CHAPTER 1

INTRODUCTION

Web services have generated great interest in researchers recently due to their diverse nature. These services are based on existing Internet protocols and open standards. They provide a flexible solution to the open problems of application integration. In current scenario, web services are platform independent software components, based on some principles of software architecture, which are available in the distributed environment of the Internet [1]. Applications are created from a set of suitable web services and these are no longer being written manually. Seamlessly web services have the potential of developing business to business interaction for enterprise application integration.

Web services are used to construct software systems from distributed components. It is difficult to build robust and secure web services as many of the technologies are immature or address only some parts of the problem. These web services deal with Service Oriented Architecture (SOA) for its designing purpose. SOA is an architecture, which follows some guidelines proposed by World Wide Web Consortium (W3C) community. SOA has focused on the collection of web services, where web services have the ability to communicate with each other at compile time/ runtime. SOA architecture provides interaction between service provider, service repository and service requestor as shown in Figure 1.1.

The fundamental SOA architecture is required to ease the interactions between a client and a web service. If the implementation of web service involves the invocation of other web services then it is necessary to combine the desired functionality with other web services. These web services can always be combined together by means of service composition to resolve critical problems, which usually exist in
current business process scenario [2]. Web services are combined with the help of composition operation to fulfill the business requirements. Those web services, which are composed by composition operation, are called composite web services and process of composition is called service composition [3]. In composite web services, business logic of the client is implemented by several services. A client invoking a composite service can itself be exposed as a web service. However, service composition suffers from several limitations of security issues such as, lack of native support for encryption and decryption, authentication etc. [4].

The service composition can be achieved with the help of different approaches such as, Business Process Execution Language (BPEL), Ontology Web Language (OWL) etc. BPEL is a language to specify business process interaction and its protocols. The interaction between business process and its partners can be achieved with the help of XML based grammar arises from web services provided by BPEL. Additionally, BPEL also identifies the states of coordination between all kinds of interactions among web services. The business interaction protocols used by business processes are called abstract processes [5]. They are used to specify message exchange among different parties involved in a business protocol. The BPEL language does not expose the internal behavior and the implementation part of the concerned parties.

The service oriented architecture with its features and benefits are explained in Section 1.1. Section 1.2 illustrates the fundamentals of web services and its flow model. The service oriented computing is demonstrated in Section 1.3. There are different web services standards available in the business world, which are explained in Section 1.4. The support of security in web services is explained in Section 1.5. The state of the art literature survey is illustrated in Section 1.6. Section 1.7 describes objectives of the thesis and outline of the thesis is illustrated in Section 1.8.
1.1 SERVICE ORIENTED ARCHITECTURE

In traditional client server approach, we had the server offering some functionality that could be used or called by the client. Web services are considered as a paradigm for distributed applications, which consist of different components such as, service broker/ service repository, service provider and service requester [5]. A service broker operates between a service provider and a service requester. A service provider circulates its services to the service broker. A service requester requests the service broker to find a suitable service as shown in Figure 1.1. The web service architecture defines an interaction between machines and exchange messages between service requesters and service providers. Service requester requests the web services and service provider provides web services. A service provider publishes a description of the web services into service broker/ service repository. The requester must be able to find the description of the web services from service broker.

Figure 1.1: Service Oriented Architecture
Service oriented architecture is capable of exchanging the messages, describing web services and its functionalities; and publishing and discovery of web services. The web service architecture provides the communications between service requester, service broker/ repository, and service provider. The services involve three fundamental operations, i.e., publish, find, and bind. These roles and operations are operated on web service artifacts. In an architectural scenario, a service provider hosts a web service, defines and publishes a service description into service repository/ service broker. The service requester uses a find operation to retrieve the service description locally or from the service repository. The service requester uses the service description to perform bind operation with the service provider and invokes the web services. Requesters and providers interact using message exchange mechanisms, which define the sequence of messages exchanged between them. The description of web services is published into service broker or discovery agencies, where provider publishes service description and the requester discovers the description. The description includes data type, operations supported by web services and web address of the web services [5].

Current software companies are accepting SOA governance frameworks to acquire the benefits of SOA architecture [6]. The governance framework has several best practices such as, impose architecture governance, set the design time governance, organize runtime governance and regiment change time governance. An impose architecture governance defines SOA architecture, classifies the infrastructure potentials, estimates and categorizes seller technologies. SOA architecture stipulates security, availability and reliability characteristics for SOA infrastructure. The design time governance set up the policies, which governs the creation and definition of web services, and builds a team, which is responsible for identifying business areas benefited by web services. Also it builds an architecture team to validate the web services and imposes its operations. The
runtime governance enhances all the IT operating functions to organize the definition and enforcement of strategy around security, availability and monitoring of deployed web services. The regiment change time governance builds and imposes a strong management process to manage and balance the web service functionalities in response to business needs. The main benefits of governance framework is to improve enterprise confidence, which are proper utilization of web services, measurable performance of web service operations and prevention of security breaches.

SOA architecture deals with web service, which has the capability to introduce service composition, service virtualization and service bus [7]. These mechanisms are applied to the SOA architecture and have the following benefits. SOA features and benefits are shown in Table 1.1.

Table 1.1: SOA Features and Benefits

<table>
<thead>
<tr>
<th>Feature(s)</th>
<th>Benefits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Services</td>
<td>It provides flexibility and exposes functionalities to improve flow of information.</td>
</tr>
<tr>
<td>Messaging</td>
<td>It provides configuration flexibility and data transfer facilities.</td>
</tr>
<tr>
<td>Message Monitoring</td>
<td>It deals with attack detection and provides security to services. It also measures the performance intelligently.</td>
</tr>
<tr>
<td>Message Control</td>
<td>Apply security and management policies.</td>
</tr>
<tr>
<td>Message Security</td>
<td>Maintain integrity and confidentiality.</td>
</tr>
<tr>
<td>Service Composition</td>
<td>Capability to merge new functionalities to</td>
</tr>
</tbody>
</table>
impose rapid changes.

<table>
<thead>
<tr>
<th>Service Discovery</th>
<th>Ability to minimize the cost and performance.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtualization</td>
<td>It enhances reliability and has the ability to extent its operations to meet business requirements.</td>
</tr>
</tbody>
</table>

### 1.2 WEB SERVICES

A web service is defined as a set of standards and protocols used for exchanging information between systems and applications with the help of existing web technologies such as, Extensible Markup Language (XML) and Hypertext Transfer Protocol (HTTP) to construct the distributed environment [8, 9]. Web service is an emerging technology in recent years due to its open standard specification. The W3C defines web service as a software system designed to support interoperable machine-to-machine interaction over a network [10]. A web service is a network accessible interface to application functionality, built using standard Internet technologies as shown in Figure 1.2 [11].

In web service flow model, web application uses web services as interface and has support of graphical user interface. The service requester interacts with web application and web application further sends incoming request to the web services for the fulfillment. A web service is an interface positioned between application code and user of that code. It acts as an abstraction layer, separating the platform and programming language specific details of how the application code is actually invoked. This standardized layer means that any language that supports the web service can access the applications’ functionality [11]. If an application can be accessed using a set of protocols over a network such as, SMTP, XML or HTTP then it is a web service in other
words. The web service flow model is 3-tier model used in different application domains such as, e-commerce, hospitality, shopping etc.

![Diagram of Web Service Flow Model](image)

**Figure 1.2: Web Service Flow Model**

The web browsers or client applications that recognize these standards can work together with the web services to perform several tasks associated with different application domains such as, weather broadcasting, search engine optimization, tour and travels activities, educational and entertainment. Web services are the new variety of web applications. They are self-contained, self-describing, modular applications that can be published, located, and invoked across the web [12]. Web services perform different tasks that can be simple requests for accessing information or complicated executing business processes to fulfill the business requirements. Once the deployment of a web service is completed; it can be invoked, merged and discovered by another web services.

The key advantage of web services is the capability to create applications with the help of reusable and loosely coupled software components, which are implicated into both technologies and business
applications. It will be possible for web services to obtain automatic and vibrant interoperability between systems to perform business tasks. Web services are usually completely distributed on the internet and utilized by the variety of communication devices. Web services offer many other benefits over distributed computing architecture such as, web services typically work beyond private networks and offer the developers a non proprietary approach to their solutions [13, 14]. Services developed will certainly have a longer life span and offer better revenue on developed service. Web solutions also allow coders to use their recommended programming languages. Furthermore, web services usually are virtually platform independent. Web service permits business logic to be exposed over the web. It provides freedom to choose the appropriate web services as per users needs. Additional application specific business logic can be included on the client side. It allows developing web services or client side applications using the languages and tools that are desired by the developers.

Web service provides not only a component based model associated with application development, it also allows reusability of web service components appropriate for different service calls. It is also possible to deploy web service as a legacy code, which are tends to be deployed over internet technologies. Web service produces new IT solutions more rapidly and at cheaper cost by focusing their code development using web service applications for core and non-core business requirements. It combines the business processes with customer requirements at low cost. With lower costs, even small business can participate in business-to-business integration. They can also enter the new market as well as widen their consumer base.

Although the simplicity of web services is an advantage in some respects, it can also be a hindrance. Web services use some methods to identify appropriate information. It ensures that web service requests are larger than requests encoded with a binary protocol. Web service binding mechanisms dynamically requires the trusted contents of the
service repository/service registry. Web services can be exposed in different areas of business world such as, e-commerce, banking system, tour and travelling system, future forecasting, weather forecasting, engineering and science education system, entertainment and movies etc. Web services are combined together to solve several complex problems existing in business scenario.

Web services have the capability and functionality to send the requests and receive the response from different web services, unknown to the consumers. For example, when a student searches jobs in Naukri.com, he has to enter his basic details such as, skills, educational qualifications and can search for appropriate job as per the profile. If the website doesn’t have suitable jobs for the profile then the website will respond with “No Match Found”. This is because the website is not a web services; it is just the website. At the same time, when person will search into google.co.in then Google has capabilities of retrieving the required result from any part of the world because of web service support. The google.co.in has millions of web services to fulfill the customer requirements. Thus the entire web applications in the world related to different business can be re-developed with the web services. Web service provides capability of fast searching and fulfillment of customer’s incoming requests. It also hides the complexity of request processing at service end. It is easy to use, deploy and manage.

Web services can be composed together to solve critical problems. There are several different ways to compose the web services such as, business process execution language, ontology web language, web service choreography interface, business process management language and web service choreography description language. Business processes execution language is an industry oriented approach to compose the web services as business processes. The composition of web services is described in detail in Chapter 2.
1.3 SERVICE ORIENTED COMPUTING

Service Oriented Computing (SOC) depending on web services is currently one of the primary drivers for IT industry. A primary goal of SOC is to manufacture a collection of software services accessible through standardized protocols, whose functionality is usually automatically discovered and built into applications or composed to build more complex service solutions [15]. The SOC paradigm uses web services to support rapid development at low cost with interoperable behavior. These web services are platform independent, autonomous entities that can be described, published, discovered, and loosely packaged in novel means. SOC paradigm is referred as a set of concepts, methods and a principle that characterizes computing in SOA architecture, therefore, web service applications can be constructed based on independent components with standard interfaces [17]. It is a new computing standard that develops different kind of services to maintain fast and easy composition of distributed services in heterogeneous environment [18].

The service oriented computing performs different operations that vary from responding simple business processes to executing complex businesses [16]. The service oriented computing assures that web services can be composed to create flexible business processes. It has been created in the different area of research such as, health services, government services and dynamic and on-demand business services. It is huge and complex concept, which incorporates different kind of technologies and sub-concepts into it.

1.4 WEB SERVICE STANDARDS

Internet is composed of variant technologies that successfully communicate through shared protocols. There are several open source software developed on the basis of those protocols, which play very crucial role in preserving the vendor interoperability in implementation of standards. There are several standards contributing important roles
in web services such as, XML, Simple Object Access Protocol (SOAP), Web Services Description Language (WSDL) and Universal Description, Discovery and Integration (UDDI) [19]. The relationship between these standards is described in Figure 1.3. The dashed line between the web service and WSDL document represents the generation of WSDL document from web service. The generated WSDL is XML based document, which further get registered into central repository/service repository/service broker called UDDI for public usage.

**Figure 1.3: Relationship between SOAP, WSDL and UDDI**

**Extensible Markup Language**

XML is a simple, very flexible text format derived from Standard Generalized Markup Language (SGML) [20]. XML also plays an important role in the exchange of variety of data on the web and elsewhere. XML describes XML documents as a class of data objects, which partially describes the behavior of computer programs. XML documents are prepared with storage units called entities, which contain either parsed or unparsed data. The parsed data is prepared
with set of characters, where some of characters form dataset, and some of them will form markup. Markups prepare a description of the document's storage layout and logical structure. The XML processor is required to process XML document, which is used to provide access to XML content and structure. XML language is a base for web services or service composition mechanisms as it is used in different protocols, description languages, central repository etc., as part of service oriented architecture.

**Simple Object Access Protocol**

SOAP is the communication protocol for a web service. It is a lightweight protocol to exchange information in a distributed environment [20]. It is an XML based protocol that consists of three parts, i.e., an envelope, set of encoding rules, and few conventions. The envelope defines a framework for describing the part of a message and method of processing messages. The encoding rules are expressing instances of applications based on defined data types. The conventions assist into representing Remote Procedure Calls (RPC) and their responses. SOAP protocol can preferably be used in combination with a variety of other protocols such as, HTTP and HTTP extension framework.

SOAP protocol has a message section, which is used for sending the requests and receiving responses between the web services. The SOAP protocol contains the tag ENVELOPE as the root element. The envelope contains two sub-elements, i.e., header and body. A header is optional element, which contains header entities. These header entities are used to provide authentication information or encoding mechanism for data security over the network [5]. The body element contains information for the last recipient in order to all recipients with their own body entities. If the ENVELOPE contains a header, the body is usually not the first element within the ENVELOPE. SOAP protocol uses the SOAP specification, which describes the behavior of XML data and also describes the way how SOAP will perform RPC
operations. SOAP also supports document style applications, where SOAP message is wrapped around an XML document. The document style applications are very flexible in nature and many new XML web services take advantage of this flexibility to build complex services that are difficult to implement using RPC only.

**Web Service Description Language**

WSDL is a language, which describes network services as a set of endpoints operating on messages with XML format specifications [20]. The operations and messages supported by WSDL are bound to a network protocol to define endpoints. WSDL provides description of endpoints and their messages. The WSDL language has its own WSDL document to describe web services to central repository. These WSDL documents have the root element called <definitions> [5]. The <definitions> contains five elements that are divided into two groups, i.e., abstract definitions and concrete definitions. The abstract definitions have three elements, i.e., <type>, <message> and <portType>, whereas concrete definitions have two elements, i.e., <binding> and <services>, to define the behavior of web services as shown in Table 1.2.

**Table 1.2: WSDL Elements and Their Roles**

<table>
<thead>
<tr>
<th>Definition(s)</th>
<th>Element(s)</th>
<th>Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Abstract Definitions</td>
<td>&lt;types&gt;</td>
<td>It is used to define the data types used by the web services.</td>
</tr>
<tr>
<td></td>
<td>&lt;message&gt;</td>
<td>It is used to define the messages used by the web services.</td>
</tr>
<tr>
<td></td>
<td>&lt;portType&gt;</td>
<td>The operations performed by the web services are defined with portType.</td>
</tr>
</tbody>
</table>
Concrete Definitions

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;binding&gt;</td>
<td>The communication protocols used by the web services are handled with binding.</td>
</tr>
<tr>
<td>&lt;service&gt;</td>
<td>It defines the location of the web services.</td>
</tr>
</tbody>
</table>

The type element contains all data type definitions relevant for sending and receiving the messages. The message element specifies data that are communicated between a service requester and a service provider. A message element may emerge multiple times and consists of a name and one or more <part> elements. The part elements refer to the defined types using a type attribute. A port type focuses on set of operations supported by WSDL language. The port type element has another sub element as a <operation>with name attribute, which specifies the order of parameters used in the operation. Additionally, four different primitives can be found in the specification such as, one way, request & response, solicit response and notification to identify the kind of operation, which has to be performed.

The one way primitive is considered as the endpoint, which receives a message. The request/ response is considered as the endpoint, which receives a message and sends a reply. The solicit response is considered as the endpoint that sends a message and receives a reply. The notification is considered as the endpoint that sends a message only. The <binding> element is introduced by WSDL specification to bind WSDL operations with SOAP protocol. The <binding> element has SOAP extension element called <soap:binding> to bind the portType to SOAP specification. A service element is responsible for set of related ports used for communication. These set of ports are responsible for locating the web services over the network. The service element is also have the URL of the web service. The URL contains the hostname, application port, and name of the web service invoked.
Universal Discovery Description and Integration

Universal Description Discovery and Integration (UDDI) is an XML based central registry for web services or business processes worldwide to list themselves on the Internet. Its main goal is to simplify all kind of online transactions by permitting companies to find each other on the web and construct their systems interoperable for e-commerce domain [21]. There are three different kinds of pages supported by UDDI, i.e., yellow pages, white pages and green pages [22]. The yellow pages help in searching as well as locating appropriate businesses, which serves a specific industry or product category or which might be located within a selected geographic region. The white pages provide details about a service provider for web services or business processes including address location and contact details etc. The green pages provide the technical information about web services or business processes that are exposed by the business requirements.

The UDDI has its own UDDI directory, which exposes set of Application Programming Interfaces (APIs) to publish web service descriptions in centrally located repository. These APIs are divided into two logical parts, i.e., Inquiry API and Publisher’s API. The Inquiry API constructs set of programs that allows searching and browsing the information in an UDDI registry for available services, whereas Publisher’s API is used in case of invocation failures. The UDDI directory also includes several methods to search appropriate web services in central repository required to build an application. For example, you can search for providers of the service in a specified geographic position or for business of the specified type [22].

The UDDI have four types of data structures, which are used to bind web services or business processes to central repository. These data structures are business entities, business services, binding templates and tModels. These data structures collectively perform binding operation between the service elements.
The business entities are considered as data structures that contain information about the company or service provider that publish the web services or business processes in central repository. It mainly contains the name of the company/service provider, discovery URLs, functionalities offered and contact details such as, address, phone number, email etc. Each business entity may contain multiple business services. The business services are used to express each offered service in business terms. The business services require at least one binding template. The binding template serves as a container for one or more binding template structures. A binding template structure describes how to access a web service. The binding template may contain one or more tModels keys. The tModels are used to describe technical specification of the web services in UDDI. These web service standards such as, SOAP, WSDL and UDDI work together to achieve and fulfill business requirements over the internet with the help of HTTP protocol. The web service standards and their key roles are shown in Figure 1.4.

<table>
<thead>
<tr>
<th>Web Service Standards</th>
<th>Key Roles</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSDL</td>
<td>Service Descriptions</td>
</tr>
<tr>
<td>UDDI</td>
<td>Publishing and Discovery</td>
</tr>
<tr>
<td>SOAP</td>
<td>Messages / Protocol</td>
</tr>
<tr>
<td>HTTP, SMTP, MQ</td>
<td>Transport Layer</td>
</tr>
<tr>
<td>Internet/ Intranet</td>
<td>Network Layer</td>
</tr>
</tbody>
</table>

Figure 1.4: Web Service Standards and Key Roles
1.5 WEB SERVICE SECURITY

Web Service is relatively new as well as an emerging area. The security issues of web services in a distributed environment certainly are the major concern. Web service security is amongst the thrust areas involving research both in industry and in academia. Security in web services can be illustrated as general security framework consisting of the following issues such as, authentication, authorization, confidentiality, non-repudiation, availability and integrity [37]. Authentication is concerned about the establishment of data security, proof of identities of particular entities involved in communications. The entity may be a user, a process or perhaps a service. Masquerading is usually a standard attack inside authentication mechanism. Authorization is a concept where, any entity is considered as an authenticated and the next issue is to ascertain, which operations the entity is allowed to perform. The authorization mechanism deals with granting and revoking protection under the law of authenticated entities. The authorization is pertaining to giving permission to end user for accessing their credentials.

The confidentiality is defined as of only senders and intended receivers are able to access the content of the message. An unauthorized person should not be able to access any message over the network. Eavesdropping is an attack in confidentiality. There are several situations when end users send a message and later repudiate it. Repudiation can also be done at the time of sending/receiving the messages. The availability states that resources and services should be accessible to authorized users only. Denial of Service (DoS) is a standard attack with availability. Data integrity can be checked using mathematical algorithms known as hashing algorithms. A hashing algorithm takes a block of files as an input and produces a much smaller part of data as an output. This output is sometimes called a digest with the data. If data is information, it is called a note digest. MD5 and SHA are the standard hashing algorithms.
1.6 STATE OF THE ART

Online web based applications can be built up with client server based architecture. This client server architecture have several drawbacks such as, less flexibility, static in nature and limited support for exposing the web application and its functionalities. These issues can be overcomed with the help of web service based architecture. The web service based architecture is an architecture, which works around self contained application logic called web services as compared to client server architecture.

Web service based architecture has the capabilities of requesting other web services, which is not possible with the client server architecture. These web services can be developed for different application domains to manage requests in a better manner. Each application domain such as, entertainment, education can have single or multiple web services or grid of web services in implementation to fulfill the service consumers’ requests, where each web services can interact with other web services.

The web services can be composed with the help of different approaches such as, business processes and ontologies. The business processes are responsible for fulfilling different kinds of business requirements from different intended users of different application domains online. Business processes play important role to fulfill the business requirements of different application domains such as, banking systems, e-commerce systems, online exam system, criminal records etc. These business processes require high security over the network, so that intended users can communicate in a secure manner and stay protected from unauthorized users.

These business processes lack into security aspects as there is limited native support for encryption and decryption mechanisms in business processes. Therefore, cryptographic mechanisms are
incorporated into business processes to secure business processes over the network [8, 9, 11, 12]. Additionally, there is a need to develop an effective security algorithm, which is fast in execution and it builds the business process highly secured as compared to the existing security algorithms [22, 23, 24]. The business processes also don’t have any capabilities of load balancing activity to reduce the load of central repository in service oriented architecture [17, 18, 19, 20, 21]. The business processes even don’t have support of two phase authentication mechanism, which can provide secure environment [13, 14, 15, 16]. In current scenario, the business processes, web service based applications/web applications have support of single phase authentication mechanism for authentication.

Our main research objective is to provide a secure framework for business processes. The secure framework can be constructed with the help of security interfaces developed around the business processes. These security interfaces provide security to business processes. An encryption and decryption modules need to be developed to provide security interfaces and assist intended users for obtaining the security over the network. In addition to secure framework for business processes, the load balancing activity is to be performed.

The load of central repository can be reduced by equipping the business processes with the broker and broker’s layer [29]. The broker is capable of managing the business processes by handling all incoming requests and forwards them to appropriate broker on broker’s layer. The broker is used to compose the set of web services for particular application domain. The secure algorithm has to be developed to provide high security to business processes. The business processes have the support of single phase authentication and authorization yet over the network [13, 14, 15, 16]. There is limited support of two phase authentication mechanism in business processes, which can provide higher security as compared to single phase mechanism.
1.7 OBJECTIVES

The research objectives for providing security to the business processes with the help of secure framework are as follows:

Integration of service oriented architecture and its guidelines with the development of web services and their building blocks in our research work.

This objective integrates service oriented architecture and its guidelines for the development of web services. The service oriented architecture illustrates the working model for the development of web services, publishes them into central repository, performs find and bind operations for each published web service. The service oriented architecture also has its own building blocks such as, SOAP, WSDL and UDDI etc. These building blocks help in the development of web services, where the service oriented architecture performs the communication between these building blocks with different operations such as, find, bind and publish.

Development of business process from web services.

The business processes are well known industry oriented approaches for service composition as compared to different service composition mechanisms. It is capable of solving many complex issues in grid computing, cloud computing, heterogeneous networks etc. It can also hide the complexity of business requirements from service consumers or application clients. The development of business processes with the help of web services is generally a complex activity. The business process can develop and deploy the workflow design with the help of BPEL language. The generalized framework is developed for business processes to simplify the complex task of business process development.
Design a framework for secure web service composition.

The developments of encryption and decryption modules help to design a secure framework over the network. The encryption module is responsible for converting plaintext into ciphertext with the available security mechanisms. The substitution method is implemented to design the encryption module. The decryption module converts the ciphertext back into plaintext with the same security mechanisms. Both encryption and decryption modules together provide better security to business processes. The cryptographic security techniques can be integrated into business processes by creating the security interfaces around the business processes. These security interfaces build secured environment over the network and protect intended users from unauthorized users.

Integration of cryptographic security techniques in business process for providing security and create a secure interface at both side of communication.

The cryptographic security techniques can be integrated into business processes by creating the security interfaces around them. This integration of security techniques secures the business processes from unauthorized users over the network. Additionally, broker’s layer is introduced to reduce the load of central repository. The broker as business processes will select appropriate broker from broker’s layer to fulfill the incoming requests from service consumer. The selected broker handles the single or grid web services.

Performance evaluation of cryptographically secure framework for business process.

The performance evaluation of cryptographically secure framework for business processes is illustrated in the thesis. The proposed multithreading based secure algorithm increases security in
business processes. The algorithm has the support of key matrices. The key matrices are the matrices of keys, where the keys will be generated randomly. The performance of multithreading based secure algorithm is assessed against the existing security algorithms. The graphical representation of encryption time analysis is also carried out on the basis of different key parameters such as, file size, file type, encryption time etc.

Comparative study of existing frameworks with proposed framework for possible inclusion in product feature of modern web services.

The comparative study of existing frameworks with proposed framework is analyzed and discussed in the thesis. Different authentication based security mechanisms such as, securepass and staganography is compared on different key parameters such as, authentication time, authentication type, confidentiality, reliability, accuracy, robustness, hardware accessibility, dictionary attack, brute force attack, spoofing attacks etc. [33, 34]. The application supporting all the aspects of proposed secure framework is also developed to access business processes in a secure manner over the network.

Development of application client or service requestor.

The existing security issues in business processes discussed above must require some kind of security framework. The security framework for business processes with the support of brokers, encryption and decryption modules, multithreading based algorithm, and two phase authentication mechanisms provide high security to business requirements. The broker provides load balancing activity, whereas cryptographic technique is used to develop the encryption and decryption modules. A file uploader and downloader application is developed and experimented that integrates all the features discussed
in aforesaid objectives. Such applications can be deployed in various business domains to secure the business communications.

1.8 OUTLINE OF THE THESIS

This thesis is organized in nine chapters to cover the web services, service composition, business processes and its security. The aspects of brokers in business processes and security framework for web service composition are also covered in the thesis. The security framework involves multithreading security and authentication mechanisms. The description and working model of integrated software developed for service security is discussed below with the concluding remarks.

Chapter 2 discusses the concept of web service, web service composition and type of service composition mechanisms such as, BPEL, OWL, Web service choreography interface, Business process management language and Web service choreography description language. The chapter provides different design goals and comparative study for different service composition languages. The description of different software tools for performing web service composition and development of business processes from web services are explained in the chapter.

Chapter 3 describes the generalized framework for secure web service composition with its different components such as, web services, business processes and BPEL engine, central repository, user interface and security interfaces. The working example along with implementation, deployment and testing activity for the implementation of secure framework are also discussed in this Chapter.

The cryptographic secure framework for business processes is proposed in Chapter 4. The components of proposed secure framework such as, encryption and decryption modules; and process description
activity are also discussed in this Chapter. An illustration is discussed about the working model for business processes as a secure framework.

The Chapter 5 discusses broker based secure web service composition. The brokers and broker’s layer are illustrated with their components. The implementation of brokers is also illustrated in this chapter. Also, the comparative analysis of broker-based and broker-less mechanisms is presented.

Multithreading based secure algorithm for business process security is introduced and discussed in Chapter 6. The algorithm is based on the concept of key matrices, which is described in this Chapter. The key matrices have the random functions to generate the random keys for encryption and decryption activities. The algorithmic representation and experimental evaluation with the support of key matrices are also explained in the Chapter.

The pattern based two phase authentication mechanism for securing the business processes along with its components is illustrated in Chapter 7. The end user has to pass first phase of authentication before performing second phase of authentication process. The comparative study of available and proposed authentication mechanisms is illustrated in this Chapter. The proposed mechanism is implemented and tested together with its working model.

Chapter 8 provides the integrated framework for secure business processes. The framework is developed, implemented and tested for an application of file uploader and downloader. The model and description of each component of integrated framework is explained and discussed in this Chapter. The framework is illustrated with an example.

Finally, Chapter 9 provides conclusion of the thesis that incorporates whole research work, outcomes and applications. It assists end users to solve critical problems, applying corresponding solution strategies,
implementation, deployment and testing mechanisms for individual objective.