USN | | | | | | | | | | |

INTERNAL ASSESSMENT TEST – III

Sub:	CRYPTOGRAPHY							Code:	21EC642
Date:	31/07/ 2024	Duration:	90 mins	Max Marks:	50	Sem:	VI	Branch:	ECE

Answer any 5 full questions

		Marks	CO	RBT
1	Describe RSA encryption and Decryption algorithm. Which are the five possible approaches to attack the RSA algorithm?	[10]	CO2	L2
2	Consider a Diffie Hellman scheme with a common prime q=11 and primitive root 2, (a) Show that 2 is a primitive root of 11. (b) If user A has public key Y_A =9, what is X_A the private key? (c) If user B has public key Y_B= 5,what is X_B private key?	[10]	CO2	L3
3	Explain the Man-In-The-Middle Attack? Define primitive root with example.	[10]	CO3	L3
4	Write a note on how ECC is used in encryption and decryption of messages.	[10]	CO2	L2
5	Consider the elliptic curve over the real numbers $y^2 = x^3 + x + 6$ over $\mathbb{Z}_{11}$, Let P = (3,10) and Q = (9,7). Find (P + Q) and 2P.	[10]	CO2	L3
6	List out distinct types of LFSR-based Keystream generator.	[10]	CO5	L1
7	With a neat diagram, explain the concept of Gifford.	[10]	CO5	L2
8	With suitable figures explain (i) Beth-piper stop and go generator, (ii) Alternating stop and go generator.	 [5] [5]	CO5	L2

CCI HOD

Solution

Ques 1. Describe RSA encryption and Decryption algorithm. Which are the five possible approaches to attack the RSA algorithm?

- Developed by Rivest, Shamir & Adleman of MIT in 1977
- It is best known & widely used public-key scheme
- Is also based on exponentiation in a finite (Galois) field over integers modulo a prime
- nb. exponentiation takes $O((\log n)^3)$ operations (easy)
- uses large integers (eg. 1024 bits)
- to encrypt a message M the sender:
  - obtains **public key** of recipient PU={e,n}
  - computes: $C = M^e \bmod n$, where $0 \leq M < n$
- to decrypt the ciphertext C the owner:
  - uses their private key PR={d,n}
  - computes: $M = C^d \bmod n$
- note that the message M must be smaller than the modulus n (block if needed)
- each user generates a public/private key pair by:
- selecting two large primes at random: p, q
- computing their system modulus n=p.q
- note ø(n)=(p-1)(q-1)
- selecting at random the encryption key e
- where 1<e<ø(n), gcd(e,ø(n))=1
- solve following equation to find decryption key d
- e.d=1 mod ø(n) and 0≤d≤n
- publish their public encryption key: PU={e,n}
- keep secret private decryption key: PR={d,n} [6]
- possible approaches to attacking RSA are:
  - brute force key search - infeasible given size of numbers
  - mathematical attacks - based on difficulty of computing ø(n), by factoring modulus n
  - timing attacks - on running of decryption
  - chosen ciphertext attacks - given properties of RSA
    - mathematical approach takes 3 forms:
      - factor n=p.q, hence compute ø(n) and then d
      - determine ø(n) directly and compute d
      - find d directly
  - exploit timing variations in operations
    - eg. multiplying by small vs large number
    - or IF's varying which instructions executed
  - infer operand size based on time taken
  - RSA exploits time taken in exponentiation
  - choose ciphertext to exploit properties of RSA to provide info to help cryptanalysis
  - can counter with random pad of plaintext
  - or use Optimal Asymmetric Encryption Padding (OASP) [4]

Ques 2 Consider a Diffie Hellman scheme with a common prime q=71 and primitive root 7,

(a) Show that 7 is a primitive root of 71.

(b) If user A has private key = 12, what is Y_A the public key?

(c) If user B has private key = 5, what is Y_B public key?

→ Let 71 be a finite commutative ring. An element 7 is a primitive root of 71 if every unit of 71 is a power of 7.

Proof. Suppose the group of units U(71) of 71 has h elements, and suppose 7 is a primitive root of 71. Then, 7, 7^2, 7^3 ,..., 7^h are all different. Thus the powers of 7 include h different units of 71. Since there

are only h units of 71, it follows that every unit of 71 is a power of 7. Conversely, if every element of U(71) is a power of 7, then there are h different powers of 71.
When there is a primitive root modulo m, then multiplication of units mod m is the same as addition of exponents of b modulo $\phi(m)$. Since modular addition is easier than modular multiplication, having a primitive root can be useful for computations.

$U_{71} = \{ [1], [7], [7]^2, \ldots, [7]^{\phi(71)-1}\}$ with $[7]^{\phi(71)} \equiv 1 \bmod 71$.

$[7]^2 \equiv 49 \bmod 71$, $[7]^8 \equiv 27 \bmod 71$, $[7]^9 \equiv 47 \bmod 71$, $[7]^{10} \equiv 45 \bmod 71$, $[7]^{20} \equiv 37 \bmod 71$,
$[7]^3 \equiv 59 \bmod 71$, $[7]^{30} \equiv 32 \bmod 71$, $[7]^{40} \equiv 20 \bmod 71$, $[7]^{50} \equiv 48 \bmod 71$, $[7]^{60} \equiv 30 \bmod 71$
$[7]^4 \equiv 58 \bmod 71$, $[7]^{70} \equiv 1 \bmod 71$
$[7]^5 \equiv 51 \bmod 71$
$[7]^6 \equiv 2 \bmod 71$
$[7]^7 \equiv 14 \bmod 71$ All powers of & generates distinct elements of U(71) hence 7 is primitive root of 71.

Let us first compute $Y_A = 7^{12} \, mod \, 71 \equiv 7^5 \cdot 7^5 \cdot 7^2 \, mod \, 71 \equiv 4 \, mod \, 71$ [3.5]
$Y_B = 7^5 \, mod \, 71 \equiv 7^2 \cdot 7^2 \cdot 7^1 \, mod \, 71 \equiv 51 \, mod \, 71$ [3.5]

Ques 3 Explain the Man-In-The-Middle Attack? Define primitive root with example.

Explanation:[6] Diagram [4]

→Suppose Alice and Bob wish to exchange keys, and Darth is the adversary. The attack proceeds as follows (Figure 10.2).

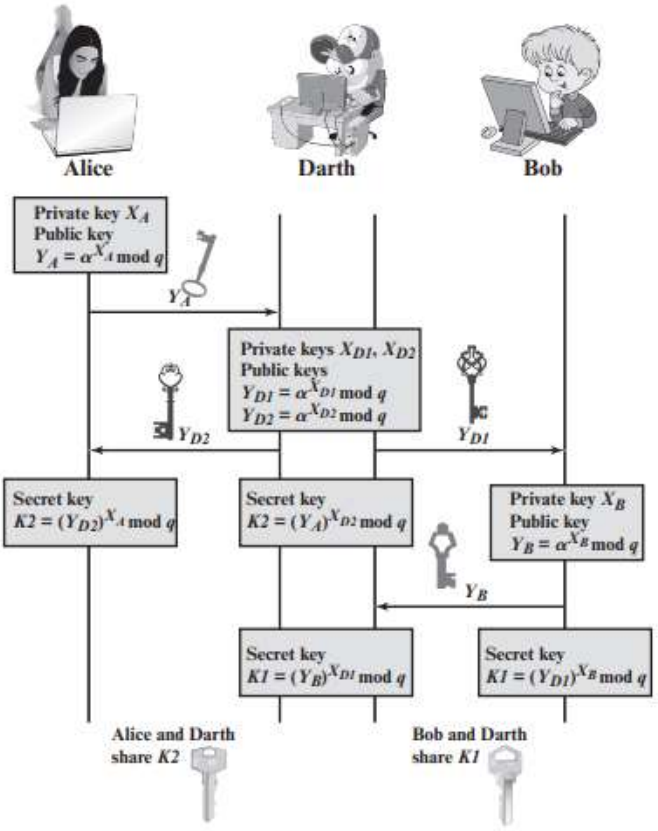

Figure 10.2 Man-in-the-Middle Attack

1. Darth prepares for the attack by generating two random private keys X_{D1} and X_{D2} and then computing the corresponding public keys Y_{D1} and Y_{D2}.
2. Alice transmits Y_A to Bob.
3. Darth intercepts Y_A and transmits Y_{D1} to Bob. Darth also calculates $K2 = (Y_A)\, X_{D2} \bmod q$.
4. Bob receives Y_{D1} and calculates $K1 = (Y_{D1})\, X_B \bmod q$.

5. Bob transmits Y_B to Alice.

6. Darth intercepts Y_B and transmits Y_{D2} to Alice. Darth calculates $K1 = (Y_B)\ X_{D1} \bmod q$.

7. Alice receives Y_{D2} and calculates $K2 = (Y_{D2})\ X_A \bmod q$.

Bob and Alice think that they share a secret key, but instead Bob and Darth share secret key K1 and Alice and Darth share secret key K2.

All future communication between Bob and Alice is compromised in the following way.

1. Alice sends an encrypted message M: E($K2$, M).
2. Darth intercepts the encrypted message and decrypts it to recover M.
3. Darth sends Bob $E(K1, M)$ or E($K1$, M'), where M' is any message.

In the first case, Darth simply wants to eavesdrop on the communication without altering it. In the second case, Darth wants to modify the message going to Bob. The key exchange protocol is vulnerable to such an attack because it does not authenticate the participants. This vulnerability can be overcome with the use of digital signatures and public-key certificates.

Ques 4. Write a note on how ECC is used in encryption and decryption of messages.

→ Consider the equation $Q = kP$ where $Q, P \in E_P(a, b)$ and $k < p$.

- It is relatively easy to calculate Q given k and P, but it is hard to determine k given Q and P.
- This is called the discrete logarithm problem for elliptic curves. [2]
- **Let us consider DH key exchange using ECC**
- Alice selects an integer X_A less than n. This is Alice's private key.
- Alice then generates a public key $P_A = X_A * G$; the public key is a point in $E_q(a, b)$.
- Bob similarly selects a private key X_B and computes a public key P_B.
- Alice generates the secret key $k = X_A * P_B$.
- Bob generates the secret key $k = X_B * P_A$.
- The two calculations in step 3 produce the same result because
- $X_A * P_B = X_A * (X_B * G) = X_B * (X_A * G) = X_B * P_A$
- The first task in this system is to encode the plaintext message m to be sent as an (x, y) point P_m.
- As with the key exchange system, an **encryption/ decryption system** requires a point G and an elliptic group $E_q(a, b)$ as parameters.
- Each user A selects a private key X_A and generates a public key $P_A = X_A * G$. [4]
- **To encrypt** and send a message P_m to B, A chooses a random positive integer k and produces the ciphertext C_m
- $C_m = \{kG, P_m + kP_B\}$ [2]
- **To decrypt** the cipher text, B multiplies the first point in the pair by B's private key and subtracts the result from the second point
- $P_m + kP_B - X_B(kG) = P_m + k(X_BG) - X_B(kG) = P_m$ [2]
- For an attacker to recover the message, the attacker would have to compute k given G and kG, which is assumed to be hard.

Ques 5. Consider the elliptic curve over the real numbers $y^2 = x^3 + x + 6$ over $\mathbb{Z}_{11}$, Let P = (2,7) and Q = (8,3). Find (P + Q) and 2P

→ To compute 2P $\lambda = \dfrac{3x_1^2 + 1}{2y_1}$ $\lambda = \frac{3\times 2^2+1}{2\times 7} \equiv \left(\frac{13}{14}\right) mod 11 \equiv \left(\frac{2}{3}\right) \bmod 11$

Multiplicative inverse of 3 mod 11 is 4. $\therefore \boldsymbol{\lambda = 2 \times 4 \bmod 11 \equiv 8}$

$x_3 = \lambda^2 - x_1 - x_2$ *and* $y_3 = \lambda(x_1 - x_3) - y_1$

$\therefore x_3 = (64 - 2 - 2) \bmod 11 \equiv 5 \equiv$

$\therefore y_3 = \{8(2 - 5) - 7\} \bmod 11 = -31 \bmod 11 = 2$, $\boldsymbol{\therefore 2P = (5,2)}$ [5]

To Compute P+Q use $\lambda = \dfrac{y_2 - y_1}{x_2 - x_1}$ $\lambda = \frac{3-7}{8-2} = \left(\frac{-4}{6}\right) mod 11 \equiv \left(\frac{7}{6}\right) \bmod 11$

Multiplicative inverse of 6 mod 11 is 2, $\therefore \boldsymbol{\lambda = 7 \times 2 \bmod 11 \equiv 3}$

$\therefore x_3 = (9 - 2 - 8) \bmod 11 = -1) \bmod 11 \equiv 10$

$\therefore y_3 = \{3(2 - 10) - 7\} \bmod 11 = -31 \bmod 11 = 2$, $\boldsymbol{\therefore P+Q = (10,2)}$ **[5]**

Ques 6. List out distinct types of LFSR-based Keystream generator.

→ The list of LFSR based keystream generators are:

a) Geffe Generator
b) Generalized Geffe Generator
c) Jennings Generator
d) Beth-Piper Stop-and-Go Generator
e) Alternating Stop-and-Go Generator
f) Bilateral Stop-and-go Generator
g) Threshold Generator
h) Self-Decimated Generator
i) Multispeed Inner-Product Generator
j) Summation Generator
k) DNRSG (dynamic random-sequence generator)
l) Gollmann Cascade
m) Shrinking Generator
n) Self-Shrinking Generator

Ques 7. With a neat diagram, explain the concept of Gifford. Diagram[4], Explanation[6]

→ David Gifford invented a stream cipher and used it to encrypt news wire reports in the Boston area.

- The algorithm has a single 8-byte register: b_0 , b_1 , ..., b_7 .
- The algorithm works in OFB; the plaintext does not affect the algorithm at all.

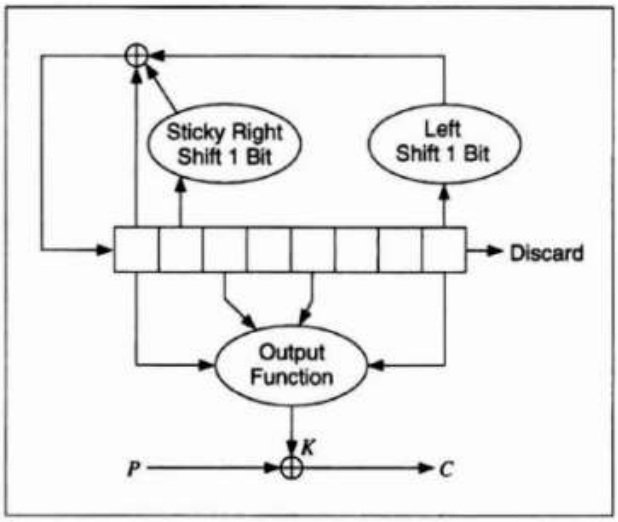

Figure 16.17 Gifford.

- To generate a key byte k_i , concatenate b_0 and b_2 and concatenate b_4 and b_7 . Multiply the two together to get a 32-bit number. The third byte from the left is k_i .
- To update the register, take b_1 and sticky right shift it 1 bit. This means the left-most bit is both shifted and also remains in place. Take b_7 and shift it 1 bit to the left; there should be a 0 in the right-most bit position. Take the XOR of the modified b_1 , the modified b_7 , and b_0 . Shift the original register 1 byte to the right and put this byte in the left-most position. The feedback polynomial wasn't primitive.

Ques 8. With suitable figures explain

(i) Beth-piper stop and go generator,

(ii) Alternating stop and go generator.

→ (i) **Beth-piper stop and go generator,** **[5]**

•BPSG generator, shown in Figure 16.9, uses the output of one LFSR to control the clock of another LFSR The clock input of LFSR-2 is controlled by the output of LFSR-1, so that LFSR-2 can change its state at time t only if the output of LFSR-1 was 1 at time t_1. No one has been able to prove results about this generator's linear complexity in the general case. However, it falls to a correlation attack.

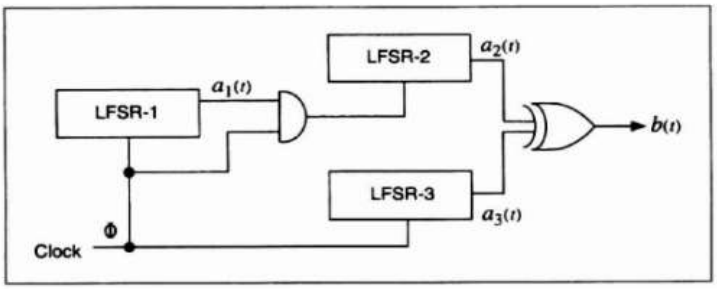

Figure 16.9 Beth-Piper stop-and-go generator.

(ii)Alternating stop and go generator. **[5]**

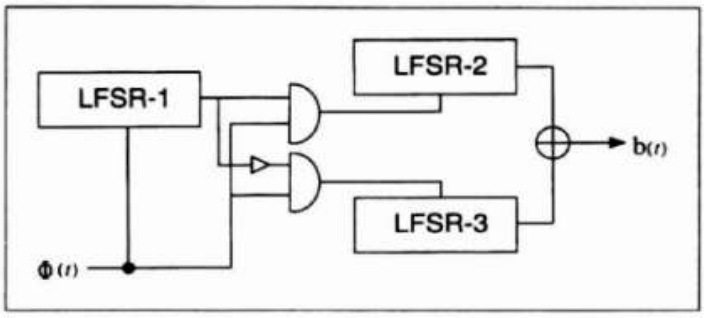

Figure 16.10 Alternating stop-and-go generator.

• ASG generator uses three LFSRs of different length. LFSR-2 is clocked when the output of LFSR-1 is 1; LFSR-3 is clocked when the output of LFSR-1 is 0. The output of the generator is the XOR of LFSR-2 and LFSR3 (see Figure 16.10).

• This generator has a long period and large linear complexity. The authors found a correlation attack against LFSR-1, but it does not substantially weaken the generator. There have been other attempts at keystream generators along these lines