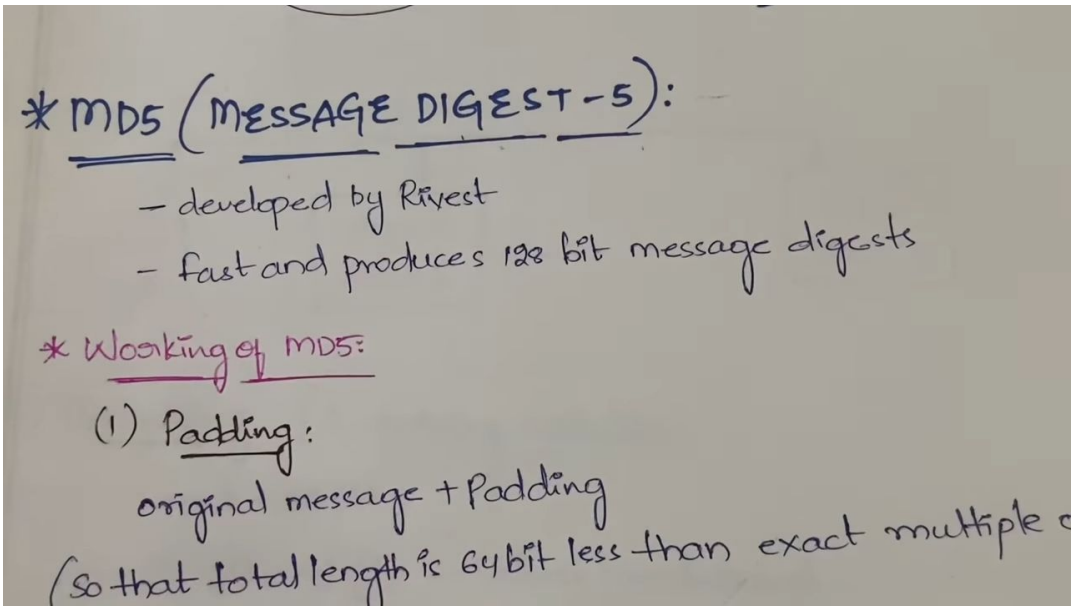
View, add and edit your notes in the app

#34 MD5 Algorithm (Message Digest 5) Working and Example |CNS

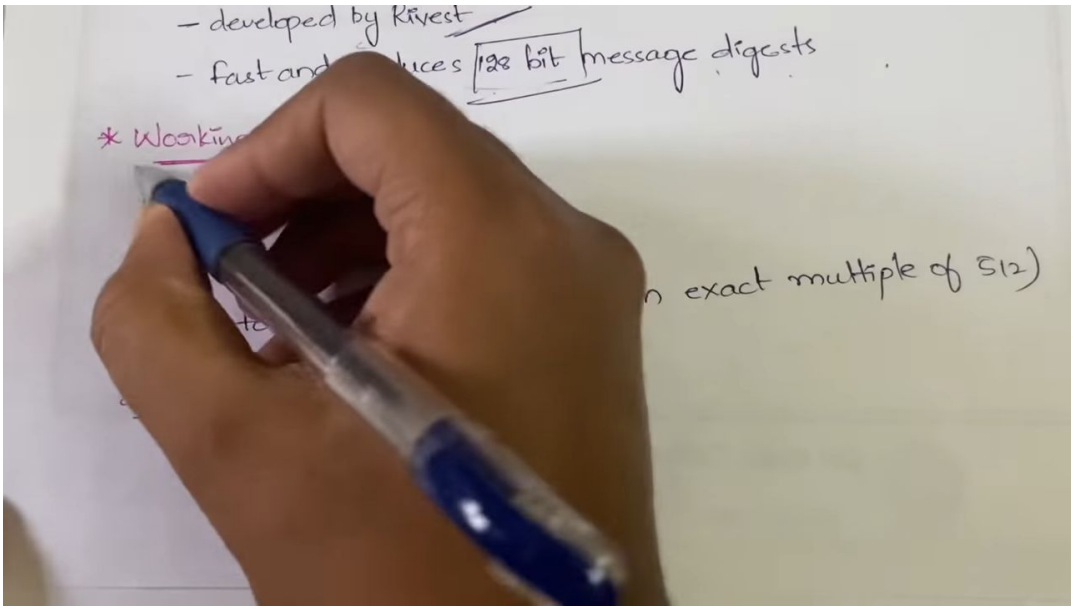
Generated on December 19, 2023

Summary

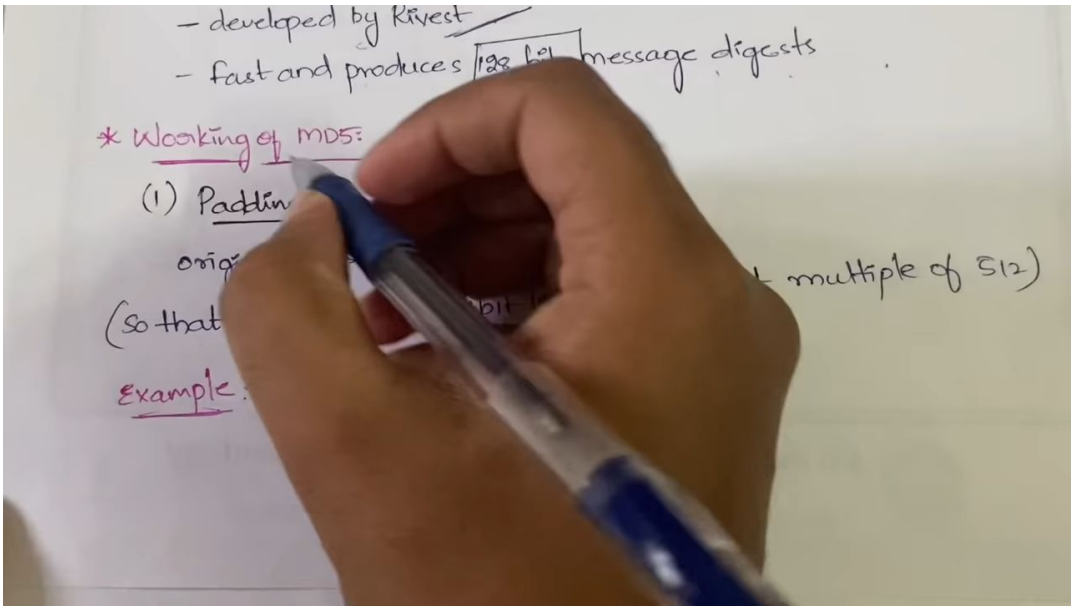
Notes	Screenshots	Bookmarks
0	13	0



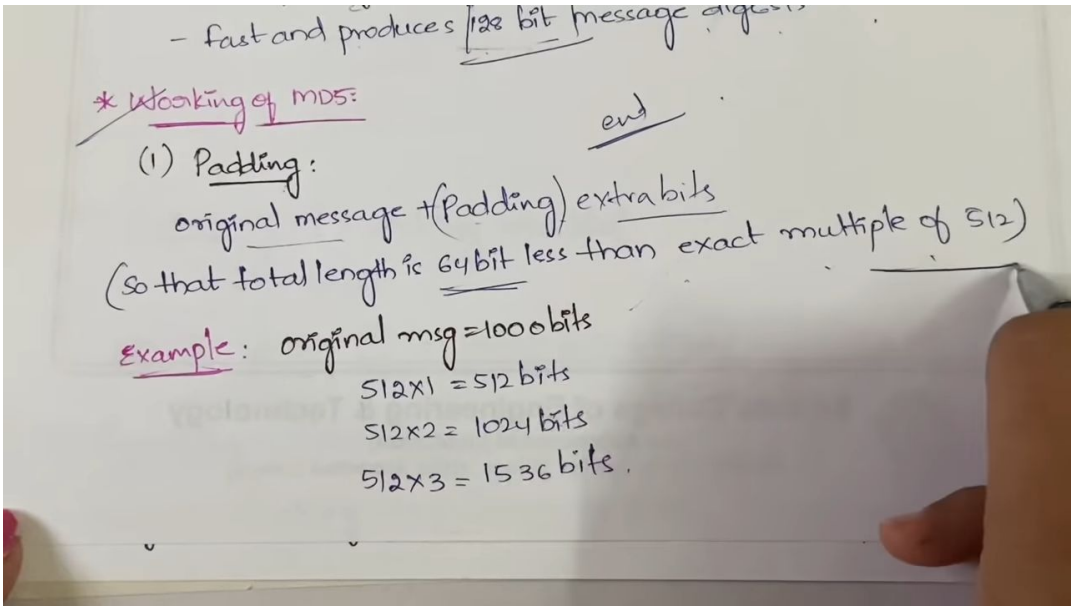
0:18


- developed by Rivest
- fast and
uces 128 bit message digests
* Wooking
exact multiple of 512)

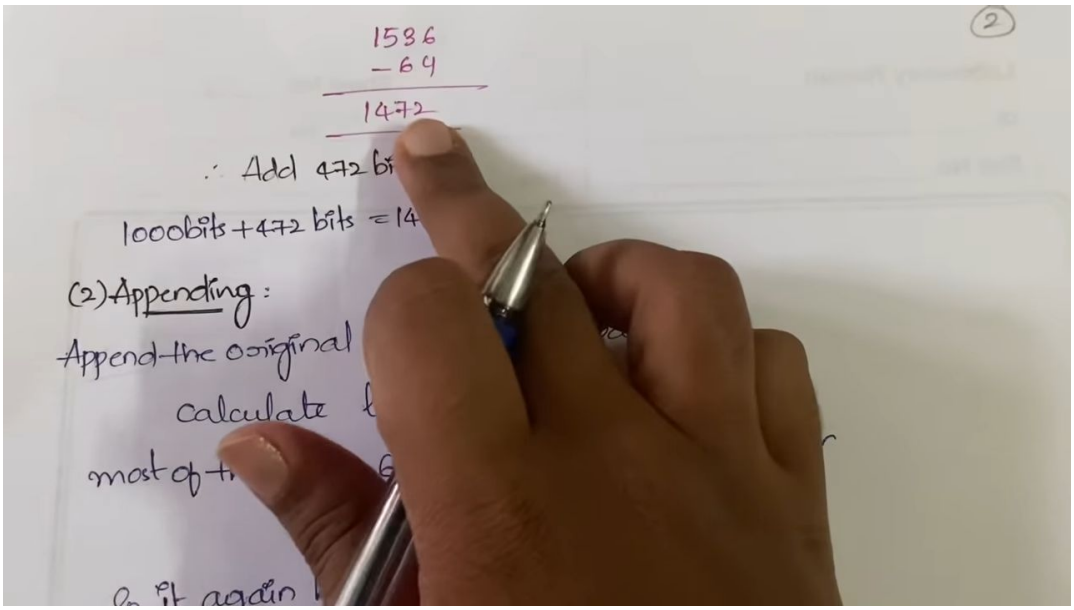
1:18


- developed by Rivest
- fast and produces 128 bit message digests
* Wooking of MD5:
(1) Paddin
orig
multiple of 512)
(so that
Example:

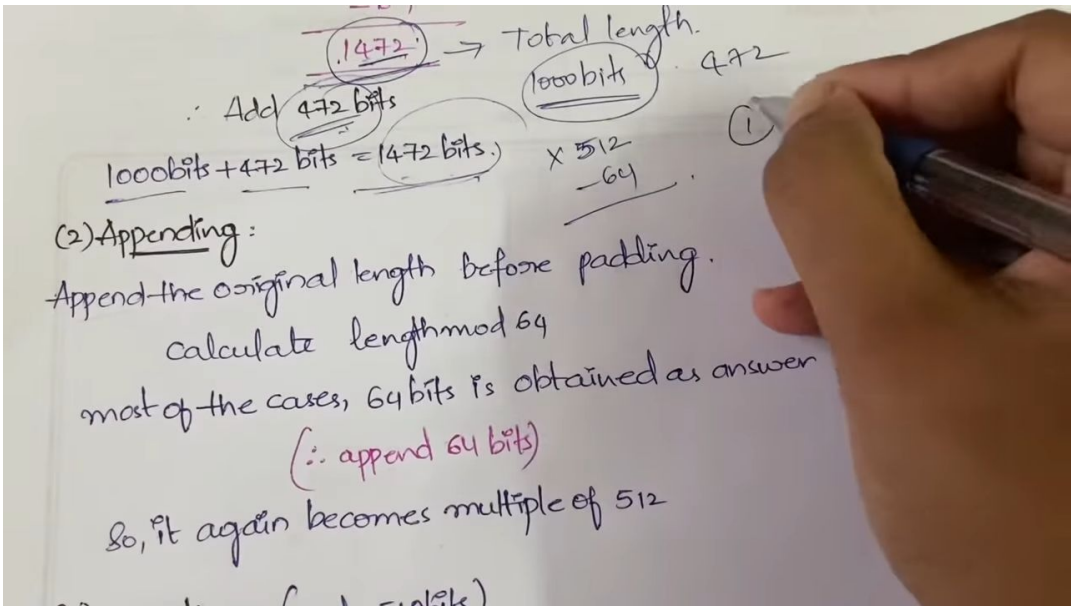
1:18


- fast and produces 128 bit message
* Working of MD5:
end
(1) Padding:
original message + (Padding) extra bits
(So that total length is 64 bit less than exact multiple of 512)
Example: original msg = 1000 bits
512 x 1 = 512 bits
512 x 2 = 1024 bits
512 x 3 = 1536 bits.

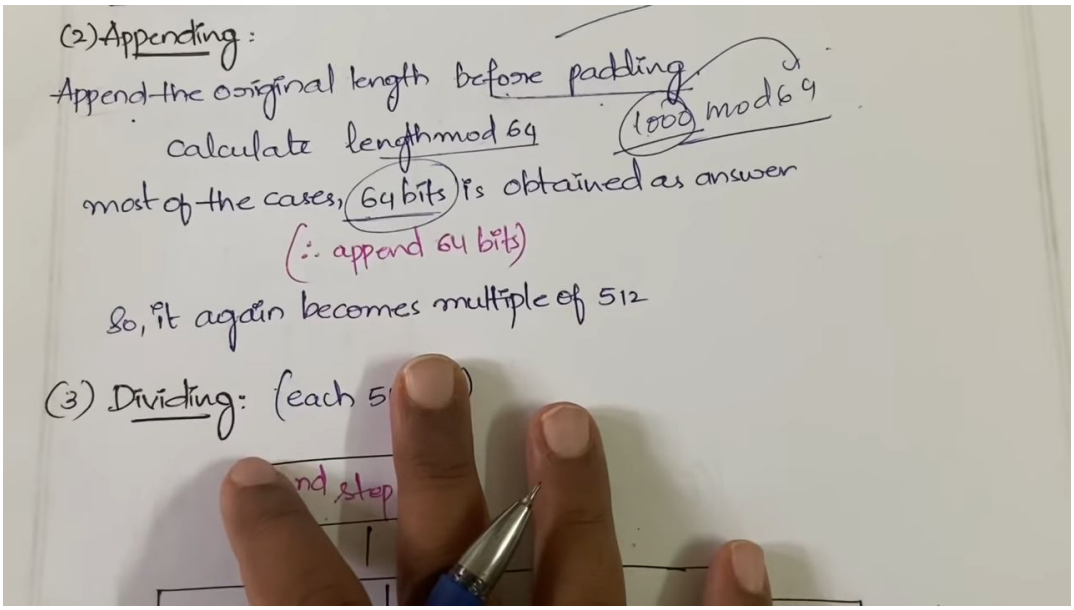
1:41


2
1536
- 64
1472
∴ Add 472 bi
1000 bits + 472 bits = 14
(2) Appending:
Append the original
Calculate
most of th

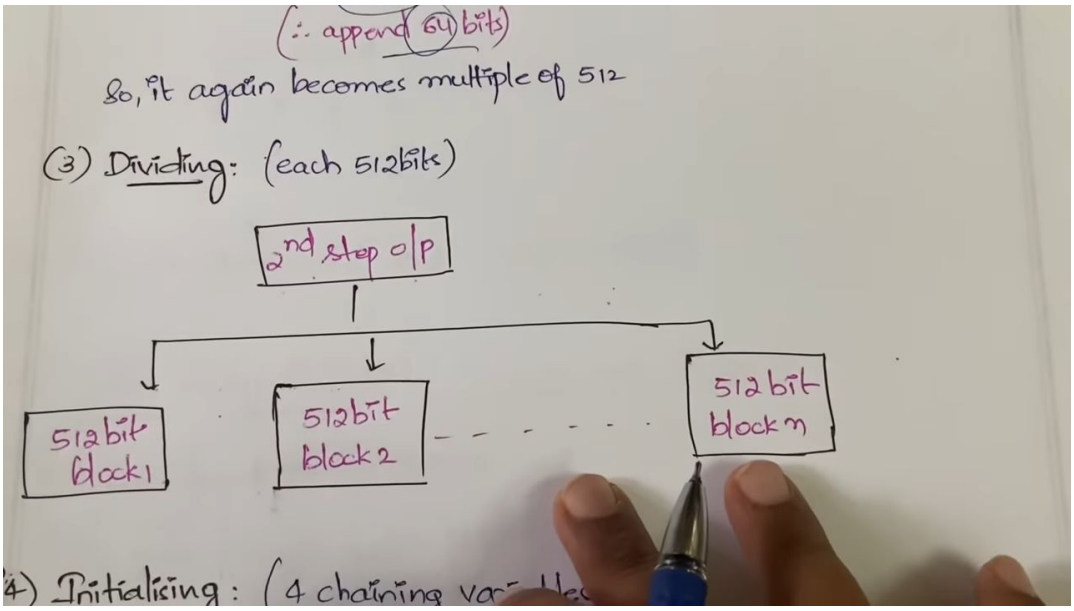
3:22


1472 → Total length.
1000 bits
472
∴ Add 472 bits
1000bits + 472 bits = 1472 bits.
× 512
−64
(2) Appending:
Append the original length before padding.
calculate length mod 64
most of the cases, 64 bits is obtained as answer
(∴ append 64 bits)
So, it again becomes multiple of 512

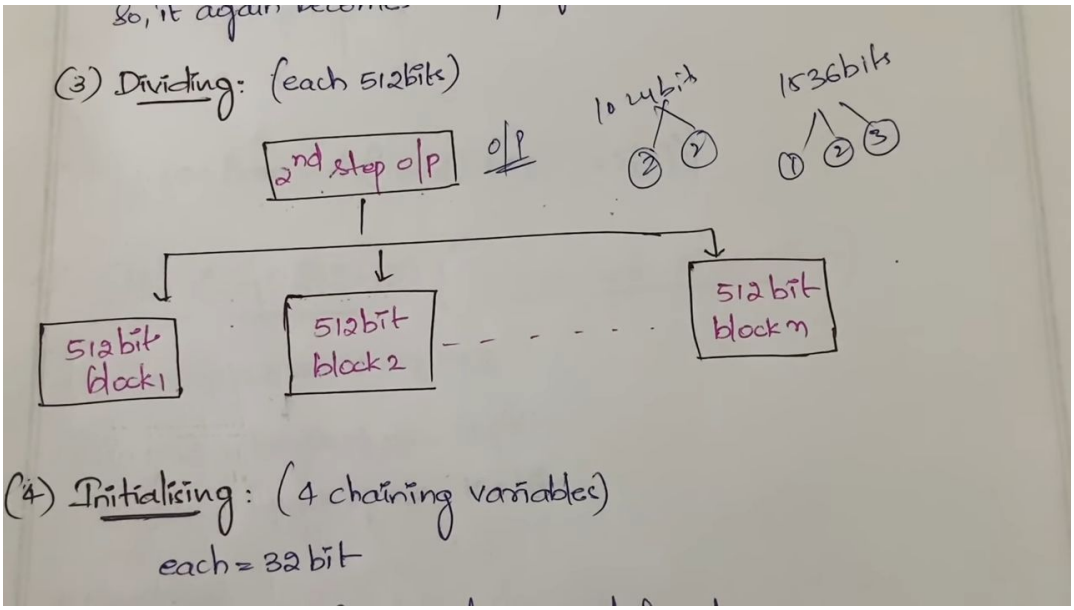
4:05


(2) Appending:
Append the original length before padding
calculate length mod 64
1000 mod 64
most of the cases, 64 bits is obtained as answer
(∴ append 64 bits)
So, it again becomes multiple of 512
(3) Dividing: (each 5

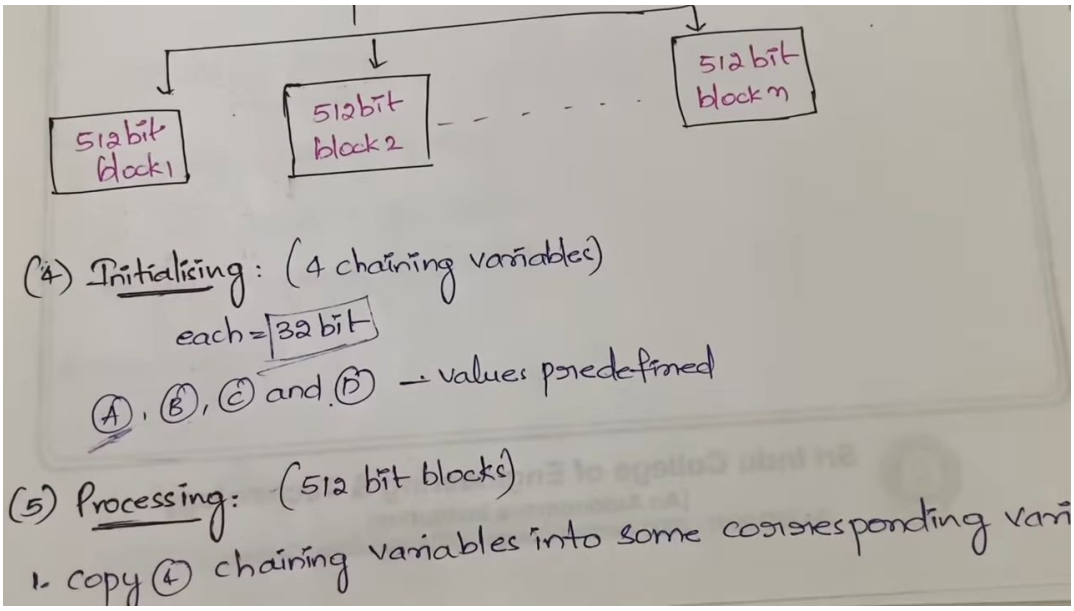
4:34


(∴ append 64 bits)
So, it again becomes multiple of 512
(3) Dividing: (each 512bits)
2nd step o/p
512 bit block 1
512bit block 2
512 bit block n
(4) Initialising: (4 chaining

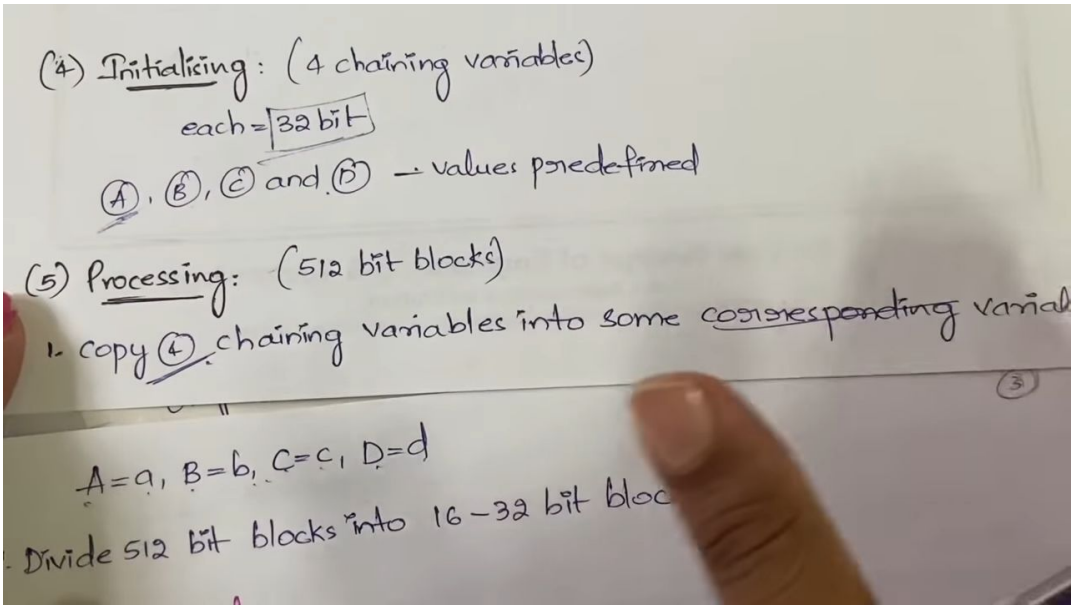
▷ 5:00


(3) Dividing: (each 512bits)
2nd step o/p
o/p
1024bits
1536bits
512 bit block 1
512bit block 2
512 bit block n
(4) Initialising: (4 chaining variables)
each = 32 bit

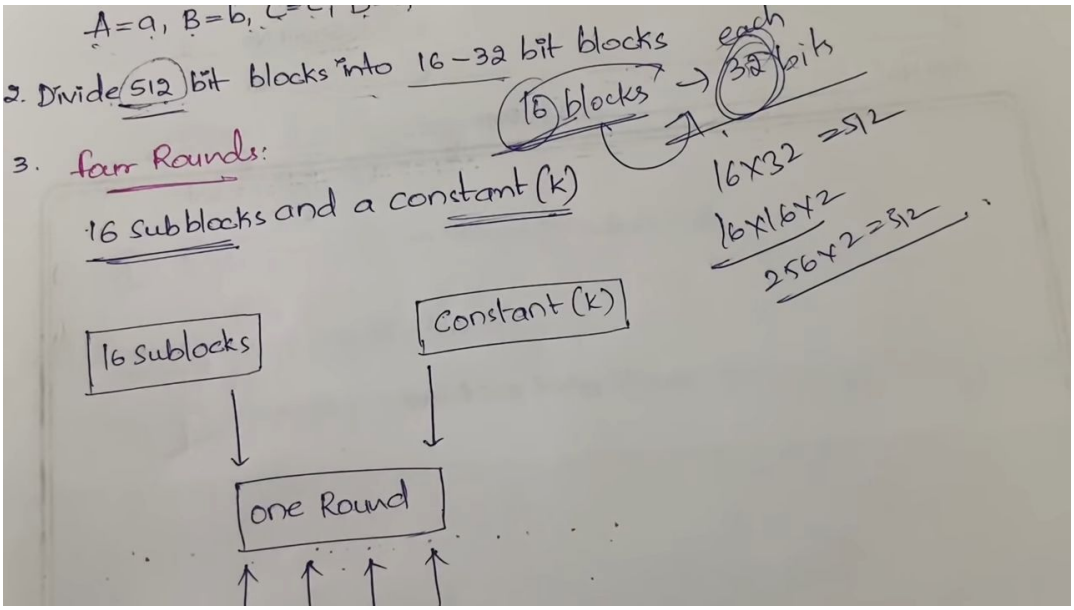
▷ 6:07


512 bit block 1
512 bit block 2
512 bit block n
(4) Initialising: (4 chaining variables)
each = 32 bit
(A), (B), (C) and (D) → values predefined
(5) Processing: (512 bit blocks)
1. copy (4) chaining variables into some corresponding vari

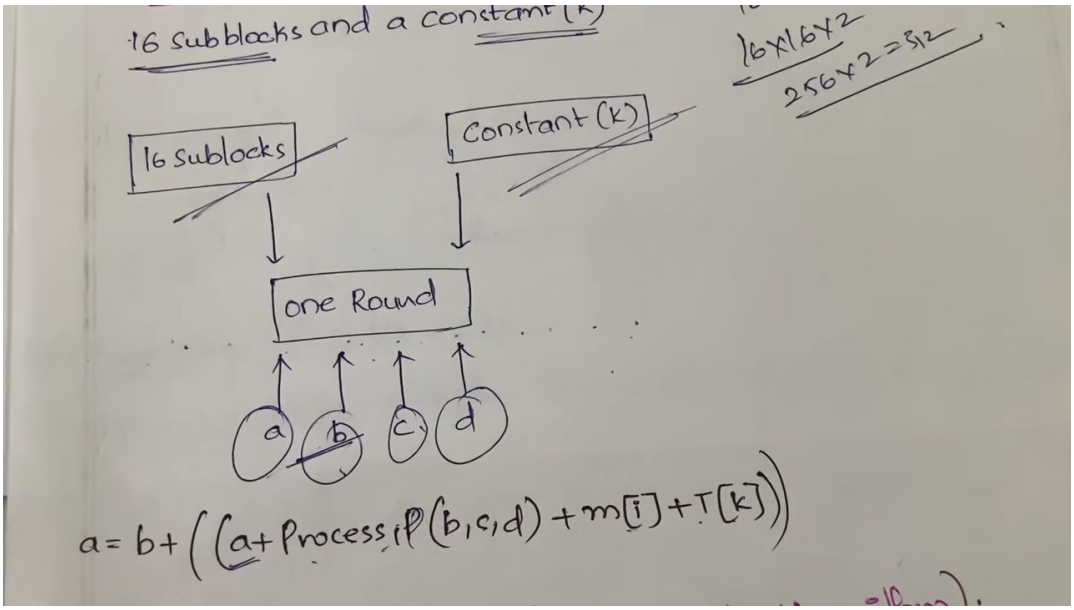
▷ 6:30


(4) Initialising: (4 chaining variables)
each = 32 bit
(A), (B), (C) and (D) → values predefined
(5) Processing: (512 bit blocks)
1. copy (4) chaining variables into some corresponding varial
A=a, B=b, C=c, D=d
Divide 512 bit blocks into 16-32 bit bloc

▷ 7:11


A=a, B=b,
2. Divide 512 bit blocks into 16-32 bit blocks each
16 blocks → 32 bits
3. four Rounds:
16 subblocks and a constant (k)
16X32 =512
16X16X2
256X2=512
16 Subblocks
Constant (k)
one Round

8:19


16 subblocks and a constant (k)
16X16X2
256X2=512
16 Subblocks
Constant (k)
one Round
a
b
c
d
a = b + ((a + Process if (b,c,d) + m[i] + T[k]))

9:04