Q1:
Plaintext: VuNguyens4010423@student.
01010110 01110101 01001110 01100111 01110101 01111001 01100101 01101110 01110011 00110100 00110000 00110001 00110000 00110100 00110010 00110011 01000000 01110011 01110100 01110101 01100100 01100101 01101110 01110100 00101110

1. θ mapping

C[x] Operation

01010110	01110101	01001110	01100111	01110101
01111001	01100101	01101110	01110011	00110100
00110000	00110001	00110000	00110100	00110010
00110011	01000000	01110011	01110100	01110101
01100100	01100101	01101110	01110100	00101110

C[0] = A[0][0] ⊕ A[0][1] ⊕ A[0][2] ⊕ A[0][3] ⊕ A[0][4]

01010110 ⊕ 01111001 = 00101111
00101111 ⊕ 00110000 = 00011111
00011111 ⊕ 00100111 = 00111000
00111000 ⊕ 01100100 = 01011100

C[0] = 01011100

C[0]	C[1]	C[2]	C[3]	C[4]
01011100	00010100	00011101	00100000	00110111

D[x] = operation

$$\begin{aligned} C[x] &= A[x,0] \oplus A[x,1] \oplus A[x,2] \oplus A[x,3] \oplus A[x,4] \\ D[x] &= C[x-1] \oplus \mathrm{rot}(C[x+1],1) \\ A[x,y] &= A[x,y] \oplus D[x] \end{aligned}$$

D[0] = C[4] ⊕ rot(C[1], 1)
D[0] = 00110111 ⊕ rot(00010100, 1)
D[0] = 00110111 ⊕ 00101000
D[0] = 00011111

D[0]	D[1]	D[2]	D[3]	D[4]
00011111	01100110	01010100	01110011	10011000

A[x,y] operation

A[x,y] = A[x,y] ⊕ D[x]

01010110	01110101	01001110	01100111	01110101
01111001	01100101	01101110	01110011	00110100
00110000	00110001	00110000	00110100	00110010
00110011	01000000	01110011	01110100	01110101
01100100	01100101	01101110	01110100	00101110

D[0]	D[1]	D[2]	D[3]	D[4]
00011111	01100110	01010100	01110011	10011000

A[x,y]	0	1	2	3	4
0	01001001	00010011	00011010	00010100	11101101
1	01100110	00000011	00111010	00000000	10101100
2	00101111	01000111	01100100	00010111	10111010
3	00111000	00100110	00110111	00000110	11110010
4	01111011	00000011	00111010	00000111	10110110

2. ϱ mapping

	x = 3	x = 4	x = 0	x = 1	x = 2
y=2	25	39	3	10	43
y=1	55	20	36	44	6
y=0	28	27	0	1	62
y=4	56	14	18	2	61
y=3	21	8	41	45	15

A[x,y]	0	1	2	3	4
0	01001001	00010011	00011010	00010100	11101101
1	01100110	00000011	00111010	00000000	10101100
2	00101111	01000111	01100100	00010111	10111010
3	00111000	00100110	00110111	00000110	11110010
4	01111011	00000011	00111010	00000111	10110110

A[0,2] = 00101111
ϱ[0,2] = 3

3 mod 8 = 3

rotl(00101111 ,3) =01111001

A[0,1] = 01100110
ϱ[0,1] = 36

36 mod 8 = 4
rotl(01100110 ,4) =01100110

01001001	00100110	10000110	01000001	01101111
01100110	00110000	10001110	00000000	11001010
01111001	00011101	00100011	00101110	01011101
01110000	11000100	10011011	11000000	11110010
11101101	00001100	01000111	00000111	10101101

3. π Mapping

ϱ[x,y]	0	1	2	3	4
0	01001001	00100110	10000110	01000001	01101111
1	01100110	00110000	10001110	00000000	11001010
2	01111001	00011101	00100011	00101110	01011101
3	01110000	11000100	10011011	11000000	11110010
4	11101101	00001100	01000111	00000111	10101101

ϱ[x,y]	0	1	2	3	4
0	ϱ[0,0]	ϱ[1,0]	ϱ[2,0]	ϱ[3,0]	ϱ[4,0]
1	ϱ[0,1]	ϱ1,1]	ϱ[2,1]	ϱ[3,1]	ϱ[4,1]
2	ϱ[0,2]	ϱ[1,2]	ϱ[2,2]	ϱ[3,2]	ϱ[4,2]
3	ϱ[0,3]	ϱ[1,3]	ϱ[2,3]	ϱ[3,3]	ϱ[4,3]
4	ϱ[0,4]	ϱ[1,4]	ϱ[2,4]	ϱ[3,4]	ϱ[4,4]

$\pi(x, y) = \varrho[y][(2x + 3y) \bmod 5]$

$\pi(0, 0) = \varrho[0][(2*0+ 3*0) \bmod 5]$
$\pi(0, 0) = \varrho(0,0)$

$\pi(1, 0) = \varrho[0][(2*1 + 3*0) \bmod 5]$
$\pi(1, 0) = \varrho(0,2)$

π[x,y]	0	1	2	3	4
0	ϱ[0,0]	ϱ[0,2]	ϱ[0,4]	ϱ[0,1]	ϱ[0,3]
1	ϱ[1,3]	ϱ[1,0]	ϱ[1,2]	ϱ[1,4]	ϱ[1,1]
2	ϱ[2,1]	ϱ[2,3]	ϱ[2,0]	ϱ[2,2]	ϱ[2,4]
3	ϱ[3,4]	ϱ[3,1]	ϱ[3,3]	ϱ[3,0]	ϱ[3,2]
4	ϱ[4,2]	ϱ[4,4]	ϱ[4,1]	ϱ[4,3]	ϱ[4,0]

π[x,y]	0	1	2	3	4
0	01001001	00110000	00100011	11000000	10101101
1	01000001	11001010	01111001	11000100	01000111
2	00100110	10001110	00101110	11110010	11101101
3	01101111	01100110	00011101	10011011	00000111
4	10000110	00000000	01011101	01110000	00001100

4. χ mapping

$$A[x,y] = B[x,y] \oplus ((\bar{B}[x+1,y]) \wedge B[x+2,y]) \quad , \quad x,y = 0,1,2,3,4$$

χ(0,0)=A[0][0] ⊕ (¬A[1][0]∧A[2][0])
01001001⊕ (¬00110000∧10000110)
01001001⊕ (11001111∧10000110)
01001001⊕10000110
χ(0,0) = 01001010

χ(1,0) = A[1][0]⊕(¬A[2][0]∧A[3][0])
00110000⊕(¬10000110∧11000000)
00110000⊕(01111001∧11000000)
00110000⊕01000000
χ(1,0) = 11110000

χ[x,y]	0	1	2	3	4
0	01001010	11110000	00001110	10000000	10011101
1	01110000	01001110	01111010	11000100	11001101
2	00000110	01011110	00100011	11110000	01100101
3	01110110	11100100	00011001	11110011	00000111
4	11011011	00100000	01010001	11110010	00001100

5. ι mapping

ι[x,y]	0	1	2	3	4
0	01001011	11110000	00001110	10000000	10011101
1	01110000	01001110	01111010	11000100	11001101
2	00000110	01011110	00100011	11110000	01100101
3	01110110	11100100	00011001	11110011	00000111
4	11011011	00100000	01010001	11110010	00001100

Q2:
4010423/K7MDENG+bPxRfiCYEXAMPLEKEY

(1) Compute kDate = HMAC("AWS4" + kSecret, Date), where Date = 20250415;
= b6d4cdb2c0b2e13044c80664fea693b4fed4fbbd3d8f59be65347741e244fbf9

(2) Compute kRegion = HMAC(kDate, Region), where Region = us-east-1;
= c1c02d715dba549f705efaa5f4da069ba6e974084e5ed70c0853e8dfd9a438dd

(3) Compute kService = HMAC(kRegion, Service), where Service = iam;
= 453ba3f807bea11837fa6aa570325f454d5088356876f07862cbddbd1a836d1e

(4) Compute kSigning = HMAC(kService, "aws4_request");
=53798ddbbcea393d0f3aefcfd1213f3d7c88384a565499508c294cc1dd905e6c

(5) Compute the signature = HexEncode(HMAC(kSigning, string to sign))
HexEncode(952e077a9eb0811daefe6892a7c2a41c26a31c0773aaf4386a3130eb2b88917b)

39353265303737613965623038313164616566653638393261376332613431633236613331
63303737336161663433383661333133306562326238383931376 2

Q3:

Keys and timestamp:

KC = 4feb87a2b9f75bcdc41d6828fd2a30e9 KS = 669c45b1684463c09e3acc8ff30fc63e nC = 39ffe3021bcad0da66bc4efc46ef0360 timestamp = 12/05/2025, 20:54:19

Ticket:

0bb0e591f781c6a0ab9e4c54194e8c88737c35676188db42ebc95ca1157ceb47392cbe6c103
53a9217d0ff699c0237e7

Authenticator:

af03bd58ac4d14fb3045db3fb7784493e70255049a765d68df28a09023efe36fcc9d39eb47aef
86dc6272e9460048325b88fa4537480059e453f61e393c9d2e7

Authenticator Decrypted:

39ffe3021bcad0da66bc4efc46ef036012/05/2025, 20:54:19

Q4:
generate random number

Random number a:

1040059817756673402666754794409345513425972868745

Random number b:

1284360465632072989599693186144990967421176239637

Sha1 Encryption

bf7d28cef3fa72713e0f9393811259ca042adf30

y=g*(mod p) output

1633059087313385488847602398570465830163027295560838825208009778033473125
0519202220590975151050725377922854233984187939246704594474082563212255176
0915417969851537581048769102105487491767608960390238936581929673402743856
9995195759571429522780399029116472228653241758975276157407542010613796142
61117199026503062

g^A (VPC public):

1689933096823421555723015901206854882164783465217727159546561772366392300
6385307435935089020764131776015872403095032211707290499292253396566183222
9874115453783064905654448838176428058468794659768554255122619371661350601
6711637059159425168461024414309656423235606907808666179175046539772157306
89071869935637374

g^B (Data Centre public):

1922441164864400450590964729889440897828385885056734340424359549598580025
70776322429427749683498726569103380370497880035539852257085411193204887967
1867495809656139021800363075231805602297587014300880639391446721507461053
5820257370733705677650872864392425449905260042739601125777749495696363512
920341025749197

Shared Secret A:

7057918281325909701577721064229116853498415218010320047538055463760378174
0938686478974104577342577217289495662579296625178493533300360725411848431
8753792438344556667466372000720374843413215039900370528404915188674848429
5430702228802219021598184160440847522756541609157635702607885733613536692
1074887839134135

Shared Secret B:

7057918281325909701577721064229116853498415218010320047538055463760378174
0938686478974104577342577217289495662579296625178493533300360725411848431
8753792438344556667466372000720374843413215039900370528404915188674848429
5430702228802219021598184160440847522756541609157635702607885733613536692
1074887839134135

Q5:
1)
e= s4010423 = 3D31B7

Ephemeral key =
B3D72C1F8EAD6B4AB9A39D531CE1EF2175A7623D0924BAF95358B028369D4ACF
5ED9D2C0A88D7E94E6CE41BE349B4D67E2C108D1AEAA3935371B7A3CF4C2A23C
1195D184B1B8FA32BFA2A75888EECFBD2D11E91A0C6ED79A1E6B8B730BCDA3FD
43C9A93BA2BB6974B81FC438C19D3AD4F6BB7DF31D4C8BA3F9913EF49C867B51

2)
Ephemeral key + e =
B3D72C1F8EAD6B4AB9A39D531CE1EF2175A7623D0924BAF95358B028369D4ACF5ED
9D2C0A88D7E94E6CE41BE349B4D67E2C108D1AEAA3935371B7A3CF4C2A23C1195D1
84B1B8FA32BFA2A75888EECFBD2D11E91A0C6ED79A1E6B8B730BCDA3FD43C9A93B
A2BB6974B81FC438C19D3AD4F6BB7DF31D4C8BA3F9913EF49C867B513D31B7

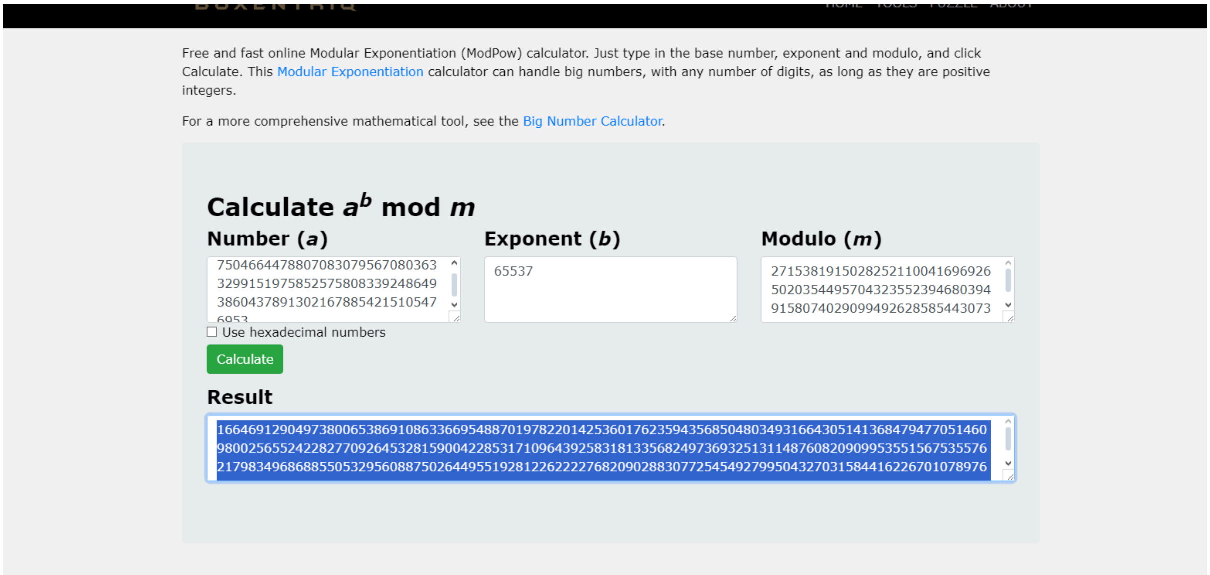
After hashed
Cfa64e36a57718a7ffa84c9227ec60f506286b94aaf455ace9fcbea0bf87029c

a^b mod m =
96aa0ab0f5eaf060e61f925bc8e713fa166b0ec9eff9c66d697079a68dca59b04ac16bfe760de5451cf5bc7ba6312d63e1c8b018bc8329000ed45919989ae18a454f58418a3814b456c2aa68172482bb7ed21e8e4fd41bd479f791e099109a9a1aa9eb4fb1f4068739dd756f740fe0862adeffb6dd0fdff0706d7c209d0ab21cd55857ad1b3e67874b79179e2d92a75b6d7d2ec7979c2b229de37273f70aacf3d6c45f80eabeda6b9d35387e6ef097d6b6b5ba273cf6f880bcc917a4b51e0a275a4b69af2d5d6016abef5acdf12dbf77f957983bb76325c2fbb36a2df482121a1387345dfb6828aa9615a4f287f6580f7815c494a070d417367104dd2d8e2dd6

3)Pre_master_secret= s4010423@student.rmit.edu.au =
5b6295b95611541897ef2591f488f94b083a892d9e75cf533e23636533e2c334b93ab469e8e3c7b7f6586a9d27777959

Converted to decimal

140654537595593535731368052775046644788070830795670803633299151975852575808339248649386043789130216788542151054769 53



166469129049738006538691086336695488701978220142536017623594356850480349316643051413684794770514609800256552422827709264532815900422853171096439258318133568249736932513114876082090995355156753557621798349686885505329560887502644955192812262222768209028830772545492799504327031584416226701078976154294581789387520327401576239775824033875835059702809448310507367629770767291199932185658247778201662151924311055150350768183527726349909996580521572358488103526989942563372052384040060097567168180333889059586010162096770873038412640576153817043603601788745813670376883388054140605740492740390253684289496457124895646789 36

4)In the SSL handshake protocol, the authentication is achieved through using a digital certificate. When a user connects, it verifies the certificate and confirming the server’s identity

5) Forward security ensures that if a private key of a server is breached and compromised, past communication will still be safe. This is possible through the usage of ephemeral key exchange