Report on SHA-512 algortihm

Submitted To

Dr. Md. Shariful Islam

Professor

Institute of Information Technology

University of Dhaka

Submitted by

Amran Hossain

BSSE 0917

Date:31-01-2021

Institute of Information Technology

University of Dhaka

Table of Contents

SHA-512

Introduction:

SHA-512 Cryptographic Hash Algorithm. A cryptographic hash (sometimes called ‘digest’) is a kind of ‘signature’ for a text or a data file. SHA-512 generates an almost-unique 512-bit(32-byte) signature for a text. The algorithm takes as input a message with a maximum length of less than bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks.

Working Principle:

SHA-512 does its work in a few stages. These stages are below:

1.Input formatting

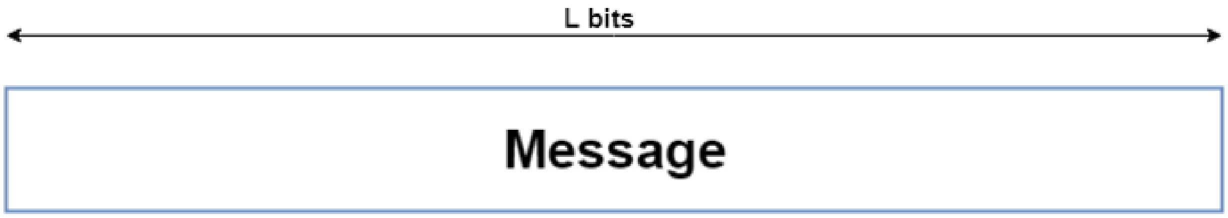
2.Hash buffer initialization

3.Message Processing

4.Output

Input formatting

SHA-512 can’t actually hash a message input of any size, i.e., it has an input size limit. This limit is imposed by its very structure as you may see further on. The entire formatted message has basically three parts: the original message, padding bits, size of original message. And this should all have a combined size of a whole multiple of 1024 bits. This is because the formatted message will be processed as blocks of 1024 bits each, so each bock should have 1024 bits to work with.



Original message

Padding bits:

The input message is taken and some padding bits are appended to it in order to get it to the desired length. The bits that are used for padding are simply ‘0’ bits with a leading ‘1’ (100000…000). Also, according to the algorithm, padding needs to be done, even if it is by one bit. So, a single padding bit would only be a ‘1’. The total size should be equal to 128 bits short of a multiple of 1024 since the goal is to have the formatted message size as a multiple of 1024 bits (N x 1024).

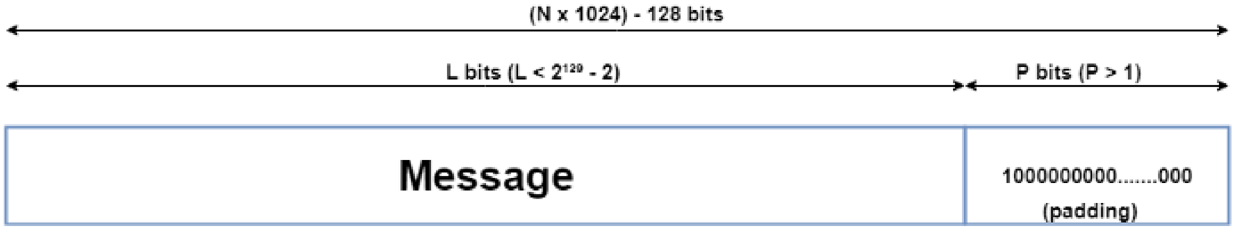


Figure: Msg with padding

Padding size:

After this, the s After this, the size of the original message given to the algorithm is appended. This size value needs to be represented in 128 bits and is the only reason that the SHA-512 has a limitation for its input message of the original message given to the algorithm is appended. This size value needs to be represented in 128 bits and is the only reason that the SHA-512 has a limitation for its input message.

Since the size of the original message needs to be represented in 128 bits and the largest number that can be represented using 128 bits is (2^{128}-1), the message size can be at most (2^{128}-1) bits; and also taking into consideration the necessary single padding bit, the maximum size for the original message would then be (2^{128}-2) Since the size of the original message needs to be represented in 128 bits and the largest number that can be represented using 128 bits is (2^{128}-1), the message size can be at most (2^{128}-1) bits; and also taking into consideration the necessary single padding bit, the maximum size for the original message would then be (2^{128}-2). Even though this limit exists, it doesn’t actually cause a problem since the actual limit is so high (2^{128}-2 = 340,282,366,920,938,463,463,374,607,431,768,211,454 bits).. Even though this limit exists, it doesn’t actually cause a problem since the actual limit is so high (2^{128}-2 = 340,282,366,920,938,463,463,374,607,431,768,211,454 bits).

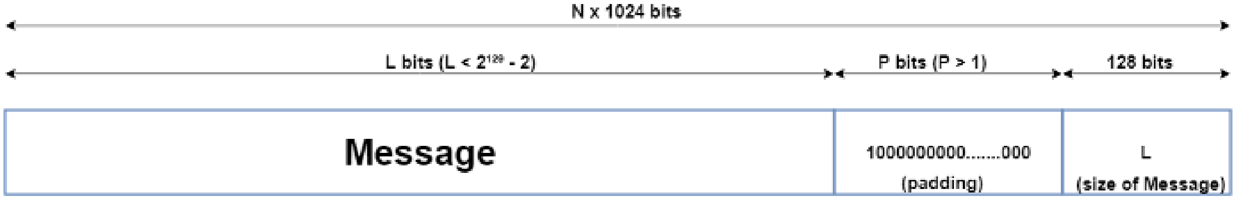


Figure: Msg with padding and size

Now that the padding bits and the size of the message have been appended, we are left with the completely formatted input for the SHA-512 algorithm.

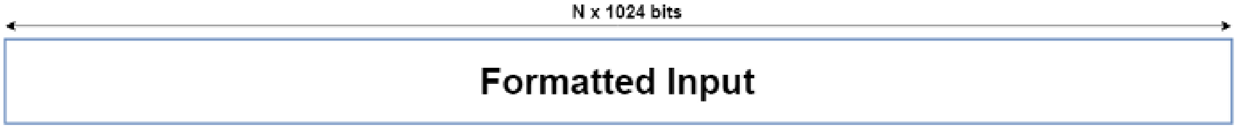


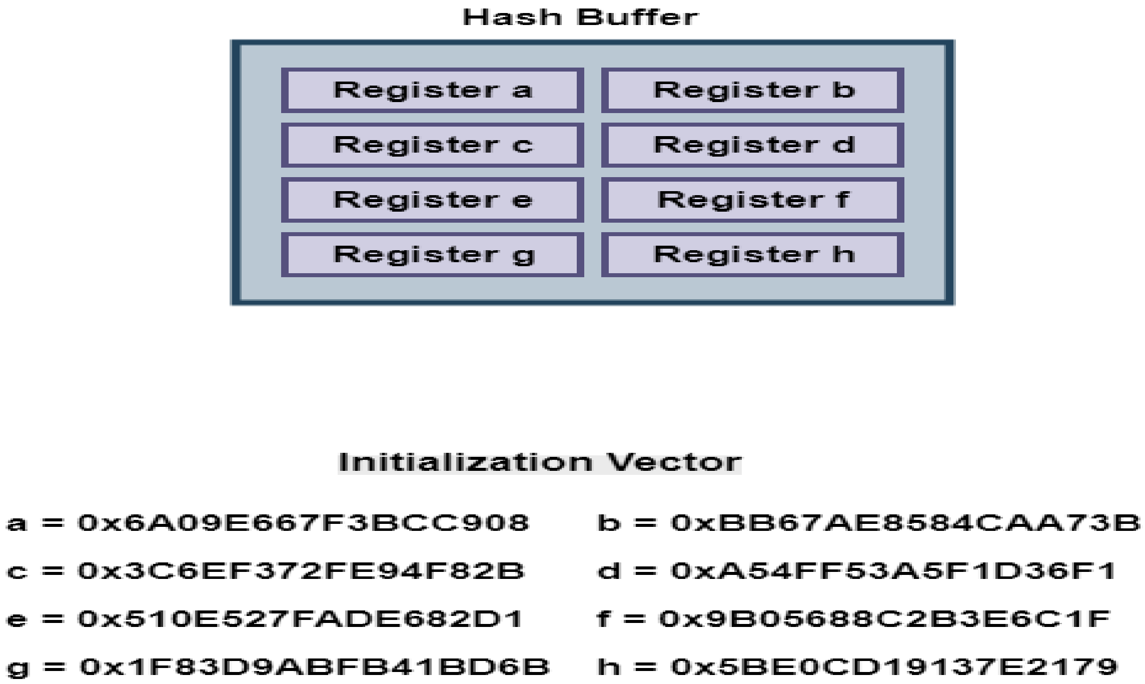
Figure: Format input

Hash buffer initialization:

The algorithm works in a way where it processes each block of 1024 bits from the message using the result from the previous block. Now, this poses a problem for the first 1024-bit block which can’t use the result from any previous processing. This problem can be solved by using a default value to be used for the first block in order to start off the process. (Have a look at the second-last diagram). Since each intermediate result needs to be used in processing the next block, it needs to be stored somewhere for later use. This would be done by the hash buffer; this would also then hold the final hash digest of the entire processing phase of SHA-512 as the last of these ‘intermediate’ results.

So, the default values used for starting off the chain processing of each 1024-bit block are also stored into the hash buffer at the start of processing. The actual value used is of little consequence, but for those interested, the values used are obtained by taking the first 64 bits of the fractional parts of the square roots of the first 8 prime numbers (2,3,5,7,11,13,17,19).

These values are called the Initial Vectors (IV). Why 8 prime numbers instead of 9? Because the hash buffer actually consists of 8 subparts (registers) for storing them.



Initialization Vector

a = 0x6A09E667F3BCC908   b = 0xBB67AE8584CAA73B

c = 0x3C6EF372FE94F82B   d = 0xA54FF53A5F1D36F1

e = 0x510E527FADE682D1   f = 0x9B05688C2B3E6C1F

g = 0x1F83D9ABFB41BD6B   h = 0x5BE0CD19137E2179

Figure: Hash buffer

Message processing:

Message processing is done upon the formatted input by taking one block of 1024 bits at a time. The actual processing takes place by using two things: The 1024-bit block, and the result from the previous processing. This part of the SHA-512 algorithm consists of several 'Rounds' and an addition operation.

So, the Message block (1024 bit) is expanded out into 'Words' using a 'message sequencer'. Eighty Words to be precise, each of them having a size of 64 bits.

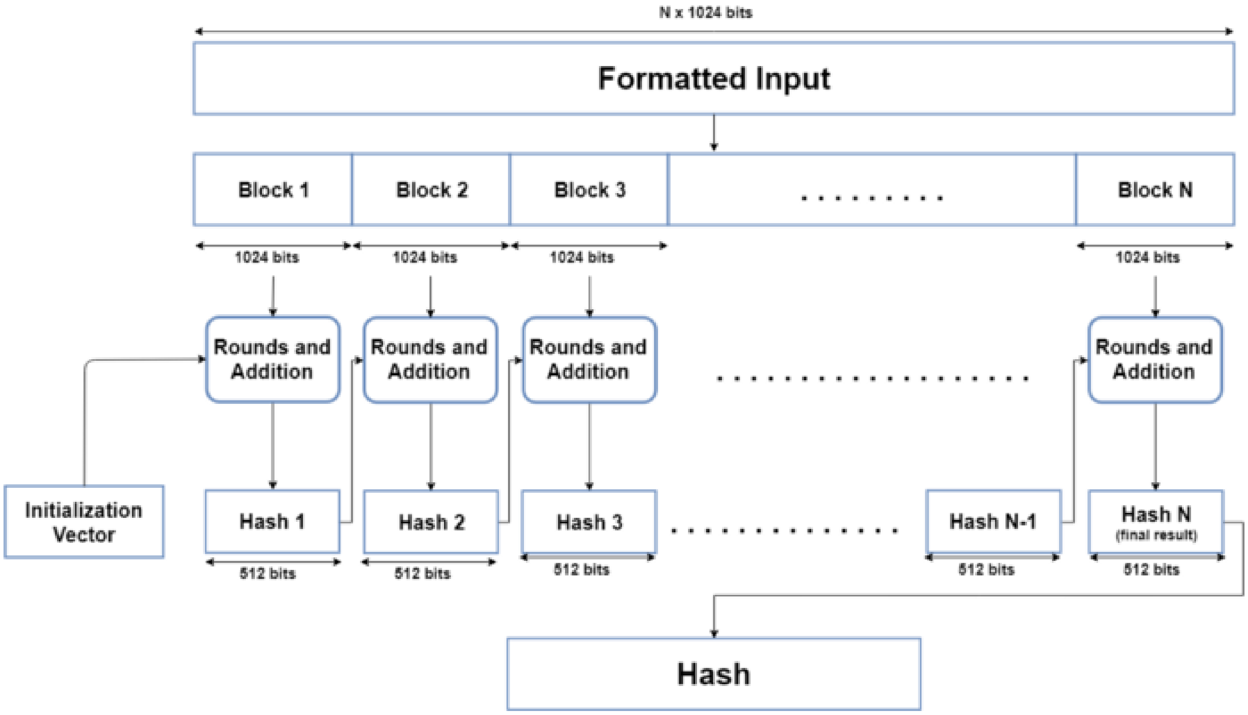


Figure   formatted input 1024 bit-block

Rounds:

The main part of the message processing phase may be considered to be the Rounds. Each round takes 3 things: one Word, the output of the previous Round, and a SHA-512 constant. The first Round doesn't have a previous Round whose output it can use, so it uses the final output from the previous message processing phase for the previous block of 1024 bits. For the first Round of the first block (1024 bits) of the formatted input, the Initial Vector (IV) is used.

SHA-512 constants are predetermined values, each of whom is used for each Round in the message processing phase. Again, these aren't very important, but for those interested, they are the first 64 bits from the fractional part of the cube roots of the first 80 prime numbers. Why 80? Because there are 80 Rounds and each of them needs one of these constants.

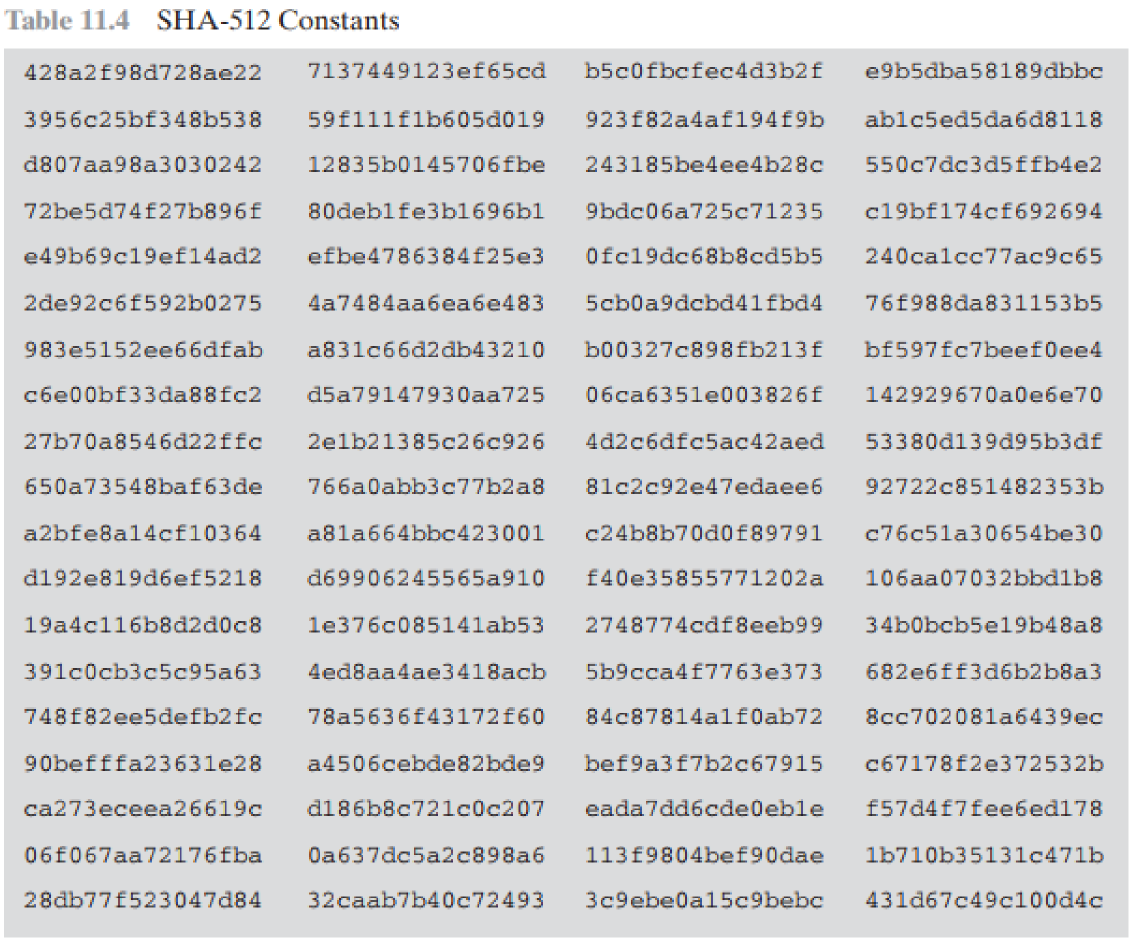


Table 11.4 SHA-512 Constants

428a2f98d728ae22	7137449123ef65cd	b5c0fbcfec4d3b2f	e9b5dba58189dbbc
3956c25bf348b538	59f111f1b605d019	923f82a4af194f9b	ab1c5ed5da6d8118
d807aa98a3030242	12835b0145706fbe	243185be4ee4b28c	550c7dc3d5ffb4e2
72be5d74f27b896f	80deb1fe3b1696b1	9bdc06a725c71235	c19bf174cf692694
e49b69c19ef14ad2	efbe4786384f25e3	0fc19dc68b8cd5b5	240ca1cc77ac9c65
2de92c6f592b0275	4a7484aa6ea6e483	5cb0a9dcbd41fbd4	76f988da831153b5
983e5152ee66dfab	a831c66d2db43210	b00327c898fb213f	bf597fc7beef0ee4
c6e00bf33da88fc2	d5a79147930aa725	06ca6351e003826f	142929670a0e6e70
27b70a8546d22ffc	2e1b21385c26c926	4d2c6dfc5ac42aed	53380d139d95b3df
650a73548baf63de	766a0abb3c77b2a8	81c2c92e47edaee6	92722c851482353b
a2bfe8a14cf10364	a81a664bbc423001	c24b8b70d0f89791	c76c51a30654be30
d192e819d6ef5218	d69906245565a910	f40e35855771202a	106aa07032bbd1b8
19a4c116b8d2d0c8	1e376c085141ab53	2748774cdf8eeb99	34b0bcb5e19b48a8
391c0cb3c5c95a63	4ed8aa4ae3418acb	5b9cca4f7763e373	682e6ff3d6b2b8a3
748f82ee5defb2fc	78a5636f43172f60	84c87814a1f0ab72	8cc702081a6439ec
90befffa23631e28	a4506cebde82bde9	bef9a3f7b2c67915	c67178f2e372532b
ca273eceea26619c	d186b8c721c0c207	eada7dd6cde0eb1e	f57d4f7fee6ed178
06f067aa72176fba	0a637dc5a2c898a6	113f9804bef90dae	1b710b35131c471b
28db77f523047d84	32caab7b40c72493	3c9ebe0a15c9bebc	431d67c49c100d4c

Figure: SHA-512 constants

Once the Round function takes these 3 things, it processes them and gives an output of 512 bits. This is repeated for 80 Rounds. After the 80th Round, its output is simply added to the result of the previous message processing phase to get the final result for this iteration of message processing.

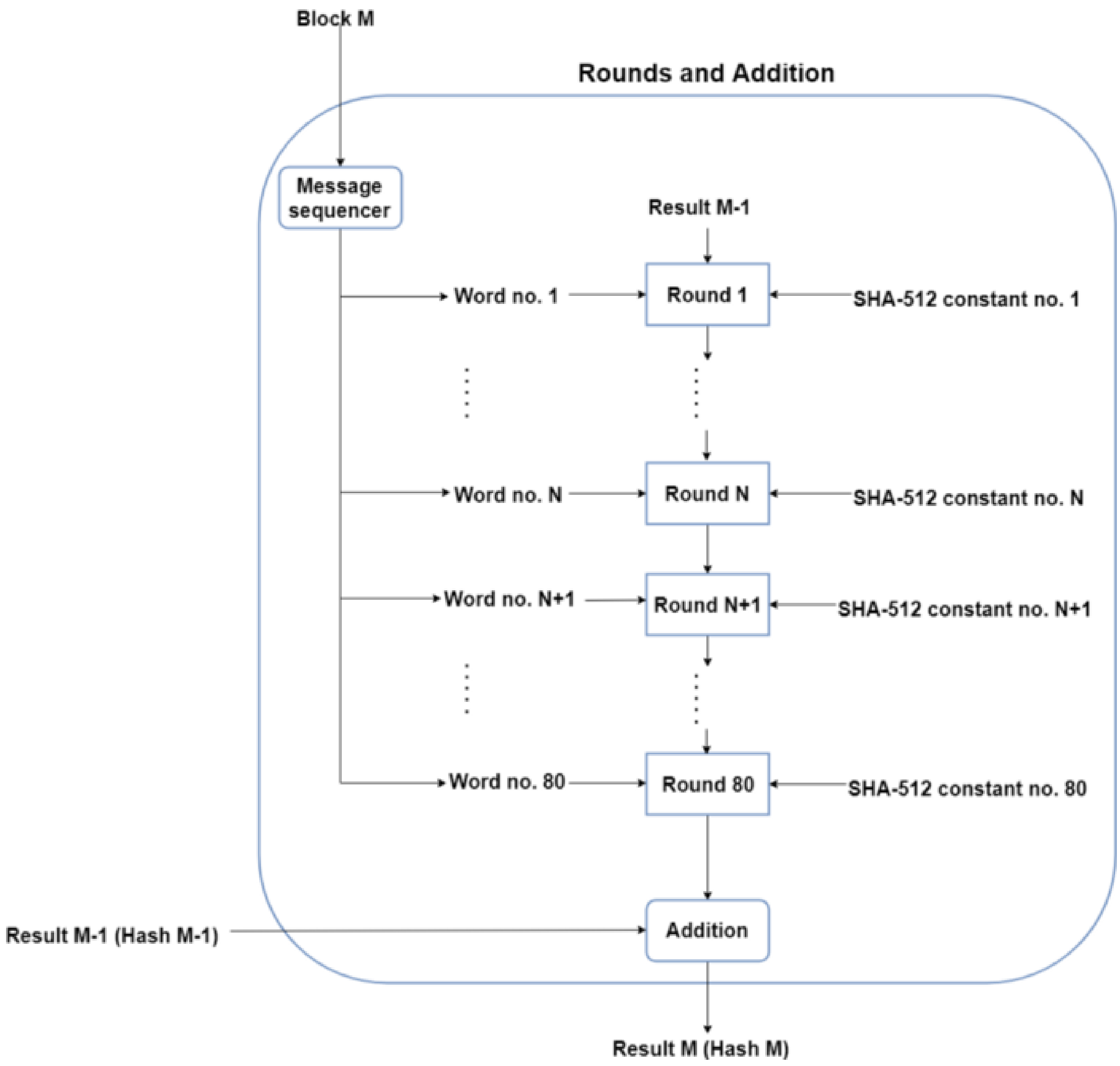


Figure : rounds

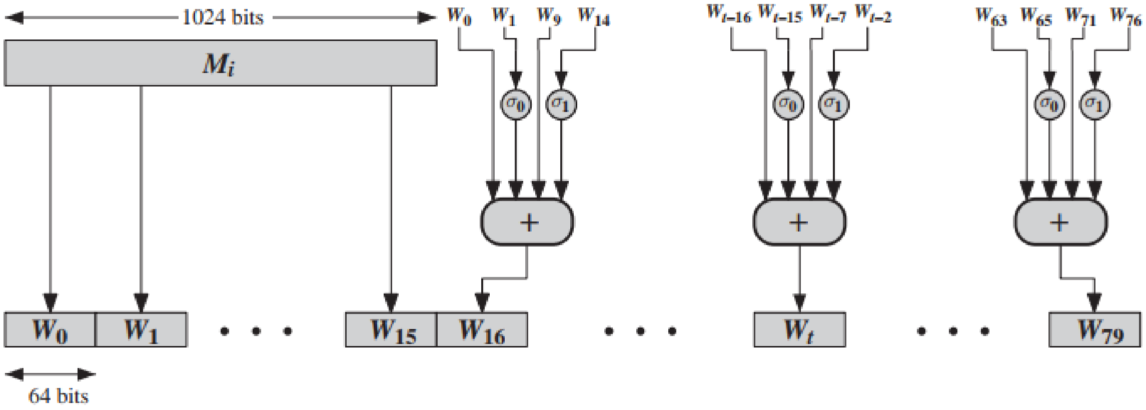


Figure 11.11 Creation of 80-word Input Sequence for SHA-512 Processing of Single Block

Outputs:

After every block of 1024 bit goes through the message processing phase, i.e., the last iteration of the phase, we get the final 512-bit Hash value of our original message. So, the intermediate results are all used from each block for processing the next block. And when the final 1024-bit block has finished being processed, we have with us the final result of the SHA-512 algorithm for our original message. Thus, we obtain the final hash value from our original message. The SHA-512 is part of a group of hashing algorithms that are very similar in how they work, called SHA-2. Algorithms such as SHA-256 and SHA-384 are a part of this group alongside SHA-512. SHA-256 is also used in the Bitcoin blockchain as the designated hash function. That's a brief overview of how the SHA-512 hashing algorithm works. I intend to go into further detail about what makes the hash functions practically irreversible (one-way) and how this is helpful for digital security.

Application

- Used as part of a system to authenticate archival video from the International Criminal Tribunal of the Rwandan genocide.
- Proposed for use in DNSSEC
- Are moving to 512-bit SHA-2 for secure password by hashing Unix and Linux vendors.