Final Assignment
Cryptography and cyber Law

Md. Shahidul Islam
ID: IT-21024

1. How does shor's algorithm threaten the security of RSA and eliptic curve cryptography (ECC) and what are the potential consequence for current digital infrastructure?

Ans: Shor's algorithm poses a serve threaten to the security of RSA and eliptic curve cryptography (ECC) because it can effiecently solve underlying mathematical problems that this crypto system rely on for security

*Threat to RSA:
→ RSA security is based on the factorial problem. The difficulty of factoring large integers Prime ($N = P \times q$).

p.t.o.

→ Shor's algorithm can factor integers in polynomial time using a quantum computer.

→ A sufficiently large computer could break RSA keys in minute, rendering all RSA based encryption and digital signatures using shor algorithm.

* Threat to ECC:

→ ECC security relies on the distance logarithm problem, Given $Q = k^{P}$. where finding k is hard.

→ Shor's algorithm can solve eliptic curve discrete logarithm problem (ECDLP) efficiently using quantum computer.

→ Ecc based system (used in Bit coin, TLS and many modern protocols) would be broken compromissing secure communication and cryptocurrencies.

* Potential consequences for digital infrastructure:

→ Mass Decryption of Data (Historical encrypted data could be decrypted if keys were harvested.

→ Broken Authentication: Digital signature (SSL/TLS, PGP, S/MIME) would no longer be secure, leading to impersonation and fraud.

→ Financial system collapse: Banking transaction blockchain systems (like Bitcoin) and smart contrast relying on RSA/Ecc would be vulnerable.

→ National security risks: Military and government communication could be exposed if quantum decryption becomes feasible.

* Miligation strategies:

→ Post quantum erytography (PQc): Transitioning

p.t.o.

IT-21024

to Quantum resistant algorithm (e.g lattice based, hash-based and code-based cryptography)

→ Quantum key Distribution : Using quantum mechanics to secure key exchange.

→ Hybrid cryptography : combining classical and post-quantum algorithm during the transition phase.

shor's algorithm fundamentally breaks RSA and ECC once large-scale quantum computer exist. The consequence would be catastrophic for digital security, necessitating an urgent shift to post quantum cryptography before quantum computer mature. NIST is already standerdizing PQC algorithm (Cry rstals - kyber, Dilithium) to Prepare for this threat.

2. Discuss the role of quantum key distribution (QKD) in future cryptographic systems. How does it differ from classical public-key encryption?

Ans:

Role of Quantum key distribution (QKD) in Future cryptographic systems:

Quantum key distribution is a method of securely exchanging cryptographic keys using the principles of quantum mechanics, particularly Heisenbergs uncertainly principle and quantum no-cloning theorem.

→ Providing unconditional security for key exchange independent of computational handness assumptions.

→ Protecting against quantum computer attacks that can break RSA or ECC.

→ securing critical infrastructure (financial,

P.T.O.

IT-21024

defence, government communication) with probable secrecy.

Difference from classical public key Encryption:

classical public key Encryption (RSA, Ecc)	Quantum key Distribution (QKD)
i) Relies on compitational hardness (factorization, discrete logs etc)	i) Relies on laws of quantum physics
ii) Breakable by quantum computer using shor's algorithm	ii) Not breakable by quantum algorithms
iii) No inherent detection mechanism	iii) Intrusion detected via quantum state disturbance.
iv) key distribution are exchangeed over classical channels using encryption	iv) key exchanged via quantum channel (optical fiber, free space)
v) Widely deployed globally.	v) Limited deployment requires specialized hardware.

QKD is not an enenyption method. Its a secune key exchange mechanism. Once keys ane shaned they can be used with symmetnic enenyption (e.g AES. white elassical public key enenyption faces obsole scene in the quantum enra, QKD offens a physics backed path to long run seeune communication, though its widespnead adoptation depends on ovencoming cunnent cost and infnastnuc-tune limitations.

3. What are the main difference between lattice based enyptognaphy and traditional number number theonetic approaches like RSA, particularly in the context of quantum nesistance?

Ans: main difference between lattice based enyptognaphy and RSA in the context p.t.o.

of quantum resistance?

i) security foundation:

→ RSA: Based on the handness of integer factorization.

→ Lattice based enyptography: Based on the handness of high dimentional latties which hand even for quantum computer.

ii) Quantum resistance:

→ RSA: vulnerable to shor's algorithm which can facton large integer in polynomial time.

→ Lattice based: No known efficient quantum algorithm exist for solving lattices problem.

iii) Performance:

→ RSA: smaller public keys but slower key

genenation and enenytion/decnyption for very large key sizes.

→Latties based: Langen key size but genenally faster communication for enenyption, decnyption and key exchanged compared to huge post-quantum safe RSA.

iv) Practical Application:

→RSA: Cunnently dominant for secune key exchanger digital signatunes and TLS/HTTPS.

→Latties based: Being standanized by NIST PQC as neplacement for RSA/Ecc in quantum nesistant system.

RSA's security collapse against large-scale quantum computers while latties based schemes are designed to nesist both classical and quantum attacks, making them central to the future of cnyptognaphy.

4. Develop a python based PRNG that uses the current system time and a custom seed value. Write complete program and corresponding output.

Ans: Python PRNG (pseudo random number generator) code is given below:

```
import time
def custom-prng (seed, count):
    current_time = int (time.time_nsc())
    combined_seed = seed. current_time
    a = 1664525
    c = 1013904223
    m = 2**32
    random_number = [ ]
    x = combined_seed
    for_in nance (count):
        x = (a*x + c) % m
        random_numbers. append (x)
    return random_numbers.
```

```
seed_value = 12345
count = 5
numbers = custom_prng(seed_value, count)
Print("custom PRNG output:")
for i, num in enumerate(numbers, 1):
    print("Random Number [i] : [num]")
```

Sample output:

```
Custom PRNG output:
Random number 1: 1885951992
Random number 2: 2910389135
Random number 3: 2475739278
Random number 4: 339947885
Random number 5: 1326364488
```

5. Explain the sieve of Enathothenes Algorithm and use it to find all prime numbers less than 50. How does it time complexity compane to trial division.

p.t.o.

Ans:

Sieve of Eratosthenes Algorithm:

The sieve of eraathosthenes is an ancient and efficient method for finding all prime numbers up to a given limit n.

Algorithm steps:

i) Create a list of integers from 2 to n.

ii) Start with the first number in the list (p=2) which is prime.

iii) Eliminate all multiple of p.

iv) Find the next number in the list that is not marked: this is the next time.

v) Repeat steps 3-4 until all multiples up to $\sqrt{n}$ have been processed.

vi) The remaining unmarked numbers are all primes.

IT-21024

Find all primes less than 50:

i) Create a list of 2 to 50:

2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50.

ii) Eliminate multiple of 2:

2, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49

iii) Eliminate multiple of 3:

2, 3, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47, 49

iv) Eliminate multiple of 5 ~~and 7~~:

2, 3, 5, 7, 11, ~~15~~, 17, 19, 23, 29, 31, 37, 41, 43, 47, 49

v) Eliminate mutiple of 7:

2, 3, 5, 7, 11, 17, 19, 23, 29, 31, 37, 41, 43, 47

vi) stop the iteration because next prime $11 > \sqrt{50}$.

vii) The last unmarked elements are all prime numbers.

Time complexity comparition:

→ sieve of enathosthenese:

$\left(\frac{n}{2} + \frac{n}{3} + \frac{n}{5} + \frac{n}{7} + \cdots\right) = O(n \log\log n)$

is very efficient for large range.

→ Trial division $O(n\sqrt{n})$ - much slowen.

C++ code:

```
#include <bits/stdc++.h>
using namespace std;
int main()
{
    int vis[n];
```

```
for(int i=2;(i<=sqrt(n);i++)
{
  if (vis[i]==false)
  {
    for(int j=i*i; j<=n; j+=i)
      vis[j]=1
  }
}
for(int i=2; i<=n; i++)
{
  if (vis[i]==false)
    cout<<i<<" ";
}
```

6. state and explain the necessary and sufficient conditions for a composite number to be a carmichael number. Then verify wheather the number n=561, and n=1105 and n=1729 are carmichael number?

Ans: A carmichael number is a composite integer n such that,

P.T.O.

$a^{n-1} \equiv 1 \pmod{n}$

for every integer a with $\gcd(a,n) = 1$

A positive composite integer n is a carmichael number if it holds:

i) n is composite

ii) n is square free

iii) for every prime p dividing n, $(p-1)$ divides $(n-1)$

Verification of given integer number:

$n = 561$

→ factorization: $561 = 3 \times 11 \times 17$

that is composite and square free.

→ compute $n-1 = 560$

→ for $p = 3$ ∴ $560/2 = 280$

for $p = 11$ ∴ $560/10 = 56$

for $p = 17$ ∴ $560/16 = 35$

∴ 561 is a carmichael number.

ii) n = 1105

→ factorization: 1105 = 5 × 13 × 17

→ composite and prime square free

→ n-1 = 1104

p = 5 ∴ 1104/4 = 276

p = 13 ∴ 1104/12 = 92

p = 17 ∴ 1104/16 = 69

∴ 1105 is a carmichael number

iii) n = 1729

→ factorization: 1729 = 7 × 13 × 19

→ composite and square number

→ n-1 = 1728

p = 7 ∴ 1728/6 = 288

p = 13 ∴ 1728/12 = 144

p = 19 ∴ 1728/18 = 96

∴ 1729 is also a carmichael number.

pts.

7. Determine whether the following are valid algebraic structure and justify your answer.

→ is the set Z_{11} with operation $(+, \cdot)$ a ring?

→ Are the sets $(Z_{5}, +)$ and $(Z_{5}, \times)$ Abelian groups?

Ans: Yes, Infact Z_{11} is a commulative ring with unity and moreover a field.

Justification:

→ $(Z_{11}, +)$ is an abelian group: closure, associatelvity, identity inverse $(-a \equiv 11 - a)$ and commutatively hold.

→ multiplication mod 11 is closed and associative and distributes over addition.

→ There is a multiplicative identity $1 \in Z_{11}$

→ Every nonzero element $a \in Z_{11}$ has a multiplicative inverse mod 11.

Yes, $(Z_{32}, +)$ is an abelian group.

Justification:

i) Z_{32} with addition modulo 32.

ii) closure associatively and commutatively follow from integer addition

iii) Identity is 0 for each a.

iv) $(Z_{32}, +)$ satisfy all group axioms and its an abelian.

No, $(Z_{35}, +)$ is not a group.

Justification:

i) multiplication mod 35 is associative and identity 1, not every element has a multiplicative inverse in Z_{35}.

ii) $5 \in Z_{35}$, $\gcd(5, 35) = 5 > 1$. so 5 has no inverse.

iii) The inverse axioms fails and (Z_{35}, X) is not a group.

P.T.O.

8. What is the remainder when -52 is reduces modulo 31?

Ans: We want to find an integer r with condition $0 \leq r < 31$ and $-52 \equiv r \pmod{31}$. compute

$$((-52) \% 31 + 31) \% 31 = 10.$$

So the remainder is 10.

9. Determine the multiplicative inverse of 7 mod 26, if it exists (use extented euclidean algorithm).

Ans: We solve $7x \equiv 1$ (modulo 26). or find integer x, y with $7x + 26y = 1$.

Use euclidean algorithm:

$$26 = 3 \cdot 7 + 5,$$
$$7 = 1 \cdot 5 + 2$$

$5 = 2 \cdot 2 + 1$

$2 = 2 \cdot 1 + 0$

Back-substitute to express 1 as a linear combination:

$1 = 5 - 2 \cdot 2$

$= 5 - 2 \cdot (7 - 1 \cdot 5) = 3 \cdot 5 - 2 \cdot 7$

$= 3(26 - 3 \cdot 7) - 2 \cdot 7 = 3 \cdot 26 - 11 \cdot 7$

Thus $-11 \cdot 7 + 3 \cdot 26 = 1$. Therefore,

$x \equiv -11 \pmod{26}$

$\therefore x \equiv -11 \equiv 15 \pmod{26}$

The multiplicative inverse of 7 modulo 26 is 15.

10. Evaluate $(-8 \times 5) \bmod 17$ and explain how to simplify negative modular multiplication.

Ans:

Step-1: Multiply the numbers:

P.T.O.

$-8 \times 5 = -40$

Step-2: Reduce modulo 17:

We find the equivalent remainder:

$-40 + 3 \times 17 = -40 + 51 = 11$

$\therefore -40 \equiv 11 \pmod{17}$

$\therefore (-8 \times 5) \bmod 17 = 11$

Explain of simplification method:

Negative numbers in modulan arithmetic can be converted to their positive equivalent before multiplying.

$-8 \equiv 9 \pmod{17}$

$9 \times 5 = 45$

Reduce 45 modulo 17:

$45 - 2 \times 17 = 45 - 34 = 11$

This match the previous result.

$\therefore$ Final answer $= 11$.

11. State and proof Bezout's theorem use it to find the multiplicative inverse of 97 modulo 385.

Ans: For integer a and b not both zero, there exist integer such that,

$$ax + by = gcd(a, b)$$

In particular, if $gcd(a, b) = 1$ there are integer x, y with $ax + by = 1$; then x is the multiplicative inverse of a modulo b.

Proof:

Let $d = gcd(a, b)$

→ d divides both a and b.

→ let m be the smallest positive number in the set of all integers of the form $ax + by$. Then m divides a and b.

P.T.O.

→ Hence $m = d$, and there exist integers x, y such that, $ax + by = d$

finding the multiplicative inverse of 97 modulo 385.

We want x such that,

$$97x \equiv 1 \pmod{385}$$

$$97x + 385y = 1$$

We use extended eucleadian Algorithm.

i) Eucledean algorithm:

$$385 = 97 \cdot 3 + 94$$
$$97 = 94 \cdot 1 + 3$$
$$94 = 3 \cdot 31 + 1$$
$$3 = 1 \cdot 3 + 0$$

so, $\gcd(97, 385) = 1$

ii) Back substitution:

$$1 = 94 - 3 \cdot 31$$
$$= 94 - 31(97 - 94 \cdot 1) = -31 \cdot 97 + 32 \cdot 94$$
$$= -31 \cdot 97 + 32(385 - 97 \cdot 3)$$

$1 = 385 \cdot 32 - 97 \cdot 127$

iii) Conclusion:

$\therefore 1 = -127 \cdot 97 + 32 \cdot 385$

$\therefore x = -127 \pmod{385}$

since the positive inverse

$-127 \equiv 385 - 127 \equiv 258 \pmod{385}$

∴ The multiplication inverse of 97 mod 385 is 258.

12. Sing Bezout identity prove that the equation $ax + by = \gcd(a, b)$ has integer solutions.

Find x such that $43x \equiv 1 \pmod{240}$.

Ans: Bezout identity statement;

For any integers a and b, there exist integers x and y such that

$ax + by = \gcd(a, b)$ p.t.o.

IT-21024

Proof:

i) Let $d = \gcd(a, b)$, so $a = da'$ and $b = db'$ with $\gcd(a', b') = 1$

ii) Since a' and b' are co-prime, there exists integers x_0, y_0 such that

$$a'x_0 + b'y_0 = 1$$

iii) multiplying through by d gives

$$a(dx_0) + b(dy_0) = d$$

$$\therefore ax + by = \gcd(a, b) \text{ (proved)}$$

finding x for $43x \equiv 1 \pmod{240}$

$$\therefore 43x - 240y = 1$$

by the Extended Euclidean Algorithm.

Compute gcd chain (Euclidean algorithm):

$$240 = 5 \cdot 43 + 25$$

$$43 = 1 \cdot 25 + 18$$

$25 = 2 \cdot 11 + 8 + 7$

$18 = 2 \cdot 7 + 4$

$7 = 1 \cdot 4 + 3$

$4 = 1 \cdot 3 + 1$

$3 = 3 \cdot 1 + 0$

$\therefore \gcd(43, 240) = 1$

Back-substitution:

$1 = 4 - 1 \cdot 3$

$= 4 - 1 \cdot (7 - 1 \cdot 4) = 2 \cdot 4 - 1 \cdot 7$

$= 2(18 - 2 \cdot 7) - 1 \cdot 7 = 2 \cdot 18 - 5 \cdot 7$

$= 2 \cdot 18 - 5(25 - 1 \cdot 18) = 7 \cdot 18 - 5 \cdot 25$

$= 7 \cdot (43 - 1 \cdot 25) - 5 \cdot 25 = 7 \cdot 43 - 12 \cdot 25$

$= 7 \cdot 43 - 12(240 - 5 \cdot 43)$

$= 7 \cdot 43 - 12 \cdot 240 + 60 \cdot 43$

$= 67 \cdot 43 - 12 \cdot 240$

$\therefore 43 \cdot 67 - 240 \cdot 12 = 1$

So $x = 67$ is a solution. $\therefore 43^{-1} \equiv 67 \pmod{240}$ p.t.o

13. Prove fermat's little theorem and explain how it is used to test for primality. Is 561 a prime number based on this test? Evaluate $5^{123} \bmod 175$ using fermat's little theorem. show all steps.

Ans: fermatt's little theorem - statements:

If p is a prime number and a is any integer with gcd $(a, p) = 1$, then

$$a^{p-1} \equiv 1 \pmod{p}$$

Proof:

Consider the non-zeros residues modulo $p: 1, 2, \cdots p-1$. multiply each by a (with gcd $(a, p) = 1$) the set $a.1, a.2, \cdots a.(p-1)$ is a permutation of $1, 2, \cdots (p-1)$ modulo p.

Taking the product of all elements in both sets and reducing mod p gives,

$$a^{p-1} \cdot (1 \cdot 2 \cdots (p-1)) \equiv 1 \cdot 2 \cdots (p-1) \pmod{p}$$

cancelling the non-zero product $1 \cdot 2 \cdots (p-1)$ which is invertible mod p yields $a^{p-1} \equiv 1 \pmod{p}$.

Is 561 prime:

take $a = 2$, $\gcd(2, 561) = 1$

$$561 = 3 \times 11 \times 17$$

By chinese remainder theorem and FLT:

$$2^{560} \equiv 1 \pmod{3}, 1 \pmod{11}, 1 \pmod{17}$$

So, $2^{560} \equiv 1 \pmod{561}$ - 561 passes FLT but is not prime. It's a carmichael number.

$5^{123} \equiv 0 \pmod{25}$

p.t.o.

$5^{123} \equiv 5^3 = 125 \equiv 6 \pmod 7$

$\therefore 5^{123} \equiv 125 \pmod{175}$

Fermat's little theorem can not be applied directly modulo 175 because 5 and 175 aren't co-prime - thats why we use CRT.

14. state and prove the chinese remainder theorem. Then solve the following system of congruences:

$x \equiv 2 \pmod 3, \; x \equiv 3 \pmod 5$

$x \equiv 2 \pmod 7$

Ans:

statement: The chinese remainder theorem states that a system of linear congruences with pairwise coprime moduli has a unique

solution modulo the product of the moduli. Then the system,

$$x \equiv a_i \pmod{m_i} \quad (i = 1, \ldots, k)$$

has a unique solution modulo $m = m_1 m_2 \cdots m_k$

Solve the system:

start with $x \equiv 2 \pmod 3$. put $x = 2 + 3k$

plug into mod 5:

$$2 + 3k \equiv 3 \pmod 5$$

$$\Rightarrow 3k \equiv 1 \pmod 5$$

$$\Rightarrow k \equiv 2 \pmod 5$$

$$\Rightarrow k = 2 + 5t$$

thus, $x = 2 + 3k$

$$= 2 + 3(2 + 5t)$$

$$= 8 + 15t$$

Now impose mod 7:

$$8 + 15t \equiv 2 \pmod 7$$

P.T.O.

$\Rightarrow 15t \equiv 6 \equiv 1 \pmod 7$

so, $t = 1 + 7s$ then,

$x = 8 + 15(1 + 7s) = 23 + 105s$

therefore the solution modulo $3.5.7 = 105$

$x \equiv 23 \pmod{105}$

15. Briefly explain the CIA triad in information security. How does each component contribute to building a secure system?

Ans: The CIA Triad - Confidentiality, Integrity and availability - is a fundamental model for designing and evaluating secure system.

i) Confidentiality:

→ Ensure information is accessible only to authorized individuals.

→ Achieved through encryption, access control and authentication.

ii) Integrity:

→ Ensure information is accurate consistant and unaltered except by authorized processes.

→ Enforced using hashing, digital signature and checksums.

→ Protects against corruption of data

iii) Availability:

→ Ensure information and systems are accessible to authorized users when needed.

→ Prevent disruption on critical operation.

16. How does steganography differ from cryptography in the context of information security and what are common techniques used for hiding data in digital media?

Ans:

steganography vs cryptography:

Cryptography scramples into unreadable ciphertext to protect context, which steganography hides the existence of the data within normal looking media.

common steganography techniques include:

→ LSB insertion (modifying least significant bits of pixel or samples)

→ Transfor-domain embedding (DCT/DWT in JPEG or video)

→ metadata hiding (using unused header fields).

Cryptography says, "You can see the message but can't read it"

steganography says " You don't even know its there.

17. What are the key difference between phising malware and denial service (DOS) attacks in terms of their method and impack on system security?

Ans:

phising: Social engineering attackers send deceptive email/messages or create fake websites to trick users. attackers goals is too credential theft, initial foothold in network.

Defences: User education, email filtering factors authentication, anti phising policies.

Malware: Software (viruses, worms, trojens, ransomware) delivered via emails, drive by downloads on malicious installers. Goal is to Data theft destruction system compromise.

Defences: Endpoint protection/AV, application white listing backup.

Denial-of-service (DOS/DDOS):

Overwhelm a service or network with traffic or exploit resources limits. Goal is to availability for legimitate users cannot access service.

Defence: Rate limiting, traffic filtering, DDOS mitigation services, redundancy.

phising target people, malwane target softwane, DDOS also target system/softwane

18. Explain how legal frameworks such as the general Data protection regulation (GDPR) help mitigate cybern attacks and protect user privacy.

Ans: Legal frameworks an cyber security:

Role of GDPR the genenal data protection regulation (GDPR) is a european union law designed to strenten the protection of personal data and privacy of individuals. It plays an important role in mitigating

p.t.o.

cyber attacks and protecting user privacy through the following ways.

1. Data Minimization: GDPR requires organization to collect only necessary personal data.

2. security measures: It mandates strong technical and organizational measures to safeguard data against unauthorised access.

3. Breath Notification: Organization must report personal data breaches to authorities within 72 hours.

4. User Rights: Individuals have rights such as access to their data, connection GDPR significantly reduces the risk and impact of cyber attacks.

19. Explain the basic working of the DES Algorithm using a simple 64-bit plain text block and a 56-bit key. Show how the initial permutation contribute to the enenyption process?

Ans:

Basic working of the DES algorithm:

The data encnyption standand (DES) is a symmetric key block ciphen that encnypts data fixed size blocks.

It openates on:

→ plain text block: 64 bits (8 bytes)

→ key: 56 bits

→ Output: 64 bit ciphentext.

1. Initial penmutation (IP):

→ The 64 bit plaintext is neannanged according to the fixed table. p.t.o.

→ If the plain text is $P = [P_1 P_2 \cdots P_{64}]$ the IP might move to position 1, P_{50} to position 2 etc.

2. Rounds (16 iteration):

The output of IP is divided into two halves:

→ left half (L_0): first 32 bits

→ Right half (R_0): Last 32 bits

i) Expansion

ii) key mixing

iii) key substitution

iv) Permutation (P-box)

v) Swap and combine

formula for each round

$L_n = R_{n-1}$

$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$

3. Final permutation (FP):

After 16 rounds the final swap is performed the two halves are joined and passed through the inverse of the initial permutation to produce the cipher text.

20. In the DES algorithm, a 64-bit plaintext block is divided into two 32 bit halves: L_0 and R_0. Given $R_0 = 0xF0F0F0F0$ and round key $k_1 = 0x0F0F0F0F$, compute the output of the first round function $f(R_0, k_1)$ assuming XOR operation only. Then, find $L_1 = R_0$ and $R_1 = L_0 \oplus f(R_0, k_1)$, where $L_0 = 0xAAAAAAAA$.

Ans: Given that,

$R_0 = 0xF0F0F0F0$,

Round key $k_1 = 0x0F0F0F0F$

p.t.o.

$L_0 = 0xAAAAAAAA$

step-1: compute $f(R_0, k)$ assuming XOR operation only.

$f(R_0, k_1) = R_0 \oplus k_1 = 0xF0F0F0F0 \oplus 0xF0F0F0F0$

Perform XOR byte-wise:

→ $F0 \oplus 0F = FF$

→ $F0 \oplus 0F = FF$

→ $F0 \oplus 0F = FF$

→ $F0 \oplus 0F = FF$

So, $f(R_0, R) = 0xFFFFFFFF$

step-2: compute $L_1 = R_0 = 0xF0F0F0F0$

step-3: compute $R_1 = L_0 \oplus f(R_0, k_1)$

$R_1 = 0xAAAAAAAA \oplus 0xFFFFFFFF$

XOR byte-wise:

→ $AA \oplus FF = 55$

→ AA ⊕ FF = 55

→ AA ⊕ ff = 55

→ AA ⊕ FF = 55

So, $R_1 = 0x55555555$

$\therefore K = 0xF0F0F0F0$, $R_1 = 0x55555555$

21. Given the input word:

$[0x23, 0xA7, 0x4C, 0x19]$

Use the following partial AES S-box to perform the subbytes transformation. Provide the resulting output word.

Row \ Col →	3	4	7	9	A	C
1		6D			C_6	
2		D_4				
4			A_1			2E
A			63		D_2	

Ans: Given input word:

$[0x23, 0xA7, 0x4C, 0x19]$

for each byte, the high nibble (4 bits) is the row and the low nibble is the column.

for each byte, the high nibble (4 bits) is the row and the low nibble is the column.

Lets find the corresponding s-box value for each byte:

- for $0x23$: row = 2, col 3 → from table row 2 col 3 = $2D_4$
- for $0xA7$: row A col 7 = 26
- for $0x4C$: row = 4, col = C, unknown

for $0x19$: row 1 col 9 = C_6

Resulting output word:

[0x04, 0x63, unknown, 0xC6]

22. In AES encryption, apply the Add round key step only. Given:

- input word: [0x1A, 0x2B, 0x3C, 0x4D]
- Round key word: [0x55, 0x66, 0x77, 0x88]

Perform XOR between the input word and the round key and write the resulting output word.

Ans:

The AddRoundKey step in AES encryption involve performing a Bitwise XOR operation between the input word and the round key word. Given,

input word: [0x14, 0x2B, 0x3C, 0x4D]

IT-21024

Round key word: [0x55, 0x66, 0x77, 0x88]

Compute the output word.

Solution (Byte wise XOR)

0x1A ⊕ 0x55 = 0x4F

0x2B ⊕ 0x66 = 0x4D

0x3C ⊕ 0x77 = 0x4B

0x4D ⊕ 0x88 = 0xC5

Resulting output word: [0x4F, 0x4D, 0x4B, 0xC5]

23. Show how mixcolumns uses the following fixed over GF(28):

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

to transform an input column. Use an example column with values [0x01, 0x02, 0x03, 0x04]

Convert the Hex to Binary.

Ans: The mixcolumns operation in AES works by multiplying each column of the state matrix by a fixed matrix over the field GF(28).

Input column

$$a_2 \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix} = \begin{vmatrix} 0x01 \\ 0x02 \\ 0x07 \\ 0x04 \end{vmatrix}$$

So the output column is $b_2 = M \cdot a$ where each entry is bytewise $\cdot$ GF(2^8) arithmatic XOR.

$$b_0 = (02 \cdot a_0) \oplus (03 \cdot a_1) \oplus (01 \cdot a_2) \oplus (01 \cdot a_3)$$

$$= 0x07$$

$b_1 = 0x04$

$b_2 = 0x09$

$b_3 = 0x0A$

$\therefore$ final result $b_2 = \begin{vmatrix} 0x07 \\ 0x04 \\ 0x09 \\ 0x0A \end{vmatrix}$

P.T.O.

24. Describle how AES-OFB modes works. How does it ensure synchronization between encryption and decryption stream?

Ans: AES-OFB (output feedback) is a mode of operation for the advanced encryption standard (AES) that turns a block cypher into a stream cipher.

Working procedure of AES-OFB mode

i) Initialization: The process starts with a unique initialization vector.

ii) Keystream Generation: The output of the AES encryption.

iii) Encryption: Each block of the keystream is xored with a corresponding block of plaintext to produce the ciphertext.

iv) Decryption: The decryption process is identical.

synchronization in AES - OFB:

→ Both senden and receiven stant with the same initialized vecton (IV) and sane secnet key.

→ keystream blocks are generated only from the Previous keystream block not from the ciphentext on plaintext.

25. Which AES modes causes ennon propagation during decryption and how does it affected the integrity of the decrypted message? Illustrate with CBC and CFB modes.

Ans:

AES modes causes ennon propagation:

Some AES modes (CBC and CFB) cause an ennon in a ciphen text block to affect multiple plaintext blocks during decryption.

i) CBC (Ciphen Block chaining) mode:

→ Decryption formula:

$$P_i = AES^{-1}(C_i) \oplus C_{i-1}$$

→ if C_i has a 1 bit ennon - P_i becomes random.

→ Ennon effect two consequetive plaintext blocks.

ii) CFB (Ciphen feedback Mode):

→ Decryption formula: $P_i = C_i \oplus AES(C_{i-1})$

→ If C_i has a 1 bit ennon → the same bit in P_i is wrong.

→ P_{it} becomes completely random

26. which AES mode would you recommend for encrypting large files with parrabel processing and why? Justify answen between ECB, CBC and CTR.

Ans: Recommend model : AES - CTR (counter mode)

Justification :

→ Pannellel Processing :

- In CTR, each block is encrypted by xoring the plaintext with a keystream generated from AES (key, counter).

→ Performance :

- No chaining between blocks - faster that CBC and for large files.

→ Security :

- Unlike ECB, does not produce identical plaintext blocks.

→ Flexibility :

→ Works efficiently for both stream like and random access data.

Why not ECB ?

P.T.O.

Reveals patterns in the data in secure.

Why not CBC?

Each block depends on the previous ciphertext block.

for large file encryption with parallel processing, AES-CTR is best because it is secure and highly pannalelizable.

27. Given a message "A" represented as m = 1, encrypt it using public key e = 5, n = 14. What is ciphertext? Then decrypt using private key d = 11.

Ans:

Given $m = 1, e = 5, n = 14, d = 11$

Encryption:

$c = m^e \bmod n = 1^5 \bmod 14 = 1$

Decryption:

$m = c^d \bmod n = 1^{11} \bmod 14 = 1$

RSA relies on $m^{ed} = m \pmod n$ when $ed \equiv 1 \pmod{\varphi(n)}$. Here, $\phi(14) = 6$ and $e \cdot d = 5 \cdot 11 = 55 \equiv 1 \pmod 6$ so decryption recovers m.

∴ ciphertext $c = 1$ and Decrypted message $m = 1$.

28. Given message hash: $H(m) = 5$, RSA Private key: $d = 3, n = 33$. Generate the digital signature,

Ans: Given hash $H(m) = 5, d = 3, n = 33$

Signature generation :

We need to generate a digital signature for a message hash.

→ The message hash is H(m) = 25

→ The RSA private key is (d, n) = (3, 33)

→ The digital signature is completed using the hash formula.

$$S = H^d \pmod{n}$$

$$S = 25^3 \pmod{33}$$

$$= 15625 \pmod{33}$$

$$\because 15625 = 33 \times 473 + 26$$

$$\therefore 15625 \pmod{33} = 26$$

∴ The digital signature is 26.

29. Aleya and Badol are using the Diffie-Hellman key exchange protocol. They agree on the following public values: Prime modulus $p = 17$, $g = 3$, Alia private key $a = 4$, badol's $b = 5$.

Compute public key of Aleya and Badol.

Ans:

Public values: Prime modulus P = 17

base, g = 3

Alias private key a = 4

Badol b = 5

∴ Aleyas public key is 13.

thus, Badol's public key, $B = g^b \pmod p$

$= 3^5 \pmod{17}$

$= 5$

∴ Badol's public key is 5.

30. A simple hash function

H(x) = (ΣASCII to characters in x) mod 11

~~Ascii~~ compute the hash value of the

p.t.o.

message "AB" and "BA" using this function. Do they produce same hash? What does this imply collision resistance in weak hash function?

Ans:

$H(x) = (\Sigma \text{ ASCII chars in } x) \bmod 100$

ASCII values : A = 65, B = 66

$\therefore H(\text{"AB"}) = (65+66) \bmod 100 = 31$

$\therefore H(\text{"BA"}) = (66+65) \bmod 100 = 31$

$\therefore$ Both result produce same Hash → collision

Collision Resistance:

The hash function is not collision resistance because it based on a simple modulo sum.

IT+2107.4

31. In a secure message system a simple message authentication code (MAC) is computed using modular operation.

MAC = (mes + sec.key) mod 17

where, mes=15, sec.key = 7. compute MAC for message. Can they forge the correct MAC easily?

Ans:

m=15, k=7

∴ MAC = (15+7) mod 17

= 22 mod 17 = 5

Suppose attenes changes m'=10 but doesn't know k then,

MAC' = (10+7) mod 17 = 0

Why forging is easy?

Because MAC is linear. If attackers knows one valid pair (m, MAC) and wants to create

p.t.o.

$m' = m + \Delta$ they can compute $MAC' = MAC + \Delta(m)$ without knowing k

32. Explain the steps involved in the TLS handshake process. How are symmetric key established securely using symmetric cryptography during the handshake?

Ans:

TLS handshake steps 2 symmetric key.

Establishment:

i) clientHello → client sends supported TLS, version, random number.

ii) serverHello → server select TLS and sends its random number.

iii) Server certificate → contains server public key

iv) key exchange →

- RSA
- Ec DHE
- Mode

v) Pre Master secret → master secret

vi) session key

vii) finished message → both confirmed to the handshake.

Assymetric encryption ensures that only the intended parties can compute the shared secret keys from which symmetric keys are derived.

33. Explain the layerd architechture of SSH. Briefly describle the roles of each layer.

Ans:

SSH refers to "secure shell" is a protocol that provides a secure channel over an unsecured network. It has a layered architechture consist of three layers.

p.t.o.

i) Transport layer:

This is the lowest layer responsible for managing the secure connection.

→ Handles encryption
→ Integrity protection

ii) User authentication layer:

This layer runs on top of the transport layer and handles client authentication.

iii) Connection protocol:

multiplexes the encrypted channel into multiple logical channels. This is the highest layer.

34. Explain the steps involved in the TLS handshake process.

Ans:

TLS handshake process steps:

i) ClientHello → Purposes connection parameters.

ii) ServerHello → Purposes connection parameters.

iii) Certificate and key exchange

iv) Generate Master secret

v) Generate session key

vi) Finished messages

vii) Secure communications

35. What is the general form of an elliptic curve equation over a finite field and why it used in cryptography?

Ans: The general form of an elliptic curve equation over a finite field is

$$y^2 = x^3 + ax + b \pmod{p}$$

Here p is a large prime number that defines the finite field, and a and b are constant such that,

P.T.O.

$4a^3 + 27b^2 \neq b \pmod{p}$

(ensure no singularities)

Use in cryptography:

Provides a group structure for Elliptic curve cryptography enabling secure key exchange.

36. How does ECC achieve the same level of security as RSA, with a smaller key size?

Ans: Elliptic curve cryptography is a public key - cryptography technique that provides the same cryptographic strength as RSA but with much smaller key sizes,

i) mathematical foundation:

→ RSA is based on integer factorization
→ Ecc is based on the Elliptic curve discrete

logarithomic Problem (ECDLP).

(ii) key size:

→ Ecc 256 bit ≅ RSA 3072 bit

→ Ecc 384 bit ≅ RSA 7680 bit

(iii) conclusion:

Ecc provides strong security with reduced key size with ECDLP.

37. Given the elliptic curve

$$y^2 = x^3 + 2x + 3 \pmod{97}.$$

determine wheather the point P = (3, 6) lies on the curve.

Ans: Given that,

Curve: $y^2 \equiv x^3 + 2x + 3 \pmod{97}$

point: P (3, 6)

L.H.S = $y^2 = 6^2 = 36 \pmod{97}$

P.T.O.

R.H.S = x^3+2x+3

$= 2^2+6+3$

$= 36 \pmod{99}$

∴ point lies on the curve.

38. Given public key ($p=23, g=5, h=8$) and message $m=10$, compute the elgamal ciphertext using random $k=6$.

Ans: Given,

$p=23, g=5, h=8, m=10, k=6$

The ciphertext is (c_1, c_2).

$c_1 = g^k \pmod{p}$

$= 5^6 \pmod{23}$

$= 8$

and, $c_2 = m \times h^k \pmod{p}$

$= 10 \times 8^6 \pmod{23}$

$= 10 \times 13 \bmod 23$

$= 130 \pmod{23} = 15 \pmod{23}$

$\therefore C_2 = 15$

$\therefore$ The Elgamal ciphertext is (8,15)

$(C_1, C_2) = (8, 15)$.

39. Explain how lightweight cryptography is important for securing IoT devices. Give on example of a lightweight encryption algorithm used in IoT.

Ans: Lightweight cryptography is specially designed to provide strong security while using minimal computational resource, memory and power. This is crucial for IoT devices such as sensors, wearables and smart applications. Traditional encryption Algorithm like AES or RSA may be too heavy for these devices. Lightweight cryptography ensures:

P.T.O.

→ Low power consumptions
→ Low memory and cpu usage
→ Adequate security.

Present ciphen - a block ciphon with 64 bit block site designed for IOT devices which is a lightweight cryptography.

40. List and briefly explain any three comm IOT-specific attacks (e.g firmware hijacking physical tempering) what mitigation shategie can be applied.

Ans: Three common IOT specifi attacks and mitigation strategies:

i) Firmware Hijacking: Attacken neplace modify device firmware with malicious code, allow them to take control, steal data on disrupt functionality

→ mitigation: use digitally signed firmware enable secure boot, and only allow update from trusted servers.

iv) Physical Tamering: IoT devices in public or unprotected areas can be physically accessed. Attackers may open the devices.

→ mitigation: Use tempen resistant coatings and disable debug ports.

iii) Botnet attacks: Malware infected to IOT devices with weak/default passwords linking them into a botnet for large scale DDOS or spam attacks.

→ mitigation: change defaults credential keep firmware updated and use firewalls.

IoT devices are vulnerable due to limited resources and weak security. Implement

secure updates, physical protection, and strong authentication can greatly reduce attack risks.

completed