MESSAGE DIGEST TECHNIQUE USED IN CRYPTOGRAPHIC HASH ALGORITHMS

Mr. Arihant D. Majalage[1], Mr. Pradip A. Chougule[2], Mr. Rajesh A. Sanadi[3]
[1](Department of Computer Science & Engg., Dr. J. J. Magdum college of Engineering, India)
[2](Department of Computer Science & Engg., Dr. J. J. Magdum college of Engineering, India)
[3](Department of Information Technology, Dr. J. J. Magdum college of Engineering, India)

ABSTRACT: *Information is most important thing between two communicating entities so we require to provide security to information. Therefore in order to provide security to achieve data integrity, data confidentiality & authentication. To achieve data integrity message digest of plaintext is calculated by applying hash function. Hash functions are used in cryptographic hash algorithms. In this paper we describe formation of message digest by applying cryptographic hash algorithms (Whirlpoo1 & Secure Hash Algorithm-512) to original plaintext, Efficiency of cryptographic hash algorithms for maximum length message & minimum length message. Performance of cryptographic hash algorithms & comparative study of message digest. Hash functions H are transformations that take variable size input M & returns fixed size string which is called hash value h. When hash function are used in cryptography have some basic requirements such as input can be any length, output has a fixed length(x) is relatively easy to compute for any given x.H(x) is one way, H(x) is collision free, a hash function H is said to e one-way if it is to hard to invert, here "hard to invert" means that given a hash value h, it is computationally infeasible to find some input x such that H(x)=h.*

Keywords*:* *Cryptographic hash function, Message Digest, Secure Hash Algorithm, Whirlpool.*

I. INTRODUCTION

The Cryptographic hash algorithms that are Whirlpool & secure hash algorithm-512 are used to provide data integrity & confidentiality. Whirlpool is a one-way hashing function designed by Paulo S.L.M. Barreto and Vincent Rijmen (also of AES fame). It was originally submitted to NESSIE (New European Schemes for Signatures, Integrity and Encryption) project and is the only hash function alongside SHA-256, SHA-384 and SHA-512 in the NESSIE portfolio. Whirlpool is based on 512-bit block cipher, which structure is similar to Rijndael (AES). It uses 512-bit keys. The block cipher is dedicated only to be used for hashing, which is very exceptional in cryptography i.e. Whirlpool block cipher will most likely never be used for standalone encryption. It is designed for both software and hardware implementations, with compactness and performance in mind. Basically whirlpool had three versions the first version the first version, WHIRLPOOL-0, was submitted to the NESSIE project. Its "tweaked" successor, WHIRLPOOL-T, was selected for the NESSIE portfolio of cryptographic primitives. A flaw in its diffusion layer reported by Shirai and Shibutani ("On the diffusion matrix employed in the Whirlpool hashing function," NESSIE public report, 2003) was fixed afterwards, and the final version (called simply WHIRLPOOL for short) was adopted by the International Organization for Standardization (ISO) in the ISO/IEC 2004 standard.

II. GOALS

The security goals for Whirlpool are:
Assume we take as hash result the value of any n-bit substring of the full Whirlpool output.
1. The expected workload of generating a collision is of the order of $2^{n/2}$ executions of Whirlpool. 2. Given an n-bit value, the expected workload of finding a message that hashes to that value is of the order of 2n executions of Whirlpool. 3. Given a message and its n-bit hash result, the expected workload of finding a second message that hashes to the same value is of the order of 2^n executions of Whirlpool. 4. Moreover, it is infeasible to detect systematic correlations between any linear combination of input bits and any linear combination of bits of the hash result. It is also infeasible to predict what bits of the hash result will change value when certain input bits are flipped, i.e. Whirlpool is resistant against differential attacks.

The security goals for secure hash function H are:
1. The security goal is to provide symmetric key encryption, asymmetric key encryption authentication & confidentiality. authentication & digital signature. 2. *One-way property*: for any given value h, it is computationally infeasible to find x such that H(x) = h. 3. *Weak collision resistance*: for any given message x, it is computationally infeasible to find y 6= x with H(y) = H(x). 4. *Strong collision resistance*: it is computationally infeasible to find any pair (x, y), such that H(x) =H(y).

III. Whirlpool

III.I Algorithm Overview

Like most hash functions, Whirlpool operates by iterating a compression function that has fixed-size input and output. Its compression function is a dedicated AES-like block cipher that takes a 512-bit hash state M and a 512-bit key K. (Hence, both the state and the key can be conveniently represented as 8 × 8 matrices with byte entries.) The iteration process adopts the well-known Miyaguchi-Preneel construction [15]. In what follows, we provide a concise description of the compression function that is most relevant to our implementation. Technical details of the algorithm can be found in [1]. At a high level, each execution of the compression function can be divided into two parts:

a. expanding the initial key K into ten 512-bit round keys, and
b. Updating the hash state M by mixing M and the round keys.

Part b consists of ten rounds, and each round consists of the following four steps (labeled W1 through W4 below) with byte-oriented operations:

W1. Non-linear substitution. Each byte in the state matrix M is substituted by another byte according to a predefined substitution, S(x) (aka S-box). W2. Cyclical permutation. Each column of the state matrix M is cyclic shifted so that column j is shifted downwards by j positions. W3. Linear diffusion. The state matrix M is multiplied with a predefined 8 × 8 MDS matrix C. 4 W4. Addition of keys. Each byte of the round key is exclusive-or (XOR) to each byte of the state. The key expansion (part a) is almost the same as the above state update, except that the initial key K is treated as the state and some pre-defined constants as the key. Hence, both parts consist of ten similar rounds. Note that Whirlpool differs from AES in that the rounds operate on 512-bit inputs rather than 128-bit inputs. Because of the larger block size, the design of the S-box and MDS matrix is also adjusted accordingly, but the general design philosophy remains the same.

III.II Whirlpool Logic

Given a message consisting of a sequence of blocks m_1, m_2, . . . , $m_.$ the Whirlpool hash function is expressed as follows:

H_0 = initial value.

$H_i = E(H_{i-1}, m_i) \oplus H_{i-1} \oplus m_i$ = intermediate values:

H_t = hash code value:

The encryption key input for each iteration i is the intermediate hash H_{i-1} value from the previous iteration, and the plaintext is the current message block mi.The output for this iteration (Hi) consists of the bitwise XOR of the current message block, the intermediate hash value from the previous iteration, and the output from W.

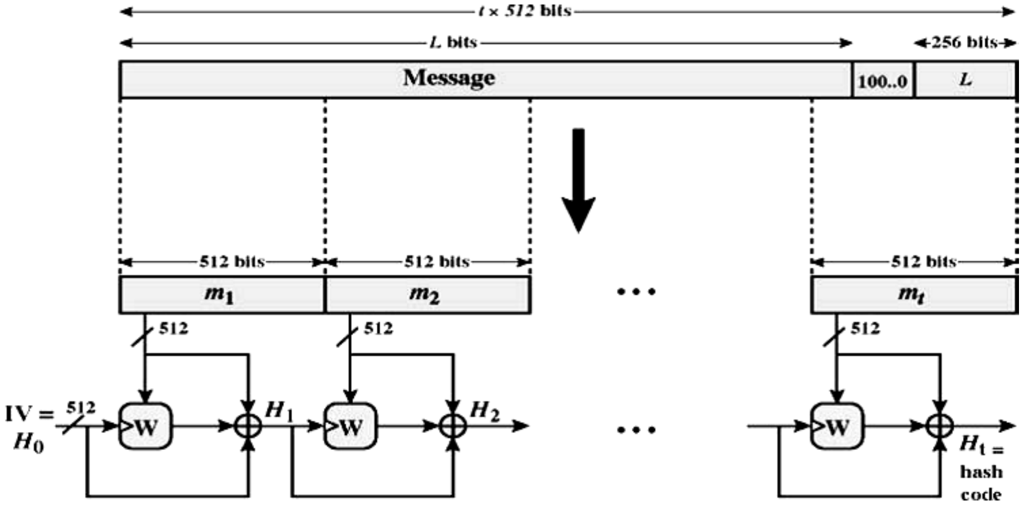

Figure 1: Message digest generation using Whirlpool.

The algorithm takes as input a message with a maximum length of less than2 ^{256} bits and produces as output a 512-bit message digest. The input is processed in 512-bit blocks. Figure 1 depicts the overall processing of a message to produce a digest.

The processing consists of the following steps:

Step 1: Append padding bits. The message is padded so that its length in bits is an odd multiple of 256. Padding is always added, even if the message is already of the desired length. For example, if the message is 256 * 3 = 768 bits long, it is padded by 512 bits to a length of 256 * 5 = 1, 280 bits. Thus, the number of padding bits is in the range of 1 to 512. The padding consists of a single 1-bit followed by the necessary number of 0-bits.

Step 2: Append length. A block of 256 bits is appended to the message. This block is treated as an unsigned 256-bit integer (most significant byte first) and contains the length in bits of the original message (before the padding). The outcome of the first two steps yields a message that is an integer multiple of 512 bits in length. In Figure 1, the expanded message is represented as the sequence of 512-bit blocks m_1, $m_{2,}$ . . . , m_t so that the total length of the expanded message is t *512 bits. These blocks are viewed externally as arrays of bytes by sequentially grouping the bits in 8-bit chunks. However, internally, the hash state Hi is viewed as an 8 * 8 matrix of bytes. The transformation between the two is explained subsequently.

Step 3: Initialize hash matrix. An 8 * 8 matrix of bytes is used to hold intermediate and final results of the hash function. The matrix is initialized as consisting of all 0-bits.

Step 4: Process message in 512-bit (64-byte) blocks. The heart of the algorithm is the block cipher W.

Block Cipher W Unlike virtually all other proposals for a block-cipher-based hash function, Whirlpool uses a block cipher that is specifically designed for use in the hash function The Whirlpool Secure Hash Function and that is unlikely ever to be used as a standalone encryption function. The reason for this is that the designers wanted to make use of a block cipher with the security and efficiency of AES but with a hash length that provided a potential security equal to SHA-512. The result is the block cipher W, which has a similar structure and uses the same elementary functions as AES [20], but which uses a block size and a key size of 512 bits. Although W is similar to AES, it is not simply an extension. In fact, AES is one version of the cipher Rijndael, which was submitted as a candidate for the AES. The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits. The AES specification uses the same three key size alternatives but limits the block length to 128 bits. AES operates on a state of 4 * 4 bytes. Rijndael with block length 192 bits operates on a state of 4 * 6 bytes. Rijndael with block length 256 bits operates on a state of 4 * 8 bytes. W operates on a state of 8 * 8 bytes. The more the state representation differs from a square, the slower the diffusion goes and the more rounds the cipher needs. For a block length of 512 bits, the Whirlpool developers could have defined a Rijndael operating on a state of 4 * 16 bytes, but that cipher would have needed many rounds and it would have been very slow [18]. As Table 1 indicates, W uses a row-oriented matrix whereas AES uses a column-oriented matrix. There is no technical reason to prefer one orientation to another, because one can easily construct an equivalent description of the same cipher, exchanging rows with columns.

IV The Secure Hash Algorithm

The Secure Hash Algorithm is a family of cryptographic hash functions published by the National institute of standards and Technology (NIST) as a U.S. Federal Information Processing standard (FIPS):

SHA-0

A retronym applied to the original version of the 160-bit hash function published in 1993 under the name "SHA". It was withdrawn shortly after publication due to an undisclosed "significant flaw" and replaced by the slightly revised version SHA-1.

SHA-1

A 160-bit hash function which resembles the earlier MD-5 algorithm. This was designed by the National Security Agency (NSA) to be part of the Digital Signature Algorithm. Cryptographic weaknesses were discovered in SHA-1, and the standard is no longer approved for most cryptographic uses after 2010.

SHA-2

A family of two similar hash functions, with different block sizes, known as SHA-256 and SHA-512. They differ in the word size; SHA-256 uses 32-bit words where SHA-512 uses 64-bit words. There are also truncated versions of each standardized, known as SHA-224 and SHA-384. These were also designed by the NSA.

SHA-3

A hash function formerly called Keccak, chosen in 2012 after a public competition among non-NSA designers. It supports the same hash lengths as SHA-2, and its internal structure differs significantly from the rest of the SHA family.

The corresponding standards are FIPS PUB 180 (original SHA), FIPS PUB 180-1 (SHA-1), FIPS PUB 180-2 (SHA-1, SHA-256, SHA-384, and SHA-512). NIST has said that FIPS 180-5 will include SHA-3.

The SHA-512 hash function takes for input messages of length up to 2^{128} bits and produces as output a 512-bit message digest (MD). Message digests produced by the most commonly used hash functions range in length from 160 to 512 bits depending on the algorithm used.

1. The length of the overall message to be hashed must be a multiple of 1024 bits. 2. The last 128 bits of what gets hashed are reserved for the message length value. 3. This implies that even if the original message were by chance to be an exact multiple of 1024, you'd still need to append another 1024-bit block at the end to make room for the 128-bit message length integer. 4. Leaving aside the trailing 128 bit positions, the padding consists of a single 1-bit followed by the required number of 0-bits.

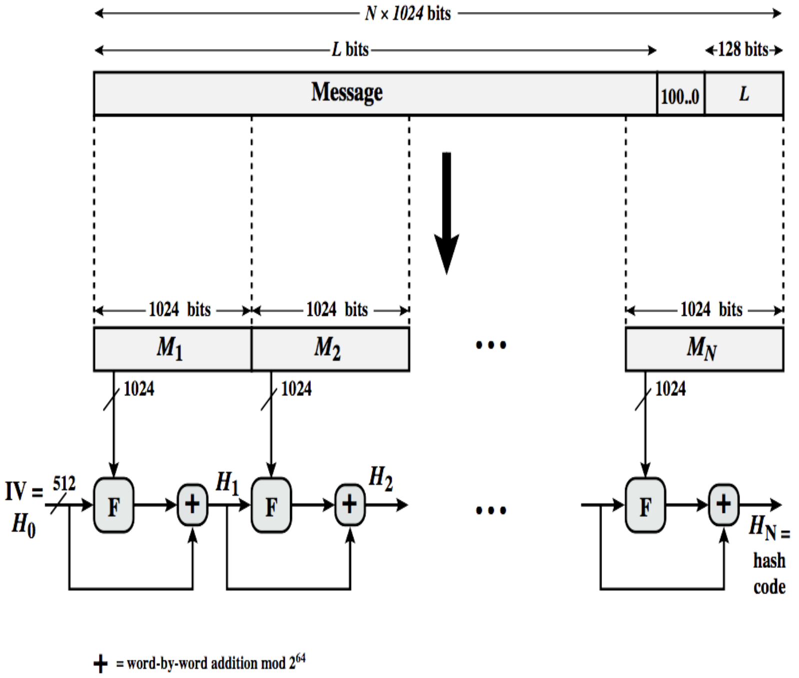

Figure 2: SHA-512 Message digests formation

Initialize Hash buffer with Initialization Vector: You'll recall from Figure 3 that before we can process the first message block, we need to initialize the hash buffer with IV, the Initialization Vector:
1. We represent the hash buffer by eight 64-bit registers. 2. For explaining the working of the algorithm, these registers are labeled (a, b, c, d, e, f, g, h). 3. The registers are initialized by the first 64 bits of the fractional parts of the square-roots of the first eight primes. These are shown below in hex:

6a09e667f3bcc908
bb67ae8584caa73b
3c6ef372fe94f82b
a54ff53a5f1d36f1

Process Each 1024-bit Message Block Mi: Each message block is taken through 80 rounds of processing.
1. The 80 rounds of processing for each 1024-bit message block are depicted in Figure 5. In this figure, the labels a, b, c, . . . , h are for the eight 64-bit registers of the hash buffer. Figure 5 stands for the modules labeled f in the overall processing diagram in Figure 2.
2. In keeping with the overall processing architecture shown in Figure 2, the module f for processing the message block M_i has two inputs: the current contents of the 512-bit hash buffer and the 1024-bit message block. These are fed as inputs to the first of the 80 rounds of processing depicted in Figure 3.

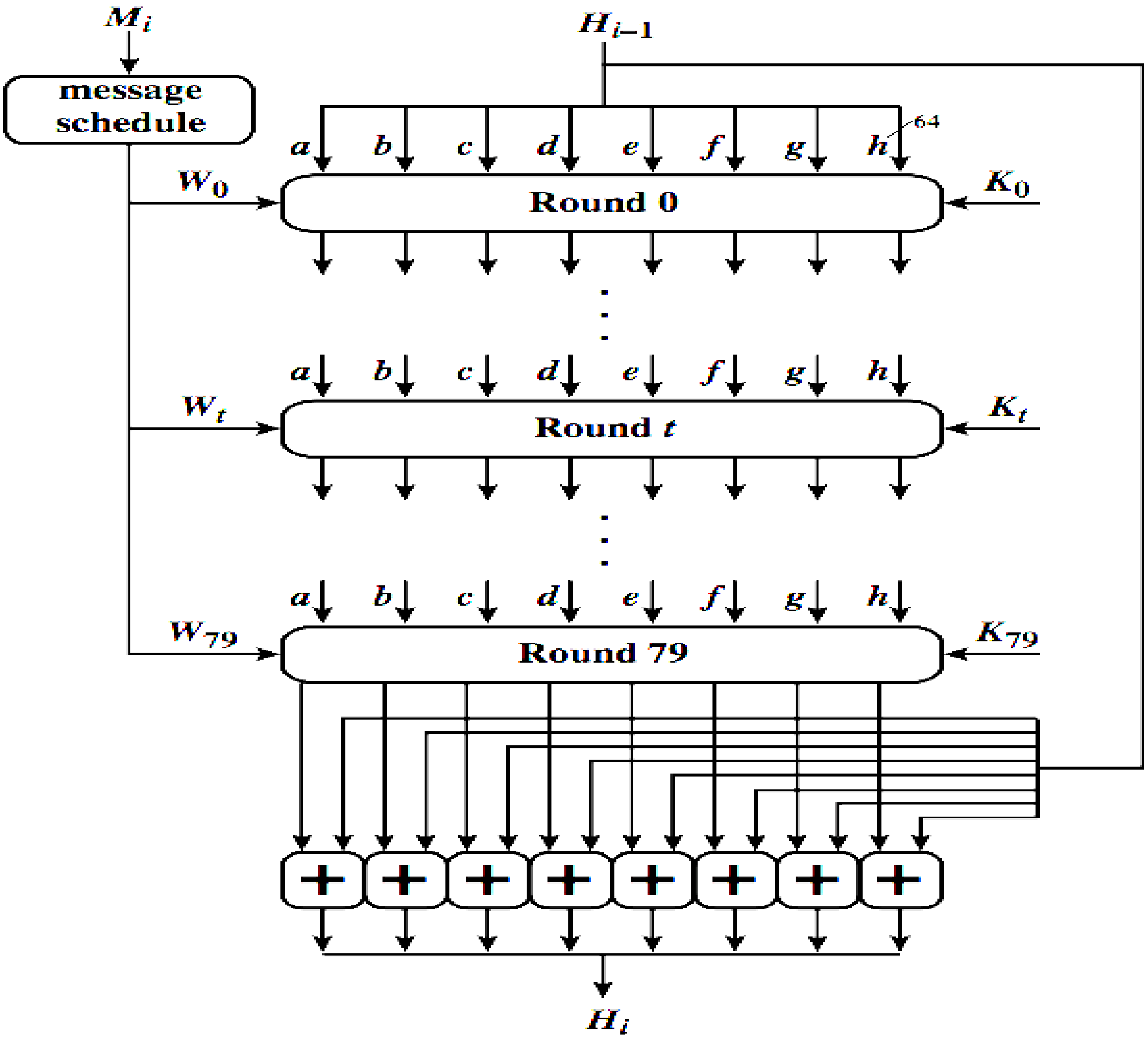

Figure 3: compression function

V COMPARISONS

The commonly-used cryptographic hash functions include the SHA-256, SHA-512 and WHIRLPOOL cryptographic hash functions. Each of these algorithms has security levels that are orders of magnitude above MD5 and SHA-1 due to their increased message digest sizes and further-refined algorithms. To compare, the best public cryptanalysis for each algorithm can be compared. For MD5, collision resistance has been broken in $2^{20.96}$ time, which is no more than a few seconds on an average computer (Xie, and Feng). For SHA-1, the hash function was broken due to hash collisions being producible with a complexity of 2^{51} operations (Manuel). For SHA-512 (SHA-2). As for SHA-512, the best public attack breaks preimage resistance for 46 of the 80 rounds (Sasaki, Wang, and Aoki). And lastly, for WHIRLPOOL, a rebound attack was disclosed presenting full collisions against 4.5 rounds in 2^{120} operations, semi-free-start collisions against 5.5 rounds in 2^{120} time and semi-free-start near-collisions against 7.5 rounds in 2^{128} time.

V.I SHA-512

SHA-512 uses compression function & Whirlpool uses a symmetric key block cipher that is based on AES to form the message digest. Maximum message size of SHA-512 is 2^{128-1} bits, Block size is 1024 bits, maximum message digest 512bits, Number of rounds 80 & word size is 64 bits. SHA-512 and Whirlpool hash functions are both 64-bit hash functions. That's why they are expected to be well suited for 64-bit processors. Whirlpool runs faster in Pentium III and Pentium 4 Prescott in single message hashing, due to a better instruction scheduling. But the effect of double message hashing is evident, the SHA-512 becomes more than 30 % faster than Whirlpool.SHA-2 support functions hashsize(), rounds(), add(LIST), addfile(HANDLE),
reset(), hexdigest(), base64digest() & operations are ADD,AND,OR,XOR,ROTATE,MOD & SHIFT.

V.II WHIRLPOOL

In case of Whirlpool 2^n executions for n-bit value. operates on messages less than 2^{256} bits in length & produces message digest of 512bits.actual size of the message digest depend upon the algorithm used. so message digest of the whirlpool consists of size 512bit,collision resistant, one way hash function & methods consists of clone,reset,base64 digest. Block Size 512 bits, Cipher key size 512bits, Number of Rounds 10, Key expansion using cipher itself with round constants as round keys. Substitution using SubByte transformations, Permutation using ShiftColumn transformations, Mixing using mix row transformations.

VI Purpose of using hash function

Using message digest data integrity is achieved is very easy and fast to check some data for validity. The algorithms are well known and implemented in most major programming languages, so they can be used in almost all environments. Whirlpool is much more scalable than most modern hashing functions. Even though is not specifically oriented toward any platform, it is rather efficient on many of them, its structure favouring extensively parallel execution of the component mappings. At the same time, it does not require excessive storage space (either for code or for tables), and can therefore be efficiently implemented in quite constrained environments like smart cards, although it can benefit from larger cache memory available on modern processors to achieve higher performance. It does not use expensive or unusual instructions that must be built in the processor. The mathematical simplicity of the primitive resulting from the design strategy tends to make analysis easier. And finally, it has a very long hash length, this not only provides increased protection against birthday attacks, but also offers a larger internal state for entropy containment, as is needed for certain classes of pseudo-random number generators.

VII Conclusion

The Whirlpool & Secure Hash Algorithm both uses message digest to provide security to data or information so from the above description we conclude that whirlpool provide more security as compared to secure hash algorithm-512 since whirlpool is based on Advanced encryption standard & SHA-512 is based on SHA family. there is no weakness in the message digest of whirlpool. Whirlpool is collision free & its performance is better as compared to secure hash algorithm & does hashing well. The secure hash algorithm is not collision free. The problem with hashing algorithms is that they quickly become outdated. To crack an algorithm it requires brute force trying all the different combinations. No attacks have been reported on earlier versions of whirlpool, but new versions are currently being produced which are able to perform better and these are likely to be more secure.

REFERENCES

[1] Barreto, P. and V. Rijmen. 2003. The Whirlpool Hashing Function. Submitted to NESSIE, May.

[2] Barreto, Paulo S.L.M. & Rijmen, Vincent: The Whirlpool Hashing Function. Available: http://paginas.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html.

[3] Black, J., P. Rogaway, and T. Shrimpton. 2002. Black-Box Analysis of the Block-CipherBased Hash Function Constructions from PGV, Proceedings, Advances in Cryptology CRYPTO 002, New York: Springer-Verlag.

[4] Damgard, I. 1989. A Design Principle for Hash Functions, Proceedings, CRYPTO 89,New York: pringer-Verlag.

[5] Paulo Barreto, Vincent Rijmen, *The Whirlpool Hashing Function page*. http://planeta.terra.com.br/informatica/paulobarreto/WhirlpoolPage.html

[6] Stallings, William, Cryptography and Network Security, Prentice Hall, 1999.