15, 2001 Mohammad al- Kahtani 6</p>	

Why XML?		
<ul style="list-style-type: none"> XML was created so that richly structured documents could be used over the web. The only viable alternatives, HTML and SGML, are not practical for this purpose: <ol style="list-style-type: none"> HTML comes bound with a set of semantics and does not provide arbitrary structure. SGML provides arbitrary structure, but is too difficult to implement just for a web browser. 		
Nov. 15, 2001	Mohammad al- Kahtani	7

<h2>SAML</h2> <h3>(Secure Assertion Markup Language)</h3>
<p>Nov. 15, 2001</p> <p>Mohammad al- Kahtani</p> <p>10</p>

An XML example		
<pre> <?xml version="1.0"?> <Country> <Summary> <Geographical_Location> Arabian Penansula </Geographical_Location> <Populations> 14 Million </Populations> <Religion> Islam </Religion> </Summary> <body> </body> </Country> </pre>		
Nov. 15, 2001	Mohammad al- Kahtani	8



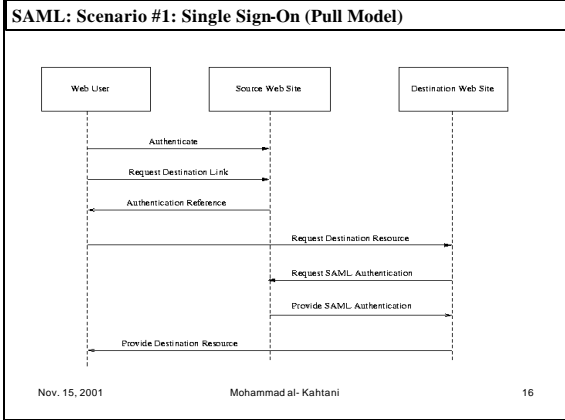
What is SAML		
<ul style="list-style-type: none"> A proposed standard for the exchange of authentication and authorization information between trust domains. SAML enables Single Sign On across trust domains 		
Nov. 15, 2001	Mohammad al- Kahtani	12

SAML Assertions

The **basic data objects** of the SAML protocol model are "Assertions" and "References" (to Assertions).

- 1. Authentication Assertion:** Asserts that the issuer has authenticated the specified subject
- 2. Attribute Assertion:** Asserts that the specified subject has the specified attribute(s).
- 3. Authorization Assertion:** Asserts that a subject has been granted specific permissions to access one or more resources.

Nov. 15, 2001 Mohammad al- Kahtani 13

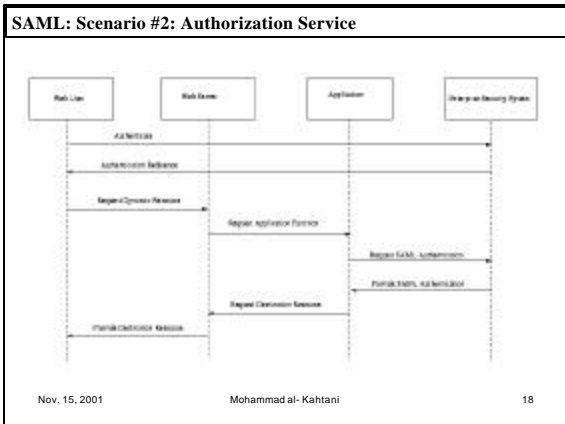
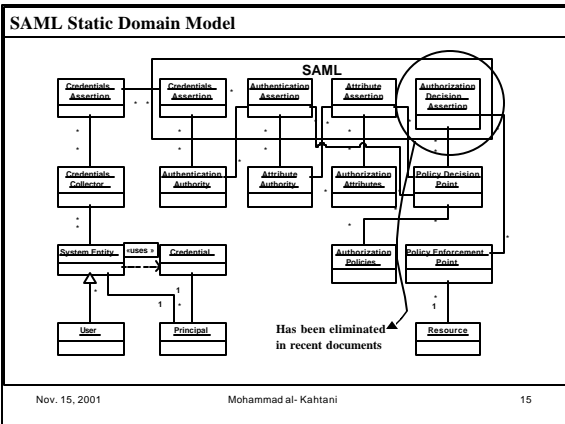
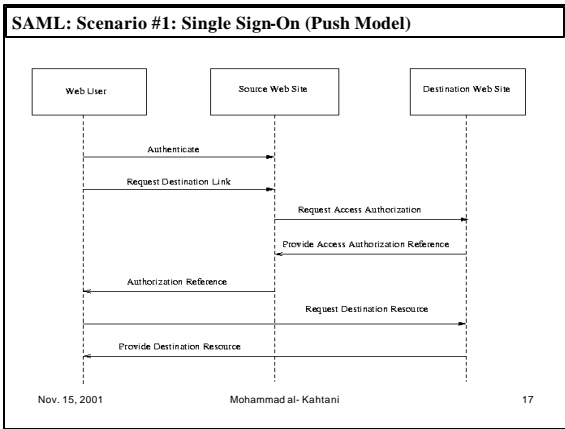


SAML

SAML Assertions may be exchanged using a variety of protocols:

1. The request protocol (defined by the <SAMLQuery> and <SAMLQueryResponse> elements)
2. HTTP
3. SMTP
4. MIME
5. ebXML
6. SOAP/XP
7. BEEP

Nov. 15, 2001 Mohammad al- Kahtani 14




SAML assertion: Basic Info

The different types of SAML assertion are encoded in a common XML package, which consists of:

Basic Information:

- A **unique identifier**: Serves as a name for the assertion.
- SAML version no.
- Date and time of issue: Optional
- Time interval for which the assertion is valid: Optional




Nov. 15, 2001 Mohammad al- Kahtani 19

SAML assertion: Advice

(Optional): Additional information that may be used to specify the assertions that were used to make a policy decision.

The Advice element is a **general container for any additional information** that does not affect the semantics or validity of the assertion itself.




SAML assertion package

Nov. 15, 2001 Mohammad al- Kahtani 22

SAML assertion: Claims

Describe the **use** of assertions to make claims for **Authorization**

1. "DecisionClaim": Access permissions specified in the request identified by the corresponding RequestID were either permitted, denied or could not be determined
2. "AuthenticationClaim": Specified subject has been authenticated
3. "AttributeClaim" element: Specified subject has the specified attribute(s) specified by a URI
4. "AuthorizationClaim" : Specified subject is authorized to perform the specified operation(s) on the specified resource(s).



Nov. 15, 2001 Mohammad al- Kahtani 20

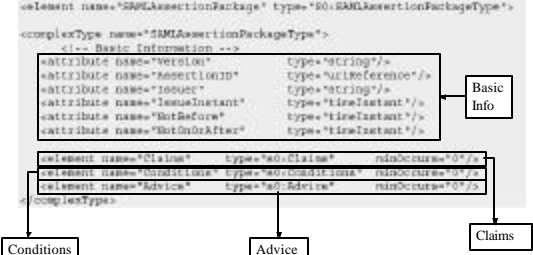
SAML AssertionPackage

The <SAMLAssertionPackage> element is specified by the following schema:

```

<element name="SAMLAssertionPackage" type="S0:SAMLAssertionPackageType">
  <complexType name="SAMLAssertionPackageType">
    <!-- Basic Information -->
    <attribute name="version" type="string"/>
    <attribute name="assertID" type="URIReference"/>
    <attribute name="issue" type="string"/>
    <attribute name="issueInstant" type="timeInstant"/>
    <attribute name="notBefore" type="timeInstant"/>
    <attribute name="notOnOrAfter" type="timeInstant"/>
    <element name="Claims" type="S0:Claims" minOccurs="0"/>
    <element name="Conditions" type="S0:Conditions" minOccurs="0"/>
    <element name="Advice" type="S0:Advice" minOccurs="0"/>
  </complexType>

```




Nov. 15, 2001 Mohammad al- Kahtani 23

SAML assertion: Conditions

(Optional): The assertion status may be dependent on:

- Additional information from a validation service.
- Other assertions being valid.

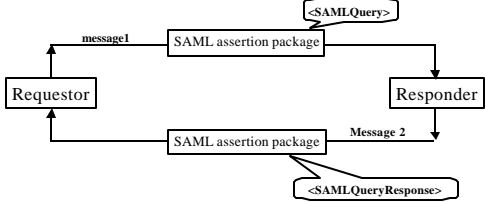


SAML assertion package

Nov. 15, 2001 Mohammad al- Kahtani 21

SAML protocol

If a suitable assertion already exists, then that assertion may be returned in response to the request, without the responder having to create a new one.



Nov. 15, 2001 Mohammad al- Kahtani 24

XACML (Extensible Access Control Markup Language)

Nov. 15, 2001

Mohammad al- Kahtani

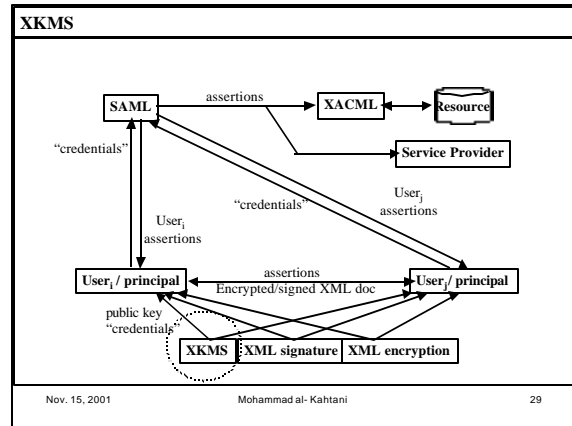
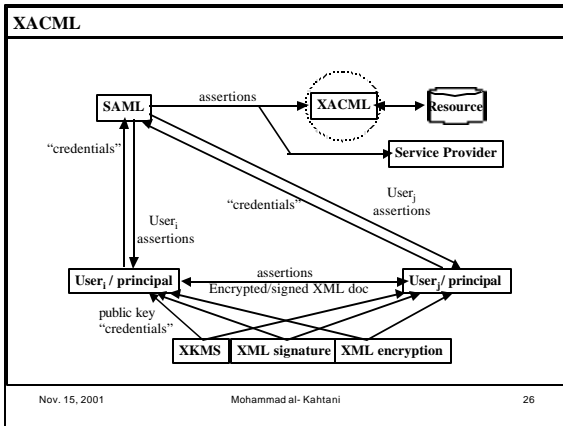
25

XKMS XML Key Management Specification

Nov. 15, 2001

Mohammad al- Kahtani

28



What is XACML

- It is an XML specification for expressing **policies** for **information access** over the Internet
- XACML targets any object that can be **referenced** using XML
- XACML allows the assignment of privileges **directly** to users
- XACML does **not** specify the action primitive at all
- XACML specifications document has not been released yet

Nov. 15, 2001

Mohammad al- Kahtani

27

What is XKMS

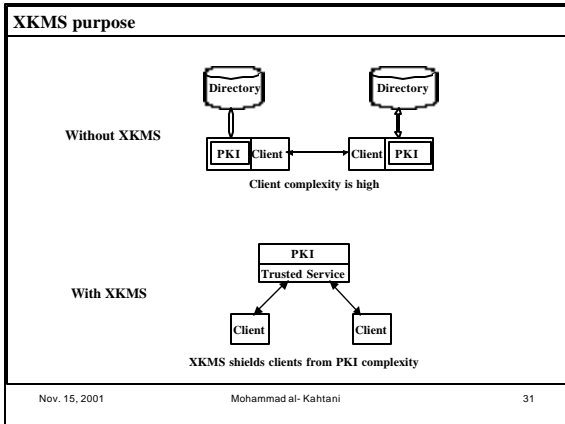
A protocol for:

- Distributing
- Registering public keys

Nov. 15, 2001

Mohammad al- Kahtani

30



XKMS: Tiered Service Model

KeyInfo: Optional element that enables the recipient's to obtain the key needed to validate the signature.

- KeyInfo may contain:
 1. Keys
 2. Names
 3. Certificates

Tiers	<ds:keyinfo> processing	<ds:keyinfo> Validation
Tier 0	Done by Application	NA
Tier 1	Done by trust service	Done by Application
Tier 2	Done by trust service	Done by trust service

XKMS provided

Nov. 15, 2001 Mohammad al- Kahtani 34

XKMS Components

- The XML Key Information Service Specification (X-KISS)
- The XML Key Registration Service Specification (X-KRSS).

X-KISS
X-KRSS

Nov. 15, 2001 Mohammad al- Kahtani 32

X-KISS : Tiered Service Model

Tier 0:

- XKMS is **not** deployed
- The client pulls PKI info from PKI server

Nov. 15, 2001 Mohammad al- Kahtani 35

XKMS sub-protocol: X-KISS

- Allows a client to delegate part or all of the tasks required to process XML Signature to a Trust service.
- The underlying PKI may be based upon a different specification such as X.509/PKIX, SPKI or PGP.

Nov. 15, 2001 Mohammad al- Kahtani 33

X-KISS : Tiered Service Model

Tier 1: (Key Locating service)

- A client receives a signed XML document
- The client requests the trust server to obtain the public key parameters

Protected by XML Signature, SSL, IPSEC

Done only if info is not available in Server A

Nov. 15, 2001 Mohammad al- Kahtani 36

X-KISS : Tiered Service Model

Tier 2: (Key Validating service)

- A client receives a signed XML document
- The client queries the trust server to determine whether the signing key is trustworthy.
- The Trust Service builds a certificate trust path, then validates each certificate in the path against the relevant CRL

The diagram shows a Client sending a 'Query' to a Trust Service. The Trust Service sends 'Get info' to a PKI Service, which returns 'Info'. The Trust Service then returns 'Validity, Key Value & binding' to the Client. A note indicates that this process is 'Done if info provided by client is not enough for the binding'. A box labeled 'Protected by XML Signature, SSL, IPSEC' is also shown.

Nov. 15, 2001 Mohammad al- Kahtani 37

XML Signature

The diagram illustrates the XML Signature process. A SAML entity sends 'assertions' to XACML, which in turn sends 'assertions' to a Resource. A Service Provider sends 'User_j assertions' to the Resource. A User_i / principal sends 'User_i assertions' to SAML and 'credentials' to the Resource. The Resource sends 'User_j assertions' to the User_i / principal. The User_i / principal sends 'assertions' to another User_i / principal, which then sends an 'Encrypted/signed XML doc' back to the first User_i / principal. A box at the bottom indicates 'XKMS XML signature XML encryption'.

Nov. 15, 2001 Mohammad al- Kahtani 40

XKMS sub-protocol: X-KRSS

- X-KRSS permits management of information that is bound to a public key pair
- 2 ways to generate a public key pair:
 1. In advance by the client, or
 2. On request by the service (to support key **recovery**)
- Services provided:
 1. Registration
 2. Revocation
 3. Key Recovery

Nov. 15, 2001 Mohammad al- Kahtani 38

Scope

- XML signature is composed of:
 1. Syntax used for representing the signature of Web resources (anything referenceable by a URI)
 2. Procedures for computing and verifying such signatures.
- XML Signatures are generated from a hash over a signature manifest (a collection of references to the objects being signed)
- XML signature does not address mechanisms for making statements or assertions.

Nov. 15, 2001 Mohammad al- Kahtani 41

XML Signature

Nov. 15, 2001 Mohammad al- Kahtani 39

XML signature

- The XML Signature data structures must be based on the RDF data model
- XML Signatures apply to any resource addressable by a locator including non-XML content
- XML Signatures may apply to a part or totality of an XML document
- XML Signatures are first class objects themselves and consequently must be able to be referenced and signed

Nov. 15, 2001 Mohammad al- Kahtani 42

Signature "element"		
<p>XML digital signatures are represented by the Signature element which has the following structure</p> <pre> <Signature> <SignedInfo> (CanonicalizationMethod) (SignatureMethod) (<Reference (URI=)? > (Transforms)? (DigestMethod) (DigestValue) </Reference>+ </SignedInfo> (SignatureValue) (KeyInfo)? (Object)* </Signature> </pre>		
Nov. 15, 2001	Mohammad al- Kahtani	43

Signature Process: Core Generation		
<p>1. Core Generation</p> <p>a) Reference Generation: Creating a Reference element for each data object to be signed</p> <p>b) Signature Generation</p>		
<pre> graph LR A[Data object] -- transform --> B[Transformed Data object] B -- Calculate digest value --> C[Data object digest] C -- Create a Reference element --> D[Reference element] E[Include: Id of the data object, transform elements, digest algorithm, DigestValue] --> D </pre>		
Nov. 15, 2001	Mohammad al- Kahtani	46

Signature element example		
<pre> [s01] <Signature Id="MyFirstSignature" xmlns="http://www.w3.org/2000/09/xmldsig#"> [s02] <SignedInfo> [s03] <CanonicalizationMethod Algorithm="http://www.w3.org/TR/2001/REC-xmldsig-core1-20010315"/> [s04] <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1"/> [s05] <Reference URI="http://www.w3.org/TR/2000/REC-xhtml1-20000126/"> [s06] <Transforms> [s07] <Transform Algorithm="http://www.w3.org/TR/2001/REC-xmldsig-core1-20010315"/> [s08] </Transforms> [s09] <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/> [s10] <DigestValue>j6w3rvEPOvKiMup4NbeVu8nk=</DigestValue> [s11] </References> [s12] <SignedInfo> [s13] <SignatureValue>MC0CFrVLIrIK...</SignatureValue> [s14] <KeyInfo> [s15a] <Key Value> [s15b] <DSAKey Value> [s15c] <P>...</P><Q>...</Q><G>...</G><Y>...</Y> [s15d] <DSAKey Value> [s15e] <Key Value> [s16] <Key Info> [s17] </Signature> </pre> <p>The key to be used to validate the signature</p> <p>Info that is actually signed</p>		
Nov. 15, 2001	Mohammad al- Kahtani	44

Signature Process: Core Generation		
<p>1. Core Generation (continues)</p> <p>a) Reference Generation</p> <p>b) Signature Generation</p>		
<pre> graph LR A[Reference element] -- Specify: SignatureMethod, Specify: CanonicalizationMethod --> B[SignedInfo element] B -- Canonicalize and then Calculate SignatureValue --> C[Signature Value] C -- Create Signature element --> D["SignedInfo {References} , KeyInfo SignatureValue"] D --> E[Signature element] </pre>		
Nov. 15, 2001	Mohammad al- Kahtani	47

Signature Process		
<p>1. Core Generation</p> <ul style="list-style-type: none"> Reference Generation Signature Generation <p>2. Core Validation</p> <ul style="list-style-type: none"> Reference Validation Signature Validation 		
Nov. 15, 2001	Mohammad al- Kahtani	45

Signature Process: Core Validation		
<p>2. Core Validation</p> <p>a) Reference Validation: The verification of the digest contained in each Reference in SignedInfo</p> <p>b) Signature Validation</p>		
<pre> graph TD A["SignedInfo {References} KeyInfo SignatureValue"] -- CanonicalizationMethod --> B[Reference element] B -- transform --> C[Data Object] C -- Obtain data object to be digested --> D[Data object digest] D -- Calculate digest value --> E[Data object digest] E -- Compare digest value --> F[Reference validation result] </pre>		
Nov. 15, 2001	Mohammad al- Kahtani	48

Topics of the Presentation

2. Core Validation

a) Reference Validation

b) Signature Validation: Cryptographic signature validation of the signature calculated over SignedInfo

```

    graph TD
      SignedInfo["SignedInfo (References)  
KeyInfo  
SignatureValue"] -- "Obtain canonical form" --> CanonicalForm["Canonical form"]
      SignedInfo -- "Obtain key info" --> CanonicalForm
      CanonicalForm -- "Confirm SignatureValue" --> SignatureResult["Signature validation result"]
      SignedInfo -- "Confirm SignatureValue" --> SignatureResult
  
```

The diagram illustrates the signature validation process. It starts with a 'Signed element' containing 'SignedInfo (References)', 'KeyInfo', and 'SignatureValue'. Two arrows labeled 'Obtain canonical form' and 'Obtain key info' point from the SignedInfo box to a 'Canonical form' box. A third arrow labeled 'Confirm SignatureValue' points from the Canonical form box to a 'Signature validation result' box. A fourth arrow, also labeled 'Confirm SignatureValue', points directly from the SignedInfo box to the Signature validation result box.

Nov. 15, 2001 Mohammad al- Kahtani 49

XML security protocols and Framework for Security Engineering

The diagram shows a vertical stack of four boxes: 'Objective', 'Model', 'Architecture', and 'Mechanism'. To the left of this stack is the text 'XACML/XKMS/ XML signature' and to the right is 'SAML'. Double-headed vertical arrows connect the stack to both 'XACML/XKMS/ XML signature' and 'SAML', indicating a bidirectional relationship or mapping between these security frameworks and the underlying architectural layers.

Nov. 15, 2001 Mohammad al- Kahtani 50

References

1. XML Base: W3C Recommendation 27 June 2001; (<http://www.w3.org/TR/2001/REC-xmlbase-20010627/>)
2. An Introduction to the Extensible Markup Language (XML) by Martin Bryan of The SGML Centre (<http://www.personal.u-net.com/~sgml/xmlintro.htm>)
3. A Technical Introduction to XML by Norman Walsh (<http://nwalsh.com/docs/articles/xml/>)
4. SAML Specifications June 20, 2001. 'draft-sstc-ff3-saml-spec-00' version (A very poorly-written unstable document): (<http://xml.coverpages.org/draft-sstc-ff3-saml-spec-00.pdf>)
5. SAML Specification Version 00 draft-sstc-saml-spec-00.doc (No date was specified): (<http://www.oasis-open.org/committees/security/docs/draft-sstc-saml-spec-00.PDF>)
6. XACML: (<http://www.oasis-open.org/committees/xacml/docs/>)
7. XML Key Management Specification (XKMS), 30 March 2001: (<http://www.w3.org/TR/xkms/>)
8. XML Key Management Specification (XKMS) by Phillip M. Hallam-Baker & Warwick Ford (VeriSign Inc.): (<http://www10.org/cdrom/posters/1129.pdf>)
9. XML-Signature Syntax and Processing: (<http://www.w3.org/TR/2001/PR-xmlsig-core-20010820/>)

Nov. 15, 2001 Mohammad al- Kahtani 51