

BITS, Pilani – Dubai Campus, Knowledge Village, Dubai.
IVth Year Second Semester 2005-2006
Degree: B.E.(Hons) Branch: C.S.E.

Course No : EA UC441 Web services and internet based distributed computing
Maxmarks=70

Time:3hrs

Part-A Answer any 10 questions –answering and marking scheme

(All questions carry equal marks)=10*2

Q1. Outline the need for XML based soap messages for invoking web services rather than going for simple RPC calls.

- Web Services were intended to solve three main problems:
 1. Interoperability
 2. Firewall traversal
 3. minimal Complexity

Drawbacks of typical RPC calls have to be explained.

Q2.I want to delete a particular service (alone) hosted by a business in a UDDI. Outline the appropriate UDDI soap message to do the same

Using a business key ,and the service of the web service that has to be deleted along with a soap message to the UDDI delete_service have to be briefed.

Q4.Outline what is meant by CPP in ebxml

The first step towards that is to request the **ebXML** Specs (Business Processes, Business Scenarios) and understand them. Once the organization has taken a look at the specs, it decides which business processes it would like to implement, following which it needs to implement a system in-house based on those standards. It could either build a new system or build on top of an existing legacy system. The whole idea is to expose a system that understands and talks **ebXML**. There are several choices available today in the form of third-party applications that can just take and assemble together an **ebXML** System. Once the system is built, the organization is ready to conduct business with other organizations. To facilitate that, it needs to publish its profile known as a Collaboration Protocol Profile (CPP) to the **ebXML** Repository for other organizations to discover. A CPP thus enables any organization to describe its profile i.e. which business processes it supports, its roles in that process, the messages exchanged, the transport mechanism for the messages. Once the CPP is published to the **ebXML** Repository, it will allow other organizations to access it and learn about the capabilities of Organization A. At any time, Organization A is free to access its own profile, review and make changes as necessary.

Q5. Outline the differences between UDDI registry and ebxml registry. Limitations of UDDI just as giving details about invoking web service only- Ebxml is a repository- CPP, business processes published- ebXML enables business to business collaboration- UDDI just for invoking a web service.

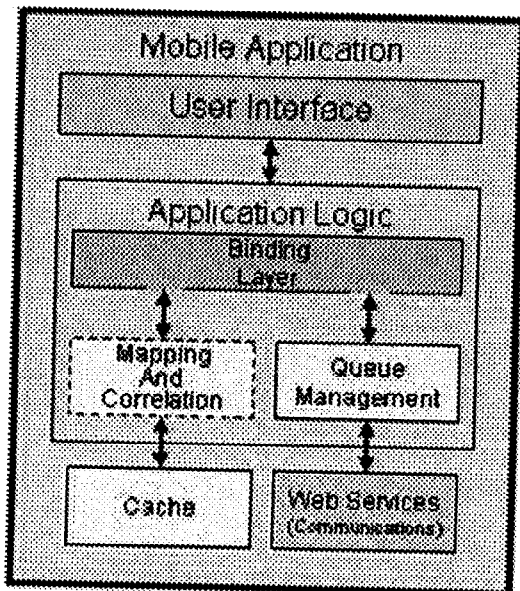
Q6. What are the advantages of XML schema over DTD?

With XML schemas, you have more power to define what valid XML documents look like.

They have several advantages over DTDs:

- **XML schemas use XML syntax.** In other words, an XML schema is an XML document.
- **XML schemas support datatypes.** While DTDs do support datatypes, it's clear those datatypes were developed from a publishing perspective. XML schemas support all of the original datatypes from DTDs. They also support integers, floating point numbers, dates, times, strings, URLs, and other datatypes useful for data processing and validation.
- **XML schemas are extensible.** In addition to the datatypes defined in the XML schema specification, you can also create your own, and you can derive new datatypes based on other datatypes.
- **XML schemas have more expressive power.** For example, with XML schemas you can define that the value of any <state> attribute can't be longer than 2 characters, or that the value of any <postal-code> element must match the regular expression $[0-9]\{5\}(-[0-9]\{4\})?$. You can't do either of those things with DTDs.

Q7. How the components of the computing model for occasionally connected applications differ from conventional client server applications?



Q8. Outline the need for qualifiers when you search information from UDDI with appropriate examples.

With appropriate examples outline how using qualifiers we can perform refined search in the UDDI-examples of using qualifiers along with UDDI soap calls like find-business have to be given

Q9. what is the difference in retrieving the contents of a XML element when you use DOM/SAX parser ?

In SAX during the invocation of callback functions the information about the contents of the element have to be provided-whereas in DOM first a tree of the parsed XML document is formed-by navigating the tree we can find the contents of the element

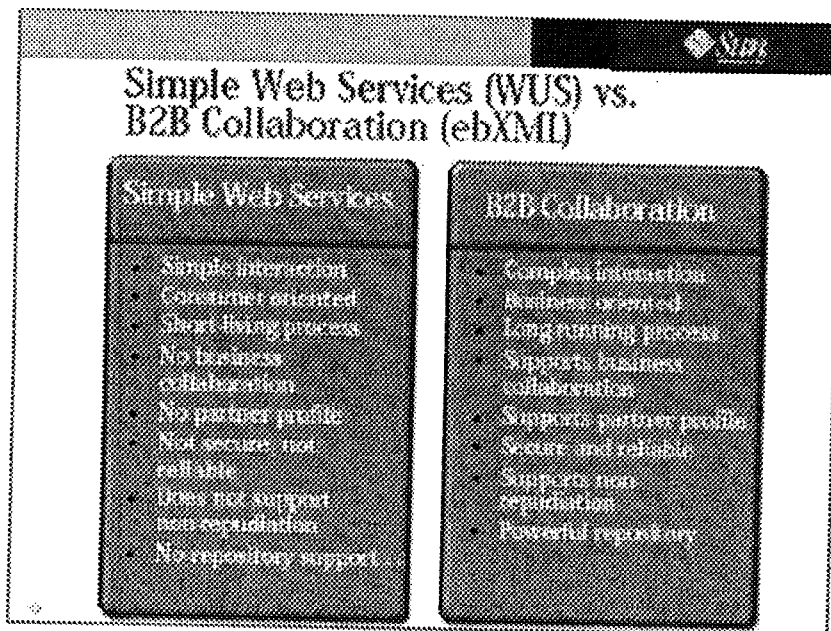
Q10. Outline the relative merits and demerits of hosting a XSLT engine in a server or browser side.

In the server side-based on the type of request server can identify whether the request is from mobile, or desktop browsers and will give a html or WAP to display the contents of XML using appropriate styles-Client need not have advanced browsers to implement XSLT engine-But the server will be overloaded due to presence of XSLT engine

Q11. What is the need for CPA in ebxml ? Justify whether CPA will be published in ebxml registry.

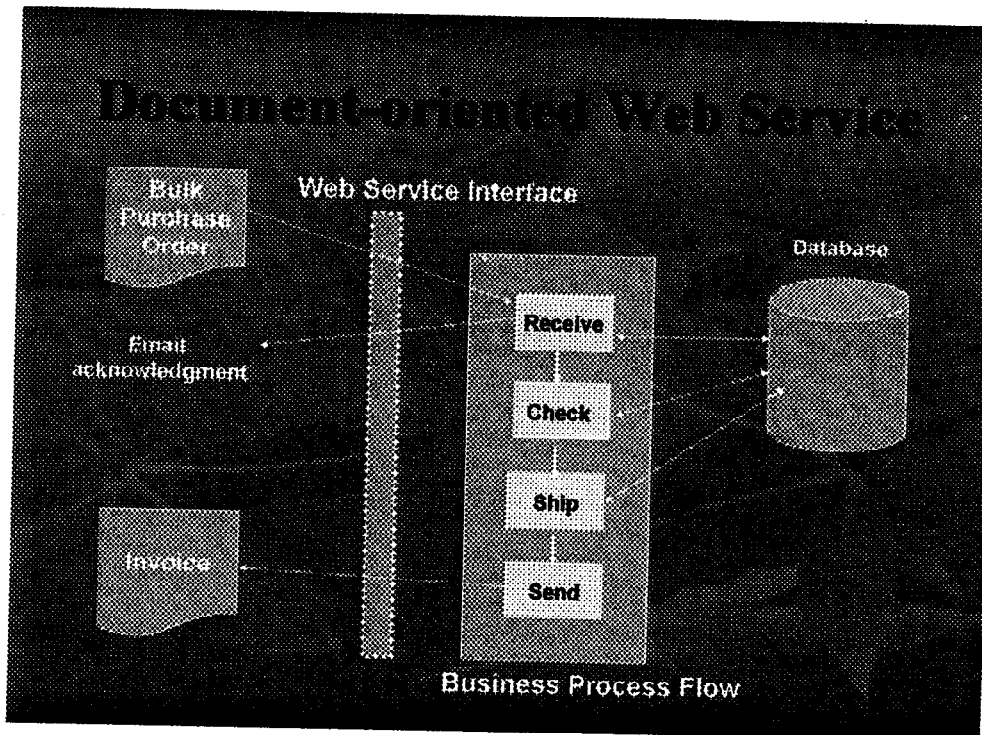
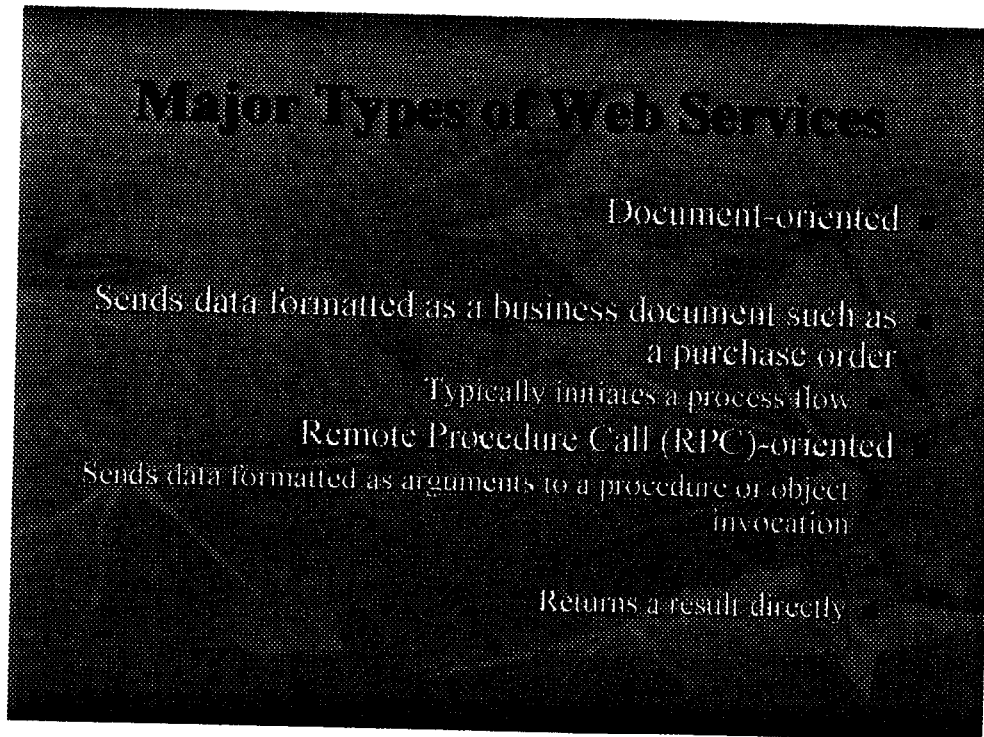
It is a collaborative protocol agreement that is derived after the agreement on the transport of ebxml message, security of the ebxml message like non-repudiation etc-brief outline of the information that is conveyed through CPA have to be provided-after CPA is exchanged then actual communication takes place between the two businesses

Q12. Outline the differences between simple web services and ebxml Simple web service.



Part-B Answer all the questions
(All questions carry equal marks =5*10)

Q1.a) Outline with relevant examples the difference between RPC centric and document-centric web services. (4M)



Similarly using diagram s simple RPC-based web service has to be explained

b) Suppose that the owner of a chain of coffee houses, called The Coffee Break, wants to expand the line of coffees that he sells. He instructs his business manager to find some new coffee suppliers, get their wholesale prices, and then arrange for orders to be placed as the need arises. The Coffee Break can analyze the prices from coffee suppliers and decide which new coffees it wants to carry and which companies it wants to buy them from. Assume that the coffee break communicates with suppliers using appropriate web services. The business manager in coffee break assigns the task to the company's software engineer. Outline with relevant interaction diagrams how using web services along with the UDDI the above objectives can be met. Make necessary assumptions. (6M)

Suppose that the owner of a chain of coffee houses, called The Coffee Break, wants to expand the line of coffees that he sells. He instructs his business manager to find some new coffee suppliers, get their wholesale prices, and then arrange for orders to be placed as the need arises. The Coffee Break can analyze the prices and decide which new coffees it wants to carry and which companies it wants to buy them from. The business manager assigns the task to the company's software engineer, who decides that the best way to locate new coffee suppliers is to search a Universal Description, Discovery, and Integration (UDDI) registry, where The Coffee Break has already registered itself. The engineer uses JAXR to send a query searching for wholesale coffee suppliers. JAXR sends messages using JAXM in the background, which ensures that the registry will be able to receive and understand it.

The UDDI registry will receive the query and apply the search criteria transmitted in the JAXR code to the information it has about the organizations registered with it. When the search is completed, the registry will send back information on how to contact the wholesale coffee distributors that met the specified criteria .

The engineer's next step is to draft a request for price lists and send it to each of the coffee distributors using JAXM. She writes an application that gets a connection to the company's messaging service so that she can send the requests. She then creates a JAXM message, adds the request, and sends it.

Each coffee distributor receives the request, and before sending out current prices, checks with its stock quote service using JAX-RPC to get the latest quotes for the relevant coffee futures. Based on the figures they get back, the distributors send The Coffee Break their newly revised prices in an XML price sheet. The vendors use an agreed upon XML schema for their price sheets because that way they can use a format that is convenient for them and that their buyers can process easily

The above interactions have to be explained with the help of diagrams.

Q2.a) Outline how a client can locate the WSDL document of a web service?

First the client will search for the business in the UDDI . Then it will find out the service hosted by the business. Once it knows the service it can get the service key. Using that it can find the binding details as well as tmodelInstance Info details. Using this it can find the URL of the wsdl document . After downloading the wsdl document it can form the soap message to invoke the web service.

(4M)

b)

?xml version="1.0" encoding="UTF-8"?>

```

<definitions name="HelloService"
  targetNamespace="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns="http://schemas.xmlsoap.org/wsdl/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:tns="http://www.ecerami.com/wsdl/HelloService.wsdl"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <message name="SayHelloRequest">
    <part name="firstName" type="xsd:string"/>
  </message>
  <message name="SayHelloResponse">
    <part name="greeting" type="xsd:string"/>
  </message>

  <portType name="Hello_PortType">
    <operation name="sayHello">
      <input message="tns:SayHelloRequest"/>
      <output message="tns:SayHelloResponse"/>
    </operation>
  </portType>

  <binding name="Hello_Binding" type="tns:Hello_PortType">
    <soap:binding style="rpc"
      transport="http://schemas.xmlsoap.org/soap/http"/>
    <operation name="sayHello">
      <soap:operation soapAction="sayHello"/>
      <input>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded"/>
      </input>
      <output>
        <soap:body
          encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"
          namespace="urn:examples:helloservice"
          use="encoded"/>
      </output>
    </operation>
  </binding>

  <service name="Hello_Service">
    <documentation>WSDL File for HelloService</documentation>
    <port binding="tns:Hello_Binding" name="Hello_Port">
      <soap:address
        location="http://localhost:8080/soap/servlet/rpcrouter"/>
    </port>
  </service>
</definitions>

```

With respect to the above WSDL description extract the relevant information to invoke the web service.

The name of the service is Hello_service. The method supported by the service is sayHello-parameter is a string-web service is RPC oriented-url of the web service <http://localhost:8080/soap/servlet/rpcrouter>

It uses SOAP over http.

POST /Hello_service HTTP/1.1

Host: localhost:8080

Content-Type: application/soap+xml; charset=utf-8 Content-Length: 150

```
<?xml version="1.0"?>
<soap:Envelope
xmlns:soap="http://www.w3.org/2001/12/soap-envelope"
soap:encodingStyle="http://www.w3.org/2001/12/soap-encoding">
```

```
  <soap:Body xmlns:m="http://www.stock.org/stock">
```

```
    <m:Sayhello>
```

```
      <m:firstname>IBM</m:firstname>
```

```
    </m:Sayhello>
```

```
  </soap:Body>
```

```
</soap:Envelope>
```

e soap response Similarly th

..have to be generated

(6M)

Q3.a) Can you encrypt a non-xml document using xml encryption. Justify with the help of an example. (4M)

Encrypting non-xml data

```
• <?xml version='1.0' ?>
• <EncryptedData xmlns='http://www.w3.org/2001/04/xmlenc#'
•   Type='http://www.isi.edu/in-otes/iana/assignments/media-types/jpeg' >
•   <CipherData>
•     <CipherValue>A23B45C56</CipherValue>
•   </CipherData>
• </EncryptedData>
```

b) Using XML based digital signing I want to digitally sign the following documents present in the web.

- <http://www.abccompany.com/index.html> would reference an HTML page on the Web
- <http://www.abccompany.com/logo.gif> would reference a GIF image on the Web
- <http://www.abccompany.com/xml/po.xml> would reference an XML file on the Web

Outline the steps involved in digitally signing all the above documents (at one step) using xml digital signature. (6M)

Calculate the digest of each resource.

In XML signatures, each referenced resource is specified through a `<Reference>` element and its digest (calculated on the identified resource and not the `<Reference>` element itself) is placed in a `<DigestValue>` child element like

```
<Reference URI="http://www.abccompany.com/news/2000/03_27_00.htm">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1" />
  <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
</Reference>
<Reference
  URI="http://www.w3.org/TR/2000/WD-xmlsig-core-20000228/signature-
  example.xml">
  <DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1"/>
  <DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>
</Reference>
```

The `<DigestMethod>` element identifies the algorithm used to calculate the digest.

3. Collect the Reference elements

Collect the `<Reference>` elements (with their associated digests) within a `<SignedInfo>` element like

```
SignedInfo Id="foobar">
  <CanonicalizationMethod
    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
  <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmlsig#dsa-
  sha1" />
  <Reference URI="http://www.abccompany.com/news/2000/03_27_00.htm">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1" />
    <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
  </Reference>
  <Reference
    URI="http://www.w3.org/TR/2000/WD-xmlsig-core-20000228/signature-
    example.xml">
    <DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1"/>
    <DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>
  </Reference>
</SignedInfo>
```

The `<CanonicalizationMethod>` element indicates the algorithm was used to canonize the `<SignedInfo>` element. Different data streams with the same XML information set may have different textual representations, e.g. differing as to whitespace. To help prevent inaccurate verification results, XML information sets must first be canonized

before extracting their bit representation for signature processing. The `<SignatureMethod>` element identifies the algorithm used to produce the signature value.

4. Signing

Calculate the digest of the `<SignedInfo>` element, sign that digest and put the signature value in a `<SignatureValue>` element.

```
<SignatureValue>MC0E LE=</SignatureValue>
```

5. Add key information

If keying information is to be included, place it in a `<KeyInfo>` element. Here the keying information contains the X.509 certificate for the sender, which would include the public key needed for signature verification.

```
<KeyInfo>
  <X509Data>
    <X509SubjectName>CN=Ed Simon,O=XMLSec
    Inc.,ST=OTTAWA,C=CA</X509SubjectName>
    <X509Certificate>MIID5jCCA0+gA...lVN</X509Certificate>
  </X509Data>
</KeyInfo>
```

6. Enclose in a Signature element

Place the `<SignedInfo>`, `<SignatureValue>`, and `<KeyInfo>` elements into a `<Signature>` element. The `<Signature>` element comprises the XML signature.

```
<?xml version="1.0" encoding="UTF-8"?>
<Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
  <SignedInfo Id="foobar">
    <CanonicalizationMethod
      Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
    <SignatureMethod
      Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
    <Reference URI="http://www.abccompany.com/news/2000/03_27_00.htm">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
      <DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</DigestValue>
    </Reference>
    <Reference
      URI="http://www.w3.org/TR/2000/WD-xmldsig-core-20000228/signature-
      example.xml">
      <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <DigestValue>UrXLDLBIta6skoV5/A8Q38GEw44=</DigestValue>
    </Reference>
  </SignedInfo>
  <SignatureValue>MC0E~LE=</SignatureValue>
  <KeyInfo>
    <X509Data>
```

```
<X509SubjectName>CN=Ed Simon,O=XMLSec  
Inc.,ST=OTTAWA,C=CA</X509SubjectName>  
<X509Certificate>  
MIID5jCCA0+gA...1VN  
</X509Certificate>  
</X509Data>  
</KeyInfo>  
  
</Signature>
```

Q4.a) With relevant example outline the need for XKMS web service .(4M)

XML encryption and digital signatures rely on Public Key Infrastructure (PKI) to help encrypt, decrypt, sign, and verify various documents. For an application to use XML encryption and/or digital signature, the application must use or integrate with a PKI solution. Various PKI solutions are available, such as X.509 (the most widely used), Pretty Good Privacy (PGP), Simple Public Key Infrastructure (SPKI), and Public Key Infrastructure X.509 (PKIX). The question is, which is the right one to use or integrate with?

Unfortunately, the answer is not as simple as choosing one of the many available PKIs and integrating with it. The real issue is how many PKIs -- along with their syntax and semantics -- must the application be aware of for it to be able to talk to other applications that may send or expect documents encoded or signed using other available PKI solutions. For example, if organization A uses an X.509 PKI solution and sends encrypted documents to organization B, which uses an SPKI PKI solution, then organization B won't be able to decrypt and use the document sent by A. For A and B to work together, one of them has to understand the other's PKI solution. If you extrapolate this scenario to a situation where multiple partners are involved, it becomes clear that all of the partners will have to be aware of each other's PKI solution, thus increasing each application's complexity many times.

b)With relevant examples outline the services provided by X-KISS and X-KRSS of XKMS.(6M)

XML Key Registration Service Specification (XKRSS)

This part of the XKMS deals with the mechanism for registering a key pair with a service provider. You can register keys with an XKMS service in two ways:

- The client generates a key pair and provides the public key, along with other information, to the service provider for registration.
- The XKMS service generates a key pair for the client, registers the public key of the pair with itself, and sends the private key of the pair to the client for its use.

The client can also tell the XKMS service to keep the private key as well. The private key is kept with the XKMS service in case the client loses its private key.

An XKRSS service specification defines four operations:

- **Register:** Information is bound to a key pair through key binding. During registration, either the client provides the public key, along with some proof of possession of the corresponding private key, or the service provider generates the key pair for the client. The service provider may request more information from the client before it registers the public key (and optionally the private key as well).
- **Reissue:** A previously registered key binding is reissued. New credentials in the underlying PKI are generated using this operation. While there is no lifespan for the key binding information used by XKMS, the credentials issued by the underlying PKI occasionally do have a time span that must be renewed periodically.
- **Revoke:** This operation allows clients to destroy the data objects to which a key is bound. For example, an X.509 certificate that's bound to an XKMS key is destroyed when this operation is called.
- **Recover:** This operation allows clients to recover the private key. For this operation to be meaningful, the private key must have been registered with the service provider. One of the ways in which the service provider may have the private key is when the key pair is generated at the server rather than the client.

XML Key Information Service Specification (XKISS)

This part of XKMS addresses the mechanism that allows client applications to authenticate encrypted/signed data.

The client authenticates the encrypted/signed data by passing the corresponding key information to the service provider. The service provider then responds with "true" or "false." "True" indicates that the public key corresponding to the private key used for signing does belong to the entity that claims to have signed the docs.

The XKISS service specification defines the following two operations:

- **Locate:** Locate resolves a `<ds:KeyInfo>` element that may be associated with XML encryption or XML signature, but it does not prove the validity of data binding in the `<ds:KeyInfo>` element.
- **Validate:** This operation does all that locate does, plus more. The locate service finds a key based on the `<ds:KeyInfo>` element, but does not assure the trustworthiness of the key binding information. The validate operation not only searches the public key corresponding to the `<ds:KeyInfo>` element, but also assures that the key binding information that it returns is trustworthy.

Q5.a) Outline the need for enabling single-sign on on to multiple web sites and web services and how it can be done using SAML.(4M)

having to remember dozens of Website usernames and passwords surely surpasses all other inconveniences. Whether you wish to chat on *JavaWorld's* forums or keep up with the news on your favorite current affairs Website, chances are you will be asked at some point to enter your username and password. Following closely behind that aggravation are the bothersome HTML forms you must fill out to obtain your user IDs in the first place.

Username and password boxes are annoying because they present an experience without close parallel in the off-line world

The Liberty Alliance Project specifications offer a possible solution to the first two of those limitations. The next-generation Liberty specifications, version 2.0 due in a few months, hope to solve the third one as well. What represents a mere inconvenience for Websites may well be a showstopper for Web services. You can almost always count on a human to punch her username and password into a Web browser. But Web services do not enjoy the benefits of human supervision: they must often invoke other Web services on their own. Even if Web services execute on behalf of a human user, they can't stop and fetch their anthropomorphic master for user access information. Rather, either they must have a way to obtain access information automatically or Web services acting together to support a common goal must trust one another prior to service-to-service invocations.

b) 1. Assume that authenticated users of company.com need access to protected resources at Travel.com in order to make travel arrangements
Company.com users need not reauthenticate to Travel.com.

. 2. Authenticated users of the company.com want to make use of an internal purchasing system to place orders for office supplies from Supplier.com.
Supplier.com need to know if the user is authorized for the purchase.

Outline how using SAML the above two scenarios can be realized. (6M)

Travel.com will initiate assertion request and company.com will give assertion response. Suitable XML based examples should illustrate the assertion request/assertion response. Similarly authorization request/response should be illustrated for supplier.com Vs Travel.com communication along with appropriate XML based messages.

BITS, Pilani – Dubai Campus, Knowledge Village, Dubai.

IV Year Second Semester 2005-2006

Degree: B.E.(Hons) Branch: C.S.E.

EAUC 441 Web services and internet based distributed computing

Date:14-5-2006 Openbook Time:50 mts

Answer all the questions (answering scheme)

Q1. a) With relevant examples outline clearly the needs for category bag and identifier bag when we try to publish a business/webservice in UDDI ? (3M)

With relevant example outline how using category bags businesses can be classified under NAICS or geographical categories.- significance of each of the category in simplified search of the business/service-how using identifier bag (DUNS based number)we can uniquely identify the business

b)Outline the difference in usage scenario of findbusiness and getbusinessdetails ?(3M)

Using relevant example outline how findbusiness depending upon the query will give rise to multiple businesses during a search criteria-whereas using getbusinessdetails which accept business key as a parameter enables to get details about the services hosted by the business.

Q2. Assume that IBM has published a web service called stockquote web service in the UDDI registry. List out the steps involved in locating the business/service and invoking the web service. Make necessary assumptions and briefly outline the soap messages involved during the above process. (6M)

Using findbusiness first try to locate the ibm business using appropriate qualifiers in the UDDI registry and get the business key-Using the business key findout the services hosted by the business and get the service keys including that of stockquote web service- once the service key is obtained get binding details for the stockquote web service-from the binding details get tmodelinstancedetails-using tmodel details you can get the wsdl document-using wsdl document you can form the soap request and soap response.all the above steps with brief explanation of soap messages in every step.

Q3. With relevant case studies outline when you go for EAI and when you go for ebxml .based development.(6M)

EAI is used to integrate various modules in a business process using web services-using sales department, purchase order processing dept and marketing dept outline how using simple web services EAI can be done-also need for EAI also has to be broughtout Ebxml for business to business collaboration have to be empasised with simple Example.(sender of purchase order/receiver of purchase order)

Q4.a)Outline the difference in usage of UDDI and ebxml registry with relevant examples(3M)

Uddi just for publishing web services in a centralized place-with an example illustrate the same- ebxml as a repository of business process, CPP,corecomponents have more features compared to uddi-brief explanation

b) What are the special requirements of ebxml messaging compared to conventional message driven web services?(3M)

conventional messaging for web service not reliable-by default no security on the otherhand ebxml messaging very reliable-by default security has to be enabled-why ?-using appropriate scenarios difference between the two types of messaging have to be explained.

Q5a).Outline clearly the difference in SSL encryption and XML encryption(3M)

SSL encryption takes place during transit because only when a computer wants to send packet to another computer at that time only network software will be invoked-it does not address the scenario of a hacker logging into the machine and steal the non encrypted information whereas using xml encryption we can encrypt the xml document and store it Safely-further there are variety of document encryption techniques are possible in xml encryption

b)Given a XML document as follows.

How the encrypted data will look when you go for complete encryption ?

Suppose you want to do encryption of element Account. How it will look like after encrypted?(3M)

```
<PatientRecord
  xmlns="http://www.medical.org/"
  xmlns:lab="http://www.lab.org/">
  <Name>John Doe</Name>
  <Account>123456</Account>
  <Visit date="10pm March 10, 2002">
    <Diagnosis>Broken second metacarpal</Diagnosis>
    <lab:Diagnosis>
      <lab:Xray>encoded xray image</lab:Xray>
    </lab:Diagnosis>
  </Visit>
</PatientRecord>
```

After encryption the encrypted document will have encrypted data tags, cipherdata tags,cipher value tags –these tags will replace the account element-also the type element in encrypteddata will specify whether an element or the content of the element has to be encrypted-

BITS, Pilani – Dubai Campus, Knowledge Village, Dubai.
IV Year Second Semester 2005-2006

Degree: B.E.(Hons) Branch: C.S.E.

EAUC 441 Web services and internet based distributed computing

Date:14-5-2006 Openbook Time:50 mts

Answer all the questions (answering scheme)

Q1. a) With relevant examples outline clearly the needs for category bag and identifier bag when we try to publish a business/webservice in UDDI ? (3M)

With relevant example outline how using category bags businesses can be classified under NAICS or geographical categories.- significance of each of the category in simplified search of the business/service-how using identifier bag (DUNS based number)we can uniquely identify the business

b)Outline the difference in usage scenario of find business and getbusinessdetails ?(3M)
Using relevant example outline how findbusiness depending upon the query will give rise to multiple businesses during a search criteria-whereas using getbusinessdetails which accept business key as a parameter enables to get details about the services hosted by the business.

Q2. Assume that IBM has published a web service called stockquote web service in the UDDI registry. List out the steps involved in locating the business/service and invoking the web service. Make necessary assumptions and briefly outline the soap messages involved during the above process..(6M)

Using findbusiness first try to locate the ibm business using appropriate qualifiers in the UDDI registry and get the business key-Using the business key findout the services hosted by the business and get the service keys including that of stockquote web service-once the service key is obtained get binding details for the stockquote web service-from the binding details get tmodelinstancedetails-using tmodel details you can get the wsdl document-using wsdl document you can form the soap request and soap response.all the above steps with brief explanation of soap messages in every step.

Q3. With relevant case studies outline when you go for EAI and when you go for ebxml .based development.(6M)

EAI is used to integrate various modules in a business process using web services-using sales department, purchase order processing dept and marketing dept outline how using simple web services EAI can be done-also need for EAI also has to be broughtout Ebxml for business to business collaboration have to be empasised with simple Example.(sender of purchase order/receiver of purchase order)

Q4. a)Outline the difference in usage of UDDI and ebxml registry with relevant examples(3M)

Uddi just for publishing web services in a centralized place-with an example illustrate the same- ebxml as a repository of business process, CPP, corecomponents have more features compared to uddi-brief explanation

BITS, Pilani – Dubai Campus, Knowledge Village, Dubai.
IV Year Second Semester 2005-2006

Degree: B.E.(Hons) Branch: C.S.E.

EAUC 441 Web services and internet based distributed computing

Date:14-5-2006 Openbook Time:50 mts

Test 2

Answer all the questions

Q1.a)With relevant examples outline clearly the needs for category bag and identifier bag when we try to publish a business/webservice in UDDI ? (3M)

b)Outline the difference in usage scenario of find business and getbusinessdetails ?(3M)

Q2.Assume that IBM has published a web service called stockquote web service in the UDDI registry. List out the steps involved in locating the business/service and invoking the web service. Make necessary assumptions and briefly outline the soap messages involved during the above process..(6M)

Q3. With relevant case studies outline when you go for EAI and when you go for ebxml .based development.(6M)

Q4.a)Outline the difference in usage of UDDI and ebxml registry with relevant examples(3M)

b) What are the special requirements of ebxml messaging compared to conventional message driven web services?(3M)

Q5a).Outline clearly the difference in SSL encryption and XML encryption(3M)

b)Given a XML document as follows.

How the encrypted data will look when you go for complete encryption ?

Suppose you want to do encryption of element Account. How it will look like after encrypted ?(3M)

```
<PatientRecord
  xmlns="http://www.medical.org/"
  xmlns:lab="http://www.lab.org/">
  <Name>John Doe</Name>
  <Account>123456</Account>
  <Visit date="10pm March 10, 2002">
    <Diagnosis>Broken second metacarpal</Diagnosis>
    <lab:Diagnosis>
      <lab:Xray>encoded xray image</lab:Xray>
    </lab:Diagnosis>
  </Visit>
</PatientRecord>
```


b) What are the special requirements of ebxml messaging compared to conventional message driven web services?(3M)

conventional messaging for web service not reliable-by default no security on the otherhand ebxml messaging very reliable-by default security has to be enabled-why ?-using appropriate scenarios difference between the two types of messaging have to be explained.

Q5a).Outline clearly the difference in SSL encryption and XML encryption(3M)

SSL encryption takes place during transit because only when a computer wants to send packet to another computer at that time only network software will be invoked-it does not address the scenario of a hacker logging into the machine and steal the non encrypted information whereas using xml encryption we can encrypt the xml document and store it Safely-further there are variety of document encryption techniques are possible in xml encryption

b)Given a XML document as follows.

How the encrypted data will look when you go for complete encryption ?

Suppose you want to do encryption of element Account. How it will look like after encrypted?(3M)

```
<PatientRecord
  xmlns="http://www.medical.org/"
  xmlns:lab="http://www.lab.org/">
  <Name>John Doe</Name>
  <Account>123456</Account>
  <Visit date="10pm March 10, 2002">
    <Diagnosis>Broken second metacarpal</Diagnosis>
    <lab:Diagnosis>
      <lab:Xray>encoded xray image</lab:Xray>
    </lab:Diagnosis>
  </Visit>
</PatientRecord>
```

After encryption the encrypted document will have encrypted data tags, cipherdata tags,cipher value tags –these tags will replace the account element-also the type element in encrypteddata will specify whether an element or the content of the element has to be encrypted-