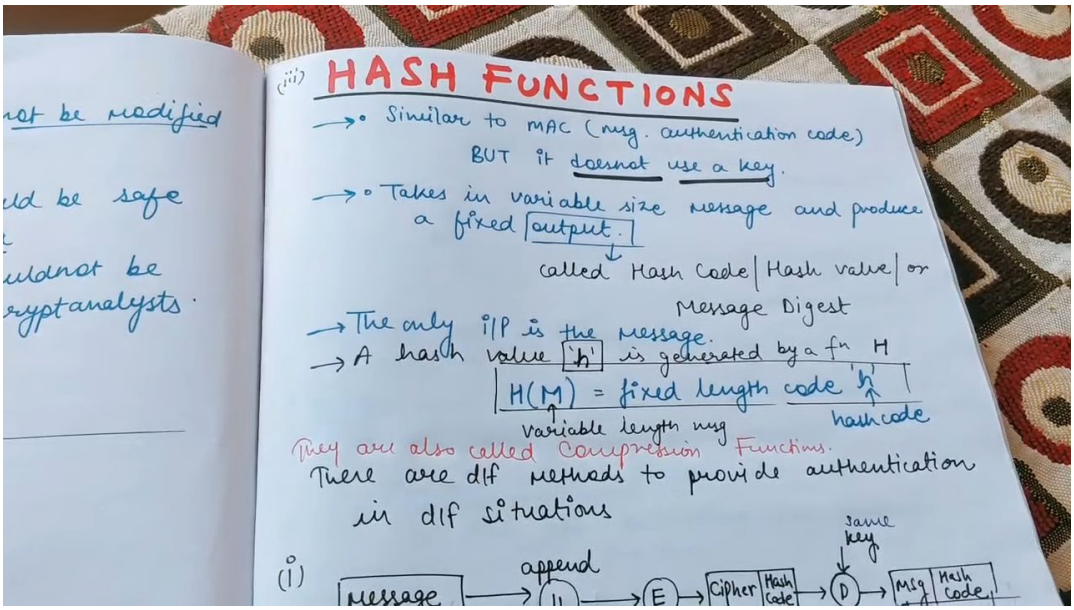
View, add and edit your notes in the app

Hash Functions in Cryptography

Generated on December 19, 2023

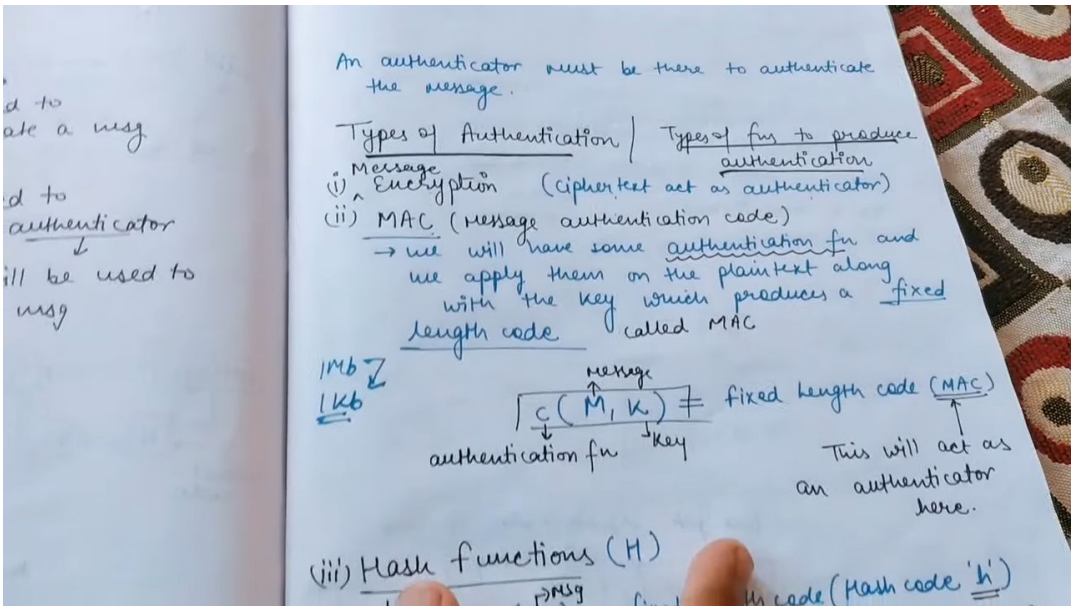
Summary

Notes	Screenshots	Bookmarks
0	18	0



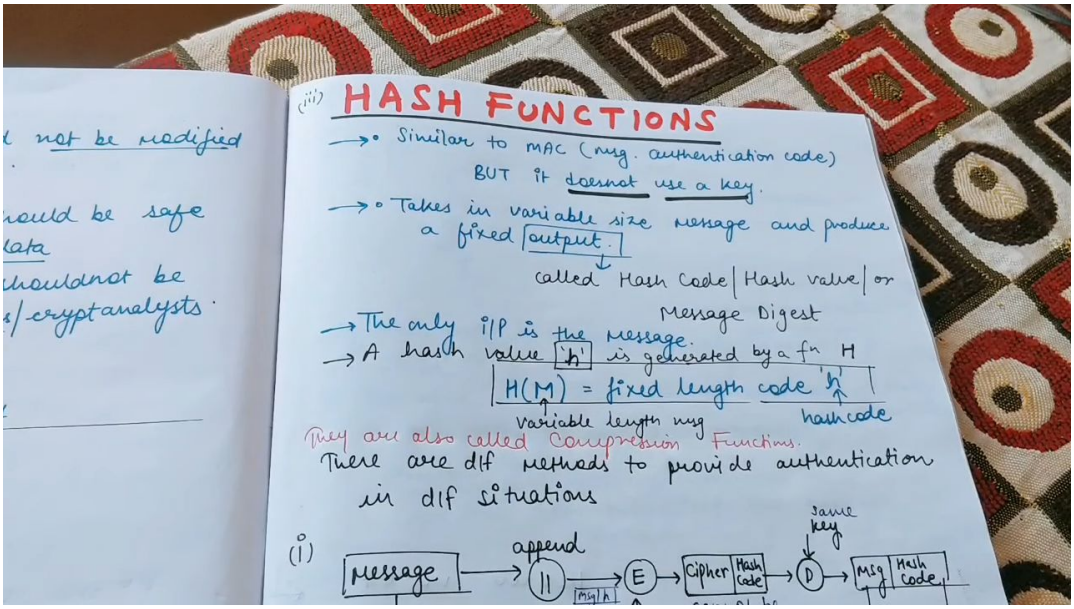
The 10 Cense 2023 Wii Summit has brought together a diverse group of speakers from around the world to share the latest achievements in science.

0:09



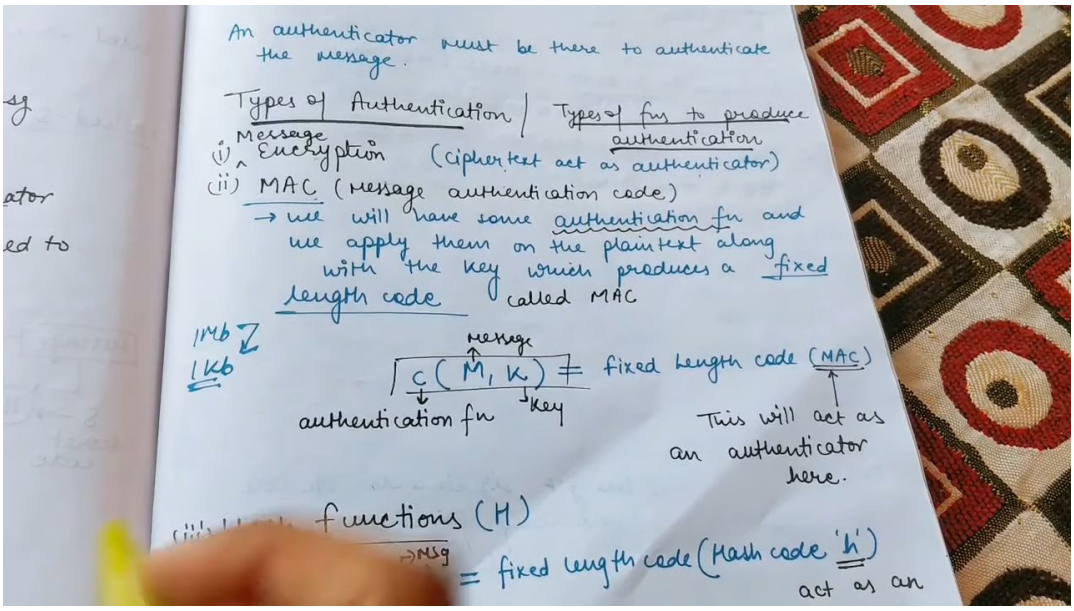
The Wii Summit has brought together speakers from around the world for 11 years to explore the role of science in understanding human life on Earth.

0:28



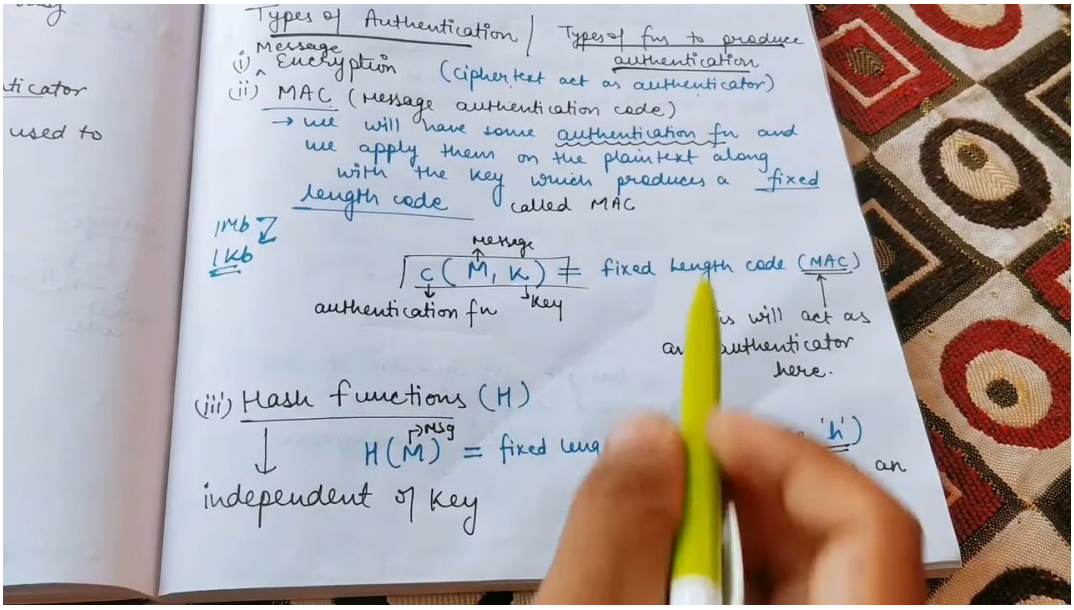
Summit's exploration of the role of science in understanding human life on Earth has inspired advancements in technology and solutions for addressing global challenges.

0:44



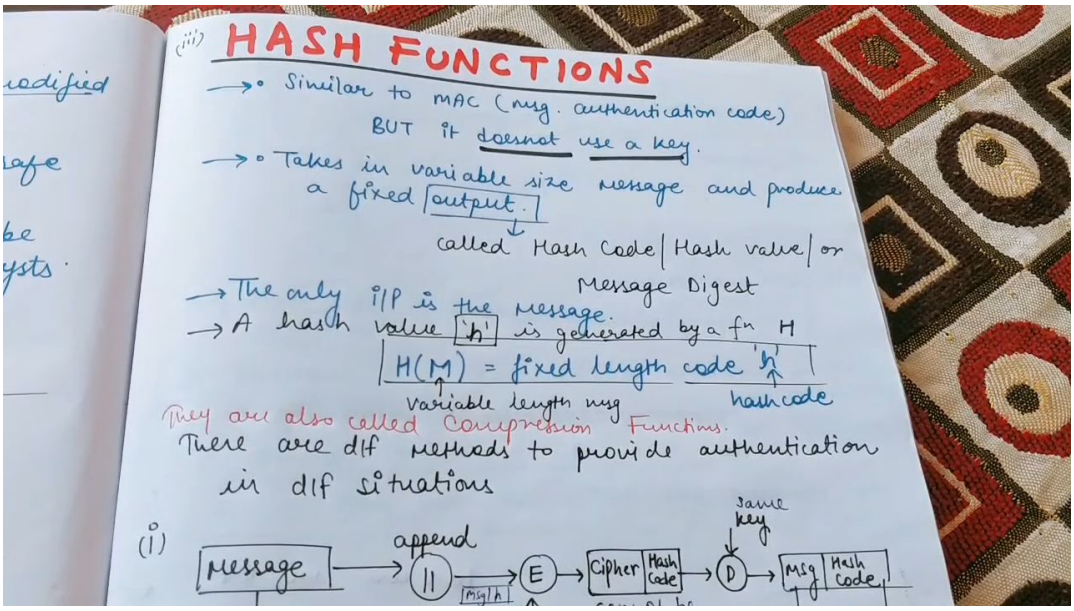
As the impacts of climate change continue to grow, it has become evident that humans need to focus on preparing for an uncertain future through both mitigation and adaptation efforts.

1:21



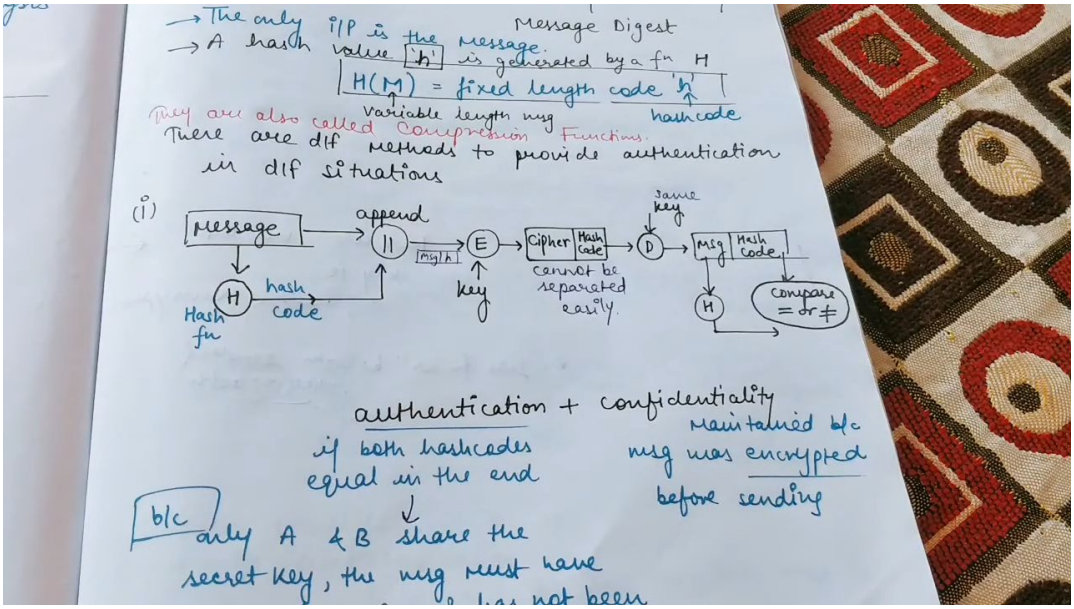
The concept of applying artificial intelligence to fuel efficiency goes beyond reducing waste and greenhouse gas emissions, preparing us for a world where critical resources may become even scarcer.

1:47



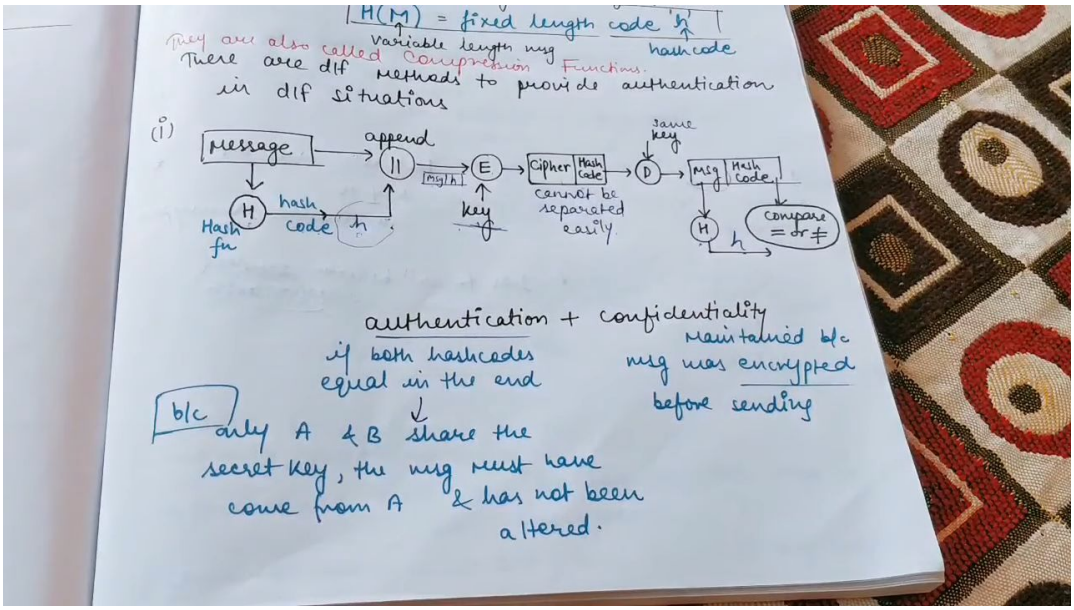
Artificial intelligence is crucial for improving productivity and reducing waste in critical infrastructure systems.

2:14

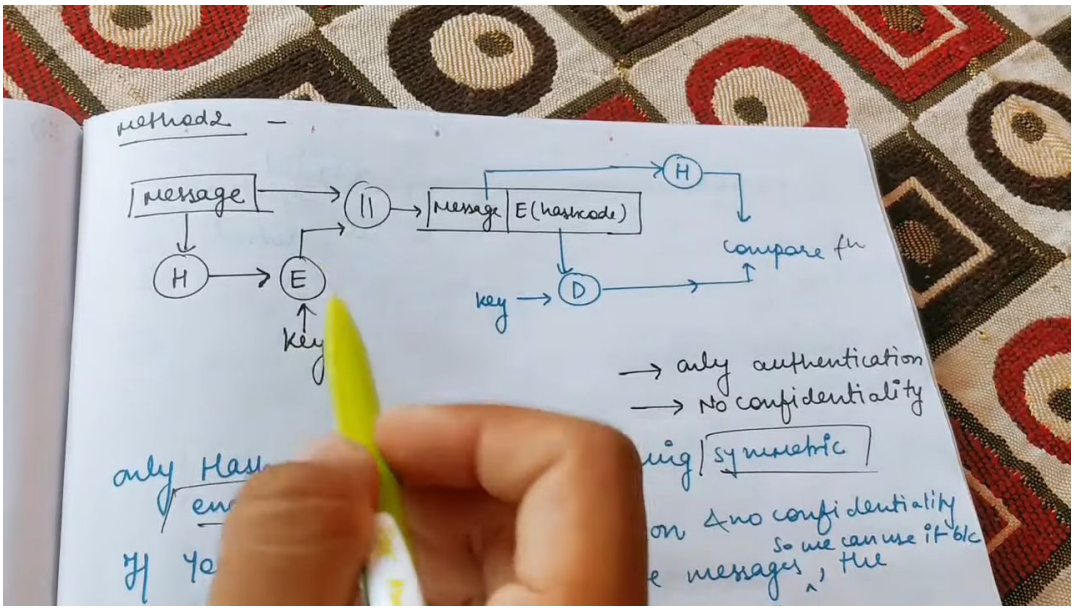


The digital seeds Bank project is set to be unveiled at this year's Wii Summit 2023, showcasing the growth and flourishing of an early seed of an idea.

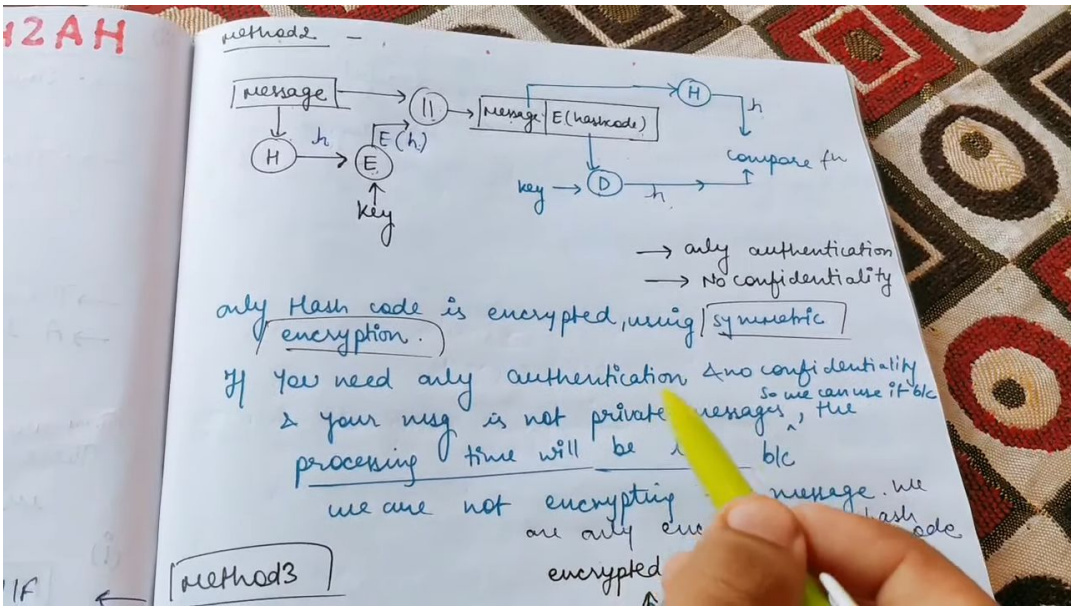
3:00


H(M) = fixed length code 'h'
variable length msg
hashcode
They are also called Compression Functions.
There are dif methods to provide authentication in dif situations
(i)
message
append
||
H
hash code
h
Hash fn
msg/h
E
key
Cipher
Hash code
cannot be separated easily
same key
D
msg
Hash code
H
h
compare = or ≠
authentication + confidentiality
if both hashcodes equal in the end
Maintained b/c msg was encrypted before sending
b/c only A & B share the secret key, the msg must have come from A & has not been altered.

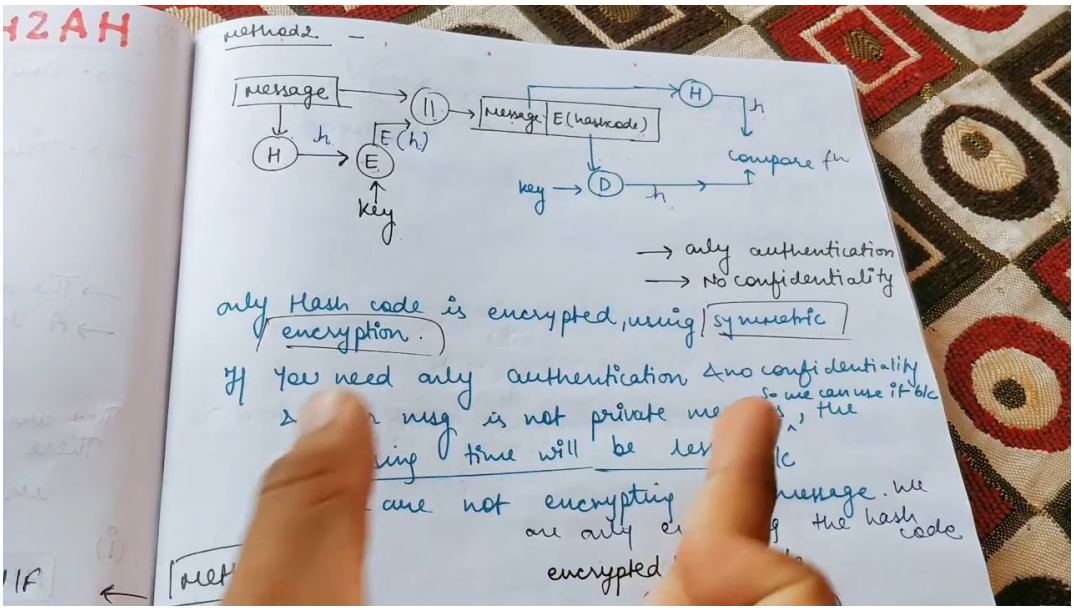
5:11


methods –
message
||
H
message
E(hashcode)
H
E
key
D
compare fn
key
→ only authentication
→ No confidentiality
only Has
msg
symmetric
en
on & no confidentiality
so we can use it b/c
Yf Yo
e messages, the

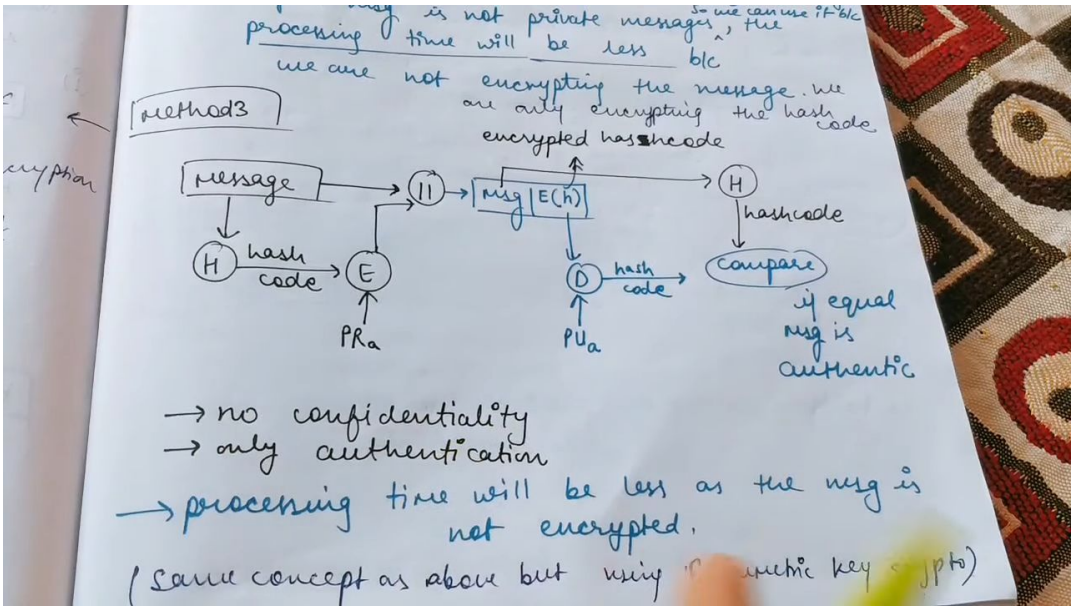
5:58


method2 –
message
H
E(h)
Key
message | E(hashcode)
compare fn
key
→ only authentication
→ No confidentiality
only Hash code is encrypted, using symmetric encryption.
If you need only authentication & no confidentiality
So we can use it bcz
processing time will be
bcz
we are not encrypting
message. we
are only
encrypted
method3

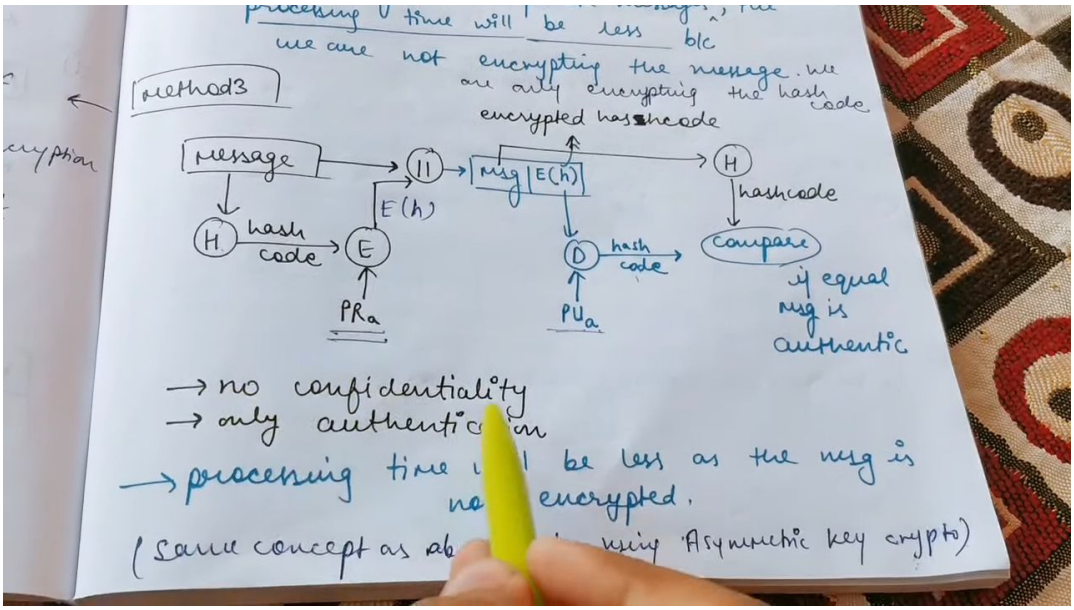
▷ 7:26


method2 –
message
H
E(h)
Key
message | E(hashcode)
compare fn
key
→ only authentication
→ No confidentiality
only Hash code is encrypted, using symmetric encryption.
If you need only authentication & no confidentiality
So we can use it bcz
are not encrypting
message. we
are only
the hash code
encrypted

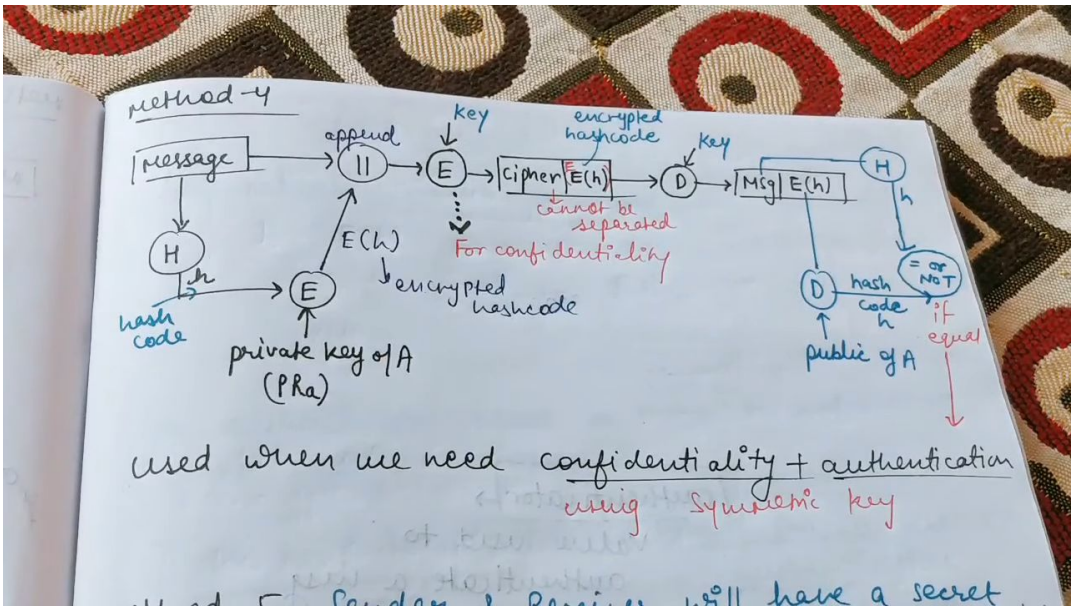
▷ 7:45


processing time will be less b/c
we are not encrypting the message. we are only encrypting the hash code
method3
encrypted hashcode
message
H
hash code
E
PRa
||
msg | E(h)
H
hashcode
D
hash code
PUa
compare
if equal msg is authentic
→ no confidentiality
→ only authentication
→ processing time will be less as the msg is not encrypted.
(Same concept as above but using

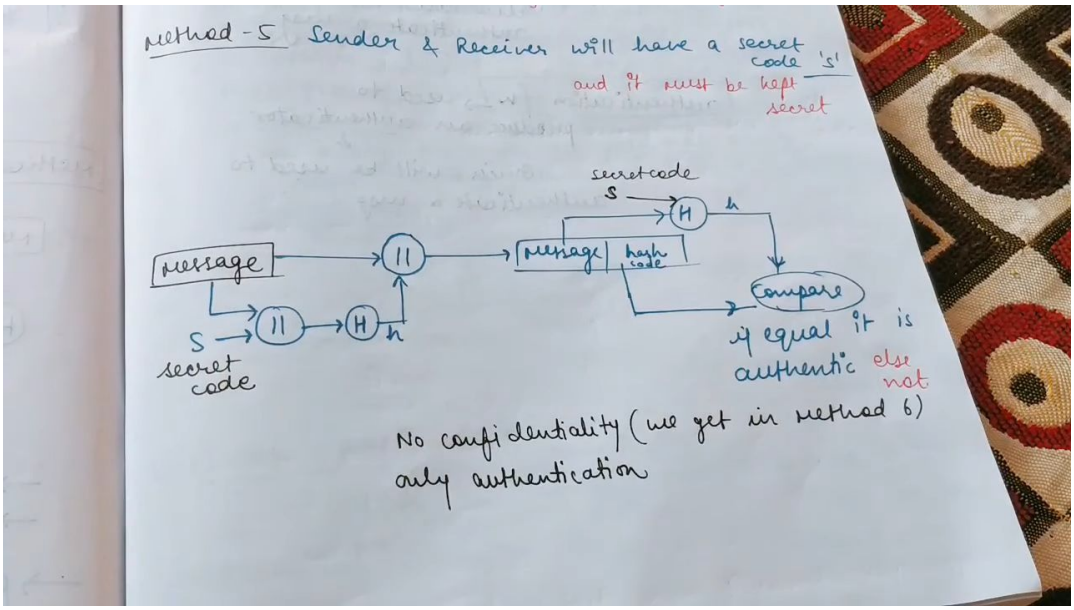
8:29


processing time will be less b/c
we are not encrypting the message. we are only encrypting the hash code
method3
encrypted hashcode
message
H
hash code
E
E(h)
PRa
||
msg | E(h)
H
hashcode
D
hash code
PUa
compare
if equal msg is authentic
→ no confidentiality
→ only authentication
→ processing time will be less as the msg is not encrypted.
(Same concept as above but using Asymmetric key crypto)

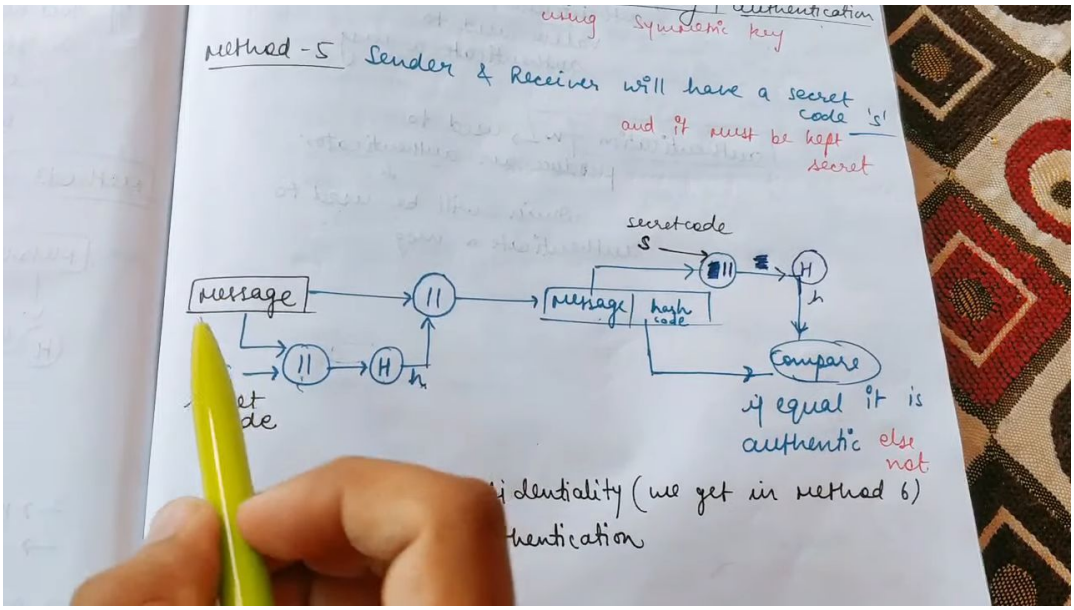
9:28


method-4
message
append
||
Key
E
encrypted hashcode
cipher | E(h)
cannot be separated
For confidentiality
Key
D
Msg | E(h)
H
h
= or NOT
D
hash code h
if equal
public of A
H
h
hash code
E
E(h)
encrypted hashcode
private key of A
(PRa)
used when we need confidentiality + authentication
using Symmetric key
will have a secret

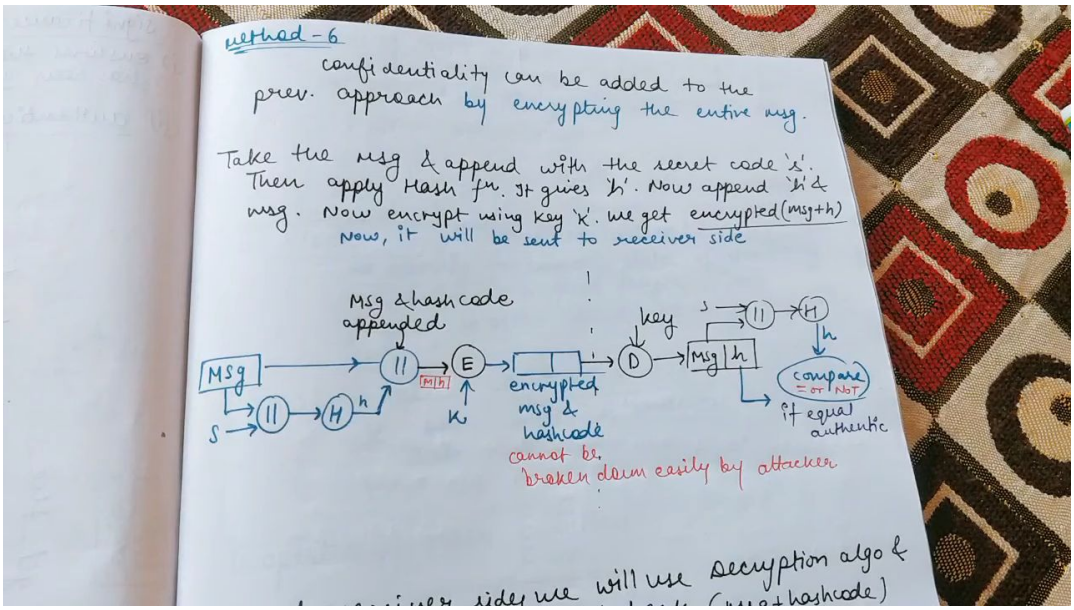
9:43


method-5 Sender & Receiver will have a secret code 's' and it must be kept secret
secret code
S
H
h
message
||
message | hash code
compare
message
||
H
h
S
secret code
if equal it is authentic else not
No confidentiality (we get in method 6)
only authentication

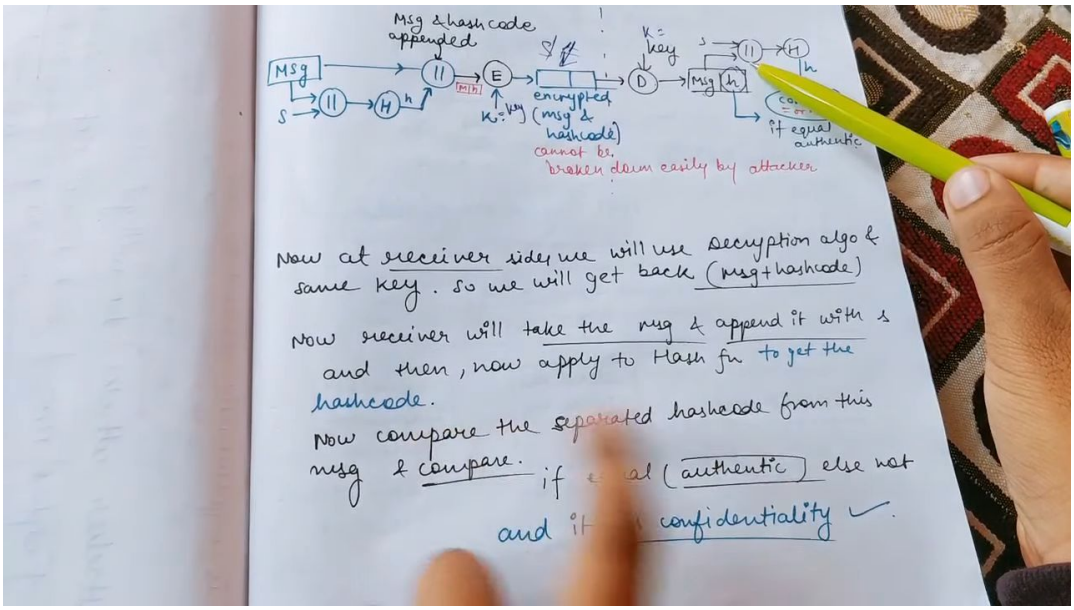
12:55


using Symmetric key
Method - 5 Sender & Receiver will have a secret code 's' and it must be kept secret
secret code
s
message
message | hash code
Compare
if equal it is authentic else not
identiality (we get in method 6)
thentication

13:05


Method - 6
confidentiality can be added to the prev. approach by encrypting the entire msg.
Take the msg & append with the secret code 's'. Then apply Hash fn. It gives 'h'. Now append 'h' & msg. Now encrypt using key 'k'. We get encrypted(msg+h)
Now, it will be sent to receiver side
Msg & hash code appended
MSG
encrypted msg & hashcode
cannot be broken down easily by attacker
key
Msg | h
compare = or NOT
if equal authentic
decryption algo &

14:15


Msg & hash code appended
MSG
S
H
h
E
encrypted (msg & hashcode)
cannot be broken down easily by attacker
K = key
D
if equal authentic
Now at receiver side we will use decryption algo & same key. So we will get back (msg+hashcode)
Now receiver will take the msg & append it with s and then, now apply to Hash fn to get the hashcode.
Now compare the separated hashcode from this msg & compare. if equal (authentic) else not
and it confidentiality

16:35