CS-369: Introduction to Cryptography and Network Security LAB

LAB Assignment IV

Course Instructor: Dr. Dibyendu Roy — Due: Dec 1, 2024, 11:59 pm

Instructions: Code must be written in C and well commented. Submission of code in any other file extension (.pdf, .docx etc) will not be considered. The file name of the code will be YOUR ROLL NO.c. Write Your Name and Roll Number on the top of your code. Your submission will not be considered if you submit late or email your submission. Note that the assignment is of 2 pages.

You need to implement the following protocol in C programming language.

1. Consider the prime number $p = 1223$ and the Elliptic curve EL: $y^2 = x^3 + 439x + 133$ over $\mathbb{Z}_p$.
2. Consider the point at infinity $\Theta = (0, 1)$.
3. Select a random point $\alpha(\neq \Theta)$ on the curve EL. This α needs to be obtained inside your code. Print α. (Output)
4. Alice and Bob have agreed on the same curve EL and the point α.
5. You code will ask for Alice's private key $n_A \in [1, 533]$ and Bob's private key $n_B \in [1, 533]$. (Input)
6. Using n_A and n_B Alice and Bob perform Diffie-Hellman key exchange on the curve EL with the point α and establish a shared secret key $SK = (x_1, y_1) \in \mathsf{EL}$. Print the SK. (Output)
7. Alice uses SHA-512 hash function and computes a key $K_A = \text{SHA-512}(x_1||y_1)$.
8. Bob uses SHA-512 hash function and computes a key $K_B = \text{SHA-512}(x_1||y_1)$.
9. Print $K_A = K_1||K_2$ and $K_B = K_1||K_2$ in the form of 64 bytes (space separated). (Output)
10. Program will ask for Alice's 128-bit message (say M_A). Input will be 16 space separated bytes in hexadecimal. For example : M_A = `00 11 22 33 44 55 66 77 88 99 aa bb cc dd ee ff`. (Input)
11. Alice will encrypt the given message M_A using Triple-AES$'-512$ bit encryption algorithm. Let the generated ciphertext be C_A. i.e., $C_A = \text{TEnc}_{\text{AES}'-512}(M_A, K_A)$.
12. $C_A = \text{TEnc}_{\text{AES}'-512}(M_A, K_A) = Enc_{AES'-256}\Big(Dec_{AES'-256}\big(Enc_{AES'-256}(M_A, K_1), K_2\big), K_1\Big)$.
13. Alice will generate a MAC for M_A using the described algorithm. The description of MAC is $\text{MAC}_A = \text{SHA-512}\Big(\big(K_A \oplus 125\big)||\text{SHA-512}\big((K_A \oplus 215)||M_A\big)\Big)$. Here the constants are in decimal.
14. Your program will display the ciphertext C_A and MAC_A in the form of bytes (space separated). (Output).
15. Alice will pass the ciphertext C_A, MAC_A to Bob. This will be passed inside your code.
16. Bob will decrypt C_A using Triple-AES$'-512$ bit decryption algorithm with his key K_B. Let the decrypted text be M_B. i.e., $M_B = \text{TDec}_{\text{AES}'-512}(C_A, K_B)$.
17. Bob will generate $\text{MAC}_B = \text{SHA-512}\Big(\big(K_B \oplus 125\big)||\text{SHA-512}\big((K_B \oplus 215)||M_B\big)\Big)$. Here the constants are in decimal.
18. Your program will display M_B and MAC_B in the form of bytes (space separated). (Output)

If your code is correct ! then $K_A = K_B$, $M_A = M_B$ and $\text{MAC}_A = \text{MAC}_B$ for every possible inputs.

Consider the below defined Subbyte function Sub (given in Figure 1) as in AES. Using the Sub (Figure 1)

	Y															
X	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

Figure 1: AES-Subbytes (Sub)

define a Subbyte′ function Subbyte′ : $\{0,1\}^8 \to \{0,1\}^8$ as per the following rule,

$$\text{Subbyte}'(x) = Sub((211 * x) + 122). \tag{1}$$

Here $+, *$ are the two binary operations of the field $\mathbb{F}_2[x]/ < x^8 + x^4 + x^3 + x + 1 >$. Using this perform the subbyte computation of AES′-256.
Consider the following matrix M (given in Equation (2), in decimal) instead of original mixcolumn matrix of AES.

$$M = \begin{bmatrix} 1 & 4 & 4 & 5 \\ 5 & 1 & 4 & 4 \\ 4 & 5 & 1 & 4 \\ 4 & 4 & 5 & 1 \end{bmatrix} \tag{2}$$

The inverse of M is given in Equation (3) (in decimal).

$$M^{-1} = \begin{bmatrix} 165 & 7 & 26 & 115 \\ 115 & 165 & 7 & 26 \\ 26 & 115 & 165 & 7 \\ 7 & 26 & 115 & 165 \end{bmatrix} \tag{3}$$

Using M we will perform our Mixcolumn′ operation in AES′-256. We will use the same key-scheduling as in AES-256 encryption algorithm with the new subbyte′ function inside the Subword function of the key scheduling algorithm.