ADVANCED ENCRYPTION SYSTEM

Yasar Azimi, Shahzada Masood Asir, Maryam Zakare

AES

AUAF

Advanced Encryption Standard (AES)

History Behind AES

- Initiated by US NIST in 1997 for robust encryption standard
- Global call for algorithms; 15 submitted
- Rijndael algorithm chosen for its balance of security, performance, and flexibility

WHAT IS AES?

An algorithm that uses the same key to encrypt and decrypt protected data. Instead of a single round of encryption, data is put through several rounds of substitution, transposition, and mixing to make it harder to compromise.

HOW IT WORKS?

- Encryption transforms plaintext to ciphertext using a key
- Goal: Make ciphertext indecipherable without key
- Representation of data as binary sequences (0s and 1s)

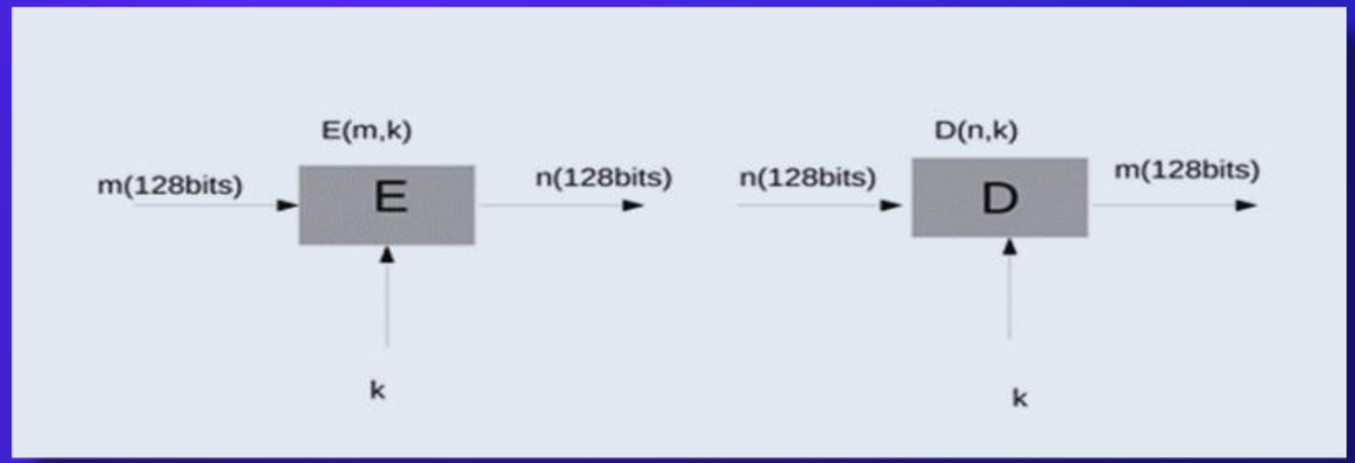
IMPORTANCE OF AES

- Used for encrypting hard drives and securing internet communications
- Integral in modern processors for efficient operation
- Concept of plaintext, ciphertext, and key
- All represented as sequences of bits
- Use of a randomly-generated key
- Bitwise exclusive or operation

UNDERSTANDING AND IMPLEMENTING AES

- The project outlines AES in five steps: substitution, shifting, and mixing for enhanced cryptographic strength. A flowchart and code snippets illustrate the process succinctly.

- AES a robust symmetric block cipher replacing DES.
- With key lengths of 128, 192, or 256 bits, AES excels in security and speed, playing a crucial role in applications like VPNs and web browsers;
- With encryption rounds varying:
- 10 for 128-bit
- 12 for 192-bit,
- 14 for 256-bit keys



ADD ROUND KEY:

- In this critical phase, each byte of the state matrix is XORed with the corresponding byte from the round key, enhancing security by integrating the derived round key into the state matrix. This process involves XORing each byte of the state matrix with its counterpart from the round key, derived from the original key through a key schedule or algorithm.

```
def add_round_key(state_matrix, round_key):
    for i in range(4):
        for j in range(4):
            state_matrix[i][j] ^= round_key[i][j]
```

SUBSTITUTE BYTES:

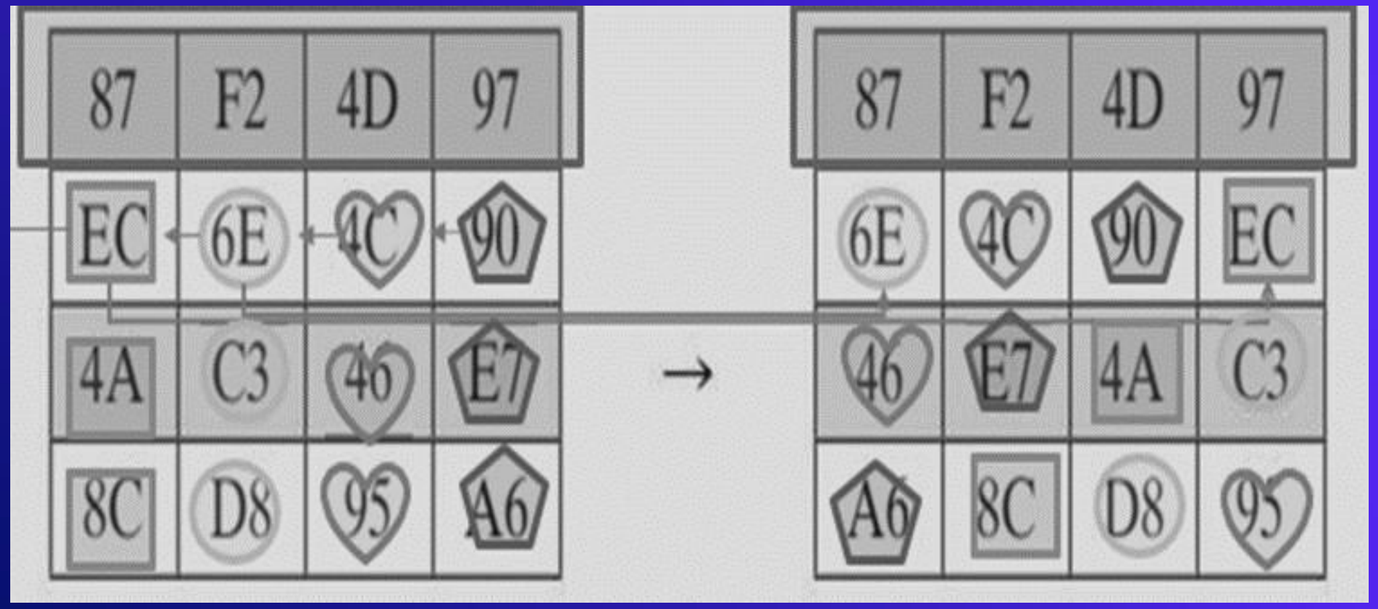
- In a key phase, each byte in the state matrix is substituted with a value from the S-Box, a 16 x 16 matrix encompassing all 256 8-bit values. This introduces non-linearity, enhancing the algorithm's resilience against cryptographic attacks and contributing significantly to the encryption process's robustness.
- A)Leftmost 4 bits of the byte à Row value
- B)Rightmost 4 bits of the byte à column value

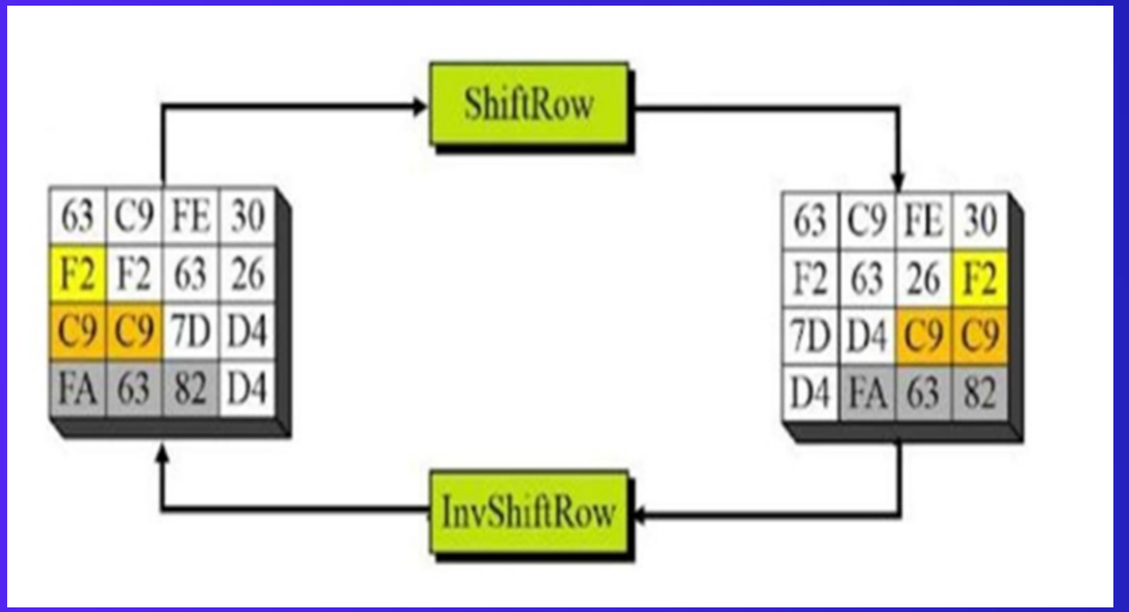
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	82	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	89	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	84	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	87	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	38	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

SHIFT ROWS:

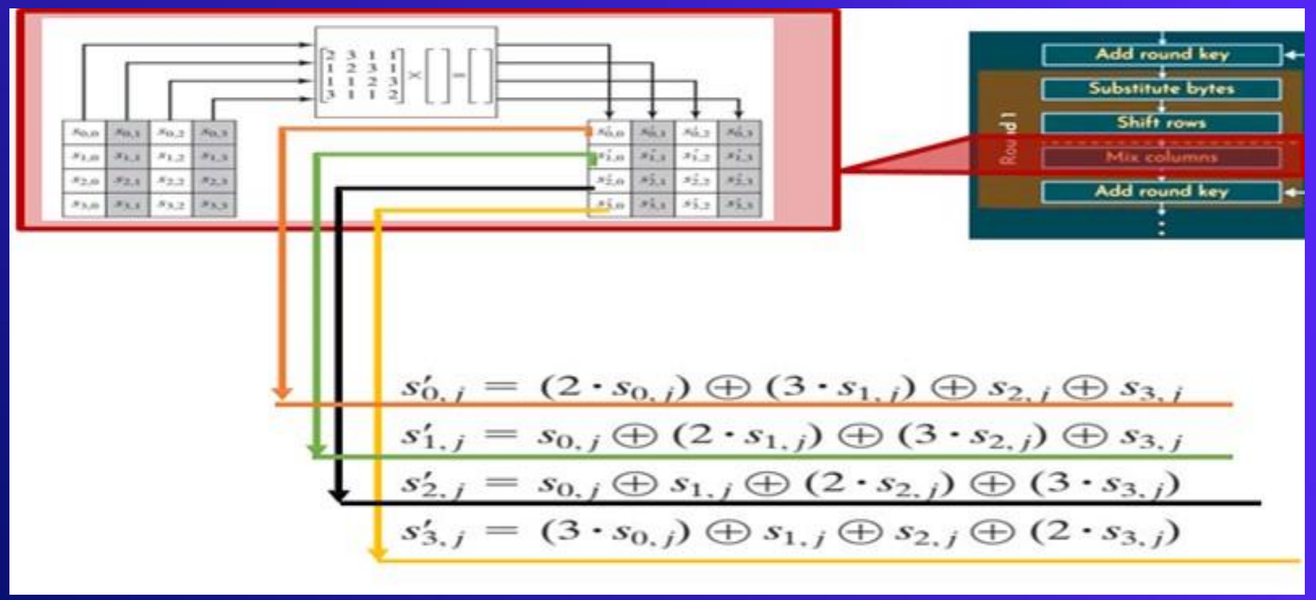
- In this step, rows in the state matrix shift left with varying offsets, ensuring each byte influences multiple bytes in the next round for enhanced diffusion and algorithmic security.
- In decryption, the InvShiftRows transformation reverses the original leftward shifts: the second row shifts one position right, the third row shifts two positions right, and the fourth row shifts three positions right. This ensures an accurate retrieval of the plaintext by reversing the shift row transformation.

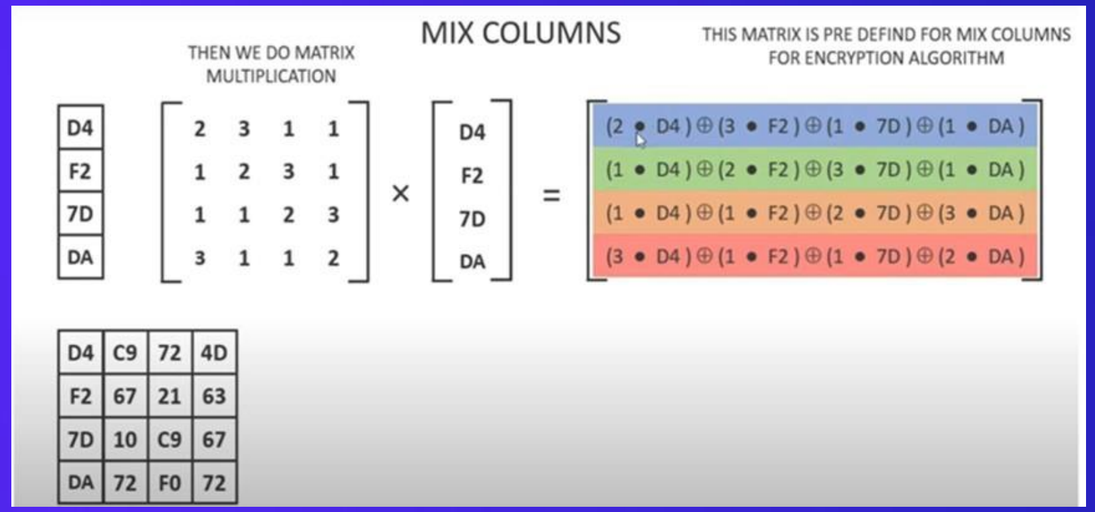




MIX COLUMNS:

- This stage mixes and diffuses values in each column of the state matrix through fixed matrix multiplication, treating each column as a polynomial in the Galois Field. It adds to the algorithm's security with both forward and inverse transformations.
- Mix Columns Forward Transformation:
- a)Each byte of a column is mapped to a new value that is a function of all four bytes in that column.
- b)The transformation can be defined by the matrix multiplication on the state matrix.





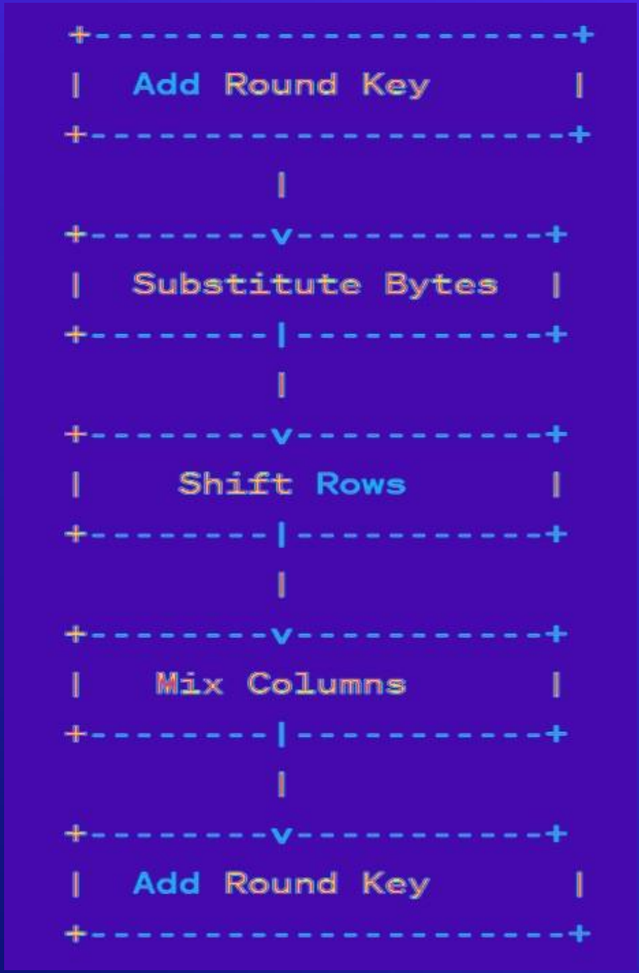
ADD ROUND KEY:

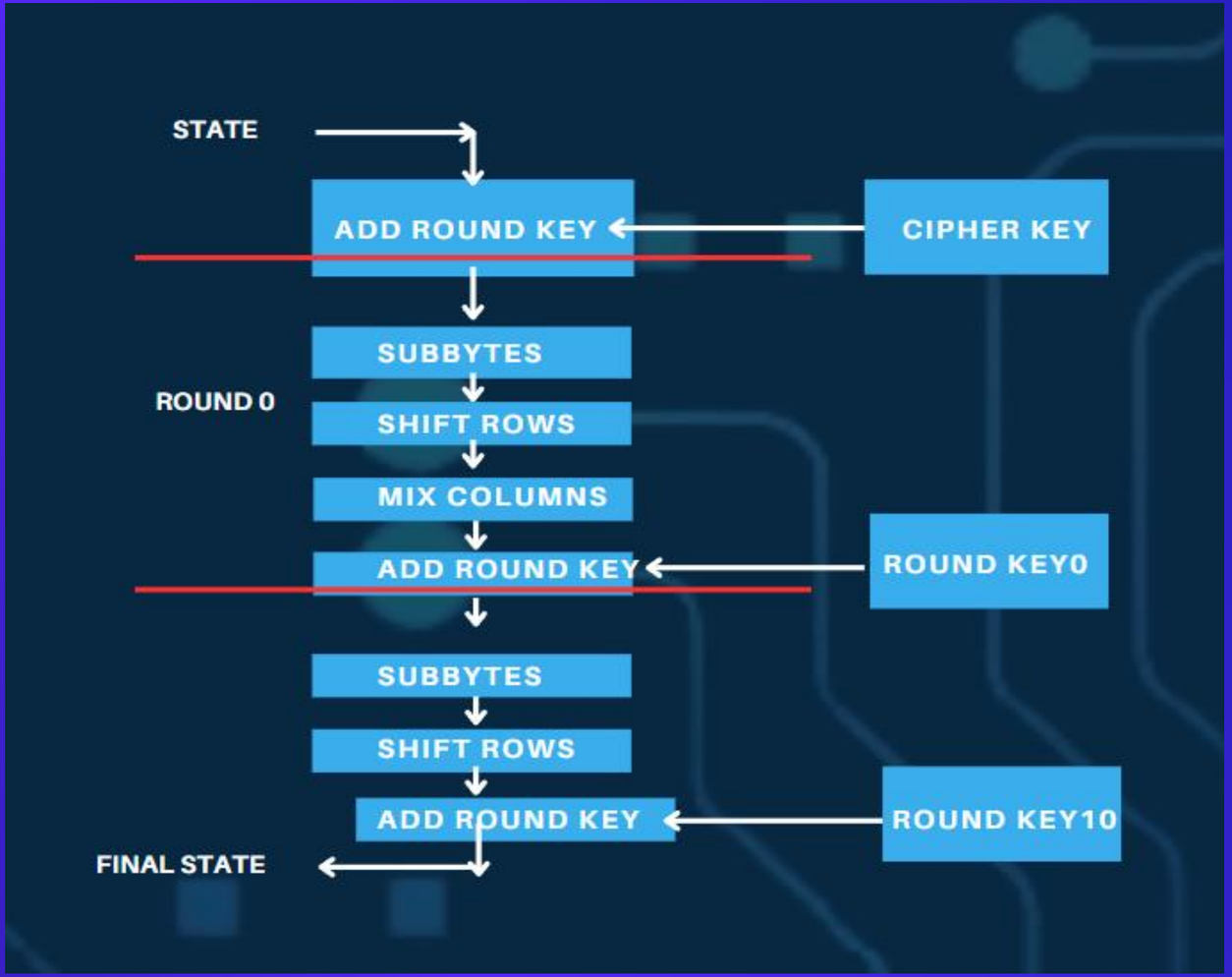
- The last key round doubly influences each state matrix byte for added security, involving XOR as in Step one Both forward and inverse transformations ensure a comprehensive and reversible cryptographic process.

47 40 A3 4C		AC 18 28 57	=	EB 59 8B 1B
37 D4 70 9F	⊕	77 FA D1 5C		40 2E A1 C3
94 E4 3A 42		66 DC 29 00		F2 38 13 42
ED A5 A6 BC		F3 21 41 6A		1E 84 E7 D2

A	B	A+B
0	0	0
0	1	1
1	1	0
1	0	1

- Flow chart: This flowchart represents the logical flow of the AES single-round implementation.

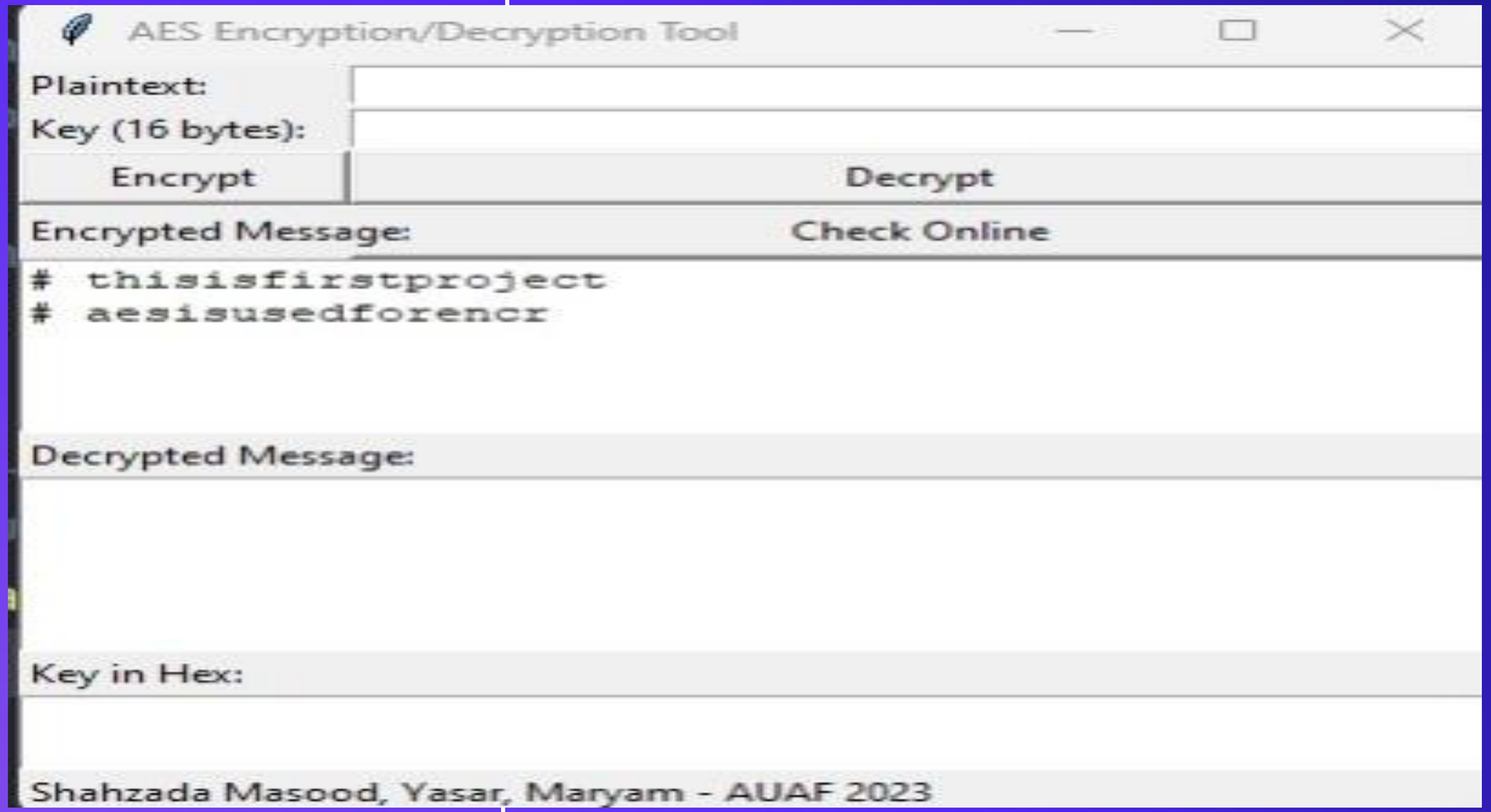




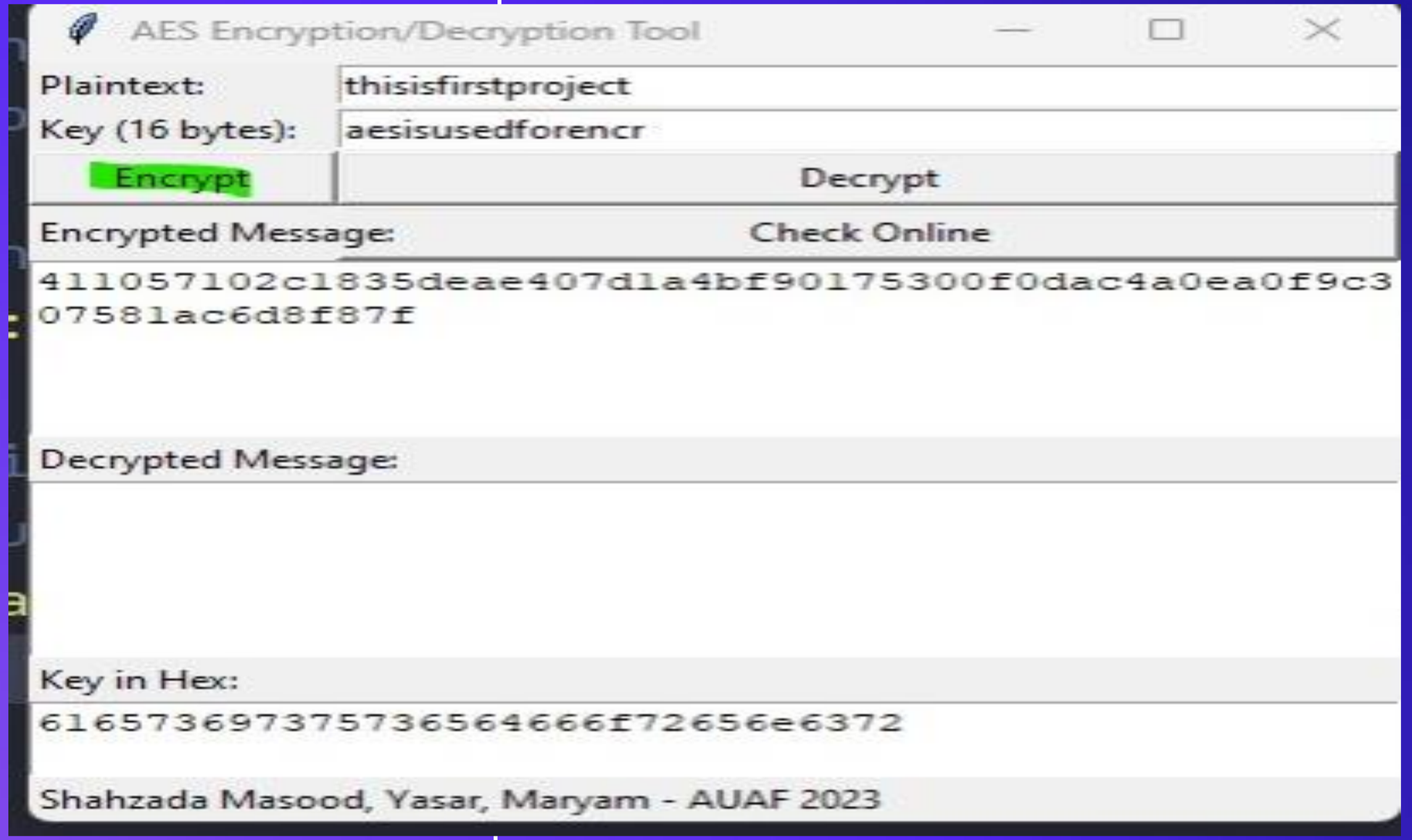
Software Objectives: how you create the software

- Importing necessary modules like subprocess, tkinter, and Cryptodome.Cipher.
- Defines functions pad_ISO10126 and unpad_ISO10126 for data padding and unpadding as per ISO 10126.
- Implements encrypt AES, ECB and decrypt AES, ECB for AES encryption and decryption in ECB mode.
- Contains encrypt message and decrypt message to handle encryption and decryption processes in the GUI.
- Features a GUI setup with Tk from tkinter for input fields, labels, buttons, and text areas.
- Includes a open website function to open a specified URL using subprocess.
- Sets up a GUI layout with widgets for plaintext, key input, and displaying encrypted and decrypted messages.
- Runs the GUI event loop with root.mainloop().

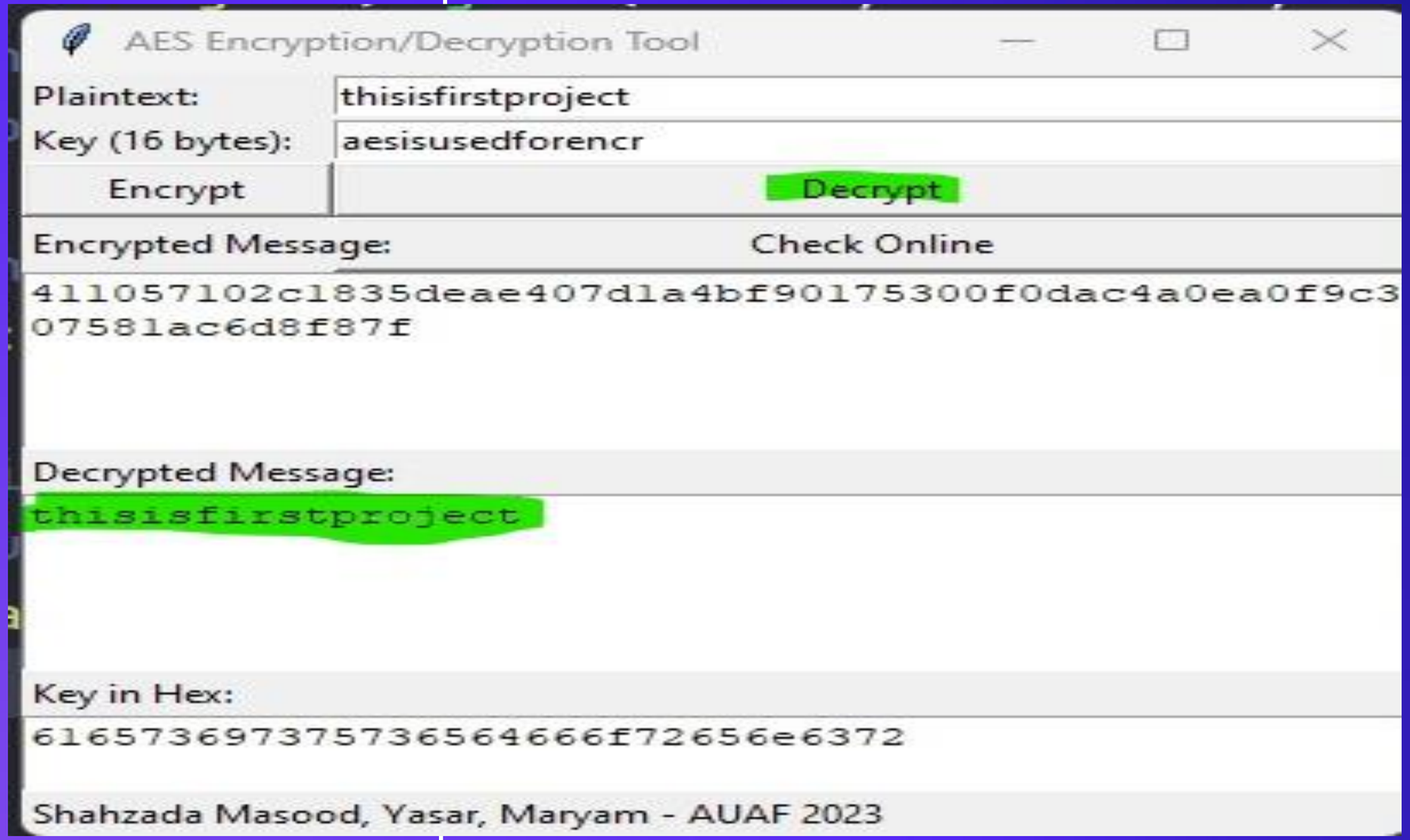
Professional GUI Interface



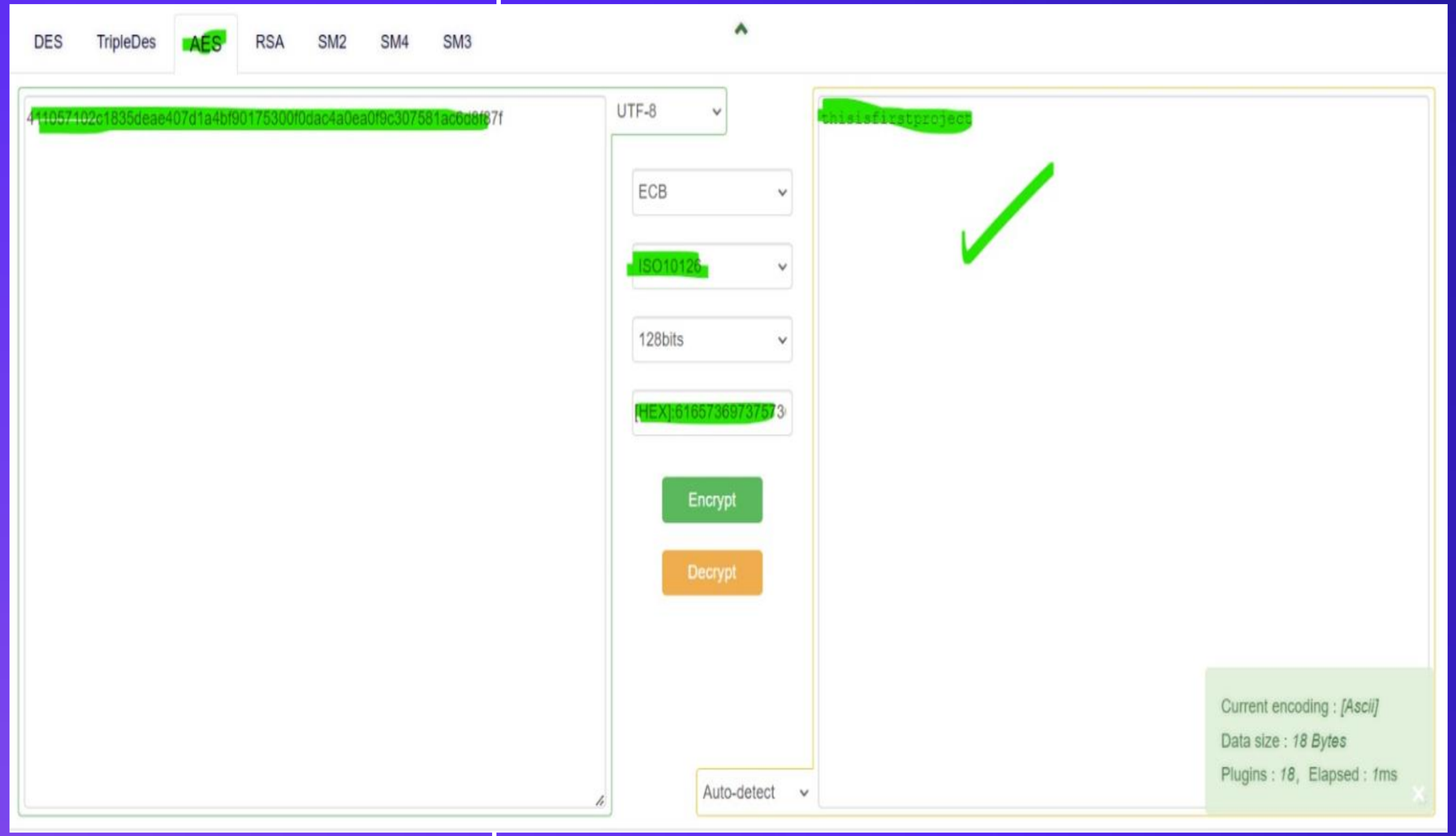
Professional GUI Interface



Professional GUI Interface



Checked online:



CONCLUSION:

01

- A single round of AES encryption involves key steps: Add Round Key, Substitute Bytes, Shift Rows, Mix Columns, and a final Add Round Key. These steps collectively introduce non-linearity, confusion, and diffusion into the process, enhancing security. The strength of AES lies in the combination of these steps across multiple rounds, creating a secure encryption scheme. Note that code snippets are illustrative, and a complete implementation requires additional functions and constants.

THANK YOU!