**CHAPTER III  
PROBLEM ANALYSIS**

MD5 algorithm provides a security string for data in database specialized corporate and applications. People need time to get used to the latest technology and started to get used to them. The public is curious about used of MD5 algorithm when it has done designed. Therefore, to avoid mistakes then we should know the details of the MD5 algorithm in terms of its use and applicability.

1. **MD5 Algorithm**

MD5 processes a variable length message into a fixed length output of 128 bits. The input message is broken up into chunks of 512 bit blocks. The message is padded so that length is divisible by 512. The padding works as follows, first a single bit, 1 is appended to the end of the message. This is followed by as many zeros as are required to bring the length of the message up to 64 bits fewer than multiple of 512. The remaining bits are filled up with a 64 bit integer representing the length of the original message, in bits.

The main MD5 algorithm operates on a 128 bits state, divided into four 32 bits word, denotes A, B, C, and D. These are initialized to certain fixed constraint. The main algorithm then operates on each 512 bit message block in turn, each block modifying the state. The processing of the message block consists of four similar stages, termed round, each round is composed of 16similar operations based on a non-linear function functions F, modular addition, and left rotation. Figure 1 illustrates one operation within a round. There are four possible functions F; a different one is used in each round:

1. *F(X,Y,Z)=(X ∧Y)∨(¬X ∧Z)*
2. *G(X,Y,Z)=(X ∧Z)∨(Y ∧ ¬Z)*
3. *H(X,Y,Z) = X ⊕ Y ⊕ Z*
4. *I(X,Y,Z) = Y ⊕ (X ∨ ¬Z)*

*⊕, ∧, ∨, ¬* denote the XOR, AND OR and NOT operations respectively.

1. **How to Work of MD5 Algorithm**

The MD5 algorithm first divides the input in blocks of 512 bits each. 64 Bits are inserted at the end of the last block. These 64 bits are used to record the length of the original input. If the last block is less than 512 bits, some extra bits are padded to the end. Each block is divided into 16 words of 32 bits each. These are denoted as M0 until M15. MD5 algorithm calculates process as follow:

1. **Appending Padding Bits**

The b-bit message is padded so that a single 1 bit is added into the end of the messages. Then 0 bits are added until the length of the message is congruent to 448, modulo 512.

1. **Appending Length**

A 64-bit representation of b is appended to the result of the padding. Thus, the resulted message is a multiple of 512 bits.

1. **Buffer Initialization**

Bit A, B, C and D will be 32-bit registers. These registers are used in derivation of the 128-bit message digest. At the beginning, they are initialized as follows:

*A = x67452301*

*B = xe f cdab89*

*C = x98badc f e*

*D = x10325476*

1. **Processing of the Message**

The heart of MD5 is an algorithm which is used for the processing of the message. The message M is divided into 512-bit blocks which are processed separately.

The algorithm consists of four rounds, each of which comprised 16 steps. Hence, 64 steps are performed in the algorithm. The algorithm is performed as follows: first, values of A, B, C and D are stored into temporary variables AA, BB, CC and DD. Then, the following operations are performed for i = 0 to 63:

*A = B+ ((A+Func(B,C,D) +Xj[k] +T[i])≪ s) A ← D, B ← A, C ← B, D ← C*

Func(X,Y,Z) is different for every round. Function F(X,Y,Z) is used for the first round (0 ≤ i ≤ 15), G(X,Y,Z) for the second (16 ≤ i ≤ 31), H(X,Y,Z) for the third (32 ≤ i ≤ 47) and I(X,Y,Z) for the final round (48 ≤ i ≤ 63). The functions are defined as follows:

*F(X,Y,Z)=(X ∧Y)∨(¬X ∧Z)*

*G(X,Y,Z)=(X ∧Z)∨(Y ∧ ¬Z)*

*H(X,Y,Z) = X ⊕ Y ⊕ Z*

*I(X,Y,Z) = Y ⊕ (X ∨ ¬Z)*

Where ∨ is a bitwise or-operation, ¬ is a bitwise complement, ⊕ is a bitwise exclusive or operation (xor) and ∧ is a bitwise and-operation. Finally, the values of the temporary variables are added to the values obtained from the algorithm.

*A = A+AA*

*B = B+BB*

*C = C +CC*

*D = D+DD*

1. **Output**

When all bit have been processed with the algorithm, the message digest of M is in A, B, C, and D. The low-order byte of A is the first byte of the digest and the high-order byte of D is its last byte.

Put simply, MD5 is a block chained digest algorithm, computed over the data in phases of 512-byte blocks organized as little-endian 32-bit. The first block is processed with an initial seed, resulting in a digest that becomes the seed for the next block. When the last block is computed, its digest is the digest for the entire stream. This chained seeding prohibits parallel processing of the blocks.

Each 512-byte block is digested in 4 phases. Each phase consists of 16 basic steps, for a total of 64 basic steps. Each step updates one word of a 4-word accumulated digest, using the entire intermediate digest as well as block data and constants. In general, each basic step depends on the output of the prior step, defeating simple parallelization of the steps. The basic structure of the steps is shown below (<<< denotes *rotate*). The accumulated digest is denoted by {A,B,C,D}, as in RFC-1321.

*A = B+((A+ F(B,C,D) + X[i++] + k1)<<<k2)*

*D = A+((D+ F(A,B,C) + X[i++] + k1)<<<k2)*

*C = D+((C+ F(D,A,B) + X[i++] + k1)<<<k2)*

*B = C+((B+ F(C,D,A) + X[i++] + k1)<<<k2)*

There are 16 steps based on each of 4 logical functions; 4 based on F are shown here. The constants k1 and k2 are not necessarily identical in basic steps, and are not relevant to this analysis. The logical functions

*F(x, y, z) = (((x) & (y)) | ((~x) & (z)))*

*G(x, y, z) = (((x) & (z)) | ((y) & (~z)))*

*H(x, y, z) = ((x) ^ (y) ^ (z))*

*I(x, y, z) = ((y) ^ ((x) | (~z)))*

The steps have optimization limitations, due to the mathematical properties of the operations used:

1. Additions can be reordered by commutative laws
2. Rotate does not distribute over addition
3. Addition does not distribute over rotation or logical



**Figure 3.1 MD5 Operation**

**REF: (**[**http://www.isi.edu/touch/pubs/sigcomm95.html**](http://www.isi.edu/touch/pubs/sigcomm95.html)**)**

1. **Advantages of MD5 Algorithm**

The advantages of MD5 algorithm analysis of which have been discussed are as follows:

1. **Utilizes a Fast Computation Algorithm**

MD5 algorithm may be the most time efficient method to use, maintain and improve the effectiveness of time used. Usually for load security in the data requires a lot of time to make a hash.

1. **Provides Collision Resistance**

Using the MD5 algorithm integrated data will not collide and crash on the server or in a database. Peoples can freely save the datato serverand expand the data without having to fear the crash data.

1. **Widespread Used**

Used by many people, because this algorithm is commonly used in the present era. Thus, to study it will be very easy because many use and references available on the internet

1. **Provide a One Way Hash**

With one way hash, then the method used for the algorithm becomes simpler and easier to use

1. **Disadvantages of MD5 Algorithm**

While it is true that the information and data secured with MD5 algorithm that can be accessed anytime and from anywhere at all, there are times when this system can have some serious dysfunction are as follows:

1. **Security Flaws and Vulnerability**

MD5 algorithm is commonly used in data security, so it is easy to be learned by others and frail against hacking activities. Ordinary people usually do not realize the purpose of the algorithm that protects data, but for an IT Professional it will be very fatal in case of data leakage

1. **Less Secure than SHA-1 Security**

MD5 algorithm uses a much simpler method than SHA-1 so that the effectiveness SHA-1 being superior and more secure.

1. **Slower**

MD5 algorithm is slightly slower than MD4, but is more conservative in design. MD5 algorithm was designed because it was felt that MD4 was perhaps being adopted for use more quickly than justified by the existing critical review. MD4 was designed to be exceptionally fast, it is at the edge in terms of risking successful cryptanalytic attack.