2 Related Work

This chapter is divided into six part to discuss the relative technology and paper which we apply them to our research. The first part is the foundation in the XML-based technique. The second is the core in this paper. The third and fourth are the infrastructure of the web services. The final part is about the security in the web services. Figure 5 shows the architecture of the related work.

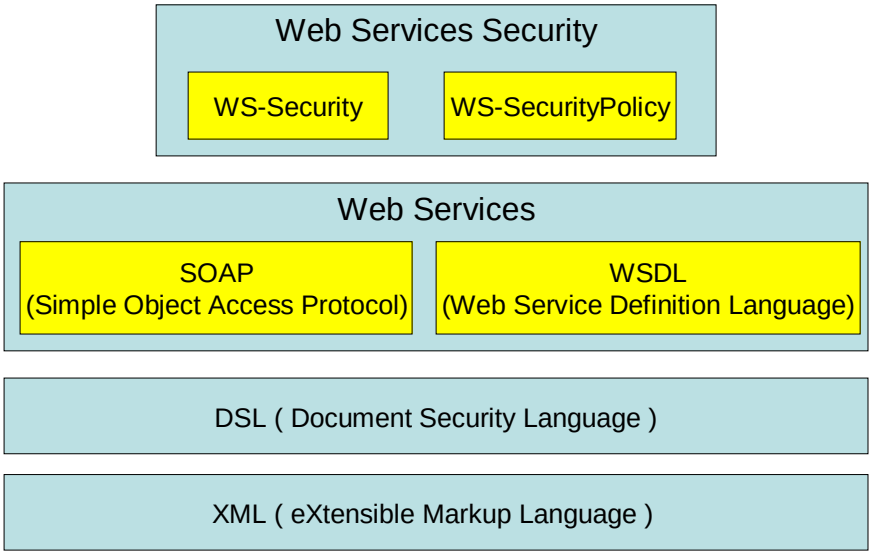

Figure 5 : The architecture of the related work

2.1. XML (eXtensible Markup Language)

XML is a markup metalanguage that was standardized by the Word Wild Web Consortium (W3C, http://www.w3.org). The main purpose of XML is to carry out data exchange between heterogeneous platforms among organizations, particularly systems connected via the Internet. XML is derived from SGML(Standard Generalized Markup Language), and it was developed to overcome the limitations of the HTML (Hypertext Markup Language). As a markup language, XML uses tags to mark a pieces of data. Each tag assigns meaning to the data associated with it. Moreover, XML is

self-describing; that is, XML not only organizes text into a hierarchy but also describes its organization directly in the text . Figure 6 shows which data is the book information and where each book begins and ends.

```
<?xml version=”1.0” encoding=”UTF-8”?>
<Books>
   <Book id="sp0001">
      <Author>John</Author>
      <BookName>Web services security</BookName>
      <Amount>21</Amount>
   </Book>
   <Book id="sp0002">
      <Author>Mary</Author>
      <BookName>Web services platform</BookName>
      <Amount>13</Amount>
    </Book>
</Books>
```

Figure 6 : The structure of XML

2.2. DSL (Document Security Language)

The purpose of the DSL is to achieve the following features : (1) the security of the XML document is ensured not by the system but by the document itself, and (2) the securing tool is configurable.

In order to accomplish these two features, a DSL document defines the syntax of the secured document to encrypt, decrypt, sign ,and verify a XML document. The structure of a DSL document is composed of five sections : key definition section, algorithm section, security pattern section, transformation description section, and digital

signature section.

1. Key definition section: It defines the type of the key, including key name, key type , and the location.
2. Algorithm section: It defines which algorithm will be executed in the DSL securing tool (see Figure 8).
3. Security pattern section: It is used to specify the combination of security algorithms and encryption and decryption keys.
4. Transformation description section: It takes the defined keys, algorithms, and security patterns in the security pattern definition to specify the actual data transformation of element-wise encryption.
5. Digital signature section: It provides users with the ability to specify how to embed digital signatures in the resulting XML document.

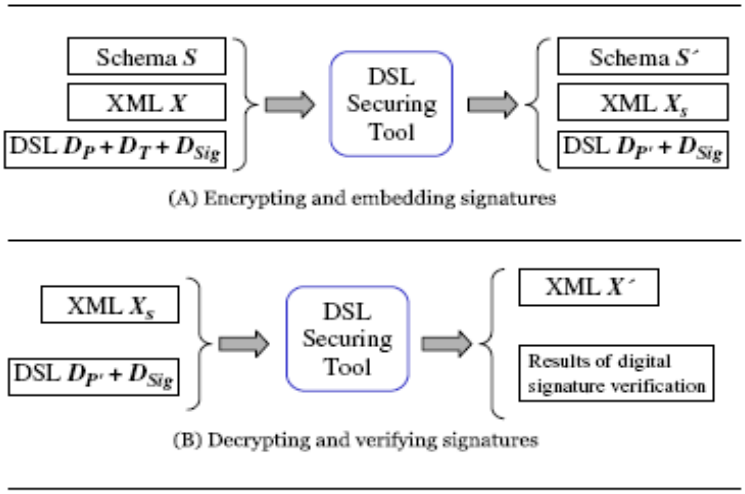

Figure 7 : The operational model for securing XML documents [11]

Because of the structure, the DSL can easily supply a security mechanism that integrates element-wise encryption and temporal-base element-wise digital signatures. In addition, the DSL also provide a security operational mode (see Figure 7) to

construct an automatic and a configurable security for XML.

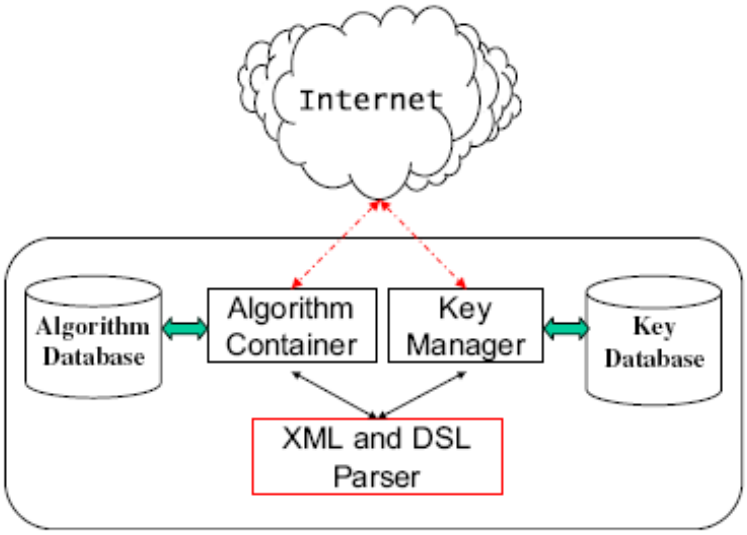

Figure 8 : The organization of the DSL securing tool [11]

2.3. SOAP (Simple Object Access Protocol)

The SOAP 1.1 specification is submitted to W3C in May 2000. Because the function is insufficient, the W3C working group produced the SOAP 1.2 Recommendation in June 2003. A SOAP message is a kind of XML document. SOAP gives a standard container to send its XML payload over any transport protocol. The most common way to exchange SOAP messages is via HTTP (Hypertext Transfer Protocol). In addition, SOAP messages can also be carried by e-mail using SMTP (Simple Mail Transfer Protocol) and by other network protocols, such as FTP (File Transfer Protocol) and TCP/IP (Transmission Control Protocol/Internet Protocol).

A SOAP message consist of a SOAP envelope which contains zero or more SOAP headers. The main purpose of the SOAP header is to carry information that can be used in the processing or routing of the message payload, such as a digital signature to guarantee the integrity of payload data or encrypted information to promise the

confidentiality of a message. The SOAP envelope also contains a SOAP body which includes the message payload or business information. Figure 9 shows the basic structure of a SOAP message.

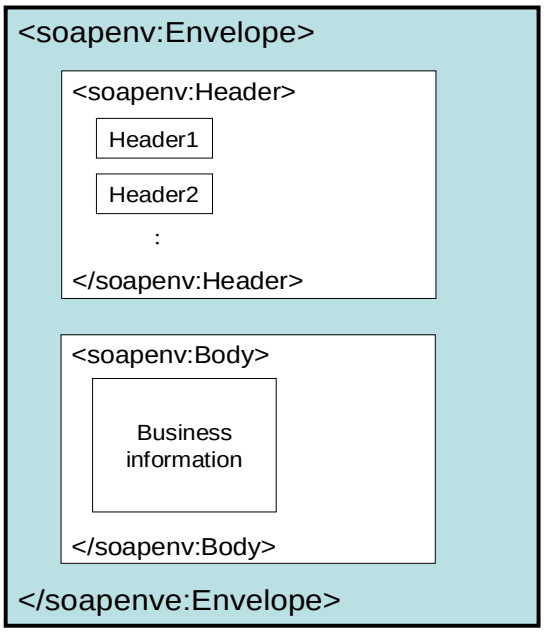

Figure 9 : The basic structure of a SOAP message

2.4. **WSDL (Web Service Description Language)**

The Web Service Description Language is an XML-based language. The latest version is 2.0. Although WSDL 2.0 is more simple and less powerful than WSDL 1.1, it probably does not become widely deployed for several years because WSDL 1.1 is now firmly entrenched in the developer community.

The main purpose of the WSDL is used to describe a service for its clients and a standard for service implementers. A WSDL 1.1 document (see Figure 10) contains six important elements: types, message, portType, operations, binding, and service, which are nested in the definitions element which the root element of a WSDL document. In the next paragraph, we will explain the function of each element briefly.

1. <types> : It is used to declare complex data types and elements.

2. <message> : It describes the messages that the Web service is exchanging.
3. <portType> : It defines a set of related <operation> that a Web service supports.
4. <operation> : It is used to declare the method.
5. <binding> : The purpose of the <binding> element is to describe how to format the message to interact with a particular service.
6. <services> : It indicates where to find a service using its <port> element children.

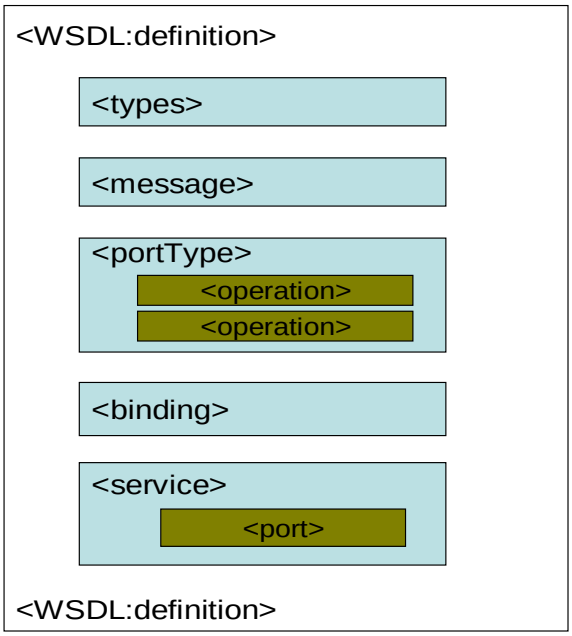

Figure 10 : The structure of a WSDL document

2.5. WS-Security (Web Services Security)

The goal of the WS-Security is to provide the means of the communication protocol for applying security to Web Services. WS-Security is developed by IBM, Microsoft, and VeriSign, and the latest version 1.1 have be released by OASIS on February 17 2006. The mechanism of WS-Security is like XML signature and XML encryption. It extensively adopted the specification of XML Signature and XML Encryption for Web Services message integrity and confidentiality. Nevertheless, their targeted data is

different. WS-Security deals with SOAP security, but XML Signature and XML Encryption secures the content of XML documents. Therefore, in order to be suitable for SOAP security, the special design for WS-Security is necessary.

WS-Security defines a SOAP security header that is a place to put security message (see Figure 11). There are three major elements to make up a <wsse:Security> header : security tokens, XML Encryption, and XML Signatures.

1. Security tokens : It is used for authentication or authorization. For example, the certification which is used to be the signature verification may be added into the <wsse:Security> header.
2. XML Encryption : The <wsse:Security> header may contain a “ReferenceList” element to point the encrypted message.
3. XML Signature : If the <ds:Signature> element is in the <wsse:Security> header, its “Reference” subelement will point the signed message.

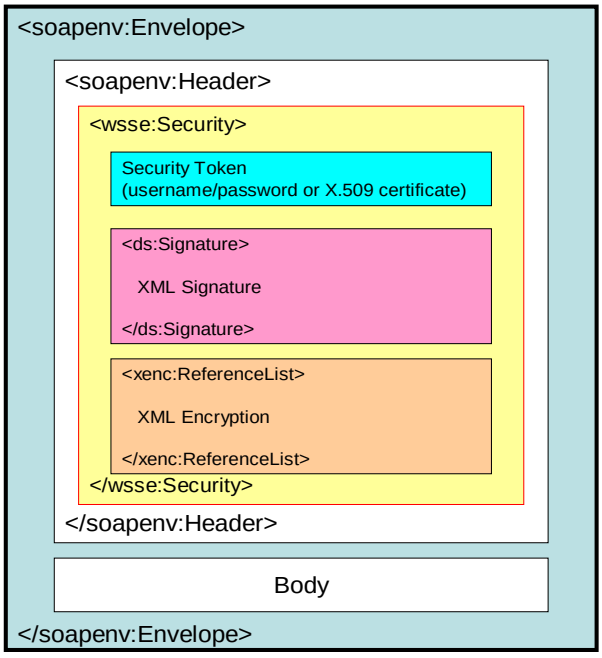

Figure 11 : The structure of a SOAP Security Header

2.6. WS-SecurityPolicy (Web Services Security Policy)

The WS-SecurityPolicy is a specification used to define security configurations for service requester and providers. It explains what needs to be protected, what security tokens to use, and what algorithms to be adopted. Beside, it also constrains content and layout of <wsse:Security> header.

The main content of WS-SecurityPolicy is consisting of four assertion types and one assertion property : protection assertion, token assertion, security binding assertion, supporting token assertion, and security binding property. Figure 12 is an example of the WS-SecurityPolicy.

1. Protection assertion : It specifies what message need to be protected.
2. Token assertion : It specifies the type of token to be used. For example, what type of the certificate will be adopted.
3. Security binding assertion : It indicates that the transport layer is used to satisfy the security requirements.
4. Supporting token assertion : It provides multiple sets of claims to be presented and adds additional tokens in a message.
5. Security binding property : It contains a collection of properties, including tokens algorithms, processing order, and broad types.

```
<wsp:Policy>
  <sp:SymmetricBinding>
    <wsp:Policy>
      <sp:ProtectionToken>
        <wsp:Policy>
          <sp:X509V3Token sp:IncludeToken=".../IncludeToken/Always" />
```

```
            </wsp:Policy>
          </sp:ProtectionToken>
          <sp:SignBeforeEncrypting />
          <sp:EncryptSignature />
        </wsp:Policy>
    </sp:SymmetricBinding>
    <sp:SignedParts>
      <sp:Body/>
      <sp:Header Namespace="http://schemas.xmlsoap.org/ws/addressing"/>
    </sp:SignedParts>
    <sp:EncryptedParts>
      <sp:Body/>
    </sp:EncryptedParts>
</wsp:Policy>
```

Figure 12 : A example of the WS-SecurityPolicy