Course: Cryptography and Network Security
Code: CS-34310
Branch: M.C.A - 4th Semester

Lecture – 12 : Message Integrity and Message Authentication

Faculty & Coordinator : Dr. J Sathish Kumar (JSK)

Department of Computer Science and Engineering

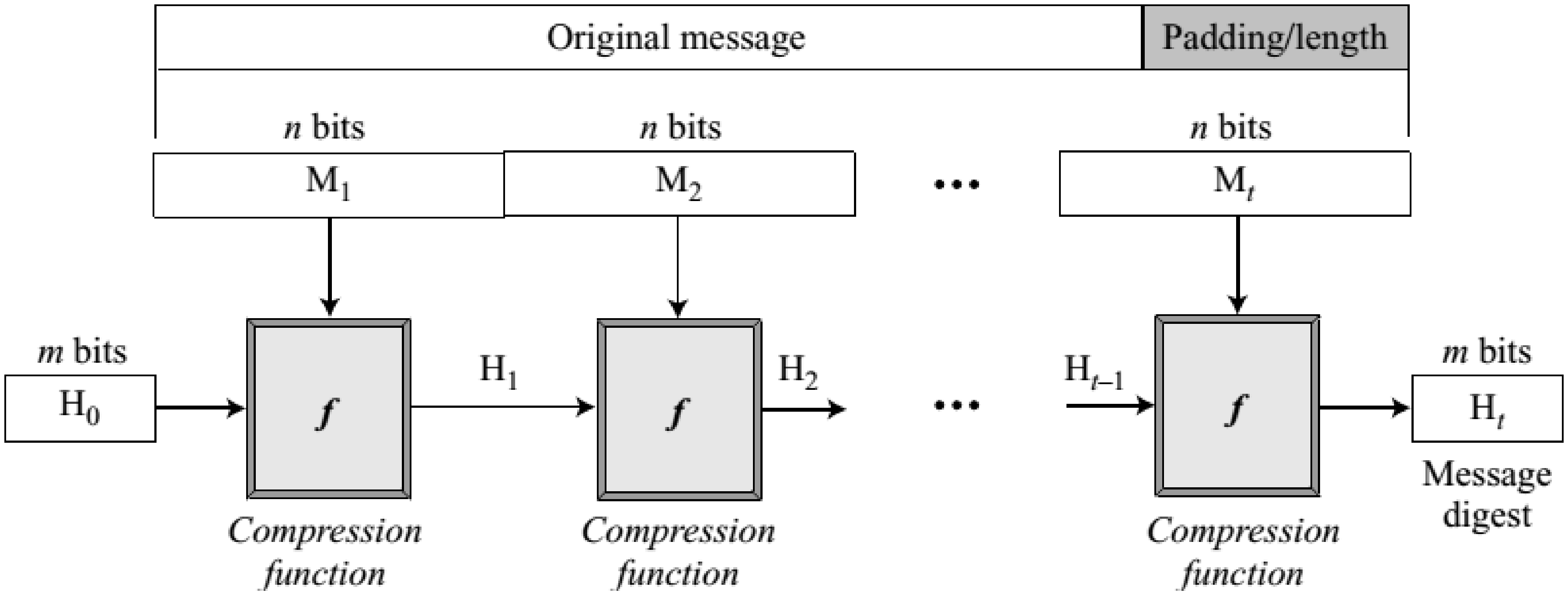
Motilal Nehru National Institute of Technology Allahabad,
Prayagraj-211004

Cryptographic Hash Functions

- All cryptographic hash functions need to create a fixed-size digest out of a variable-size message.
- Creating such a function is best accomplished using iteration.
- Instead of using a hash function with variable-size input, a function with fixed-size input is created and is used a necessary number of times.
- The fixed-size input function is referred to as a compression function.
- It compresses an n-bit string to create an m-bit string where n is normally greater than m.
- The scheme is referred to as an iterated cryptographic hash function.

Merkle-Damgard Scheme

- The Merkle-Damgard scheme is an iterated hash function that is collision resistant if the compression function is collision resistant.



Merkle-Damgard Scheme

1. The message length and padding are appended to the message to create an augmented message that can be evenly divided into blocks of n bits, where n is the size of the block to be processed by the compression function.
2. The message is then considered as t blocks, each of n bits. We call each block M_1, $M_2, \ldots, M_t$. We call the digest created at t iterations $H_1, H_2, \ldots, H_t$.
3. Before starting the iteration, the digest H_0 is set to a fixed value, normally called IV (initial value or initial vector).
4. The compression function at each iteration operates on H_{i-1} and M_i to create a new H_i. In other words, we have $H_i = f(H_{i-1}, M_i)$, where f is the compression function.
5. H_t is the cryptographic hash function of the original message, that is, h(M).

Two Groups of Compression Functions

- In the first approach, the compression function is made from scratch:
  - Message Digest (MD): Several versions (MD2, MD4, and MD5)
  - The last version, MD5, is a strengthened version of MD4 that divides the message into blocks of 512 bits and creates a 128-bit digest.
  - It turned out that a message digest of size 128 bits is too small to resist collision attack.

- In the second approach, a symmetric-key block cipher serves as a compression function.
  - Secure Hash Algorithm (SHA)
  - Also referred to as Secure Hash Standard (SHS) four new versions: SHA-224, SHA-256, SHA-384, and SHA-512.

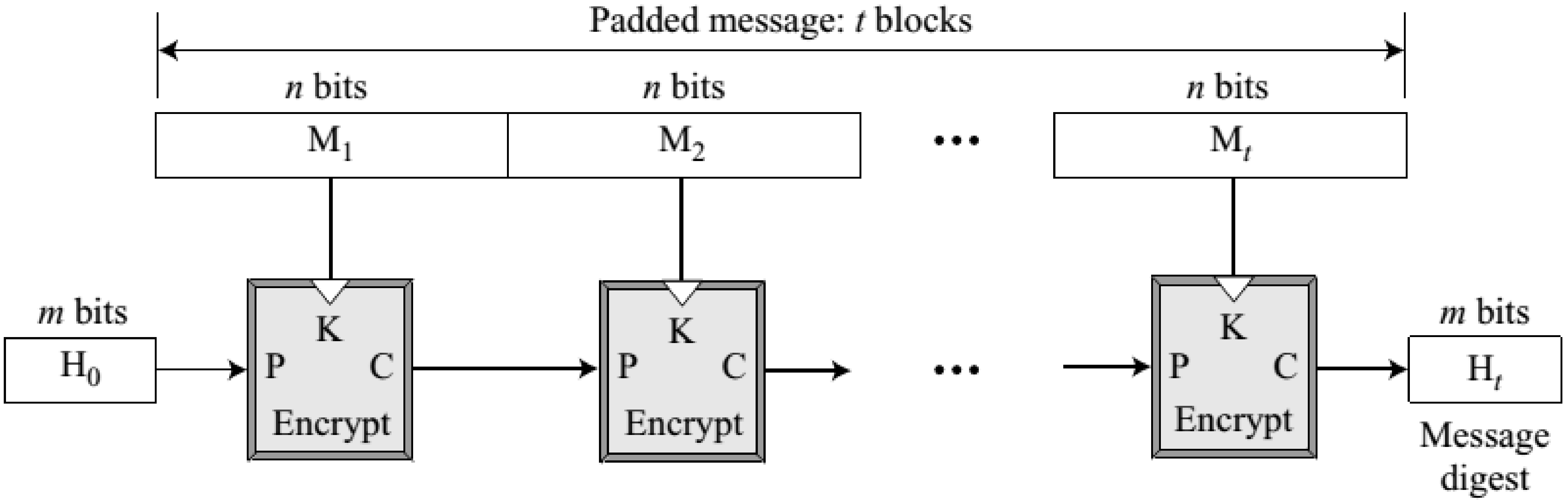
Characteristics of Secure Hash Algorithms (SHAs)

Characteristics	*SHA-1*	*SHA-224*	*SHA-256*	*SHA-384*	*SHA-512*
Maximum Message size	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$	$2^{128} - 1$	$2^{128} - 1$
Block size	512	512	512	1024	1024
Message digest size	160	224	256	384	512
Number of rounds	80	64	64	80	80
Word size	32	32	32	64	64

Rabin Scheme

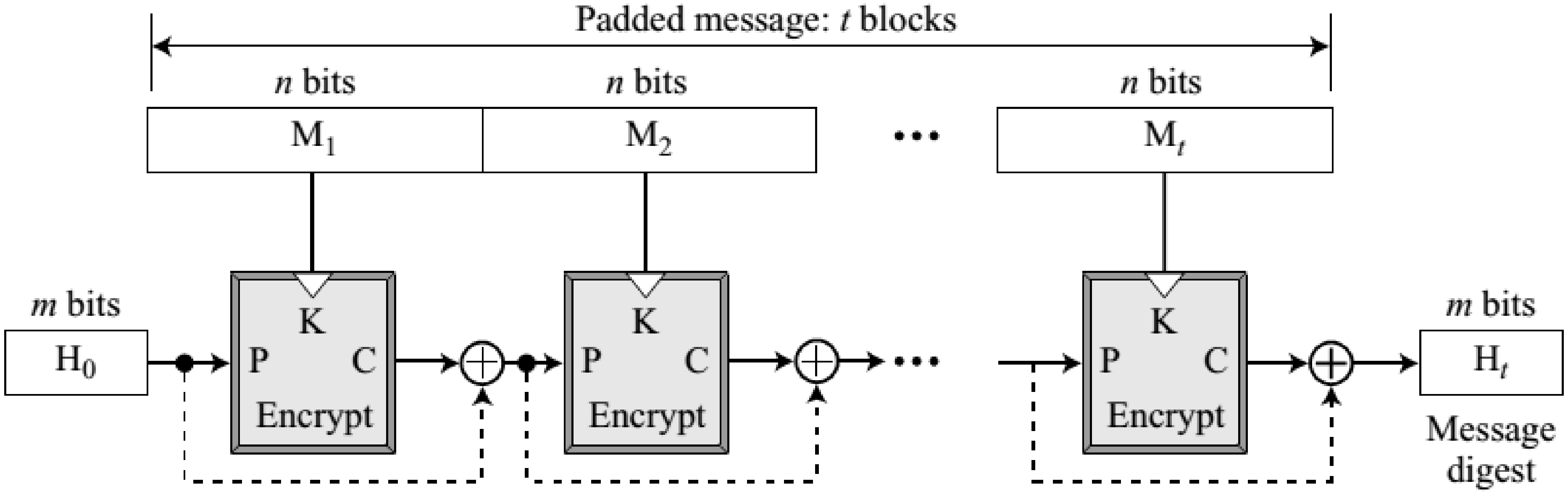
- Rabin scheme is based on the Merkle-Damgard scheme.
- The compression function is replaced by any encrypting cipher.
- The message block is used as the key; the previously created digest is used as the plaintext.
- The ciphertext is the new message digest.
- Note that the size of the digest is the size of data block cipher in the underlying cryptosystem.
- For example, if DES is used as the block cipher, the size of the digest is only 64 bits.
- Although the scheme is very simple, it is subject to a meet-in-the-middle attack

Rabin Scheme



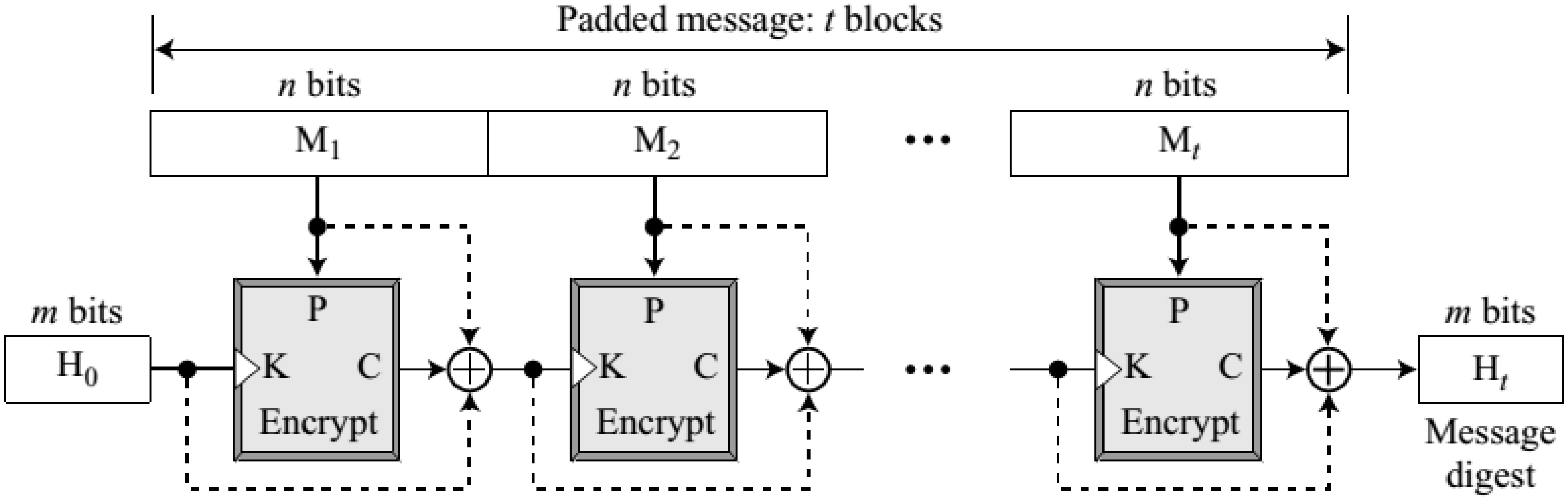
Davies-Meyer Scheme

- The Davies-Meyer scheme is basically the same as the Rabin scheme except that it uses forward feed to protect against meet-in-the-middle attack.



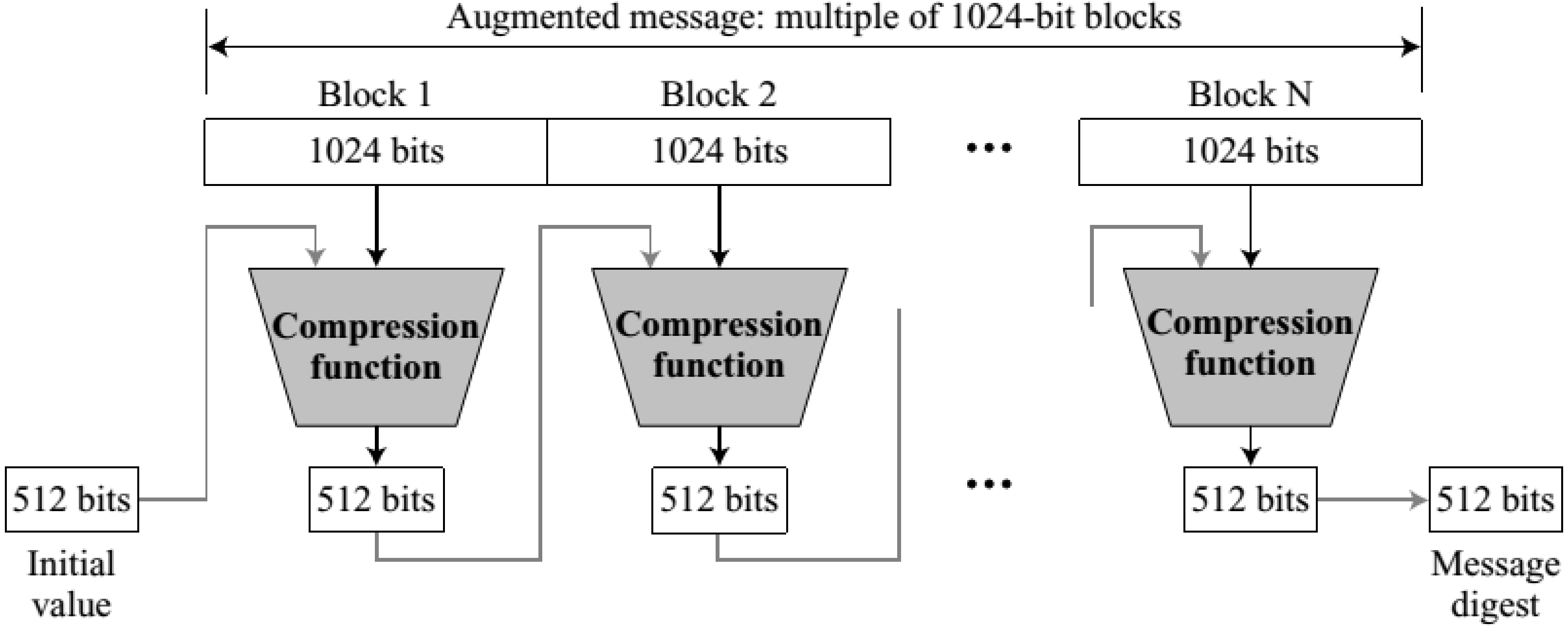
Miyaguchi-Preneel Scheme

- To make the algorithm stronger against attack, the plaintext, the cipher key, and the ciphertext are all exclusive-ored together to create the new digest.



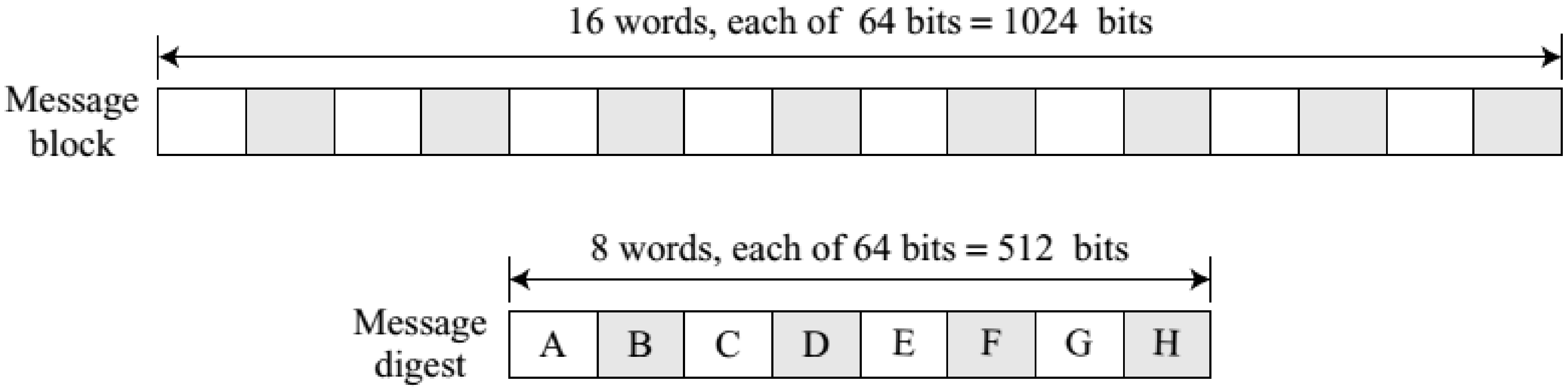
SHA-512

- SHA-512 creates a digest of 512 bits from a multiple-block message.
- Each block is 1024 bits in length.



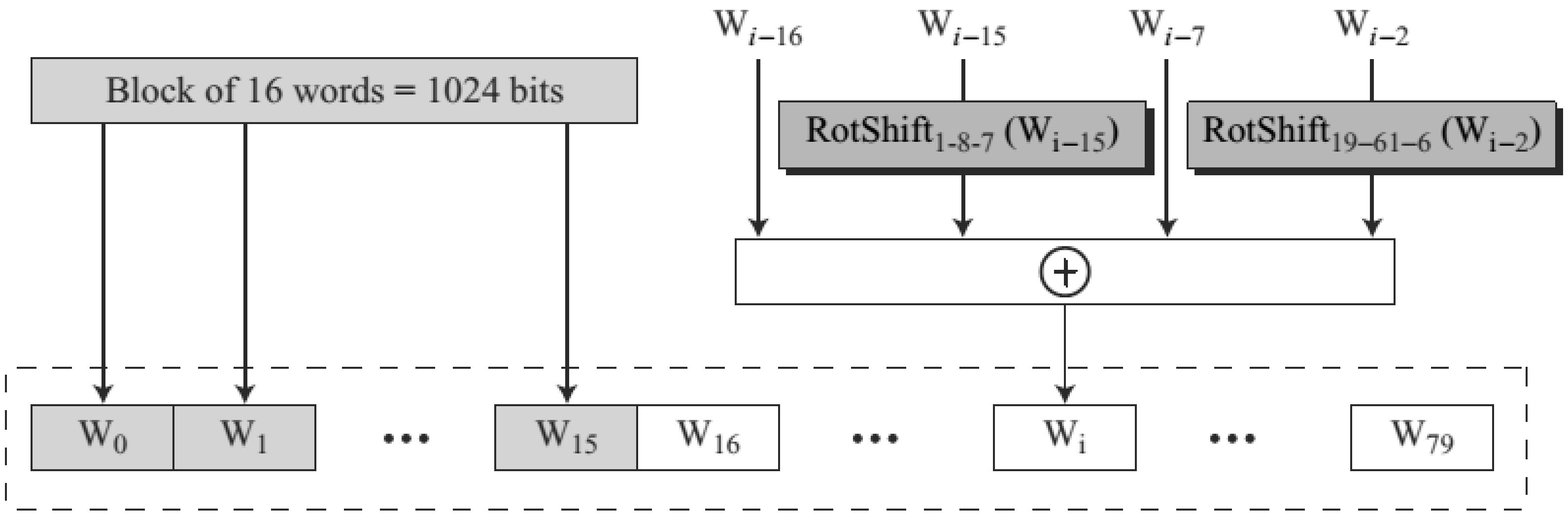
SHA-512

- SHA-512 operates on words; it is word oriented. A word is defined as 64 bits.
- The message digest is also made of 64-bit words, but the message digest is only eight words and the words are named A, B,C, D, E, F, G, and H.



SHA-512: Word Expansion

- A block is made of 1024 bits, or sixteen 64-bit words.
- 16-word block needs to be expanded to 80 words, from W0 to W79



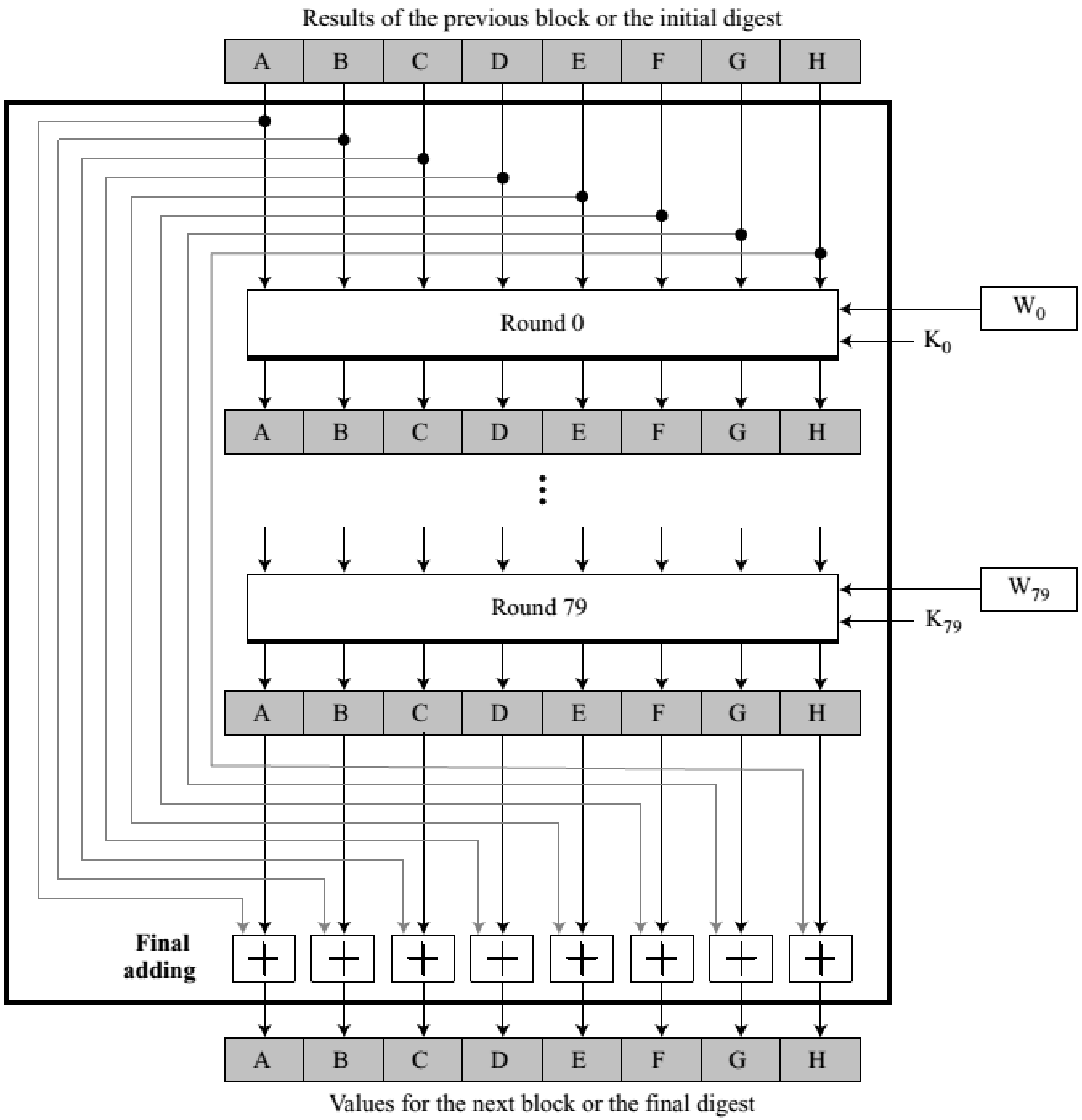
$\text{RotShift}_{\text{1-m-n}}\,(x)$: $\boldsymbol{RotR_l(x)} \oplus \boldsymbol{RotR_m(x)} \oplus \boldsymbol{ShL_n(x)}$

$\boldsymbol{RotR_i(x)}$: Right-rotation of the argument x by i bits
$\boldsymbol{ShL_i(x)}$: Shift-left of the argument x by i bits and padding the left by 0's.

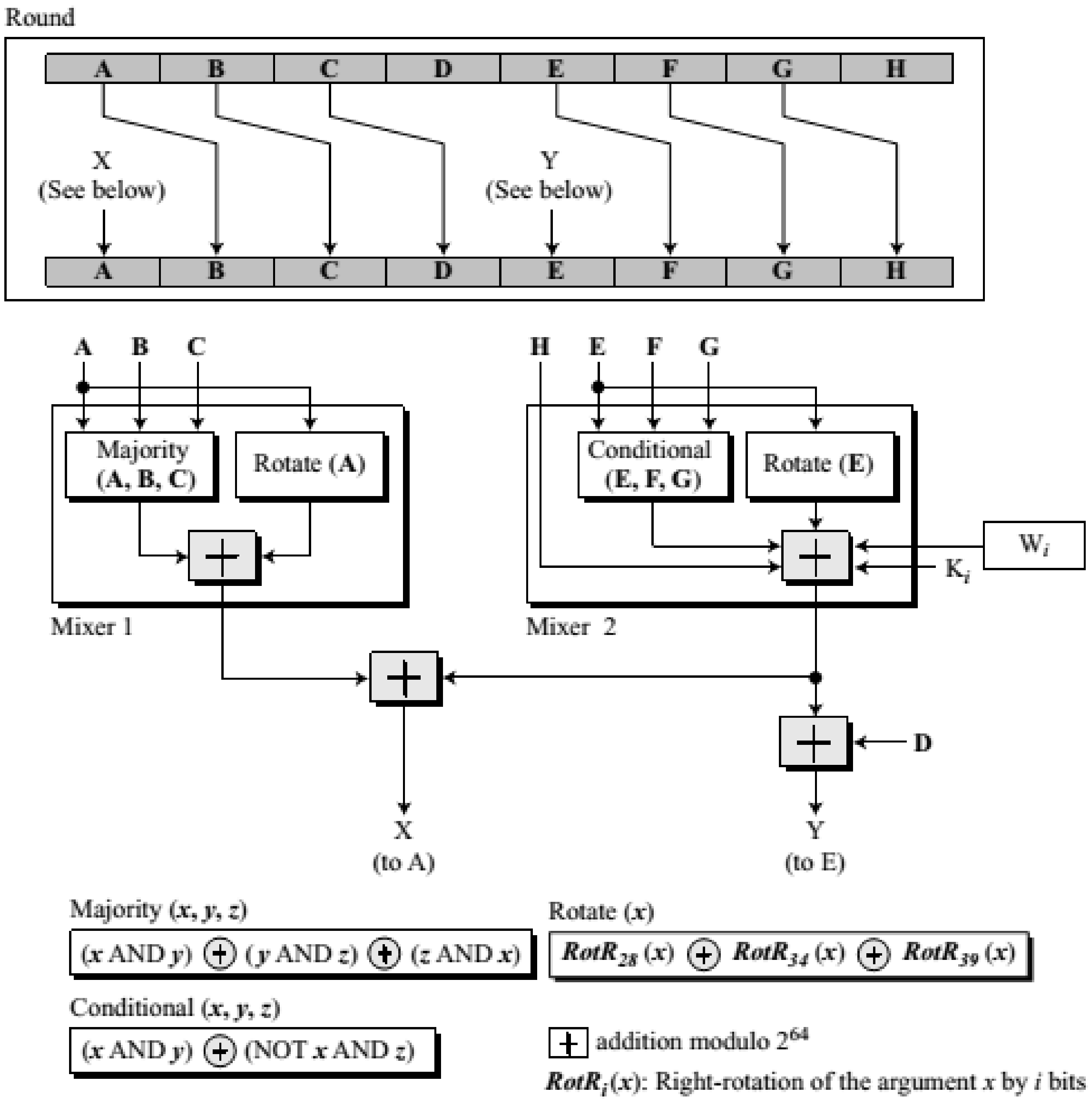
SHA-512: Compression Function

- SHA-512 creates a 512-bit (eight 64-bit words) message digest from a multiple-block message where each block is 1024 bits.
- The processing of each block of data in SHA-512 involves 80 rounds.
- In each round, the contents of eight previous buffers, one word from the expanded block (Wi), and one 64-bit constant (Ki) are mixed together and then operated on to create a new set of eight buffers.
- At the beginning of processing, the values of the eight buffers are saved into eight temporary variables.
- At the end of the processing (after step 79), these values are added to the values created from step 79. We call this last operation the final adding

SHA-512: Compression Function



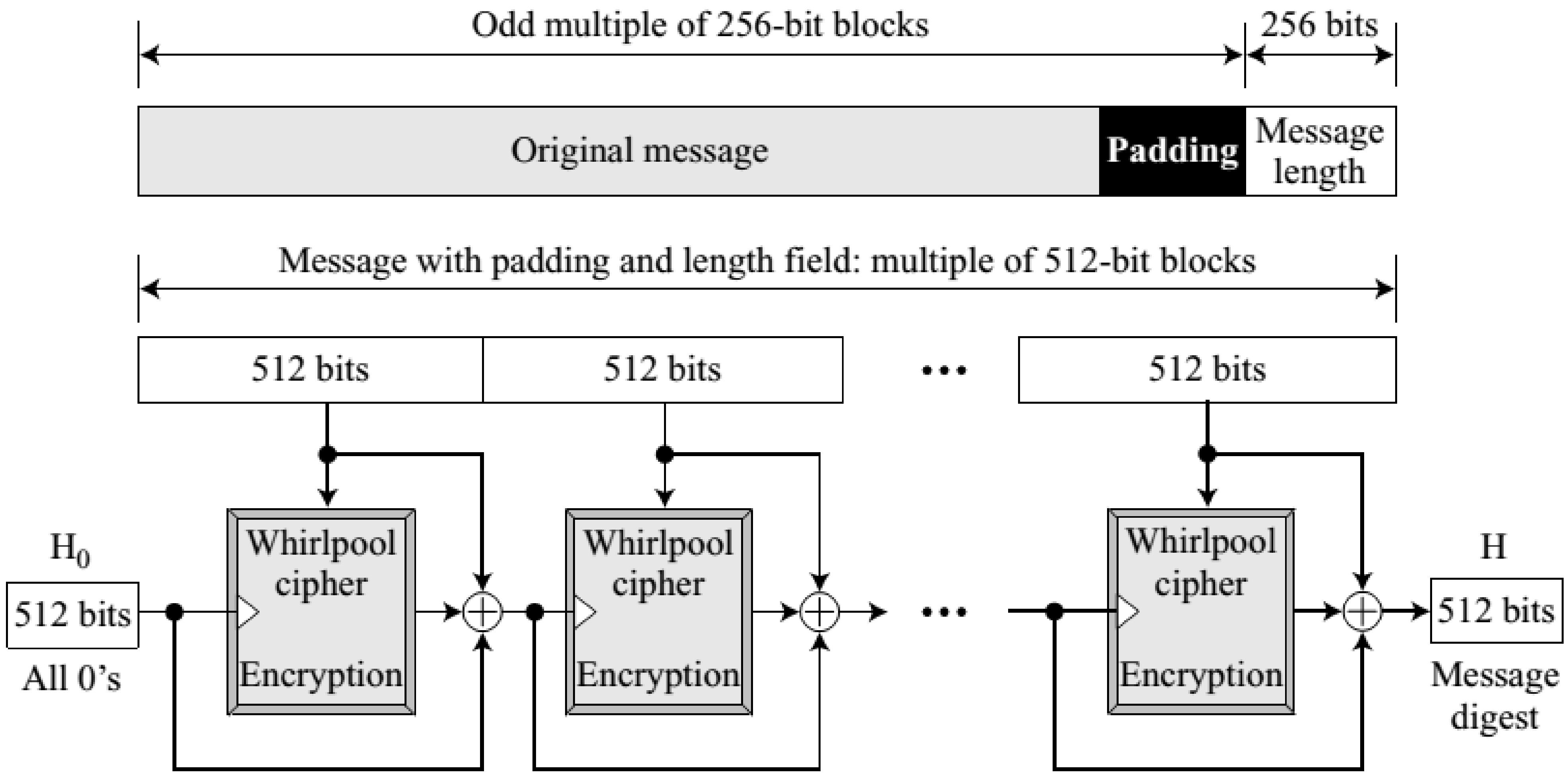
Structure of Each Round



WHIRLPOOL

- Whirlpool is the New European Schemes for Signatures, Integrity, and Encryption (NESSIE).
- Whirlpool is an iterated cryptographic hash function, based on the Miyaguchi-Preneel scheme, that uses a symmetric-key block cipher in place of the compression function.
- The block cipher is a modified AES cipher that has been tailored for this purpose.

WHIRLPOOL



Odd multiple of 256-bit blocks
256 bits
Original message
Padding
Message length
Message with padding and length field: multiple of 512-bit blocks
512 bits
512 bits
• • •
512 bits
H_0
512 bits
All 0's
Whirlpool cipher
Encryption
Whirlpool cipher
Encryption
• • •
Whirlpool cipher
Encryption
H
512 bits
Message digest