CHAPTER 6
MULTITHREADING BASED SECURE ALGORITHM

6.1 INTRODUCTION

The business process requires SAML language to provide data security over the network because the business processes themselves don’t have the support of security mechanisms such as, encryption and decryption. These security mechanisms can be integrated into the business processes with the help of cryptography. Cryptography has a support of different private key based block cipher algorithms such as, Data Encryption Standard (DES), triple DES and Advance Encryption Standard (AES) etc. to provide security to end users. DES is an algorithm for encrypting the data with private key mechanism. DES is a block cipher mechanism, which encrypts and decrypts one block at a time instead of bit level encryption or decryption used in stream cipher mechanism. DES algorithm has some limitations such as, low security and usage of single key mechanism, which can be overcome with triple DES algorithm [114]. Triple DES algorithm processes a data block thrice a time with different keys before converting plaintext into the ciphertext. With the help of multiple key support, triple DES is more secure than DES algorithm. The sequential thrice data block processing capabilities of triple DES algorithm builds the algorithm slow in execution. Therefore, it is required to design an algorithm, which is as secure as triple DES and as faster as DES in execution.

The multithreading based secure algorithm is proposed to provide more security as compared to DES and triple DES. The related literature survey is described in Section 6.2. The multithreading based secure algorithm for securing data processing in business processes is illustrated in Section 6.3. The experimental evaluation is presented in Section 6.4 and the summary of the Chapter is discussed in Section 6.5.
6.2 LITERATURE SURVEY

Several private key algorithms such as, DES, triple DES, RC2, RC6, UR5 and UMARAM, have been discussed and compared based on their execution time [114]. The DES and triple DES are 64 bit block cipher algorithms with different key sizes such as, 64 bit and 192 bit respectively. RC2 is another block cipher mechanism with a 64 bits block and a variable key size that range from 8 to 128 bits. The RC6 is also block cipher mechanism derived from RC5. It was designed to fulfill the requirements of the AES competition. RC6 has a proper block size of 128 bits and key sizes of 128, 192 and 256 bits. The UR5 is a new symmetrical encryption algorithm, which has series of transformations based on S-BOX, XOR Gate, and AND Gate. The UR5 algorithm encrypts a plaintext of size 64 bits by a key size of 64 bits. It uses eight rounds for encryption and decryption mechanisms. The UMARAM is a new symmetrical encryption algorithm, where key generation process generates 16 keys in 16 rounds. The first key is used in first round of encryption and decryption mechanisms and so on. The UMARAM algorithm uses a key size of 512 bits to encrypt a plaintext of 512 bits during the 16 rounds. The S-Box is used to map the input code to another code at the output. It is a matrix of \(16 \times 16 \times 16\). The S-Box consists of 16 slides, and each slide has 2D of \(16 \times 16\). The numbers from 0 to 255 are arranged in random positions in each slide.

The data security over the network using non-functional parameters such as, confidentiality, authentication and integrity mechanisms are explained [115]. The confidentiality can be defined as mechanism in which, the data transmitted over the network has to be accessed only by the authorized users. The authentication is a mechanism in which end users are authenticated by some security checks provided by pre-defined systems. In current web scenario, the username and password as single phase authentication mechanism is used to authenticate end users. The integrity is defined as a mechanism
in which the accuracy of data transmitted over the network is to be maintained. It also expanded as the only authorized users are allowed to modify the transmitted information. The unauthorized user in between the sender and receiver is not allowed to alter the given message. A comparative study is carried out, which states that the private key algorithms are faster than public key algorithms [115]. The security aspects of private key algorithms are better than public key algorithms in terms of performance. For example, Blowfish has better performance than other common encryption algorithms, Cipher Block Chaining (CBC) is much better than Electronic Codebook (ECB) in terms of protection and AES has a better performance than triple DES and DES etc. The private key or symmetric key algorithms and public key or asymmetric key algorithms are compared on the basis of encryption time and memory usage for textual data [116]. The encryption time is considered as the time that an encryption algorithm uses to produce a ciphertext from a plaintext. They also have illustrated that the RSA has very smaller output byte as compared to AES and DES algorithms. RSA algorithm consumes more time as compared to the time consumed by AES and DES algorithms. The variation in memory usage is also observed, i.e., the memory usage doesn’t increase in accordance with file size increases. The comparison of DES and triple DES is explained on the basis of performance issues and security aspects [117].

The DES algorithm encrypts and decrypts data in 64 bit blocks, using a 64 bit key, although the effective key strength is only 56 bits, as 8 bit is used for parity. This algorithm uses a 64 bit block of plaintext as input and provides the output as 64 bit block of ciphertext. The DES has 16 rounds of processing such that the single block will be processed 16 times before producing the first block of ciphertext [118]. It has been found that the number of rounds are exponentially proportional to the amount of time required to find a key using brute force attack. The increased number of rounds will enhance the security exponentially. The authors state that DES algorithm consumes least
encryption time and AES algorithm has least memory usage. The encryption time difference is minor in case of AES algorithm as compared to DES algorithm. RSA consumes longest encryption time and high memory usage.

The triple DES algorithm is much secure as compared to DES algorithm considering large key size of 192 bits. The triple DES algorithm processes the data block thrice the DES algorithm with encryption, decryption and encryption processes in specific order with block size of 64 bits [115]. The triple DES algorithm is slow in execution due to three times serial processing of single block as compared to the DES algorithm [116, 117]. The triple DES processing is shown in Figure 6.1.

![Figure 6.1: Triple - DES Algorithm](image-url)
There are different approaches for security aware web service composition such as, syntactic-based and semantic-based approaches [119]. In the syntactic approach, a central coordinator or the orchestrator is used to invoke and combine the atomic activities and compose available web services. In the semantic approach, a central coordinator is substituted and assigned a complex task based on the definition of the semantics.

A broker based framework has been introduced for integration and adaptation of web services [120]. The broker mainly performs service tracking and selection of appropriate web services. They have focused on non-functional characteristics such as, response time, service cost, availability and reliability. The automation of web service composition has been explained [121]. The automation means that either the method can generate the process model automatically, or the method can locate the correct services, if an abstract process model is provided. The process of automation has different phases such as, presentation of web services, generation of business processes, evaluation and execution of composite services and development of workflow design. The web service policy and web service security mechanisms are already been used to propose a secure framework for business processes [122]. Here, XML encryption is employed to wrap confidentiality. XML signature is used to provide integrity and security token for authentication.

The Aspect Oriented Extension for BPEL (AO4BPEL) has a support for adaptable and modular web services. These adaptable web services are implemented with the help of an aspect aware orchestration engine. The development of business processes in accordance with end user’s security requirements and service provider capabilities have been explained [123]. The decentralized secure framework has been presented to execute composite web service in secure manner [124]. The authentication and authorization patterns can
be integrated with business processes based on access control mechanisms [125].

The DES algorithm is vulnerable to brute force attack, and therefore, it has certain limitations in security perspectives [117]. In other case, triple DES is much secure, but requires much CPU power consumption as compared to the DES algorithm [117]. Therefore, there is a requirement of an algorithm, which can be as secure as triple DES and as faster as DES algorithm.

6.3 MULTITHREADING BASED SECURE ALGORITHM

The proposed multithreading based secure algorithm (MBSA) combines and enhances the higher security aspects with fast execution process. The proposed algorithm has support of key matrices. The key matrices are the matrix of different keys used in encryption and decryption mechanisms. The key matrices at sender side and receiver side are shown in Figure 6.2(a), and Figure 6.2(b), respectively. The concept of key matrices work in such a way that the different key pairs will be used for different data blocks. The key matrices provide much security and robustness to the proposed algorithm over the network. Suppose the data block DB1 and DB2 are available for processing then the key pairs for both the data block will be different in MBSA, which is similar to triple DES. The different key pairs are available in key matrices.

![Figure 6.2 (a): Key Matrices at Sender Side](image)

![Figure 6.2 (b): Key Matrices at Receiver Side](image)
With comparison to triple DES, in MBSA, the first data block DB1 is processed with K1 – K2 – K3 key pair, DB2 is processed with K2 – K3 – K1 key pair and DB3 is processed with K3 – K1 – K2 key pair before converting the plaintext blocks (DB1, DB2 ... DBN) into the ciphertext blocks (CB1, CB2 ... CBN) at sender side. At receiver side, CB1 is processed with K3 – K2 – K1 key pair, CB2 is processed with K1 – K3 – K2 key pair, CB3 is processed with K2 – K1 – K3 key pair before converting the ciphertext blocks into the plaintext blocks. It is observed that MBSA algorithm has different key pair for different data blocks, which is not possible with the triple DES algorithm. The triple DES has only single and key pair K1 – K2 – K3 for all the data blocks at sender side and K3 – K2 – K1 for all the cipher blocks at receiver side. In MBSA, the data blocks DB1, DB4, DB7….DBX will be processed with same key pair. The data blocks DB2, DB5, DB8….DBY will be processed with the same key pair. The data blocks DB3, DB6, DB9….DBZ will be processed with the same key pair. Here X, Y and Z can be any multiple of number. The organization of key matrices and the concept of multithreading are discussed in the following subsections.

6.3.1 Multithreading and Key Matrices

The concept of key matrices provide security to MBSA in a way that if unauthorized user has the key pair for decrypting the ciphertext into plaintext then it requires more time to decrypt it because the different key pairs is used for different blocks. In triple DES, it is easy to decrypt the ciphertext into plaintext as same key pair is used for all data blocks, whereas the proposed MBSA algorithm is used to decrypt the ciphertext into plaintext with different key pair for all the data blocks. Hence, MBSA has overcomed the limitations of existing algorithms such as, DES and triple DES.

The design of key matrices is carried out in such a way that if we provide the key pair of first block to sender then sender will generate all the key pairs by rotating first key pair in anti-clock wise direction.
If we provide key pair of first block to receiver then receiver will generate all the key pairs by rotating first key pair in clock wise direction. The concept of multithreading in MBSA has support of multiple threads, where each thread is responsible for processing of different data blocks. The processes and threads are the mechanism of parallelizing a data processing for its fast execution and less memory consumption requirements. The processes are independent execution units that contain their own state information, use their own address spaces, and only interact with each other through Inter Process Communication (IPC) managed by the operating system. All the threads within a process share the same state and same memory space, and can communicate with each other directly.

In MBSA, multiple threads are used to encrypt and decrypt multiple data blocks in parallel such that thread T1 will process the DB1 block, T2 will process the DB2 block and T3 will process the DB3 block in parallel. The multithreading concept states that increasing the number of threads in a process speeds up execution. The multithreading based secure architectural diagrams for encryption and decryption processes are illustrated in Figure 6.3 and Figure 6.4, respectively. In MBSA, the plaintext is divided into 64 bits of block size and different independent blocks are ready for processing. Each 64 bit block is processed with different threads in parallel with different key pairs. Suppose we have different threads as T1, T2...,TN and data blocks as DB1, DB2 .... DBN (plaintext blocks) at sender side. The sender side and receiver side mechanisms are shown in Figure 6.5 and Figure 6.6, respectively.

The sender side processing generates three ciphertext blocks such as, CB1, CB2 and CB3. These three blocks will be processed further at the receiver side. The outcome of receiver side processing is the original plaintext blocks. The algorithmic representation of MBSA at both sender and receiver side are described in subsequent subsections.
Figure 6.3: Encryption in MBSA Algorithm
Figure 6.4: Decryption in MBSA Algorithm
Figure 6.5: MBSA Algorithm Processing at Sender Side

Figure 6.6: MBSA Algorithm Processing at Receiver Side
6.3.2 Algorithmic Representation

The algorithmic representation of proposed multithreading based secure algorithm is illustrated in subsequent subsections. The representation includes the sender side and receiver side activities.

6.3.2.1 Sender Side Process

The plaintext is divided into different data blocks at sender side. Each data block will be processed with different threads in a multithreading environment. Each data block will be encrypted by different key pairs managed by key matrices. Each data block is encrypted using proposed MBSA algorithm and corresponding ciphertext block is generated. All the ciphertext blocks will then be appended as single ciphertext block. In proposed algorithm, N is number of data blocks/ number of threads and K is the key in the algorithm. The value of $i \in \{1\ldots N\}$ and $j \in \{1\ldots 3\}$ in the proposed algorithm. The proposed algorithm has used the linear data structure called array for processing and storage activities. The algorithm for sender side process is as follows:

Begin

Divide plaintext into data blocks

Read data block $DB_i$

Take key pair $K_j$

Start thread $T_i$

Perform all three operations

$\{\text{encryption} \rightarrow \text{decryption} \rightarrow \text{encryption}\}$

Repeat above steps for all data blocks. Each data block is processed by separate thread.
Generate ciphertext data blocks \( CB_i \) and append ciphertext data blocks together as single ciphertext block.

End.

The appended ciphertext block will be send to the receiver over the network. The deciphering of ciphertext block requires lots of time for any unauthorized users, even when the keys are known to unauthorized user.

### 6.3.2.2 Receiver Side Process

The appended ciphertext block is divided into different ciphertext blocks at receiver side. Each ciphertext block will be processed with different threads in multithreading environment. Each ciphertext block will be decrypted by different key pair managed by key matrices. Each cipher block is decrypted using proposed MBSA algorithm and corresponding plaintext block is generated. All plaintext blocks will be merged as single plaintext block. Once the original plaintext block (request from service consumer) is generated at receiver side then the web services will fulfill the incoming request and get the responses. The algorithm for receiver side process is as follows:

Begin

*Divide appended single ciphertext block into ciphertext blocks*

*Read cipher block \( CB_i \)*

*Take key pair \( K_j \) {In reverse order}*

*Start thread \( T_i \)*

*Perform all three operations*

*{decryption \arrow{encryption} \arrow{decryption}*}
Repeat above steps for all cipher blocks. Each cipher block is processed by separate thread.

Generate plaintext data blocks DBi and append plaintext data blocks together as single plaintext data block.

The appended plaintext block is forward to different web services to fulfill the incoming request and collect the respective response.

End.

6.4 EXPERIMENTAL EVALUATION

The experimental evaluation is carried on Windows Server 2008 with 4 GB RAM and Intel Core i3 Processor. The experiment is evaluated on block size of 64 bits and key size of 192 bits for triple DES and MBSA algorithm both. The experiment is tested with the environment of business processes, developed by set of web services followed by SOA architecture. The encryption and decryption processes in proposed system are based on ECB mode for data processing [126]. The experiments are performed on different key parameters such as, different file size and different file type with triple DES and MBSA both. The comparative performance of triple DES algorithm and MBSA algorithm is shown in Table 6.1.

<table>
<thead>
<tr>
<th>S. No.</th>
<th>File Type</th>
<th>File Size (in MB)</th>
<th>Time Complexity (in Sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Triple DES</td>
</tr>
<tr>
<td>1</td>
<td>Text / Word / Image File</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>2</td>
<td>Text / Word / Image File</td>
<td>5</td>
<td>140</td>
</tr>
</tbody>
</table>
As per experimental evaluation listed in table 6.1, the time complexity of existing triple DES algorithm is greater than time complexity of proposed MBSA Algorithm. The triple DES algorithm needs more time for encryption and decryption as compared to the MBSA algorithm. The observation concludes that the proposed system is faster than existing triple DES algorithm. The graphical performance comparison between triple DES and MBSA algorithms is shown in Figure 6.7.

![Graph showing time complexity comparison between Triple DES and MBSA](image)

**Figure 6.7: Time Complexity of Triple DES and MBSA Algorithm**
6.5 SUMMARY

The multithreading based secure algorithm for business processes has been illustrated and tested in this Chapter. This chapter focuses on providing high security to business processes over the network by introducing an MBSA algorithm, which has fast encryption and decryption time because of multiple threads support. Additionally, the algorithm has support of key matrices, which manages different key pair for different data blocks. The algorithmic representation at sender and receiver side is also described in the current chapter. The experimental evaluation on encryption time for triple DES and MBSA algorithms is analyzed on some important parameters such as, file size and file type. The graphical representation is also demonstrated in the chapter, which states that MBSA algorithm is as faster as DES and as secure as triple DES algorithm.