CHAPTER 4
CRYPTOGRAPHIC TECHNIQUE FOR BUSINESS PROCESSES SECURITY

4.1 INTRODUCTION

The web services, which are composed by composition mechanism, are called composite web services and the process of composition is called service composition [74]. There are different composition languages available for service composition such as, BPEL, OWL, WSCI, BPMN and WS-CDL. There are various tools available for the designing and development of static composition such as, Intalio Business Process Designer, Microsoft Biztalk Server, and Bea WebLogic.

The BPEL language has a set of advantages over different service composition languages such as, good control over workflow design, most usable language in the industry and support for industry best practices. The BPEL also have set of disadvantage such as, it does not have support of data security, encryption and decryption mechanisms for its data, transferred over the network. There are various cryptography techniques available, which need to be incorporated with business processes to provide the data security. These cryptography techniques have the support of cryptographic security algorithms such as, private key and public key cryptography. The private key algorithm involves the use of only one key, which is used for both encryption and decryption, whereas the public key cryptography involves the use of two keys for both encryption and decryption processes.

The required literature survey is explained in Section 4.2 and the proposed framework for securing business processes with the help of cryptographic techniques is discussed in Section 4.3. The working
example of business process is explained in Section 4.4. The summary of the chapter is available in Section 4.5.

4.2 LITERATURE SURVEY

The proxy based approach for web service composition using business processes focuses on non-functional requirements such as, reliability, availability, robustness etc. [76]. A framework has been proposed, which enables us to systematically and automatically instruct the business processes. In a proposed framework, when one or more of their partner services do not provide satisfactory result then the request for service is redirected to a proxy web service. The failed or slow services are substituted by some available and more responsive equivalent services. The main objective of proxy based web services is to discover and bind the web services based on their behavior. The QoS parameters and security issues are concerned in web service composition with non-functional requirements such as, response time, service cost, availability, reliability etc. [77]. The response time is a time period between services invoked and the services get finished. The service cost is considered as a price that a service requesters have to pay for invoking the web services. The availability is defined as the possibility that the service is available at some period of time and the reliability is described as the possibility that a request is correctly responded within the expected time.

The Business Process Modeling Notation (BPMN) is a graphical language that assists business analysts to create notations to define abstract business process in workflow design. The main objective of BPMN is to offer a notation, which can be easily understandable by all business users. A methodology to transform the BPMN model into a business process block diagram is explained, which has the capability to process the business information directly to the BPEL language [78]. The BPMN assists business analysts to create initial drafts for business processes and also, helps the developers for implementing the
technology, so that the execution of those business processes can be flexible in nature. It is also useful to those who will manage and monitor these business processes. The BPMN has the support for basic elements such as, connecting objects and flow objects. The connecting objects are used to inter-connect flow objects through different types of arrows, whereas the flow objects are basic elements such as, events, activities and gateways that are used to create Business Process Diagram (BPD).

The security aspects such as, authentication and authorization in business processes with broker architecture have been explained properly with the help of SAML language [79]. It constructs the secure composite web service broker, which doesn't focus on encryption and decryption mechanisms. In SAML language, the web service provider submits SAML tokens to security servers for making security decisions. The security server has the support for web service security and web service policy mechanisms for providing the security at web service level. Web service security mechanism describes the improvement in SOAP messaging to provide quality of protection through message integrity, message confidentiality and single message authentication. Based on web service security, web service policy provides a grammar for expressing web services security policies.

The web service policy includes a set of security policy assertions to support the web service security specification defined in web service security policy. The secure web service broker is responsible for receiving an input or request from web services, which further implements composition of several web services. The request contains a description of the requested web services. Additionally, the secure web service broker receives a set of general and specific security constraints to be satisfied by the resulting composition. The secure web service broker first performs the creation of an appropriate workflow that models the business process to generate required services.
Web service composition through BPEL process is carried out with the support of security features such as, authorization [80]. The broker based architecture for business processes is built and secured with the help of security constraints. The security constraints and capabilities are described with the help of OWL ontologies and SAML language. OWL ontologies include descriptions of classes, properties, and their instances, as well as formal semantics for deriving logical consequences. The SAML is used to provide the authentication and authorization in web service composition in addition to BPEL language. The BPEL directly doesn’t have the support for encryption and decryption for its data but has support for authentication and authorization. There are two different approaches for securing the web services over the network such as, syntactic based and semantic based [81]. In the syntactic approach, a central coordinator, i.e., the orchestrator is devised to invoke and combine the atomic activities and compose available web services. In the semantic approach, a central coordinator is substituted and assigned the complex task based on the definition of the semantics.

A broker based framework is introduced for integration and adaptation of web services [82]. The broker mainly performs service tracking and selection of appropriate web services. The author has focused on non-functional characteristics such as, response time, service cost, availability and reliability. The automation of web service composition has been explained, in which either the method can generate the process model automatically, or the method can locate the correct web services. The web service security and web service policy mechanisms are used to propose a secure framework for business processes [83]. The XML encryption and XML signature mechanisms are also employed into design to provide confidentiality and integrity, respectively. There is also the support of security token to incorporate the authentication mechanism. The Aspect-Oriented extension for BPEL (AO4BPEL) is implemented as an aspect aware orchestration for BPEL, which supports more adaptable and modular web services.
Above survey describes the limitations of business processes in terms of data security over the network. Therefore, it is required to combine the features of cryptographic techniques with business processes to develop the security interfaces [86, 87]. These cryptographic techniques prevent the unauthorized access to business process.

4.3 SECURE BUSINESS PROCESS FRAMEWORK

The working model of proposed secure framework for business processes is shown in Figure 4.1. The web service repository is available in private and public domain of internet as a freeware or on payment basis. The service provider develops different web services \{WS1, WS2, ..., WSN\} and composes them to develop the business process to fulfill the current business needs. These web services are responsible for accessing their data from corresponding databases \{DB\} as per their requirements. The cryptography technique protects data from theft or alteration and it can also be used for user authentication. The cryptographic technique such as, Caesar Cipher can be used to develop security interfaces surrounding business processes. The security interfaces are an interface, which is designed and developed with the help of cryptography techniques. The security interfaces have support of encryption and decryption to secure the business processes and its data over the network.

4.3.1 Encryption and Decryption Modules

The security interfaces have two major sub modules, i.e., ENC module (encryption module) and DEC module (decryption module), which are responsible for securing business processes. ENC module is responsible for encryption process of plaintext whereas DEC module is responsible for decryption process of ciphertext. Once the security interfaces is developed at both the ends of business processes, the business processes can be registered into central repository by service provider. Once business processes are registered, the service consumer
can search processes into repository as per their need as an application client. Once the business processes are searched, the service consumer starts an interaction with the business processes directly and uses their functionalities.

Figure 4.1: Secure Business Process Framework

The business processes interact with different web services for consumer needs. The web services can interact with different databases for accessing data on the same time. The ENC module is used to encrypt the incoming plaintext request into ciphertext with the help of key K1 at sender end. The encrypted ciphertext request is sent to
business processes for processing. The business processes forward the encrypted ciphertext to the corresponding web services. The same key K1 is used at the receiver end to decrypt the ciphertext back into plaintext request by DEC module. At receiver end, once the plaintext is generated then web services can fulfill all the incoming requests in a plaintext format. The response of sender’s request is also encrypted at receiver end with the help of ENC module and key K2 before sending it to business processes. The encrypted ciphertext response is sent to business processes. The business processes route response to the corresponding service consumer. At sender end, once the encrypted response is collected, the DEC module decrypts the ciphertext response back into a plaintext with the help of same key K2. The management of business processes is carried out by BPEL process manager and the execution is carried out with the help of BPEL engine resided in BPEL process manager.

4.3.2 Process Description

The procedure for the securing business processes is described below. The complete procedure has three different stages namely service consumer, service provider and service consumer again. The service consumer activity focuses on encryption of plaintext into ciphertext, forwards the ciphertext to business processes etc. The service provider activity focuses on interaction of business process with the web services, and interaction of web services to the databases etc.

4.3.2.1 Service Consumer

The service consumer is responsible for sending requests to service provider through business processes over the network in secure manner. The security can be developed with the help of security interfaces at both the ends of communication around the business processes. The service consumer needs to perform encryption on plaintext request before sending it on the network.
Begin

Send plaintext to security interface.

Read plaintext by interface.

Perform encryption of plaintext.

Generate ciphertext from plaintext request.

Forward ciphertext to business process.

Processes send the ciphertext to security interface (receiver end).

End.

The application client sends plaintext data to the security interface for generating the ciphertext. The encryption operation based on Caesar Cipher mechanism is used to generate the ciphertext from plaintext. The generated ciphertext is send to the business processes in encrypted format over the network. The business process sends the encrypted request to the receiver side security interface. The service provider has following algorithmic steps to perform

4.3.2.2 Service Provider

The service provider is responsible for reading the incoming requests from service consumer and forwards the request to different web services connected. It also responsible for collecting the cumulative response from different web services in plaintext format and perform the encryption on it. The encrypted response is send back to business processes for further redirecting to service consumer.

Begin

Read ciphertext at security interface (receiver end).

Perform decryption of incoming ciphertext.
Generate plaintext from ciphertext.

Forward plaintext to web services.

Search particular web service(s) for fulfilling the incoming request.

Match the requirements of the incoming request with web service parameters.

Collect and compose all the responses together from all the web services.

Send back plaintext responses to security interface (receiver end).

Read plaintext from web services at security interface (receiver end).

Perform encryption of all responses and generate single ciphertext response from plaintext responses.

Send back ciphertext response to business processes.

Read ciphertext response from security interface (receiver end).

Business Process sends ciphertext response to security interface (sender end).

End.

The encrypted request is read by receiver side security interface. The security interface performs decryption operation based on Caesar Cipher mechanism and generates the original plaintext request. The plaintext request is handled by available web services. These web services collect all the responses in order to fulfill the incoming requests. The responses are further encrypted before sending back to the business processes over the network. The business process collects the encrypted responses and sends them to the sender side security
interface for the application client or service consumer. The service consumer has following algorithmic steps to perform.

Begin

Read incoming ciphertext response at security interface (sender end).

Perform decryption and generate plaintext from ciphertext.

Send the plaintext response back to service consumer.

Service consumer receives plaintext and utilizes the outcome/result.

End.

The sender side security interface reads the incoming encrypted responses and decrypts them to generate the original plaintext responses received by the web services through business processes over the network.

4.4 AN ILLUSTRATION

The implementation phase describes how web services, business processes and security interfaces can be developed together to build the secure framework for business processes. The business process requires XML description files and Business Process Diagram (BPD) for interaction between the web services and business processes. For its implementation, we have considered the scenario of movies running in multiplexes as discussed in Chapter 3.

The implementation is illustrated as follows. Firstly, the consumer will request for movie time for particular movie as plaintext. The plaintext will be encrypted into ciphertext at sender side security interface as shown in Figure 4.2.
The ciphertext is ready to be passed to business process. The working model of secure business process is shown in Figure 4.3. Once the business processes is deployed to BPEL engine, the ciphertext is passed to business process and computed at sender side security interface as shown in Figure 4.4.
The business process forwards this encrypted movie name to associated web services through the receiver side security interface. The receiver side security interface decrypts the ciphertext into plaintext and gets the movie name as original message. The plaintext at receiver side is further send to the different web services for computing the movie time in theaters as shown in Figure 4.5 and Figure 4.6, respectively. In Figure 4.5 and Figure 4.6, the PVR and INOX web services are taking READY as input, respectively.

**S_PVR_WS endpoint**

<table>
<thead>
<tr>
<th>Operation:</th>
<th>Movie_Time</th>
<th>HTML Form</th>
<th>XML Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reliable Messaging</td>
<td>Include In Header</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WS-Security</td>
<td>Include In Header</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 4.5: Movie Name as Input to PVR Web Service**
Both the web services, PVR and INOX, will compute the movie time for the incoming request of movie name in parallel. The web service responses will be sent back to receiver side security interface for encryption process such that it composes the data secure over the network at business process level. The encrypted movie time will further move to business process, such that business process will send them back to consumer. The encrypted movie time at business process level received from receiver side security interface is shown in Figure 4.7 and Figure 4.8.

The encrypted movie times from different web services at business process level over the network represents our efforts to build the business process secure. At business process level, the data is secure from any unauthorized access and malicious attacks. The encrypted movie time will be sent back to sender side security interface, where the security interface will compute the movie time as plaintext from the incoming ciphertext.

The process of decryption at sender side security interface is shown in Figure 4.9 and Figure 4.10, such that consumer will receive
the movie time in plaintext for PVR and INOX web services and book the show.

AxResponse:
output2: WHQ_R_FORFNQLJKW

**Figure 4.7: Encrypted Movie Time from PVR Web Service at Business Process Level**

AxResponse:
output2: HOHYHQ_R_FORFNQLJKW

**Figure 4.8: Encrypted Movie Time from INOX Web Service at Business Process Level**

**Figure 4.9: Decryption Process for the Computation of Movie Time in Plaintext for PVR Web Service**

**Figure 4.10: Decryption Process for the Computation of Movie Time in Plaintext for INOX Web Service**
4.5 SUMMARY

The secure framework for business processes over the network has been evaluated and tested in this chapter. This chapter focuses on the expansion of security interfaces as ENC module (encryption module) and DEC module (decryption module). The ENC and DEC modules are developed with the help of Caesar Cipher technique. The algorithmic description of business processes has been explained in subsequent sections. The algorithmic representation at service consumer and service providers ends have been tested and discussed in this chapter. The chapter also illustrated the working example including the complete process of building the business processes secured from unauthorized users.