Enhanced Moving Target Defense Mechanisms to Handle Cyber Attacks

CHAPTER 7

Improved Data Chunking using Non-sequential Storage

Unknown cyber attack handling mechanisms are enhanced to increase the detection rate. Moving target defense mechanisms are found more appropriate in detecting the unknown cyber attacks. The four moving target defense mechanisms are

i. **Smart Motion Adaptation/Management – Game Theory**
ii. **Robust Cryptographic Authentication – Mouse Dynamics**
iii. **Data Chunking and Decentralization**
iv. **Decoys**

Enhanced data chunking and decentralization are discussed in this chapter. Data chunking and decentralization play a significant role in efficiently storing and securing the data. The work proposed in this research work is content defined chunking. This method is integrated with SHA-512 (Secure Hash Algorithm) and AES (Advanced Encryption Standard). The traditional way of storing the chunked files is also rearranged for more security.

Data Chunking[84] is one of the moving target defense mechanisms. In this phase, data chunking is implemented and improvised to defend against cyber attacks.

7.1. Proposed Method using Improved Data Chunking

The main idea of this phase is to develop a method which ensures strict verification of users before accessing the files stored in the database [12] and to prevent unauthorized access of data. The primary focus of this model is to provide easy access to the legitimate user which ensures data availability, secure data storage through encryption and decryption. Data chunking [76]provides security features like authentication, secure storage[53] and integrity.

In this research work, server based application is developed using Content Defined Chunking (CDC). Content Defined Chunking [9] is found to be appropriate here because it will dynamically chunk the content of the file depending on the file size.

Along with that, Secure Hash Algorithm (SHA-512)[77] is used for key generation and Advanced Encryption Standard (AES) algorithm[81][93][94] is used for encryption and decryption of the files chunked.

In this phase, Improved data chunking using non-sequential storage of chunked files is introduced. Data Chunking is proven secured data storage recently. The chunked files stored in the database are encrypted, which eliminates hacking problem. In order to understand the functioning flow of data chunking and decentralization, the existing method is implemented and then the improvisation is done. In the existing method, it is very easy to hack the files stored in the database, as it uses the traditional way of storing the chunked files. The method proposed in this phase re-arranges the traditional approach. The entry point is secured with the password and key verification which is given for every user at the time of registration. Even if the attacker compromises the entry level security process, it will not be able to view the content of the files that are chunked, encrypted and stored according to a non-sequential storage. If the key matches then the author will be allowed for entry. Following that, the probability value will be calculated using markov chain model for every event. If the calculated probability value lies within the given range, the entry will be decided as authorized user or non-infected files. Otherwise login will be rejected at the entry point itself. The proposed method consists of four steps discussed below.

7.2. Phases of this Research Work

The proposed work consists of the following phases and it is given below in detail. Usually, data chunking[41] will chunk and store the files in a sequential order like file 1 in server 1, file 2 in server 2. In this research work non-sequential order, i.e., random storage of files is introduced.

7.2.1. Flow diagram of the Proposed Method

The flow diagram of the proposed method is given below in diagram.7.1

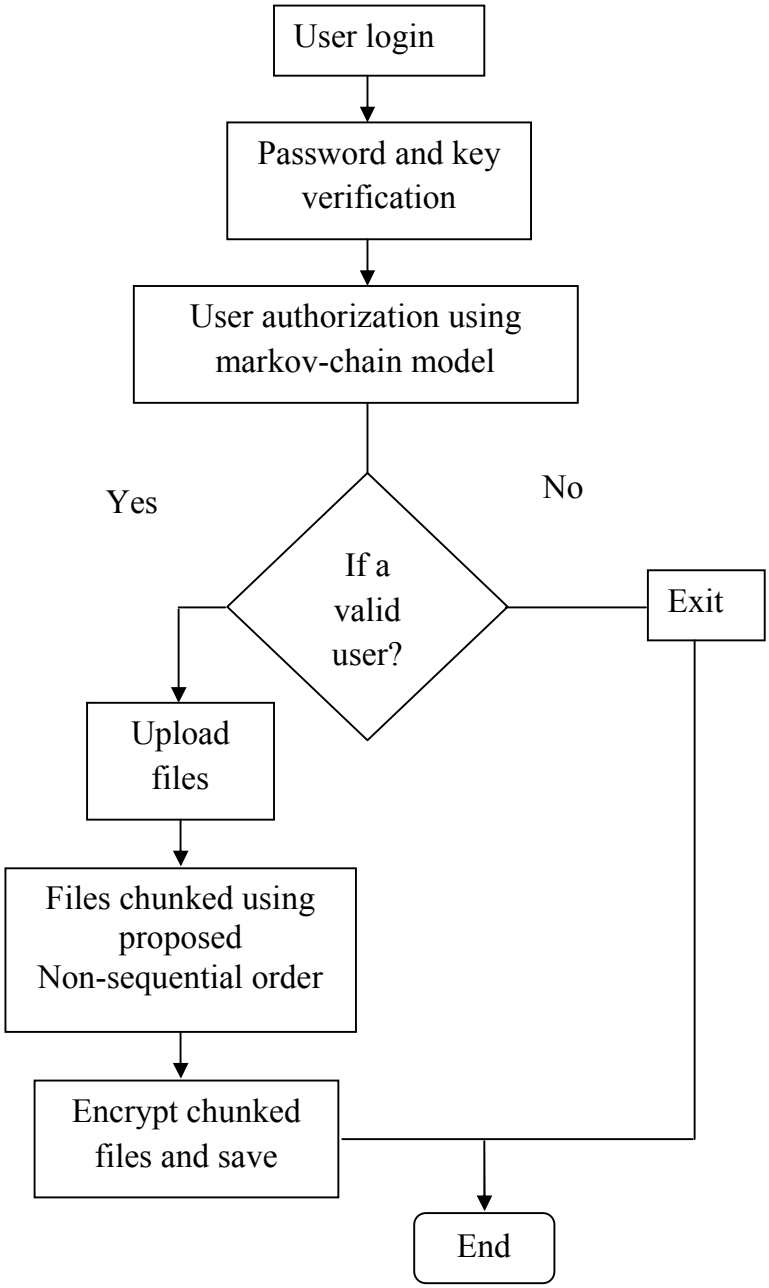

Figure.7.1 Flow chart of the improved data chunking

7.2.2. Steps involved in this Research Work

This research work consists of the following steps:

Step.1: Initialize Server

Connection is established by starting the server, is shows the availability of the server for the client for accessing.

Step.2: The user registration process

The user can register his details to the server.

Step.3: Key generation process

The unique key will be generated for every single user at the time of registration.

Step.4: Attack detection using markov-chain model

The cyber attacks like active and passive attacks will be detected using markov-chain model for user authorization.

Step.5: The chunking process

The existing sequential and proposed non-sequential order of chunking will be executed.

Step.6: The encryption process and proposed non-sequential storage

The files chunked will be encrypted and it will be stored according to the proposed unique combination.

7.2.2.1. Registration and key generation

User registration and key generation are the initial process of this proposed method. After the server is initiated, the user can register him/her with the details requested. Once the user is registered he/she will be allotted with unique key. Secure Hash Algorithm-512[30][34][43][52][74] is used to generate the key. It is a cryptographic hash function algorithm which helps to transform the text file as the value in a fixed length and to compress the message called a message digest. The algorithm acquires 2128 bits as an input which uses 1025 bit blocks to generate 512 bits of the message digest as output. Compression function and message schedule are the two components of SHA-512. The processing steps are Append padding bits, Append length, Initialize hash buffer, 80 rounds and the output.

The key generating process using SHA-512 is given in figure.7.2.

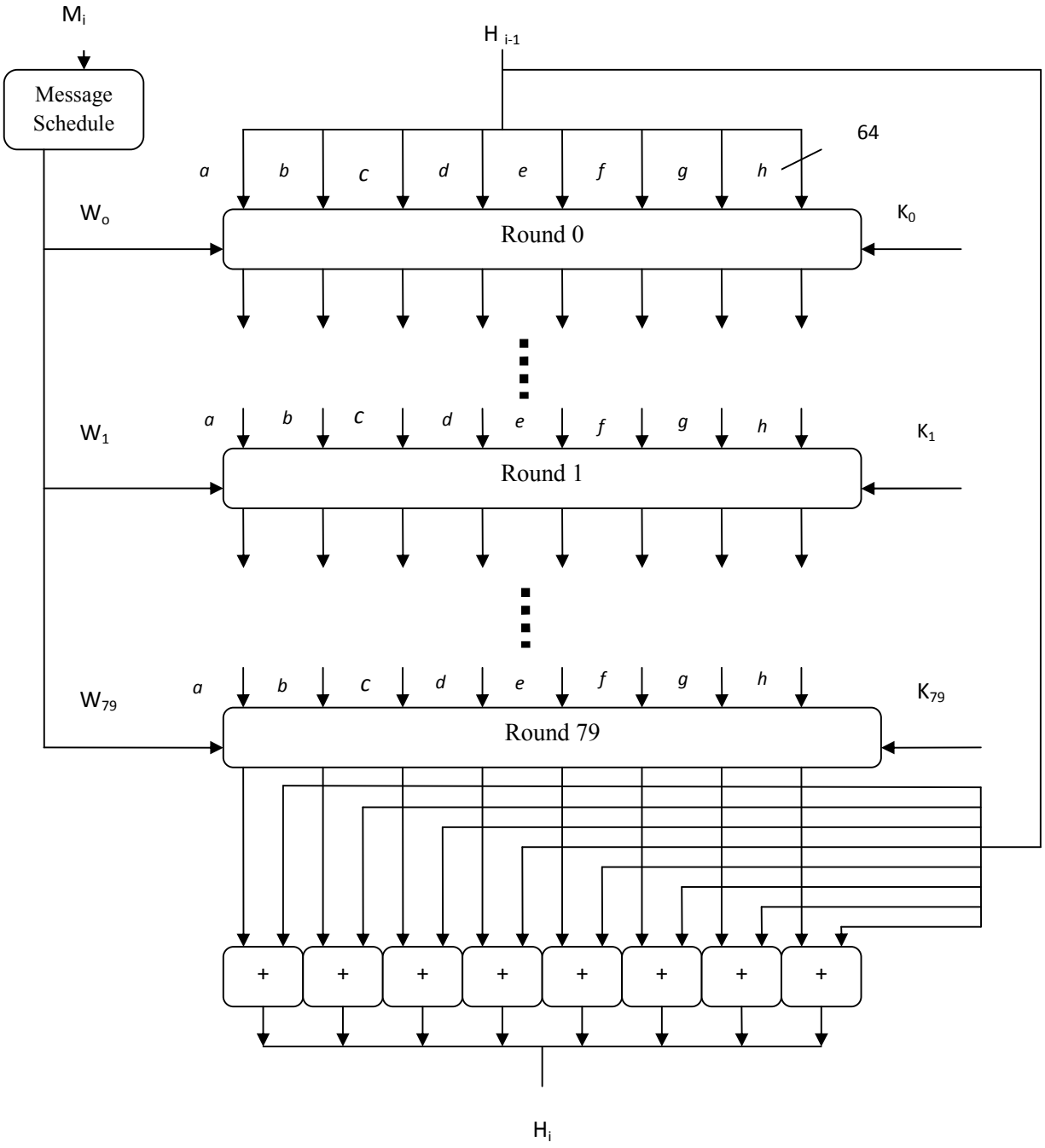

Figure.7.2 Key generating process of SHA-512[97]

7.2.2.2. Attack detection using Markov-Chain Model

In this phase, the cyber attacks are detected using the markov-chain model. Markov-chain model is used for anomaly-detection. The unusual event or occurrences from normal profile will be observed by anomaly-detection techniques which are considered as attacks. It is capable of differentiating the known and unknown pattern from normal data pattern. Markov chain helps in envision the characteristics of state transition which happens in the future. It is an efficient method in analyzing the probability of state transition. In this proposed method, markov chain model is used to detect processes like network state monitoring and abnormal activity prediction by evaluating the probability of any occurrences which helps in predicting the unauthorized users and attacks.

The Markov chain model[60] is created using the historic data of the systems' normal profile activities. The observed data are analyzed to surmise the probability where the markov chain model supports the observed activities. Robustness of the Markov chain model for cyber attack detection are computationally intensive due to the use of the Bayes' parameter estimation for learning the norm profile and a likelihood ratio test for inferring anomalies.

Markov chain model is a special type of discrete-time stochastic process with the following assumptions.

$$\Pr\{X_{t+1} = i_{t+1} \mid X_t = i_t, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0\} = \Pr\{X_{t+1} = i_{t+1} \mid X_t = i_t\},$$

for any t € T and i_t € s. That is, the probability distribution of the state at time *t+1* depends on the state at time t, and does not depend on the previous states leading to the state at time t. Thus, a markov chain model considers only 1-step transition probabilities.

7.2.2.3. Chunking process

In Data chunking , the files will be split into chunks in various sizes smaller than the original file which helps to transmit the data. This method suits for copying the larger files into the disk and it can be used for separating the files into chunks. The chunked file will be given name with some extension in a sequential order, so that there will not be any complexity in accessing the files in the future. No further indexing or maintenance is required after the files are being chunked. After the files have been chunked into two or more files, it will be stored in different servers in a sequential order [34]. If there are three servers, the files uploaded will be split into three files and it will be stored in those three servers in a predefined order like file 1 in server l, file 2 in server 2 and file 3 in server 3.

7.2.2.4. Encryption Process

In this method, after the chunking process is over, the chunked files will be encrypted before storing it into the database. The Advanced Encryption Standard (AES) algorithm[37][42][68] is used for encryption and decryption process. AES is a block cipher which uses 128-bit block for encryption and 128-bit block for decryption. It also uses a key length as 128,192 or 256 bits . The transformation is consecutively predefined as 10, 12 and 14 iterations for data block process as correspondingly AES-128, AES-192 and AES-256 which are termed as rounds. A key expansion algorithm is used to obtain the cipher text, whereas the algorithm will be utilized separately for encryption and decryption. All the functions except the last round for encryption uses the following four transformations such as SubBytes, ShiftRows, MixColumns and AddRoundKey. The encryption process will have the reverse transformation for decryption as InvSubBytes, InvShiftRows, and InvMixColumns. The structure of the encryption can be formulated precisely from the structure of the decryption. The processing steps of the AES algorithm are given in figure.7.3.

Following is the sequential order of steps involved in AES Algorithm:

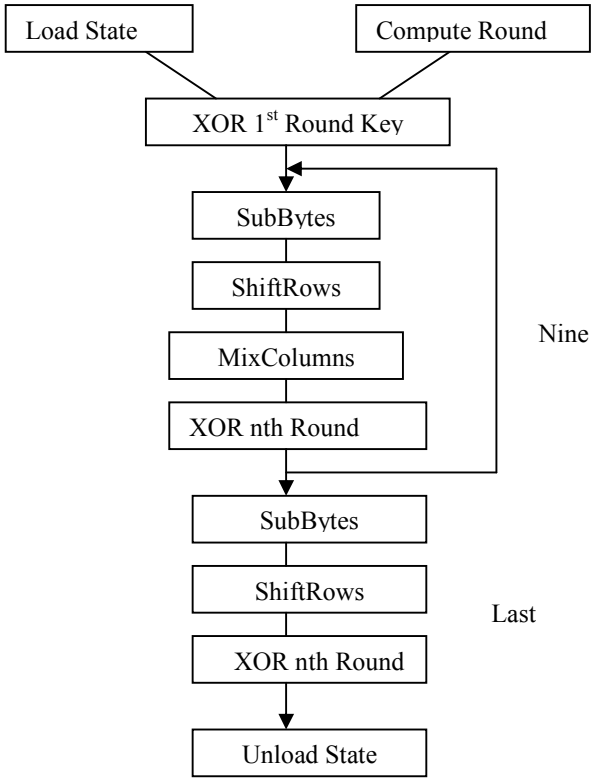

Figure.7.3 Processing steps of AES Algorithm

7.2.2.5. Proposed Non-sequential Storage

In the proposed method non-sequential order is introduced. This method uses three servers say server1, server2 and server3. The chunked files can be stored in the servers in twelve combinations like,

Server1, server 2 and server 3 (1, 2, 3)

Server 1, server 3 and server 2 (1, 3, 2)

Server 1, server 2 and server 3 (1, 2, 3)

Server 1, server 3 and server 3 (1, 3, 3)

Server 2, server 1 and server 3(2, 1, 3)

Server 2, server 3 and server 1(2, 3, 1)

Server 2, server 1 and server 1(2, 1, 1)

Server 2, server 3 and server 3(2, 3, 3)

Server 3, server 1 and server 2(3, 1, 2)

Server 3, server 2 and server 1(3, 2, 1)

Server 3, server 1 and server 1(3, 1, 1)

Server 3, server 2 and server 2 (3, 2, 2)

Among these combinations, the unique combinations are selected to avoid the storage of file redundancy in the same server. The five unique combinations used in this research work are:

Server 1, server 3 and server 2

Server 2, server 1 and server 3

Server 2, server 3 and server 1

Server 3, server 1 and server 2

Server 3, server 2 and server 1

The chunked files will be stored in unique combination of servers given above.

7.2.3. Proposed Algorithm

The algorithm of the proposed research work is given in Table.7.1

Table.7.1. Proposed Algorithm of Improved Data Chunking

```
U →User
D →Database
S →Server
Initialize server
 for each user logging in
      register and generate key
      upload file
      perform  verification
      execute markov-chain model
   if user authorized then
     chunk file using proposed non-sequential
order
      perform encryption
     store it is 'S'
  else
     exit
  end if
end
```

7.3. Performance Metrics

The performance of the proposed method is evaluated using the following performance metrics. They are:

i. False Acceptance Rate

ii. False Rejection Rate

iii. Cyber Attack Detection Rate

False Acceptance Rate

The false acceptance rate is a fraction of negative entry or unauthorized user was incorrectly identified as positive entry or authorized user and it will be calculated using the following formula:

$$\text{False Acceptance Rate} = \frac{number\ of\ \ false\ accep\text{tan}ces}{number\ of\ \ client\ \ accesses}$$

False Rejection Rate

The false rejection rate is a fraction of positive entry or authorized user that was incorrectly identified as negative entry or unauthorized user and it will be calculated using the following formula:

$$\text{False Rejection Rate} = \frac{number\ of\ \ false\ rejections}{number\ of\ \ client\ \ accessses}$$

The proposed method is intended to detect the cyber attacks. The attack detection rate is calculated using the following equation:

$$\text{Detection Rate} = \frac{Number\ of\ \ attacks\ \text{det}ected}{Number\ of\ \ attacks\ injected}$$

7.4. Simulation Environment

The proposed method is implemented using JAVA 1.7 as front end and mysql as a back end for experimentation. The proposed method is evaluated for performance in terms of attack detection rate, false acceptance rate and false rejection rate. The experiments were conducted for 24 participants.

7.5. Results and Discussions

The aim of this section is to evaluate the efficiency of the research work in terms of some performance metrics.

Table.7.2. Results of FAR and FRR for every 50 users

Users	False Acceptance Rate		False Rejection Rate	
	Proposed Non-sequential Order	**Sequential Order**	Proposed Non-sequential Order	**Sequential Order**
50	0.88	0.91	0.34	0.33
100	0.84	0.87	0.30	0.36
150	0.82	0.84	0.27	0.30
200	0.77	0.80	0.22	0.35

The results taken for every 50 users registering with the proposed method is given in Table.7.2. The above table shows the false acceptance rate and false rejection rate evaluated using the proposed method. The attack detection rate is given below in Table.7.3.

Table.7.3. Cyber attack detection rate

Attack Types	Sequential Order	Proposed Non-sequential Order	Percentage of Improvement
Active Attacks	84%	87%	**3%**
Passive Attacks	76%	79%	**3%**

The attack infected files are uploaded along with the normal files to the proposed method. The proposed method detects 70% infected files before chunking process is done and the results are shown in Table.7.4.

Table.7.4 Detection Rate of infected file by the proposed method

File Type	Detection Rate	
	Sequential Order	**Proposed Non-sequential Order**
Normal Files	100%	**100%**
Attack Infected Files	73%	**80%**

7.6. Chapter Summary

The main idea of this phase is to develop a model which ensures strict permission and verification in accessing the files stored in the database. This model is developed by rearranging the existing sequential order in saving the chunked files into the server. The proposed method provides security features like user authentication, secure communication and Integrity. The model developed defends against cyber attacks namely active and passive attacks. The primary focus of this model is to provide easy and secured access to the legitimate users to ensure availability, secure storage through encryption and decryption. The various levels of security are assigned and compared with existing method. The system model is developed to evaluate the performance of the data chunking for security and preventing data or information from cyber attacks. This model also helps in detecting the cyber attacks accurately.