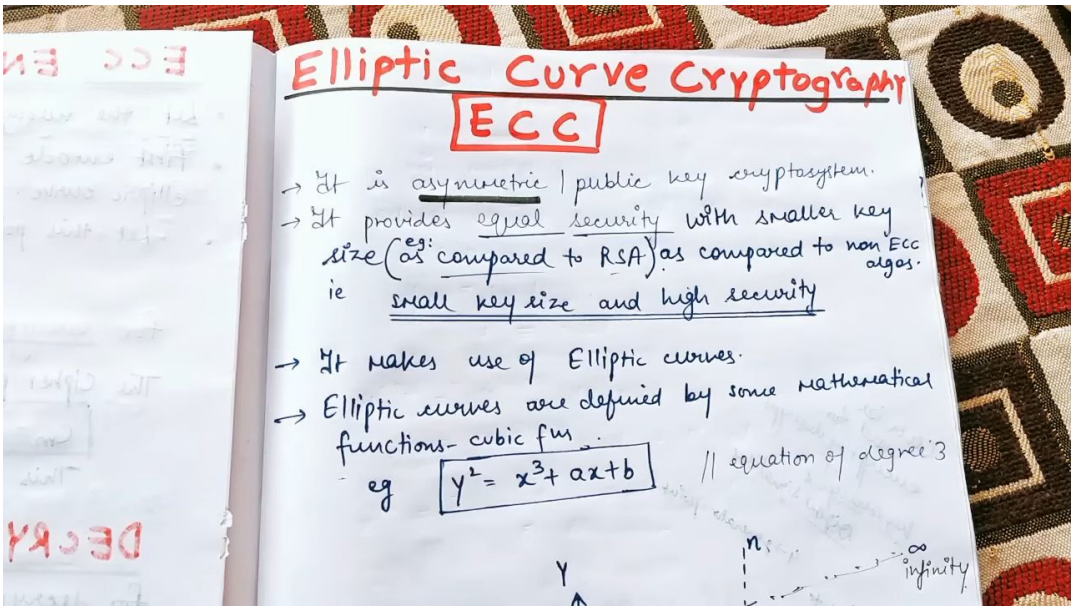
View, add and edit your notes in the app

Elliptic Curve Cryptography | ECC in Cryptography and Network Security

Generated on December 19, 2023

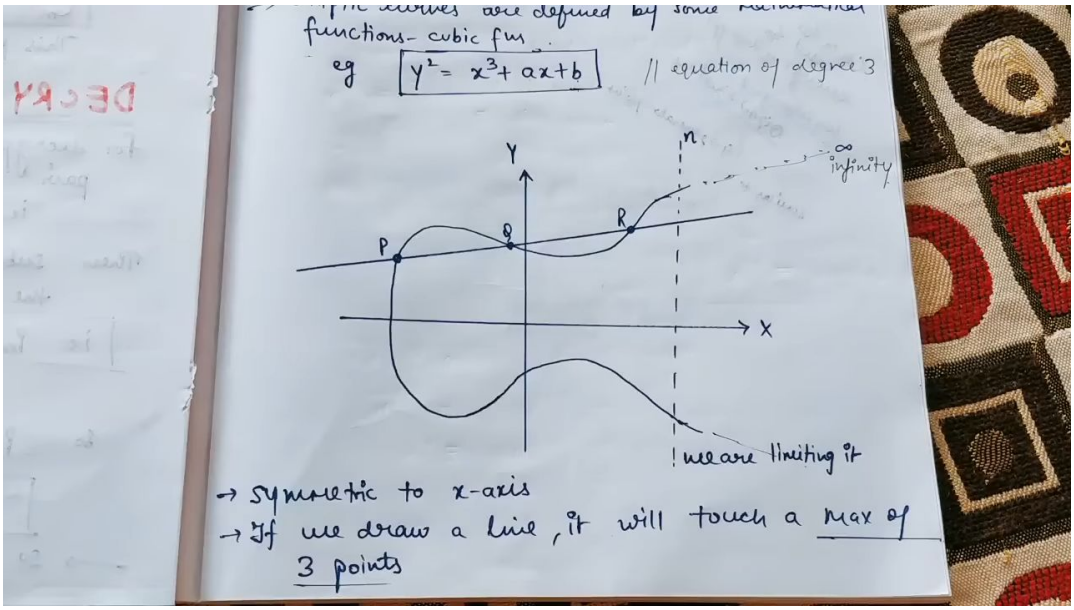
Summary

Notes	Screenshots	Bookmarks
0	16	0



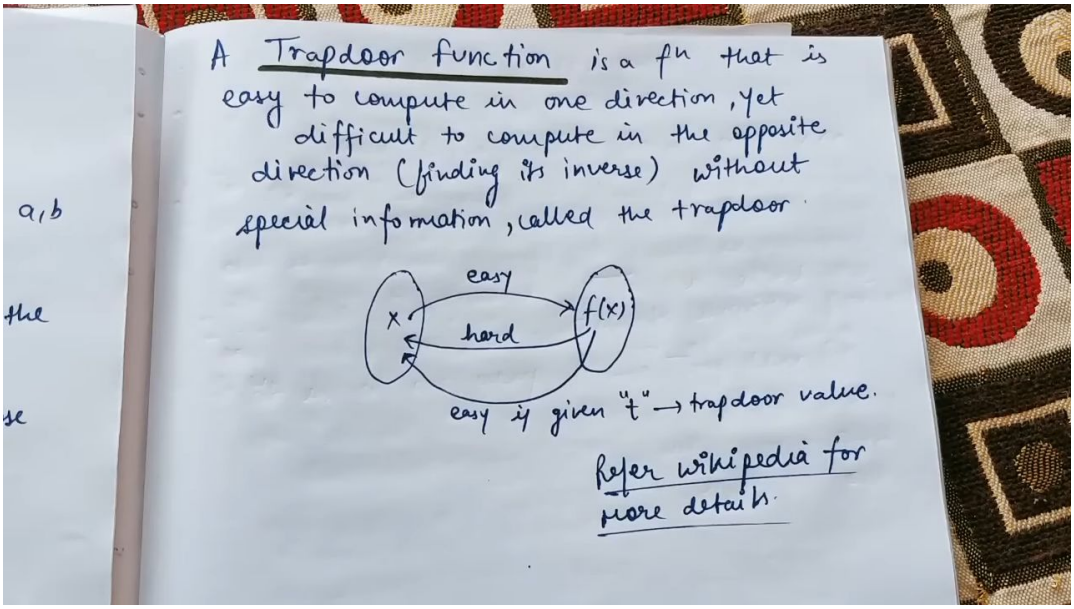
The study of the 10 most powerful empires and reserve currencies over the last 500 years reveals fascinating patterns of rise and decline.

0:08



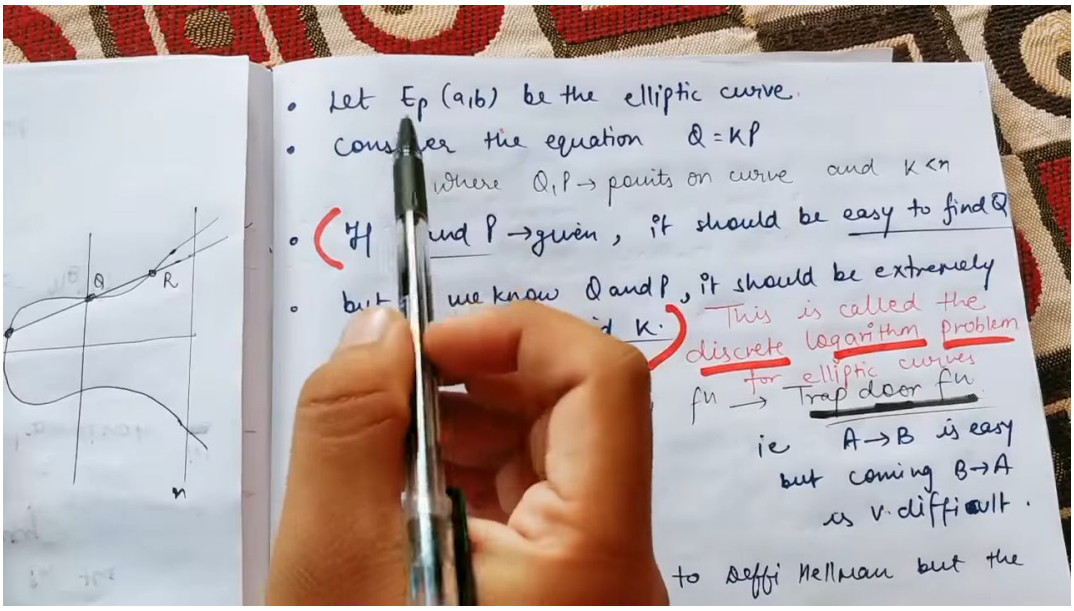
A new study has developed a comprehensive method for measuring the power of different empires, using eight key metrics to determine their total strength.

1:43



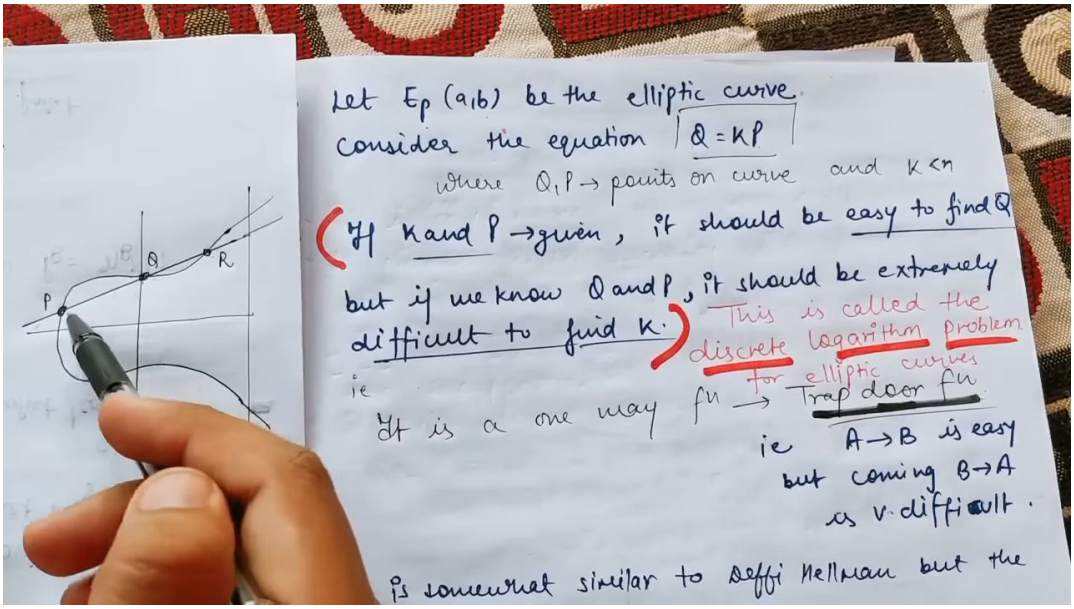
The link between education, innovation, technology development, and currency reserve status is crucial for understanding a country's economic rise and decline.

2:49



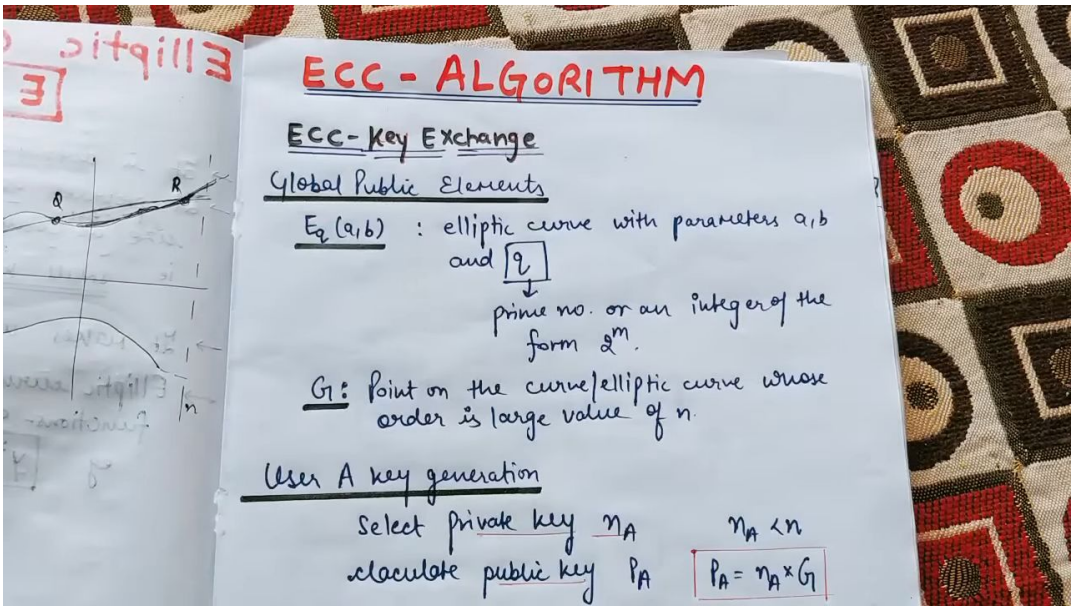
The uneven distribution of wealth in times of increased prosperity can lead to financial bubbles, internal conflict, and potential revolutions for wealth redistribution.

4:00

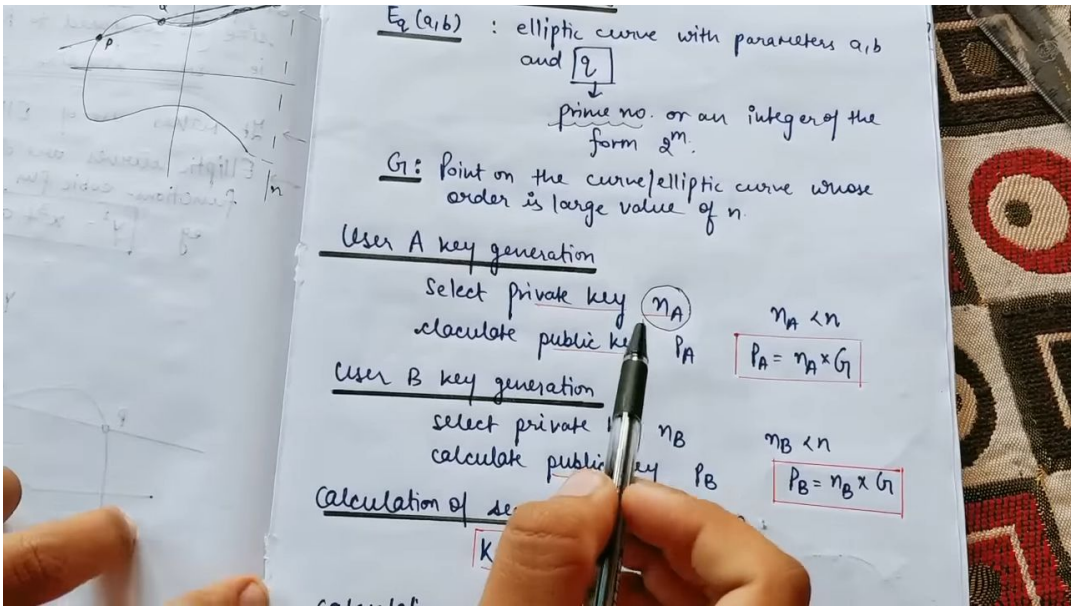


Internal conflict and power struggles within the Empire may lead to a redistribution of wealth and potential civil unrest.

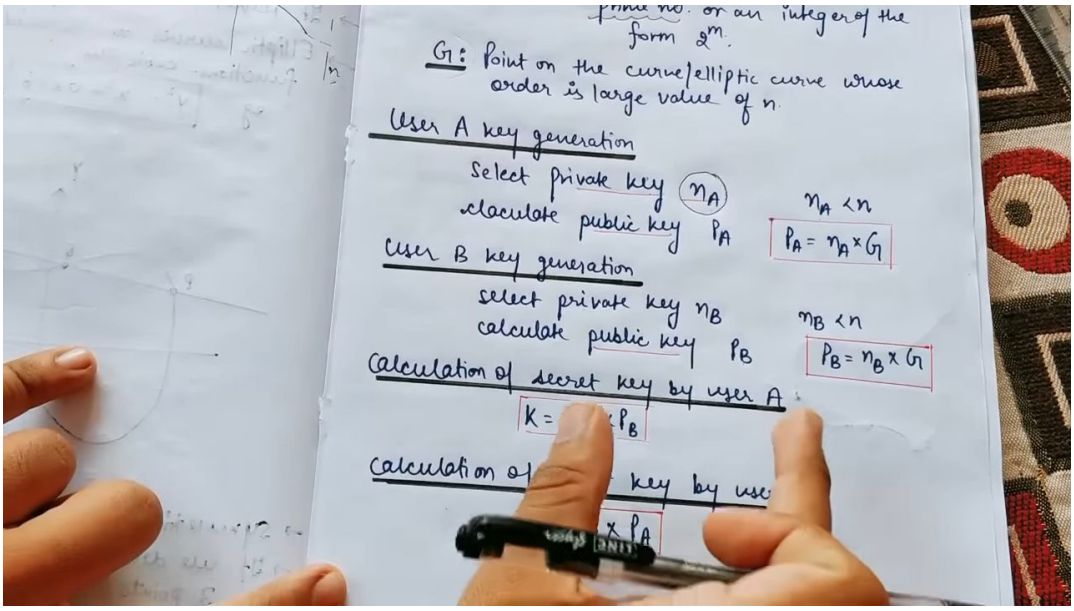
4:18


ECC - ALGORITHM
ECC - Key Exchange
Global Public Elements
$E_q(a,b)$: elliptic curve with parameters a, b and q
prime no. or an integer of the form 2^m.
G: Point on the curve/elliptic curve whose order is large value of n.
User A key generation
Select private key n_A $n_A < n$
calculate public key P_A $P_A = n_A \times G$

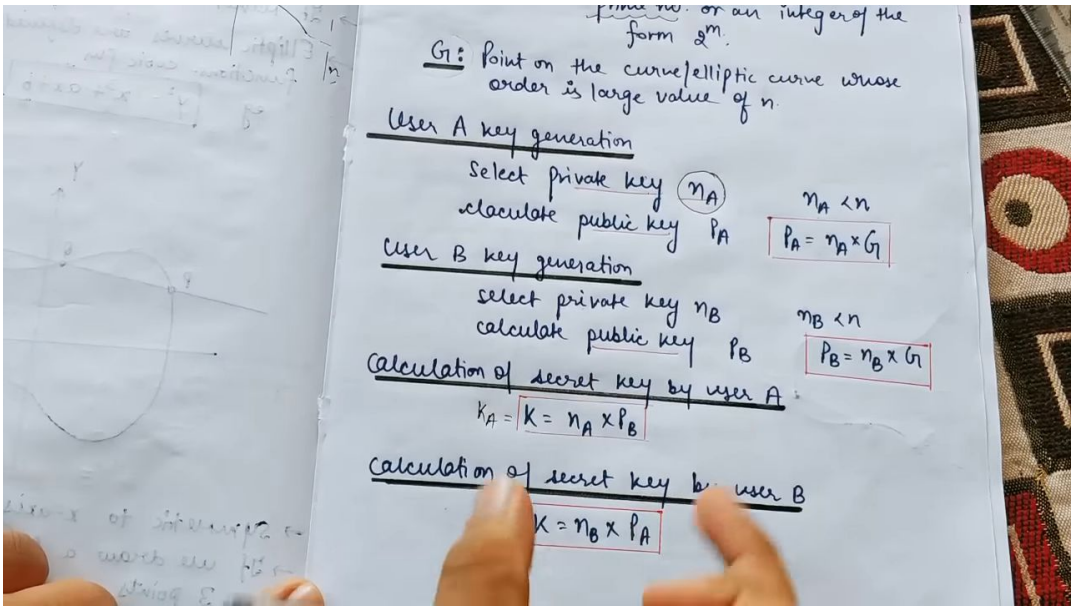
5:48


$E_q(a,b)$: elliptic curve with parameters a, b and q
prime no. or an integer of the form 2^m.
G: Point on the curve/elliptic curve whose order is large value of n.
User A key generation
Select private key n_A $n_A < n$
calculate public key P_A $P_A = n_A \times G$
User B key generation
select private key n_B $n_B < n$
calculate public key P_B $P_B = n_B \times G$
Calculation of

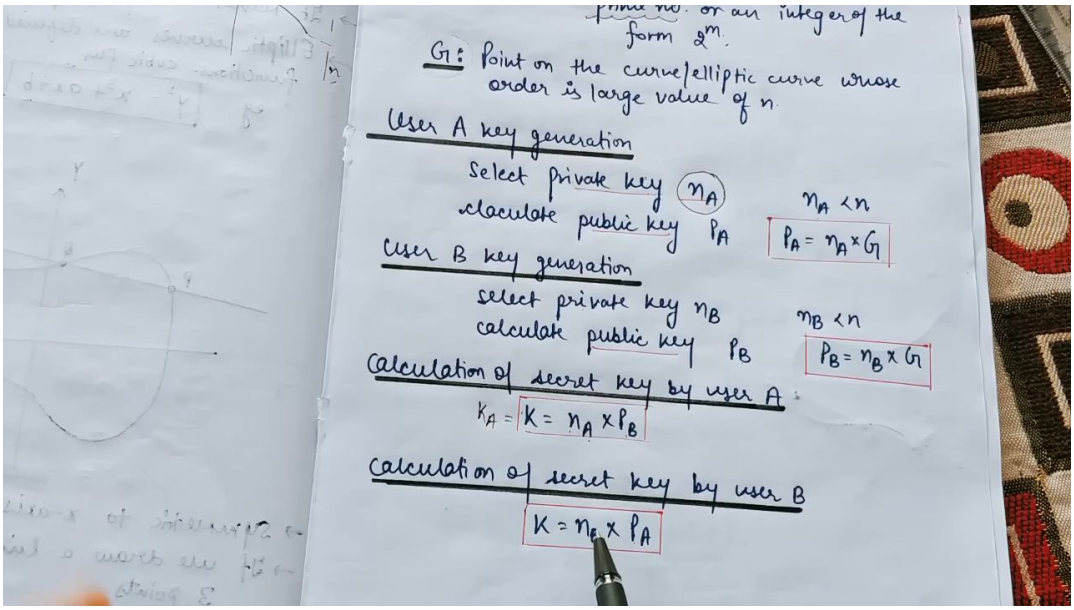
7:19


form 2^m.
G: Point on the curve/elliptic curve whose order is large value of n.
User A key generation
Select private key n_A
$n_A < n$
calculate public key P_A
$P_A = n_A \times G$
User B key generation
select private key n_B
$n_B < n$
calculate public key P_B
$P_B = n_B \times G$
Calculation of secret key by user A
$K = \times P_B$
Calculation of key by use
$\times P_A$

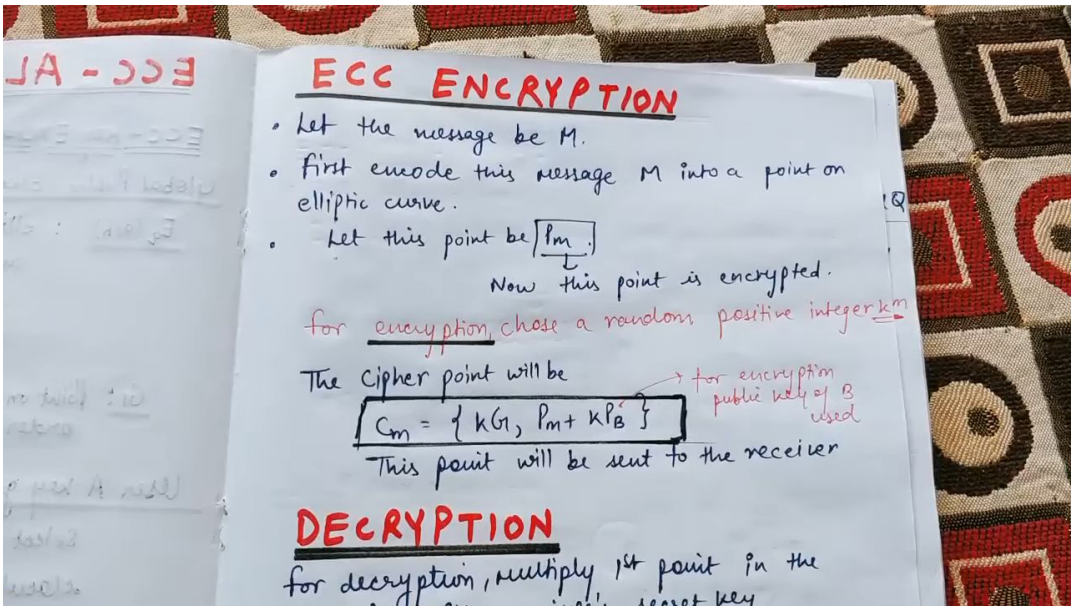
8:42


form 2^m.
G: Point on the curve/elliptic curve whose order is large value of n.
User A key generation
Select private key n_A
$n_A < n$
calculate public key P_A
$P_A = n_A \times G$
User B key generation
select private key n_B
$n_B < n$
calculate public key P_B
$P_B = n_B \times G$
Calculation of secret key by user A
$K_A = K = n_A \times P_B$
Calculation of secret key b user B
$K = n_B \times P_A$

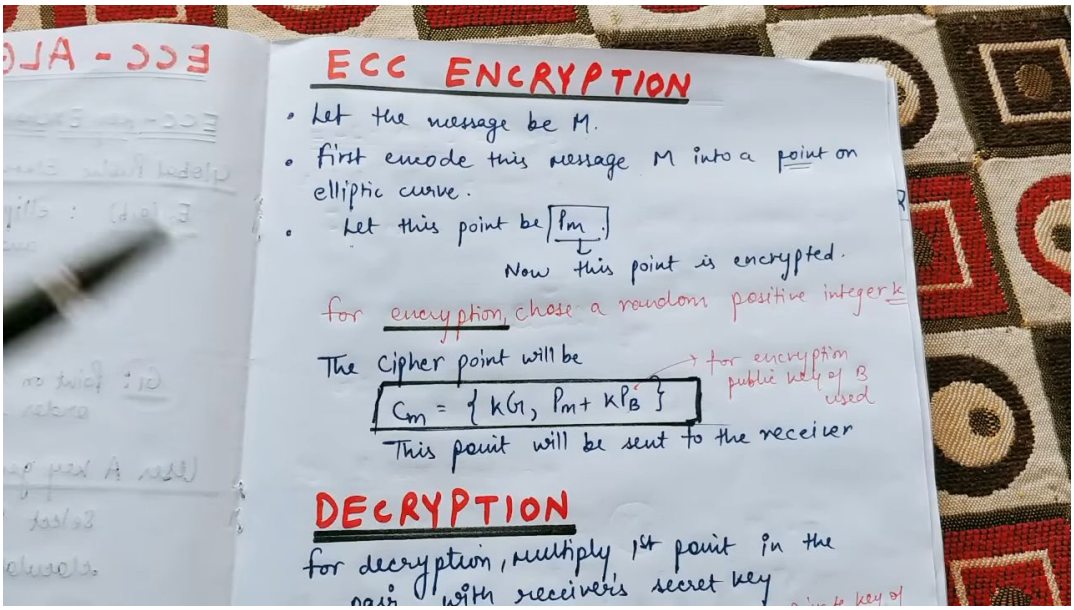
8:54


form 2^m.
G: Point on the curve/elliptic curve whose order is large value of n.
User A key generation
Select private key n_A
$n_A < n$
calculate public key P_A
$P_A = n_A \times G$
User B key generation
select private key n_B
$n_B < n$
calculate public key P_B
$P_B = n_B \times G$
Calculation of secret key by user A
$K_A = K = n_A \times P_B$
Calculation of secret key by user B
$K = n_B \times P_A$

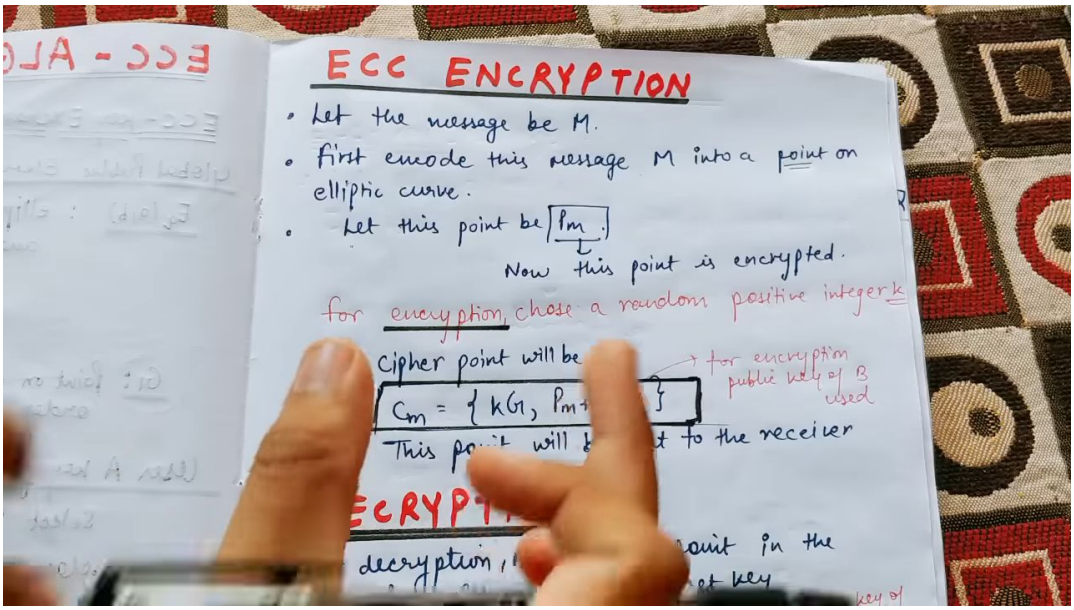
9:36


ECC ENCRYPTION
Let the message be M.
First encode this message M into a point on elliptic curve.
Let this point be P_m.
Now this point is encrypted.
for encryption, chose a random positive integer k_m
The cipher point will be
$C_m = \{kG, P_m + kP_B\}$
for encryption public key of B used
This point will be sent to the receiver
DECRYPTION
for decryption, multiply 1st point in the

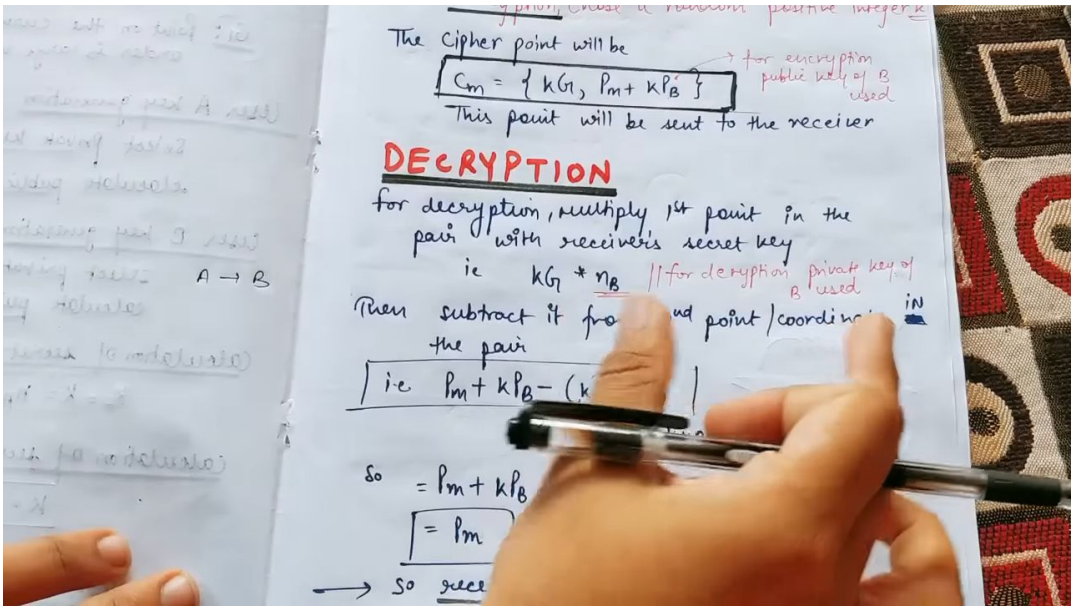
10:12


ECC ENCRYPTION
Let the message be M.
First encode this message M into a point on elliptic curve.
Let this point be Pm.
Now this point is encrypted.
for encryption, chose a random positive integer k
The cipher point will be
Cm = { kG, Pm + kPB }
for encryption public key of B used
This point will be sent to the receiver
DECRYPTION
for decryption, multiply 1st point in the

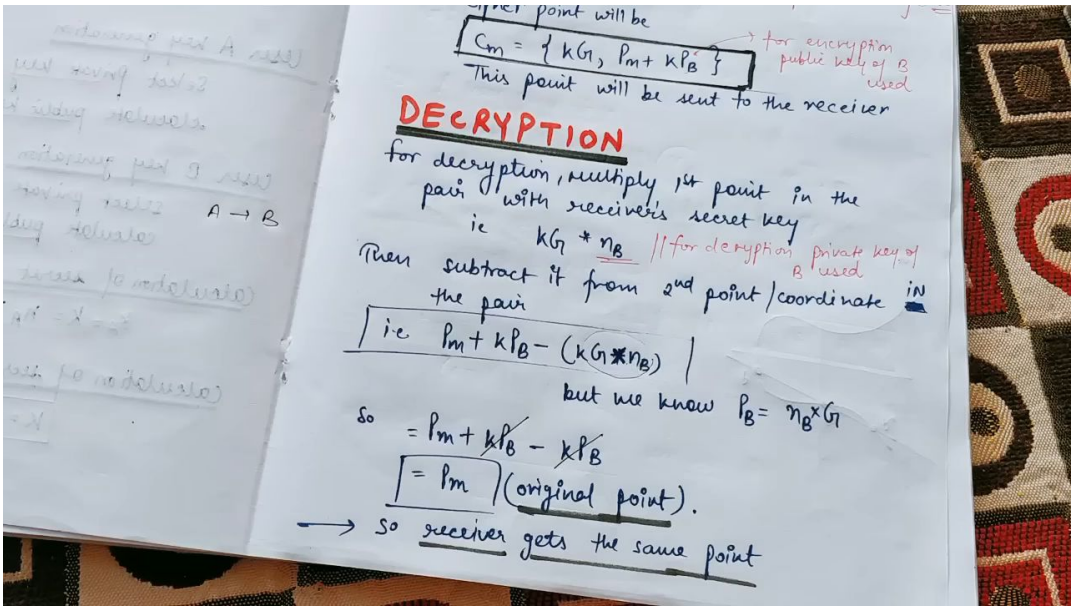
11:27


ECC ENCRYPTION
Let the message be M.
First encode this message M into a point on elliptic curve.
Let this point be Pm.
Now this point is encrypted.
for encryption, chose a random positive integer k
Cipher point will be
for encryption public key of B used
to the receiver
decryption,

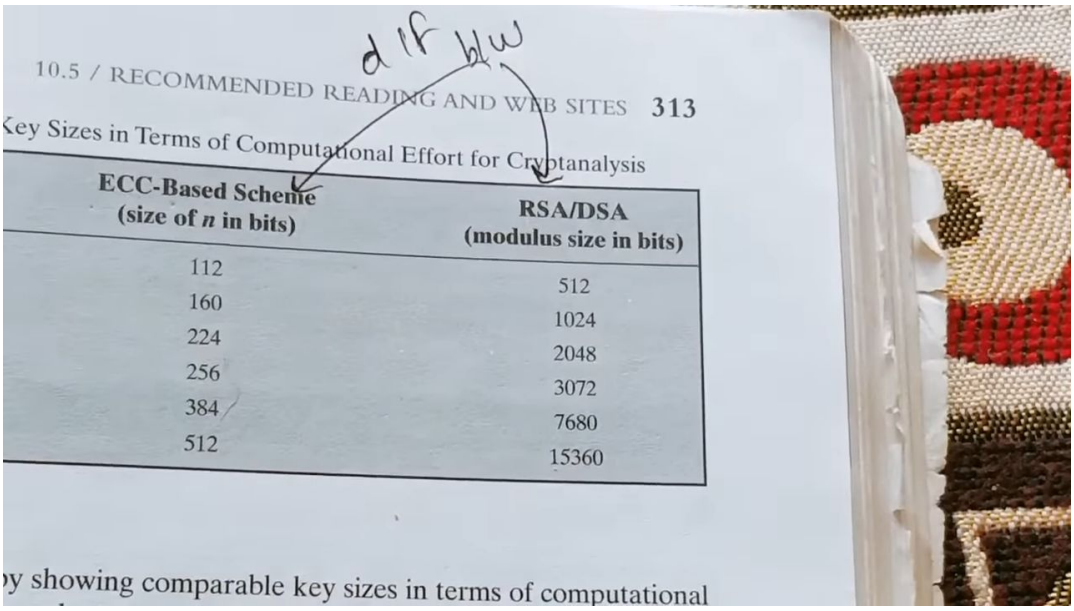
11:30


The Cipher point will be
$C_m = \{kG, P_m + kP_B\}$
for encryption public key of B used
This point will be sent to the receiver
DECRYPTION
for decryption, multiply 1st point in the pair with receiver's secret key
ie $kG * n_B$ //for decryption private key of B used
Then subtract it from 2nd point/coordinate in the pair
ie $P_m + kP_B - (k$
so $= P_m + kP_B$
$= P_m$
so

12:06


Cipher point will be
$C_m = \{kG, P_m + kP_B\}$
for encryption public key of B used
This point will be sent to the receiver
DECRYPTION
for decryption, multiply 1st point in the pair with receiver's secret key
ie $kG * n_B$ //for decryption private key of B used
Then subtract it from 2nd point/coordinate in the pair
ie $P_m + kP_B - (kG * n_B)$
but we know $P_B = n_B \times G$
so $= P_m + kP_B - kP_B$
$= P_m$ (original point).
so receiver gets the same point

13:00



Key Sizes in Terms of Computational Effort for Cryptanalysis

ECC-Based Scheme (size of n in bits)	RSA/DSA (modulus size in bits)
112	512
160	1024
224	2048
256	3072
384	7680
512	15360

by showing comparable key sizes in terms of computational

13:29