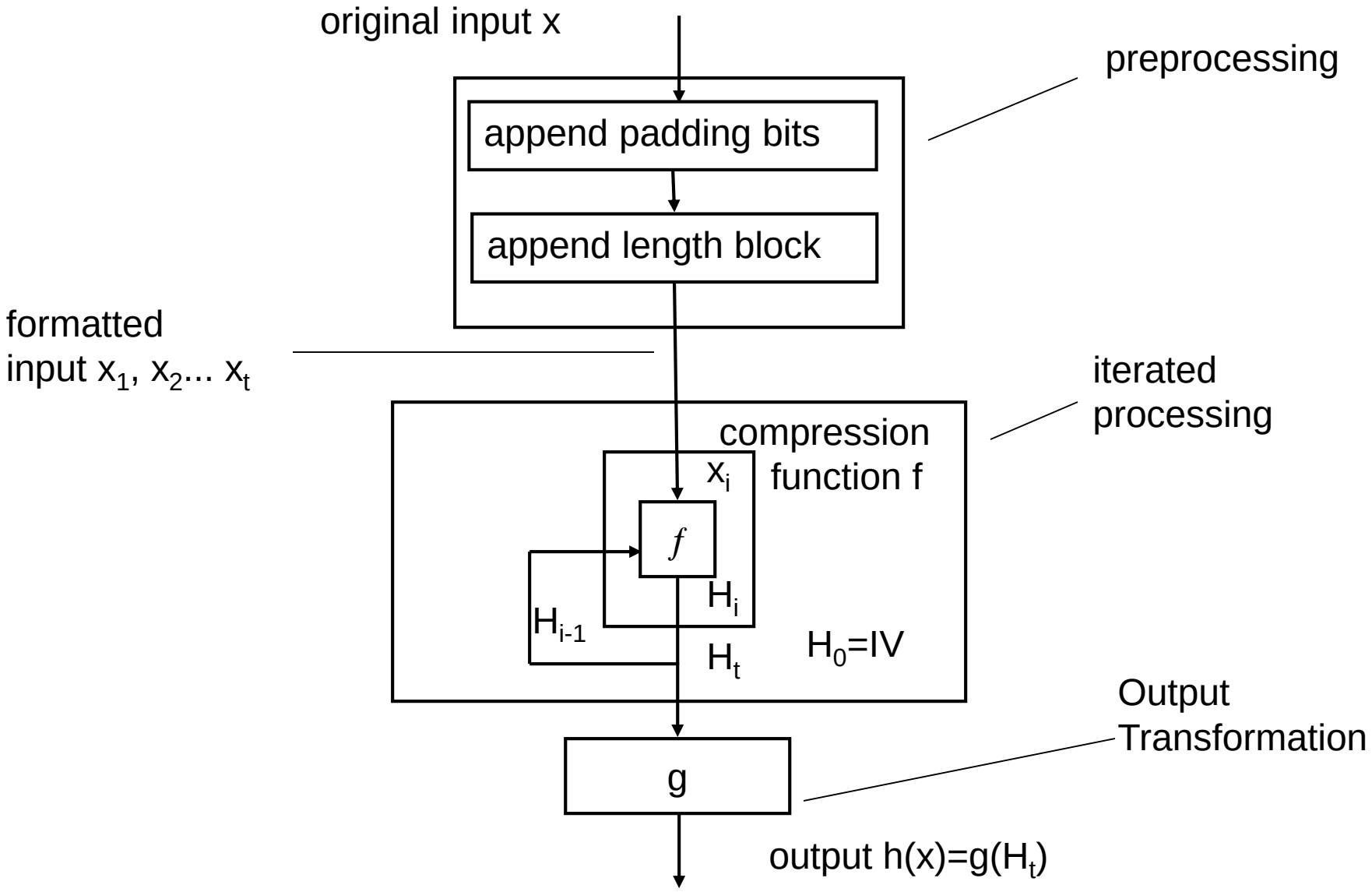
CS557: Cryptography

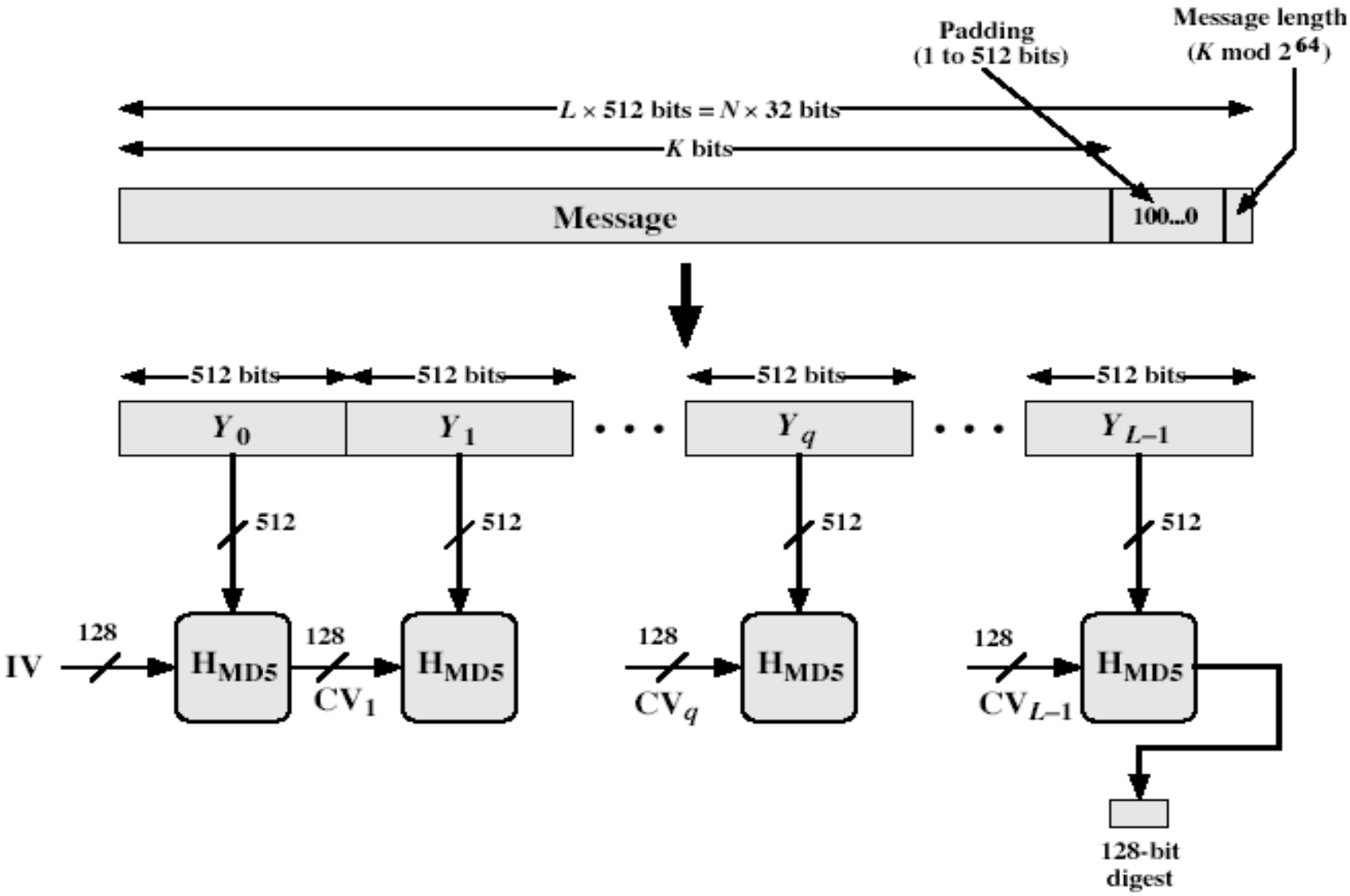
Cryptographic Hash Function III

S. Tripathy
IIT Patna

Detailed view of Hash Function



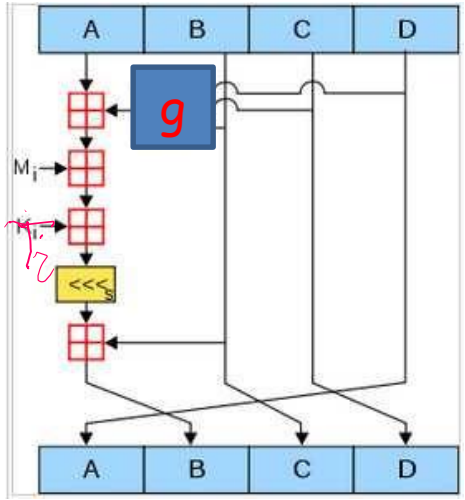
MD5 Overview

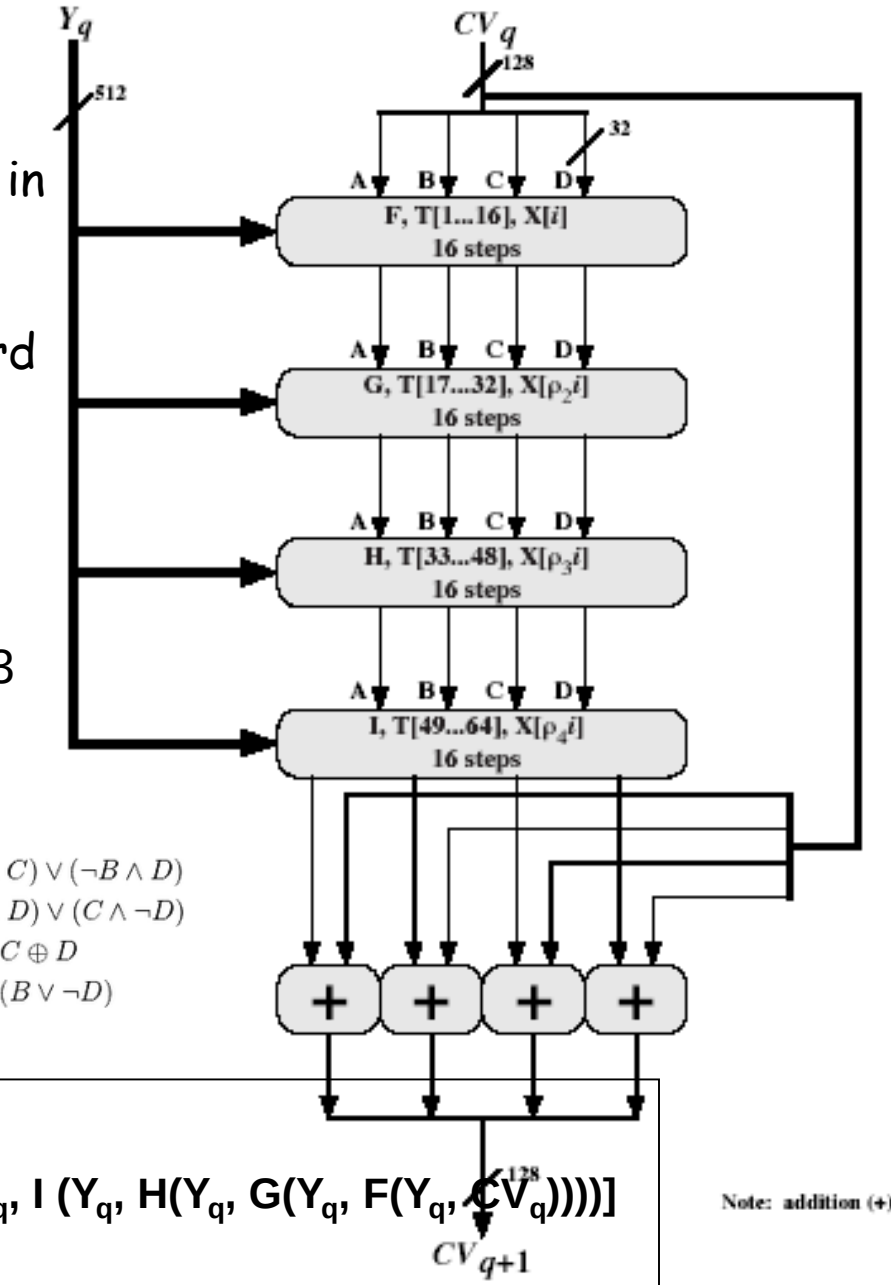


Truth Table for Logical function g

b	c	d	F	G	H	I
0	0	0	0	0	0	1
0	0	1	1	0	1	0
0	1	0	0	1	1	0
0	1	1	1	0	0	1
1	0	0	0	0	1	1
1	0	1	0	1	0	1
1	1	0	1	1	0	0
1	1	1	1	1	1	0

IV: h1 = 0x67452301, h2 = 0xefcdab89, h3 = 0x98badcfe, h4 = 0x10325476





$$F(B,C,D) = (B \wedge C) \vee (\neg B \wedge D)$$
$$G(B,C,D) = (B \wedge D) \vee (C \wedge \neg D)$$
$$H(B,C,D) = B \oplus C \oplus D$$
$$I(B,C,D) = C \oplus (B \vee \neg D)$$

$CV_0 = IV$

$CV_{q+1} = SUM_{32}[CV_q, I(Y_q, H(Y_q, G(Y_q, F(Y_q, CV_q)))))]$

$MD = CV_{L-1}$

Note: addition (+) is mod 2^{32}

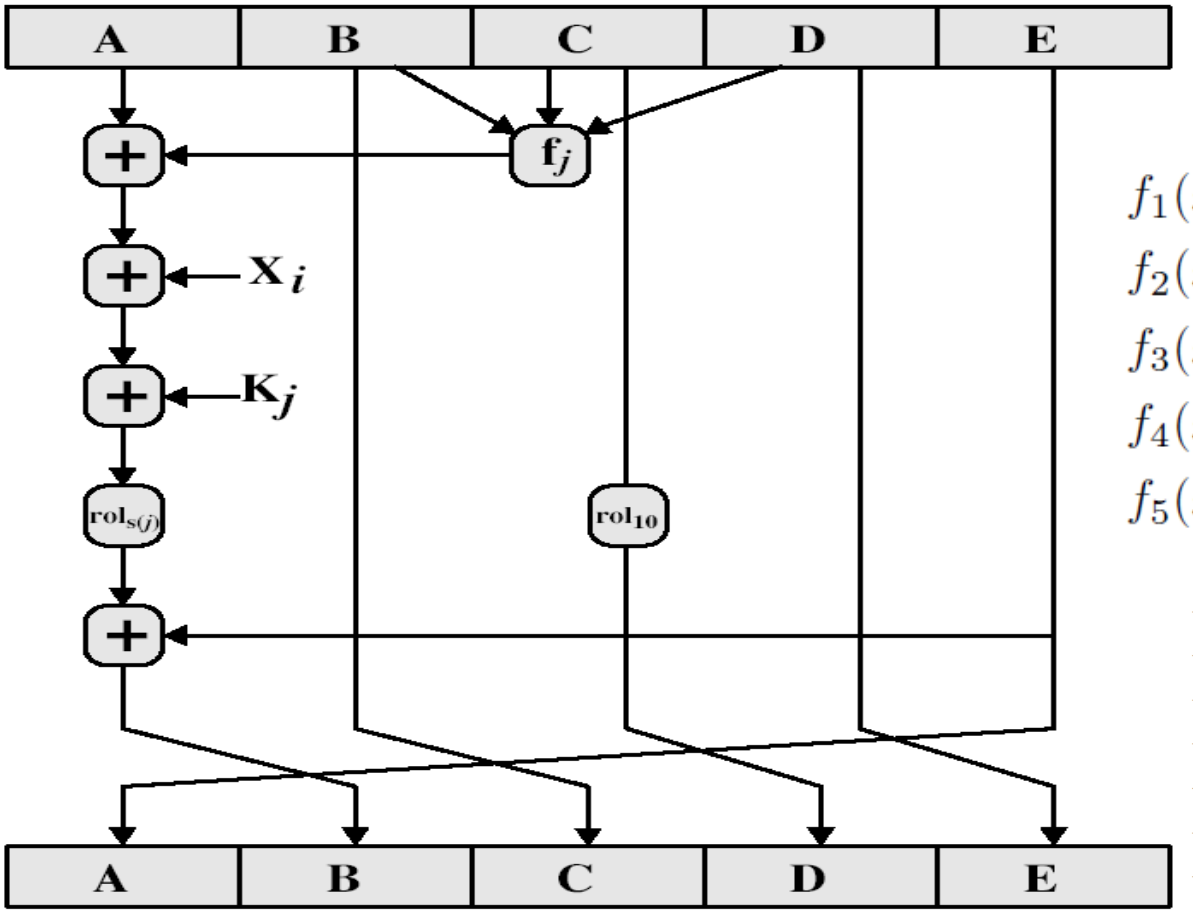
Figure MD5 Processing of a Single 512-bit Block

RIPEMD-160 Overview

RIPE (RACE Integrity Primitives Evaluation) in 1992

1. pad message so its length is 448 mod 512
2. append a 64-bit length value to message
3. initialise 5-word (160-bit) buffer (A,B,C,D,E) to (67452301,efcdab89,98badcfe,10325476,c3d2e1f0)
4. process message in 16-word (512-bit) chunks:
   - use 10 rounds of 16 bit operations on message block & buffer – in 2 parallel lines of 5
   - add output to input to form new buffer value
5. output hash value is the final buffer value

RIPEMD-160 Round



$$A := (A + f(B, C, D) + X + K)^{\lll s}.$$

$$f_1(x, y, z) = x \oplus y \oplus z,$$
$$f_2(x, y, z) = (x \wedge y) \vee (\neg x \wedge z),$$
$$f_3(x, y, z) = (x \vee \neg y) \oplus z,$$
$$f_4(x, y, z) = (x \wedge z) \vee (y \wedge \neg z),$$
$$f_5(x, y, z) = x \oplus (y \vee \neg z).$$

added constants (hexadecimal)

$K(j) = \texttt{00000000}_{\texttt{x}}$	$(0 \le j \le 15)$	
$K(j) = \texttt{5A827999}_{\texttt{x}}$	$(16 \le j \le 31)$	$\lfloor 2^{30} \cdot \sqrt{2} \rfloor$
$K(j) = \texttt{6ED9EBA1}_{\texttt{x}}$	$(32 \le j \le 47)$	$\lfloor 2^{30} \cdot \sqrt{3} \rfloor$
$K(j) = \texttt{8F1BBCDC}_{\texttt{x}}$	$(48 \le j \le 63)$	$\lfloor 2^{30} \cdot \sqrt{5} \rfloor$
$K(j) = \texttt{A953FD4E}_{\texttt{x}}$	$(64 \le j \le 79)$	$\lfloor 2^{30} \cdot \sqrt{7} \rfloor$
$K'(j) = \texttt{50A28BE6}_{\texttt{x}}$	$(0 \le j \le 15)$	$\lfloor 2^{30} \cdot \sqrt[3]{2} \rfloor$
$K'(j) = \texttt{5C4DD124}_{\texttt{x}}$	$(16 \le j \le 31)$	$\lfloor 2^{30} \cdot \sqrt[3]{3} \rfloor$
$K'(j) = \texttt{6D703EF3}_{\texttt{x}}$	$(32 \le j \le 47)$	$\lfloor 2^{30} \cdot \sqrt[3]{5} \rfloor$
$K'(j) = \texttt{7A6D76E9}_{\texttt{x}}$	$(48 \le j \le 63)$	$\lfloor 2^{30} \cdot \sqrt[3]{7} \rfloor$
$K'(j) = \texttt{00000000}_{\texttt{x}}$	$(64 \le j \le 79)$	

RIPEMD-160 verses MD5 & SHA-1

- brute force attack harder (160 like SHA-1 vs 128 bits for MD5)
- not vulnerable to known attacks, like SHA-1 though stronger (compared to MD4/5)
- slower than MD5 (more steps)
- all designed as simple and compact
- SHA-1 optimised for big endian CPU's vs RIPEMD-160 & MD5 optimised for little endian CPU's

Secure Hash Algorithm (SHA-1)

- SHA was designed by NIST & NSA in 1993, revised 1995 as SHA-1
- US standard for use with DSA signature scheme
  - standard is FIPS 180-1 in 1995, also Internet RFC3174
  - the algorithm is SHA, the standard is SHS
- produces 160-bit hash values
- now the generally preferred hash algorithm
- based on design of MD5 with key differences

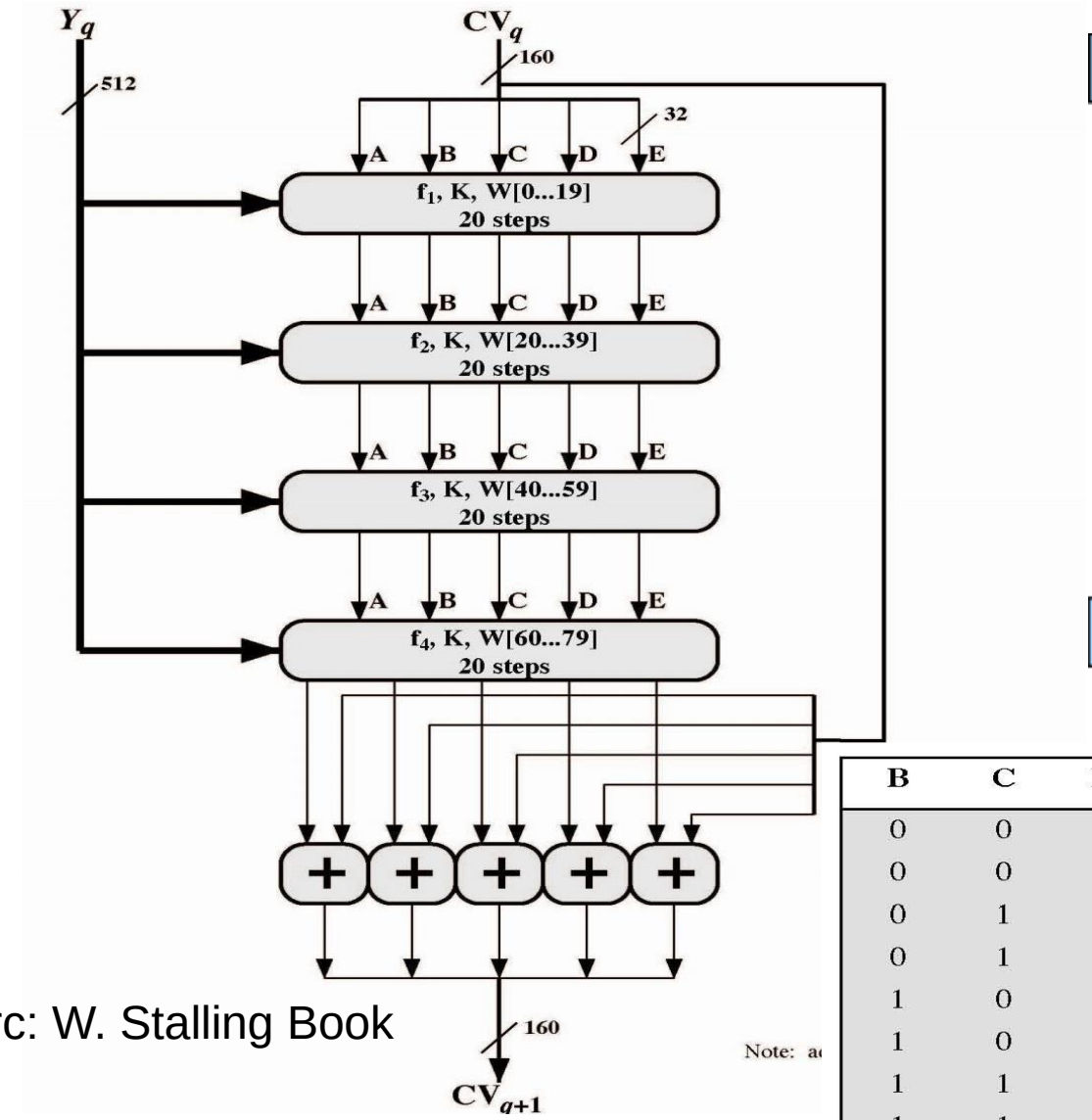
SHA-1 Compression Function

- each round has 20 steps which replaces the 5 buffer words thus:

  $(A,B,C,D,E) \leftarrow (E+f(t,B,C,D)+(A<<5)+W_t+K_t),A,(B<<30),C,D)$

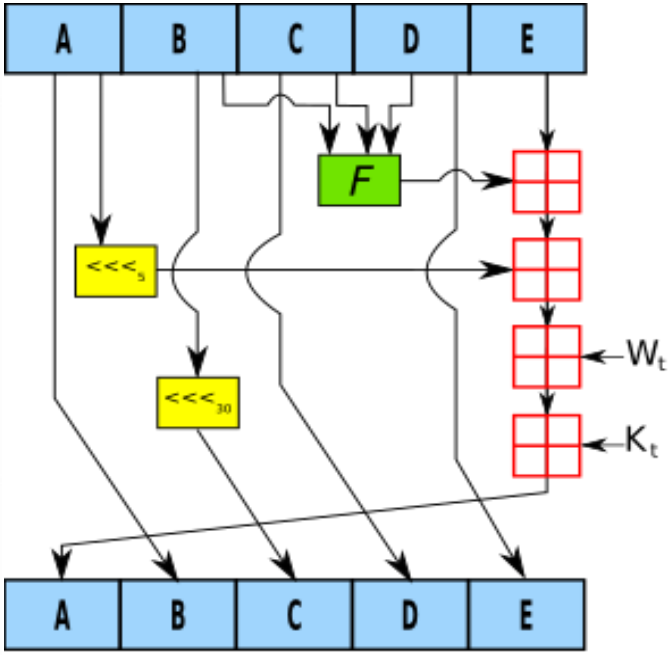
- t is the step number
- $f(t,B,C,D)$ is nonlinear function for round
- W_t is derived from the message block
- K_t is a constant value is integer part of
  - $2^{30}\text{sqrt}(2)$ for $0 \le t \le 19$
  - $2^{30}\text{sqrt}(3)$ for $20 \le t \le 39$
  - $2^{30}\text{sqrt}(5)$ for $40 \le t \le 59$
  - $2^{30}\text{sqrt}(10)$ for $50 \le t \le 79$

SHA-1



Src: W. Stalling Book

One iteration within the SHA-1 compression function



Src: Wikipedea

B	C	D	$f_{0..19}$	$f_{20..39}$	$f_{40..59}$	$f_{60..79}$
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	0	1	0	1
0	1	1	1	0	1	0
1	0	0	0	1	0	1
1	0	1	0	0	1	0
1	1	0	1	0	1	0
1	1	1	1	1	1	1

SHA-1 versus MD5

- brute force attack is harder (160 vs 128 bits for MD5)
- a little slower than MD5 (80 vs 64 steps)
- both designed as simple and compact
- optimised for big endian CPU's (vs MD5 which is optimised for little endian CPU's)

- Collision failure found in 2005 in 2^{33} operations

Revised Secure Hash Standard

- NIST have issued a revision FIPS 180-2
- adds 3 additional hash algorithms
- SHA 2: SHA-256, SHA-384, SHA-512
  - designed for compatibility with increased security
  - structure & detail is similar to SHA-1

- SHA-3 : Winner selected from solicitations in 2012

Collision resistance

- Def: *A Collision for H: M→T is a pair* $(x,x') \varepsilon M$ *s.t* $x \neq x'$ but $H(x) = H(x')$
  - *As* $|M| > |T|$ *collision exists*
    - Pigeon hole principle
  - *A function H: M→T is collision resistant hash function (CRHF) if it is hard to find even single collision pair (x x')*

Finding Collision

- How to find a collision (for 256 bit output)
  - try 2^{130} randomly chosen inputs
  - 99.8% chance that two of them will collide
- This works no matter what H is, but it takes too long
  - If a computer calculates 10,000 hashes/sec, it would
  - take 10^{27} years to compute 2^{128} hashes

Table Comparison of SHA Properties

	SHA-1	SHA-256	SHA-384	SHA-512
Message digest size	160	256	384	512
Message size	$< 2^{64}$	$< 2^{64}$	$< 2^{128}$	$< 2^{128}$
Block size	512	512	1024	1024
Word size	32	32	64	64
Number of steps	80	80	80	80
Security	80	128	192	256

Notes: 1. All sizes are measured in bits.
2. Security refers to the fact that a birthday attack on a message digest of size n produces a collision with a workfactor of approximately $2^{n/2}$.

- Thanks