Assignment—

final

S·M Nasimul Hasan

IT-21026

Cryptography and Cyber Law

IT - 21026

S M Nasimul Hasan

1

Shor's Algorithm: Shor's algorithm, a quantum computing algorithm, poses a significant threat to the security of RSA and elliptic curve cryptography (ECC) by efficiently solving the mathematical problems these algorithms rely on. These systems ensure the confidentiality, authenticity, and integrity of everything from emails to online banking and e-commerce.

Shor's algorithm poses a direct and threat to widely used in public-key cryptography

schemes like RSA and elliptic curve cryptography because it enables a quantum computers to efficiently solve problems.

RSA: Based on factoring large composite numbers. Shor's algorithm uses quantum period-finding and Quantum Fourier Transform to break RSA efficiently.

ECC: Relies on the difficulty of discrete logarithms on elliptic curves. Shor's algorithm can solve this problem too, allowing recovery of private keys from public ones.

IT-21026

Potential consequences for Digital Infrastructure:

(i) Breakdown of Internet Security:

- Protocols like TLS/SSL (used in HTTP websites) will become insecure.
- Attackers can intercept and decrypt web traffic exposing sensitive data.

(ii) Data Vulnerability:

- Harvest now, Decrypt later
- Retrospective Decryption
- Compromised confidentiality.

(iii) Compromise of Financial System:

- Online banking and crypto currencies depend on RSA/ECC
- Quantum computers could forge signature, steal assest and manipulate

NASIMUL

(iv) Erosion of Privacy:

. Personal and sensitive data exchanged through insecure networks could be expressed, leading to identify theft and privacy breaches.

(v) Urgency for Cryptographic migration.

. It is not addressed, there may be sudden collapse in digital security often referred to as "Q-Day".

Shor's algorithm is not just a theoritical threat - it is ticking time bomb for all cryptographic systems based on RSA and Ecc.

NASIMUL RAHMAN

2

Quantum Key Distribution (QKD) is a secure communication method that utilizes principles of quantum mechanics to generate and distributes cryptographic keys.

Role of QKD in future cryptographic Systems.

1. Quantum-Safe Key Exchange:

- Secure key sharing using quantum physics.
- Resistant to quantum attacks.

2. Unconditional Security:

- QKD's security is based on laws of Quantum physics.

- Any attempt to eavesdrop on the quantum channel distrubs the system, alerting the legitimate parties.

NASIMUL

3. Eavesdropping Detection:

- Compromised keys are discarded.

4. Post Quantum Ready:

- Used with symmetric encryption.

5. Quantum - Internet Backbone:

- Core tech for future secure quantum networks.

6. Physics-Based Trust:

- Security from laws of nature, not hard math.

Nasimul

3

Lattice based cryptography and traditional number theoretic cryptography (such as RSA and ECC) differ significantly in their mathematical foundations, security assumption and resistance to quantum attacks.

Lattice based cryptography offers a significant advantage over traditional number theoretic approaches like RSA in terms of quantum resistance.

Here's more differences detailed:

1. Security foundation

- Traditional (RSA/ECC):
  - Relies on the difficulty of factoring numbers of solving the discrete logarithm problem

Nasimul

- Lattice based cryptography:
  - Based on problems in high dimension al geometry, such as the shortest vector problem (SVP) and LWE.

- Quantum Resistance:
  - Traditional (RSA/ECC):
    - Vulnerability to quantum algorithm like Shor's algorithm, which can solve their underlying problems in polynomial time.
  - Lattice Based Cryptography:
    - Considered quantum-resistant, as no efficient quantum algorithms are known to solve lattice problems.

Nasimul

4

Answer:

Python PRNG code:

```
import time

def custom_prng(seed, count):
    current_time = int(time.time_ns())
    combined_seed = seed ^ current_time

    a = 1664525
    c = 1013904223
    m = 2**32
    random_numbers = []

    x = combined_seed

    for _ in range(count):
        x = (a*x + c) % m
        random_numbers.append(x)

    return random_numbers

seed_value = 12345

count = 5
```

Nasimul

```
numbers = custom_prng(seed_value, count)

print("Custom PRNG Output:")

for i, num in enumerate(numbers, 1):

    print(f"Random Number {i}: {num}")
```

Sample Output:

Custom PRNG Output:

Random Number 1: 1885991992

Random Number 2: 2910389135

Random Number 3: 2475739278

— o —

Question — 5 :

* Sieve of Eratosthenes is an algorithm to find prime numbers by excluding the multiples of other numbers.

Finding primes under fifty.

Nasimul

Step-1: ~~Divide~~ see who are Divisible by 2:

1, 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~~, 27, ~~28~~, 29
~~30~~, 31, ~~32~~, 33, ~~34~~, 35, ~~36~~, 37, ~~38~~, 39, ~~40~~, 41,
~~42~~, 43, ~~44~~, 45, ~~46~~, 47, ~~48~~, 49, ~~50~~.

prime → 2

Step-2 Take next valid num and exclude multiples. ~~num~~ = 3.

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

prime → 3

Step-3: take '5' and do same;

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

prime-5

Step-4: take 7

7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, ~~49~~

Step-5: take 11, but $11^2 = 121 > 50$

so algorithm stop.

prime numbers: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47.

Nasimul

Time complexity:

(i) Siene : $O[n\{\log(\log n)\}]$

(ii) Trial Division: $O(n\sqrt{n})$.

6. Answer:

A charmichael numbers is a composite integers n such that

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

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

conditions for charmichael numbers.

(i) n is composite

(ii) n is square-free

(iii) for every prime p dividing n $(p-1)$ divides $(n-1)$ i.e $p-1 \mid n-1$

Verification for the given numbers:

* Checks korselts criterion

1. $n = 561$

Vasimal

. Factorization: $561 = 3 \times 11 \times 17$ → composite and square free

. Compute $n-1 = 560$

. For $p = 3$: $p-1 = 2.2 \mid 560$ $(560/2 = 280)$

. for $p = 11$: $p-1 = 10.10 \mid 560$ $(560/10 = 56)$

. For $p = 17$: $p-1 = 16.16 \mid 560$ $(560/16 = 35)$

All conditions satisfied. 561 is a charmichael number.

2. $n = 1729$

factorization: $1729 = 7 \times 13 \times 19$ → composite and square.

$n-1 = 1728$

For, $p = 7$: $p-1 = 6.6 \mid (1728/6) = 288$

$p = 13$: $p-1 = 12.12 \mid (1728/12) = 144$

$p = 19$: $p-1 = 18.18 \mid (1728/18) = 96$

conditions satisfied 1729 is a carmichael number.

Navimul

(7). Yes, In fact $\mathbb{Z}_n$ is a commutative ring and moreover a field.

Justification: $(\mathbb{Z}_n, +)$ is an abelian group: closure, associatively, identity 0, inverses ($-a \equiv 11 - a$), and commutativity holds.

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

- There is a multiplicable identity $1 \in \mathbb{Z}_n$.

11 is prime $\longrightarrow$ $\mathbb{Z}_{11}$ is a field.

$*_8(\mathbb{Z}_{37}, +)$ is an abelian group.

Justification:

$\mathbb{Z}_{37} = \{0, 1, \ldots, 37\}$ with addition modulo 37.

→ Closure, associativity and commutivity follow from integer additions.

Identity is 0! For each a, additive inverse from integer additions.

Naimul

Thus $(\mathbb{Z}_{31}, +)$ is an abelian group.

But not $(\mathbb{Z}_{35}, \times)$ is a group.

8. Ans: We want r with $0 \le r \le 31$

and $-52 \equiv r \pmod{31}$

Compute:

$$-52 + 3 \cdot 31 = -52 + 62 = 10$$

So,

$$-52 \equiv 10 \pmod{31}$$

Alternate: $-52 \equiv -52 + 31 = -21 \equiv -21 + 31$

So, the remainder is 10.

Navimul

[9]. We solve $7x \equiv 1 \pmod{26}$ or find integers x, y with $7x + 26y = 1$

Use the Euclidean algorithm:

$$26 = 3 \cdot 7 + 15,$$

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

$$5 = 2 \cdot 2 + 1,$$

$$2 = 2 \cdot 1 + 0$$

to express 1 as a linear combination:

$$1 = 5 - 2 \cdot 2$$

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

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

Thus $-11 \cdot 7 + 3 \cdot 26 = 1$. Therefore $x \equiv -11 \pmod{26}$

$$x \equiv -1 \equiv 15 \pmod{26}$$

The multiplicative inverse of 7 modulo 26 is 15.

Nasimul

10

Step-1: Multiply the numbers:

$$-8 \times 5 = -40$$

Step-2: Reduce Modulo 17:

We find that equivalent remainder,

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

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

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

11

For integers a and b to both zero, there exist integers x, y such that,

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

~~In particular~~ Proof Idea: Apply the Euclidean algorithm, back-substitute to express the last

Naimul

non zero remainder (the gcd) as a linear combo of a & b

Inverse: $97^{-1} \pmod{385} = 258$ (since $97 \cdot 258 \equiv 1 \pmod{385}$):

12. Bezout for solvability:

of $ax+by=c$

Claim. $ax+by=c$ has integers solutions if $\gcd(a,b) \mid c$

Why. If $d = \gcd(a,b)$, Bezout gives $ax_0+by_0=d$

Multiply both sides by c/d to get one solution.

conversely any solution implies $d \mid c$

General Solution: If (x_0, y_0) is one solution and $d = \gcd(a,b)$, then all solutions are:

$x = x_0 + \frac{b}{d}t, \quad y = y_0 - \frac{a}{d}t, \quad t \in \mathbb{Z}$

Nasimul

13. fermats Little Theorem (FLT), Primality Test.

FLT: If p is prime and gcd $(a, p) = 1$, then $a^{(p-1)} = 1$. Or use the permutation $\{a, 2a, \ldots, (p-1)\}$ modulo p.

Use for Testing: If $a^{n-1} \not\equiv 1 \pmod{n}$ for some base a coprime to n, the n is composite. If it does hold, n is only a probable prime (There are pseudoprimes).

561 = 3.11.17 is composite and a even a charmichael number. It passes FLT for all a comprime to 561. So FLT alone cannot certify it prime.

Nainmul

Evaluate $5^{?}$: FLT reduces exponents mod $p-1$

$p > 13$

$5^{100} \equiv 5^{(12 \cdot 8 + 4)} \equiv (5^{12})^{8} \cdot 5^{4}$

$\equiv 1 \cdot 625 \equiv 1 \pmod{13}$

14. Chinese Remainder Theorem.

Statement: For pairwise coprime $n_1, n_2, \ldots n_k$ the system $x \equiv a_i \pmod{n_i}$ has a unique solution modulo $N = \prod n_i$

Construction: Let $N_i = N/n_i$, compute $M_i \equiv N_i^{-1} \pmod{n_i}$.

Then,

Nasimul

$$x \equiv \sum_{i=1}^{k} a_i N_i M_i \pmod{N}.$$

Why it works: Each term is 0 modulo the other moduli and a_i modulo n_i, so the sum matches all congruences.

Uniqueness follows from counting modulo N.

Nasimul

[19] CIA Triad:

- Confidentiality: Only authorized parties see data. Controls: Encryption at rest / in transit, access control, classification, DLP.

- Integrity: Data is accurate/untampered. Controls: hashes, MACs, digital signatures, checksums, immutability logs.

- Availability: Systems / data are usable when needed. Controls: Redundancy, failover, rate-limiting, WAF/DoS mitigation, Backups.

Together they balance secrecy, correctness, and uptime - comprising one can endanger others.

Nasimul

[16]:

Steganography vs Crytography:

Cryptography:

Cryptography hides information, meaning (ciphertexts look random). Even if seen, attacker needs key.

That means crytography is just changing the appearance of a text rather than hiding it completely. There are many cryptographic systems we used. Ceaser cipher, RSA, SHA. These are some systems to encrypt confidential data.

Nasimul

Steganography: It hides existence of an information or data. Secret embedded inside a carrier (image/video/audio/text).

Meaning we don't change the appearance Rather we completely hide it over a cover file.

Popular Stego Techniques are:

- Image LSB. (replace pixel)
- Palette manipulation,
- DCT coefficient- tweaking in JPEG.
- Echo/Phase coding in audio

Best practice is stego + crypto together.

Navinul

17. Phishing, Malware, DoS - method and impact.

- **Phishing:** Tricks users to reveals creds/2FA or run payloads:

Phishing is like putting a trap that looks like the original website or application. But originally a fake one to fool users and steal their sensitive informations.

- **DoS:** Denial of Service is a not like a hacking. It's an attack used to down or stop a service to fail to provide service.

DoS/DDoS is about flooding resources/

Nasimul

bandwidth.

Impact: Down time, lost revenue.

SLAs.

- **Malwares:** Malicious codes or softwares programs (Trojan, ransomware, spyware) to get inside a system and corrupt it.

Impact: Data Lost, extortions, persistence.

- **Defense:** User training, anti-malware/EDR, patching, zero-trust, email security, rate limits/CWT, WAF, anomaly detection

Navimul

18. How GDPR mitigate attacks and protects privacy.

- Data Minimization: It reduces blast radius or impact radius if the data is getting breached by outsiders.

- Lawful basis and consent: Restrict uncontrolled processing / sharing accross platforms. To share it requires consents or permissions.

- Security by design and by defaults (Art. 25) mandates appropriate technical

Nasimul

- **Breach Notification : (72 h)**

faster response whenever the data is getting tried to breach out. It alerts as soon as the attack comes.

- **Data Subject Right : (Access, erasure)**

force better data lifecycle controls.

- Fines incentive governance. Net effect: fewer copies, better controls, clearer accountability.

Nasimul

19 DES workflow (64-bit block, 56-bit key).

- Apply Initial Permutation:

The 64 bit plaintext is rearranged according to a fixed table.

- This steps does not involve the key.

It simply changes bit positions to prepare for rounds.

- Example: If the plaintext is $p = [P_1, \ldots, P_{64}]$

The IP might move P58 to position 1.

P50, to position 2 etc.

Rounds:

- IP divided into two part's
  - Left half (32 bits)
  - Right half (32-bits)

Nasimul

a. Expansion.

b. key mixing

c. substitute (S-boxes)

d. permutation (p-box)

e Swap and combine.

Formula for each round:

$$L_n = R_{n-1}$$

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

Then comes the 3rd Roud that is Final permutation (FP).

Here the processed two halves are joined.

Narimul

20

Given,

$R_0 = 0x$ FOFOFOFO

Round key : $k_1 = 0x$ OFOFOFOF

$L_0 = 0x$ AAAA AAAA

Sol Step 1 : Compute $f(R_0, k_1)$ assuming XOR operations only.

$f(R_0, k_1) = R_0 \oplus k_1 = 0x$ FOFOFOFO $\oplus$ 0x0F0F0F0F

Perform XOR byte wise

- F0 $\oplus$ 0F = FF
- F0 $\oplus$ 0F = FF
- F0 $\oplus$ 0F = FF
- F0 $\oplus$ 0F = FF

Nasimul

So, $f_0(R_0, k_2) = 0x\ FFFFFFFF$

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

Step-3: Compute $R_1 = L_0 \oplus F(R_0, k_1)$

$R_1 = 0x\ AAAAAAAA \oplus 0x\ FFFFFFFF$

XOR - byte wise:

$AA \oplus FF = 55$

$AA \oplus FF = 55$

$AA \oplus FF = 55$

$AA \oplus FF = 55$

So, $R_1 = 0x55555555$

$L_1 = 0x\ F0F0F0F0$, $R_1 = 0x\ 55555555$

Nasimul

21

Given input word:

[0x23, 0xA7, 0x4C, 0x10]

Partial AES S-box table:

Row \ Col	3	4	7	9	A	C
1	6D	.	.	C_6	.	.
2	D4	.	.	.	.	.
4	.	A1	.	.	2E	.
A	.	.	C3	D_2	.	.

for each byte, the high nibble (4-bits) is

Nasimul

the row and the low nibble is the column.

Lets find the corresponding value for each byte:

- For 0x23: row = 2, col = 3 → From table row 2 and col 3 = D4
- For 0xA7: row = A, col = 7 → From table row 2 col 7 = 63
- For 0x4C: Row = 4 col = C, row 4 col C = not given.
- For 0x19: row = 1, col = 9 → From table

Resulting word output:

[0x D4, 0x63, unknown 0xC6]

Since 0x4C is not present, we indicate unknown for no mapping given.

Vaishnavi

22.

The addRoundKey step in AES encryption include performing a bitwise XOR operation between the input word and the round key word given,

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

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

Compute the Output word:

Solution (byte-wise XOR)

$0x14 \oplus 0x55 = 0x4F$

$0x2B \oplus 0x66 = 0x4D$

$0x3C \oplus 0x77 = 0x4B$

$0x4D \oplus 0x88 = 0xC5$

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

Nasimul

[23] Ans:

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

Input column;

$$a = \begin{vmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{vmatrix} = \begin{vmatrix} 0x01 \\ 0x02 \\ 0x03 \\ 0x04 \end{vmatrix}$$

So the output column is $b = M \cdot a$

where each entry is bytewise GF(2^8) arithmetic XOR.

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

Navimul

$$= 0x02 \oplus 0x01 \oplus 0x03 \oplus 0x04$$

$$= 0x03$$

$$b_1 = (01 \cdot a_0) \oplus (01 \cdot a_1) \oplus (02 \cdot a_2) \oplus (03 \cdot a_3)$$

$$= 0x01 \oplus 0x02 \oplus 0x06 \oplus 0x06$$

$$= 0x09$$

$$b_3 = (03 \cdot a_0) \oplus (01 \cdot a_1) \oplus (01 \cdot a_2) \oplus (02 \cdot a_3)$$

$$= 0x03 \oplus 0x02 \oplus 0x03 \oplus 0x08$$

$$= 0x0A$$

The final result will $b = \begin{bmatrix} 0x03 \\ 0x04 \\ 0x09 \\ 0x0A \end{bmatrix}$.

Nasimul Kurniawan

24)

AES - OFB (Output Feedback) is a mode of operation for the Encryption standard (AES) that turns a block cipher into a stream cipher.

Working procedure of AES-OFB mode

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

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

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

(iv) Decryption: The Decryption process is identical to the encryption process. The same IV is used to generate the exact key stream.

Synchronization in AES-OFB:

Both sender and reciever start with the same initialization vector (IV) and same secret key.

keystream blocks are generated only from the previous keystream block not from the ciphertext or plaintext.

IT-21026

(iv) Decryption: The Decryption process is identical to the encryption process. The same IV is used to generate the exact key stream.

Synchronization in AES-OFB:

Both sender and receiver start with the same initialization vector (IV) and same secret key.

Keystream blocks are generated only from the previous keystream block not from the ciphertext or plaintext.

25

AES modes causing error propagation:

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

(i) CBC (Cipher Block Chaining) Mode:

- Decryption formula: $P_i = AES^{-1}(C_i) \oplus C_{i-1}$
- if C_i has a 1 bit error → P_i becomes random
- Error effect two consecutive plaintext blocks

(ii) CFB (Cipher Feedback) Mode:

- Decryption Formula: $P_i = C_i \oplus AES(C_{i-1})$
- If C_i has a '1 bit error → the

same bit in Pi is wrong.

- Pi+1 becomes completely random.

26

Recommended mode: AES-CTR (Counter mode)

Justification:

- Parallel processing:

→ In CTR, each block is encrypted by XORing the plaintext with a key stream generated from AES(key, counter).

- Performance:

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

- Security:

→ Unlike EBC, CTR does not produce identical ciphertext for identical plaintext blocks.

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

IT-21026

.Why not EBC?

Reveals pattern in the data insecurely.

.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 parallelizable.

21026

27.

Solution:

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

Encryption:

$$c \equiv M^e \bmod n = 1^5 \bmod 14 = 1$$

Decryption:

$$M = C^d \bmod n = 1^{11} \bmod 14 = 1$$

RSA relies on $M^{ed} \equiv M \pmod n$ when

$ed \equiv 1 \pmod{\phi(n)}$. 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

$M = 1$

IT21026

[28]

Ans:

Given : hash H(M) = 5, d = 3, n = 33

Signature Generation (sign with private key).

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

- The message hash is H(M) = 5
- The RSA private key is (d, n) = (3, 33)
- The digital signature is completed using the hash formula,

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

$$S = 5^3 \pmod{33}$$

$$= 125 \pmod{33}$$

$\because 125 = 3 \times 33 + 26$

$\therefore 125 \pmod{33} = 26$

$\therefore$ The digital signature is 26.

IT-21026

29

Public values, Prime modulus $p = 17$

base, $g = 3$

Alias private key, $a = 4$

Badol $b = 5$

$\therefore$ Alias public key, $A \equiv g^a \pmod{p}$

$= 3^4 \pmod{17}$

$= 13$

$\therefore$ Aleya's public key is 13

Thus Badols public key, $B = g^b \pmod{p}$

$= 3^5 \pmod{17}$

$= 5$

$\therefore$ Badol's public key is 5.

IT-21026

30

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

ASCII values : A = 65, B = 66

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

$H("BA") = (66+65) \bmod 100 = 31$

∴ Both result produce same hash → collision

Collision Resistance:

The hash function is not collision resistance because its base on a simple modulary sum.

IT-21026

31

Computing MAC for message where given

MAC = (mes + sec. key) mod 17

mes = 15, sec. key = 7

MAC = (15+7) mod 17

= 22 mod 17

= 5

Suppose attackers 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 M' = M + Δ they can compute MAC' MAC' ≡ MAC + Δ (mod 17). without even knowing k.

IT-21026

B2

TLS handshake steps for symmetric key.

Establishment:

(i) Client Hello → Client sends supported TLS version, random number

(ii) Server Hello → Server selects TLS and sends its random number

(iii) Server certificate → Server certi contains server public key.

(iv) Key exchange : →

• RSA • ECDHE • Mode

(v) Pre Master secret → Master secret

(vi) Session key

(vii) Finished message → both confirmed to the handshake.

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

IT-2902G

[33] Ans:

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

(i) Transport Layers

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

- Handle 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 encripted channel into multiple logical channels. this is the highest layer.

IT-21026

34.

Explain steps involved in the TLS handshake process:

Ans:

(i) ClientHello: → proposes connection parameters.

(ii) ServerHello → Proposes connection parameters.

(iii) Certificate and key exchanges.

(iv) Generate master key.

(v) Generate session key.

(vi) Finished messages.

(vii) Secure communications.

These are the necessary steps that are involved in the TLS handshaking process.

IT21026

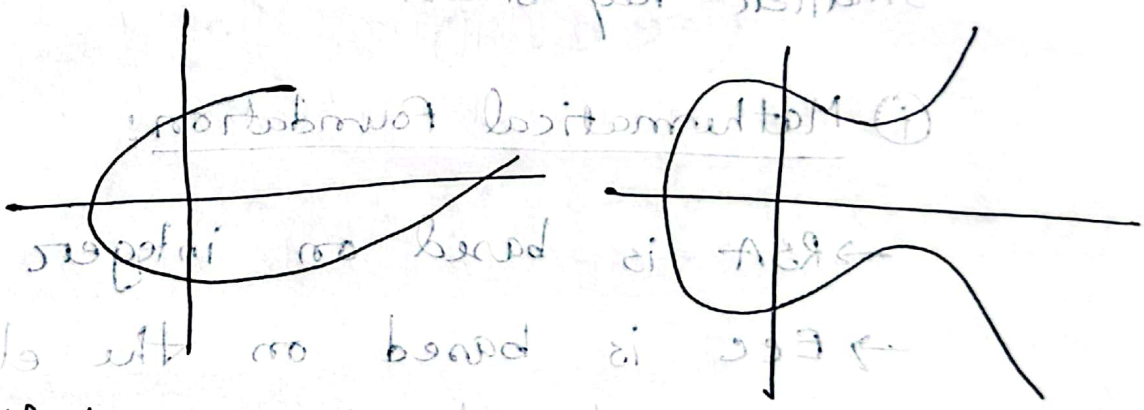
35

Ans:

The general form equation of an elliptic curve 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 constant



Such that

$$4a^3 + 27b^2 \not\equiv b \pmod{p}$$

ensures no singularities

Use in cryptography:

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

9-7-24026

[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 provide the same cryptographic strength as RSA but with much smaller key sizes.

① Mathematical foundation:

→ RSA is based on integers factorization

→ ECC is based on the elliptic curve discrete logarithmic problem (ECDLP).

② Key size:

→ ECC provides strong security with reduced key size with ECDLP.

IT. 21026

37.

Given the Elliptic curve $y^2 = x^3 + 2x + 3$ mod 97. Determine whether the point $P = (3, 6)$ lies on the curve.

Ans:

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

point = $(3, 6)$

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

R.H.S = $x^3 + 2x + 3 = 27 + 6 + 3 = 36 \pmod{97}$

As, LHS = R.H.S

The point lies on the curve.

IT-21026

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)

$c_1 = g^k \pmod p$

$= 5^6 \pmod{23}$

$= 8$

$\therefore c_1 = 8$

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

$= 10 \times 8^6 \pmod p$

$= 10 \times 13 \pmod{23}$

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

$\therefore c_2 = 15$

$\therefore$ The elgamal ciphertext is,

$(c_1, c_2) = (8, 15)$

IT-21026

39. * Lightweight cryptography is specially designed to provide strong security while using minimal computational resources, memory and power.

This is crucial for IOT and smart application. Traditional encryption like AES or RSA may be too heavy as an algorithm for these devices like sensors and wearable.

Lightweight cryptography ensure:

- Low power consumption.
- Low memory and CPU.
- Adequate Security.

Present Cipher - a block cipher with 64 bit block size designed for IOT devices which is a lightweight cryptography.

[40] List and Briefly explain any three common IOT specific attacks (eg: firmware hijacking, physical tempering) what mitigation strategies can be applied?

Ans:

Three common IOT specific attacks and mitigation strategies:

① Firmware Hijacking:

Attackers replace or modify device firmware with malicious code, allow them to take control, steal data, or disrupt functionality.

→ Mitigation: Use digitally signed firmware enable secure boot and only allow update from trusted servers.

IT-21026

(ii) Physical Tampering: IoT devices in public or unprotected areas can be physically accessed. Attackers may open the devices.

→ Mitigation: Use tamper resistant casings and disable debug ports.

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

→ Mitigation: Change default credentials keep firmware updated and use firewalls.

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

IT-21026

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

→ Mitigation: ~~Use tamper resistant~~ ✗

(iii) Botnet Attacks: Malware infected IoT devices with weak / default passwords, linking them into botnet for large-scale DDoS or spam attacks.

→ Mitigation: Change default credentials
Keep firmware updated and use firewalls.

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