Cryptography and Cyber Law

Final Assignment

Yeasin Arafat
ID: IT21041

Ans: to the Q: No: 1

Shor's algorithm threaten's RSA and ECC because it can efficiently solve the # heard mathematical Problems. These cryptosystem rely on:

i) RSA depends on difficulty of integers factorization.

ii) ECC refies on the hardness of the eliptic curve discrete logarithm problem.

Potential consequences for current digital infrastructure:

i) Massive Security breach:

All ~~ercrypted~~ communication, banking transactions, emails, VPN

connections using RSA/ECC could be decrypted.

ii) Lots of confidentiality: Even Past communication store today could be decrypted.

iii) Compromised authentication: Digital signature based on RSA/ECC could be forged, certificate, and blockchain systems.

Ans to the Q: NO: 2

Role of quantum key distribution (QKD):

Establishes symmetric keys with information-theoretic security by encoding bits on quantum states, eavesdropping includes detectable errors.

Differ from classical public-key encryption (PKC):

1) Security from physics, not computational hardness,

2) Needs a quantum channel and authenticated classical channel, limited distance/rate, specialized

Ans to the Q: No: 3

Lattic-based crypto vs number-theoretic (RSA/DH/ECC) in context of quantum resistance.

i) Assumption: Lattice problems (LWE/RLWE/SIS) vs factoring/dicreate log.

ii) Quantum: No known efficient quantum attacks on lattice problem, believed PQ-secure, RSA/ECC fall to shor.

iii) Math/OPS: Mostly linear algebra mod q. good performance.

Ans. to the Q. No: 4

Python based PRNG that uses the current system time and a custom seed value.

Code:

```
import time
A = 6364136223846793005
B = 1442695040888963407
MOD = 1<<64
def
def lcg(seed):
    x = seed % MOD
    while True:
        x = (A * x + c) % MOD
        yield x
```

```
def mix_seed(user_seed: int) -> int:
    t = time.time_ns()
    s = (t ^ ((user_seed + 0x9E3779B9
        ^ R4A7C15) * 0xBF58476
        D1CE4E5B9)) & (MOD-1)
    return s.
if __name__ == "__main__":
    user_seed = 12345
    s = mix_seed(user_seed)
    g = lcg(s)
    print("mixed_seed =", s)
    for _ in range(10):
        v = next(g)
    print(v, v/MOD)
```

Ans: to the Q: No: 5

Sieve of Eratosthenes (Prime < 50).

Alogrithm: ~~Mark~~ Mark multiple of each unmarked number starting from 2 up to $\sqrt{n}$.

Prime < 50 are:

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

Complexity:

Time → $O(n \log n)$ vs trial division $O(n\sqrt{n})$ to list primes up to n.

Explanation of Eratosthenes:

i) List numbers from 2 to n.

ii) Start with the first unmarked number p. initially, prime.

iii) Mark all multiples of p as composite.

iv) Find the next unmarked number.

v) Repeat until $p^2 > n$.

vi) All unmarked numbers left are primes.

Ans to the Q: No: 6

A Carmichael number is a composite number n that satisfies Fermat's Little Theorem for all integers a that are coprime to n:

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

They are called absolute Pseudoprimes.

Necessary and sufficient condition

i) n is square-free.

ii) n has at least three distinct Prime factors.

iii) For every Prime divisor p of n:

$$(n-1) \bmod (p-1) = 0$$

verify and check the number if it charmicheal -

i) 561 → 3, 11, 17, all (P-1) | 560 → carmicheal

ii) 1105 → 5, 13, 17, all (P-1) | 1104 → carmicheal

iii) 1729 → 7, 13, 19, all (P-1) | 1728 → carmicheal

Therefore, All three are carmichael numbers.

Ans: to the Q: No: 7

Valid Algebric Structure -

→ Z_{11} with operation $(+, \cdot)$: Yes it's ring (actually a field) because 11 is prime.

→ $(Z_{37}, +)$: Abelian group (cyclic of order 37)

→ $(Z_{35}, \times)$: Not a group (zero divisors, non-units) like 5 has no inverse

The units $U(35)$ of $\varphi(35) = 24$ so from an Abelian group.

Ans: to the Q: No: 8

Reminder of $-52 \mod 31$:

Find the least non-negative residue r with $0 \leq r < 31$ such that $-52 \equiv r \pmod{31}$

Method 1 (add multiples of 31):

$\rightarrow -52 + 31 = -21$ (still negative)

$\rightarrow -21 + 31 = 10$ (in range of 0-30)

$\therefore r = 10$

method 2 (Euclidian division):

Find q, r with $-52 = 31q + r$, $0 \leq r < 31$

Take, $q = -2$;

$-52 = 31(-2) + 10$ as $r = 10$.

Therefore -

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

$(-52 \bmod 31) = 10$

Ans: to the Q. No: 9

Multiplicative inverse of 7 mod 26.

Extended Euclid:

$$26 = 3 \cdot 7 + 5$$

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

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

$$1 = 3 \cdot 26 - 11 \cdot 7$$

So, $-11 \cdot 7 \equiv 1 \pmod{26}$

inverse is 15 (since $-11 \equiv 15$)

check: $7 \cdot 15 = 105 \equiv 1 \pmod{26}$

Ans:

(10) Evaluating ~~(-8×5 = -40)~~ . $-8 \times 5 \bmod 17$.

i) Multiply → $-8 \times 5 = -40$

ii) Now add multiples of 17 until result is in 0-16

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

∴ Result is 11.

Tips for negative modular multiplication:

i) Replace negative numbers with their positive equivalent mod m.

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

Then $9 \times 5 = 45 \equiv 11 \pmod{17}$

(11)

Bezoutis Theorem: For integers a, b (not both zero), their exist integers x, y such that:

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

Proof idea: Use the extended Eucledian algorithm to express gcd (a, b) as a linear combination of a and b.

Finding inverse of 97 mod 385:

1. $\gcd(97, 385) = 1 \rightarrow$ Inverse exists
2. Extended Euclid:

$$1 = 385(32) - 97(127).$$

3. So, $97(-127) \equiv 1 \pmod{385}$.
4. Positive inverse: $-127 \equiv 258 \pmod{385}$

∴ Inverse of 97 mod 385 is 258.

Ans.

(12) Given Equation -

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

We need the modular inverse of ~~43 mod~~ 43 mod 240.

Steps (Extended Euclid):

1. $240 = 43(5) + 25$
2. $43 = 25(1) + 18$
3. $25 = 18(1) + 7$
4. $18 = 7(2) + 4$
5. $7 = 4(1) + 3$
6. $4 = 3(1) + 1$
7. Back-substitute $\rightarrow 1 = 43(67) - 240(12)$

$\therefore x \equiv 67 \pmod{240}$

$\therefore x = 67$. (Ans)

Ans: to the Q: No: 13

Shor's algorithm threaten's RSA and ECC because it can efficiently solve the # heard mathematical Problems. These crayplosystem rely on:

i) RSA depends on difficulty of integers factorization.

ii) ECC refies on the hatdness of the elliptic curve discrete logarithum problem.

Potential consequences for carrent digital infrastructure:

i) Massive Security breach:

All erercypted ~~communicat~~ communicatior banking transactions, emails, VPN

connections using RSA/ECC could be decrypted.

ii) Lots of confidentiality: Even Past communication store today could be decrypted.

iii) Compromised authentication:

Digital signature based on RSA/ECC could be forged, certificate, and blockchain systems.

Method 2:

i) $N = 3 \cdot 5 \cdot 7$

$= 105$

ii) $N_1 = 35$.

Find y_1 with $35y_1 \equiv 1 \pmod 3$

$35 \equiv 2 \pmod 3$, so,

$2 \times 2 \equiv 1$

$y_1 \to 2$.

contribution: $2 \cdot 35 \cdot 2 \to 140$

iii) $N_2 = 21$. $21y_2 \equiv 1 \pmod 5 \to 21 \equiv 1 \pmod 5$

$y_2 \to 1$.

contribution: $3 \times 21 \times 1 = 63$

iv) $N_3 = 15$. $15y_3 \equiv 1 \pmod 7$

$15 \equiv 1 \pmod 7$

$y_3 \equiv 1$.

contribution: $2 \times 15 \times 1 = 30$.

Sum $140 + 63 + 30 = 233$. Reduce mod 105:

$233 - (2 \cdot 105)$

$\to 233 - 210$

$\to 23$.

$\therefore x \equiv 23 \pmod{105}$.

connections using RSA/ECC could be decrypted.

ii) Lots of confidentiality: Even Past communication store today could be decrypted.

iii) Compromised authentication:

Digital signature based on RSA/ECC could be forged, certificate, and blockchain systems.

Ans: to the Q: No: 18

Shor's algorithm threaten's RSA and ECC because it can efficiently solve the # heard mathematical Problems. These cryptosystem rely on:

i) RSA depends on difficulty of integers factorization.

ii) ECC refies on the hardness of the elliptic curve discrete logarithm problem.

Potential consequences for current digital infrastructure:

i) Massive Security breach:

All encrypted ~~communicat~~ communication banking transactions, emails, VPN

(17) Phishing vs malware vs DOS.

i) Phishing: Social-engineering attacks to trick users into revailing credentials or clicking malicios links.

Impact: Credential theft, account compromise, data breaches.

ii) Malware: Malicious software (viruses, trojan, ransomware) installed to steal data, damage system, or create backdoors.

Impact: Data loss, Unauthorized access, lateral movement.

iii) Denial-of-service (DOS): overwhelm resources to make services unavailable.

Impact: Down time, revenue loss, reputational damage.

(16)

Steganography: Hides the existance of a message in another file (image, audio, Video).

Example: least significant bit (LSB) image Pixel modification, audio echo hiding, DCT coefficient changes in JEPG.

Goal: Secret message looks like normal media.

Cryptography: Scrambles message into Unreadable ciphertext using an algorithm + key. Presence of message is obrious but content are protected.

Key difference:

Steganography conceals existance.

Cryptography conceals content.

(15) CIA triad in information security -

i) Confidentiality : Keep data secret from unauthorized parties.

controls: encryption, access control, strong authentication, network segmentation.

ii) Integrity : Ensure data is correct and unmodified.

controls: cryptographic hashes, digital signature, checksum, input validation.

iii) Availability : Ensure authorized users can access system and data when needed.

controls : Redundancy, backups, load blancing Pacching and monitoring, disaster recovery plan.

generate identical keystream.

(v) Error effect: 1-bit error in ciphertext only affect the same bit in plaintext (no propagation).

(25) AES modes and error propagation -

1) CBC (cipher block chaining):

→ 1-bit in ciphertext → entire current block corrupted + 1-bit flipped in next block.

→ High error spread, so integrity is badly affected.

2) CFB (Cipher Feedback):

→ 1-bit error → corrupts current block segment and

(24) AES - OFB mode -

i) Treats AES as a keystream generator.

ii) Process:

IV → AES (key) → Keystream Block

↓

Cipher text ← XOR with plaintext

iii) Decryption: Same keystream XOR with ciphertext → plaintext.

iv) Synchronization: Both end must use same key and IV to

Ans to the Q: NO: 23

Role of quantum key distribution (QKD)

Establishes symmetric keys with information-theoretic security by encoding bits on quantum states, eavesdropping includes detectable errors.

Differ from classical public-key encryption (PKC):

1) Security from physics, not computational hardness;

2) Needs a quantum channel and authenticated classical channel, limited distance/rate, specialized.

(27) SubBytes with the given partial AES-S-box—

input word: [0x23, 0xA7, 0x4C, 0x19]

i) 0x23 → row 0x2, col 0x3 → 0xD4

ii) 0xA7 → row 0xA, col 0x7 → 0x63

iii) 0x4C → row 0x4, col 0xC → 0x2E

iv) 0x19 → row 0x1, col 0x9 → not shown in the Partial table (for reference, the standard AES S-box maps 0x19 → 0xD4)

∴ output: [0xD4, 0x63, 0x2E, 0xD4].

connections using RSA/ECC could be decrypted.

ii) Lots of confidentiality: Even Past communication store today could be decrypted.

iii) Compromised authentication:

Digital signature based on RSA/ECC could be forged, certificate, and blockchain systems.

(30) Simple hash function example:

Hash Rule: Sum of ASCII values of character mod 100.

→ 'AB': 65+66 = 131.

= 131 mod 100

= 31

→ 'BA': 66+65 = 131

→ 131 mod 100

→ 31

collison: Different inputs → same hash (3

Implication:

weak hash → easy to find collisons → Bad for integrity and digital signature.

Ans to the Q: NO: 29

Role of quantum key distribution (QKD)

Establishes symmetric keys with information-theoretic security by encoding bits on quantum states, eavesdropping includes detectable errors.

Differ from classical public-key encryption (PKC):

1) Security from physics, not computational hardness,

2) Needs a quantum channel and authenticated classical channel, limited distance/rate, specialized

Ans to the Q: NO: 28

Role of quantum key distribution (QKD)

Establishes symmetric keys with information-theoretic security by encoding bits on quantum states, eavesdropping includes detectable errors.

Differ from classical public-key encryption (PKC):

1) Security from physics, not computational hardness,

2) Needs a quantum channel and authenticated classical channel, limited distance/rate, specialized

(27) Given that -

$M = 7$

$e = 5$

$n = 14$

$d = 11$

Encryption:

$$C = M^e \bmod n$$

$$= 7^5 \bmod 14$$

$$= 7.$$

Decryption:

$$M = C^d \bmod n$$

$$= 7^{11} \bmod 14$$

$$= 7.$$

∴ ciphertext = 7.

Decrypted message = 7. (same as original

→ Each block is independent; Perfect for parallel encryption or decryption.

→ Allows random access to encrypted data.

ii) ECB: Not secured (pattern leakage).

iii) CBC: secure but sequentially. CBC can't fully parallelize encryption.

Among them CTR is Fast, Parallel, no pattern leak, works for large files and streaming.

flips same bit in next segment.

→ Less spread than CBC but still alters multiple bytes

Impact: Both CBC and CFB cause decrypted data corruption if cipher text changes, making them unsuitable where bit errors are common useless integrity checks (MAC) are used

(26) Best AES Modes for large files with Parallel processing -

i) CTR (counter) Mode: Best choice.

→ Encrypts counters with AES to Produce keystream.

(36) Why Ecc gives same security with smaller keys -

i) Ecc security relies on ECDLP, which currently has no sub-exponential time algorithm.

ii) Because ECDLP is harder per bit, smaller key sizes give same security.

iii) Example : 256-bit Ecc ≈ 3072-bit RSA

384-bit Ecc ≈ 7680-bit RSA

iv) Benefits: Faster encryption/decryption, less bandwidth, lower storage and power use.

(35) The Equation of Elliptic curve -

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

condition: $4a^3 + 27b^2 \not\equiv 0 \pmod{p}$

→ Why used in Kryptography -

i) Security based on Elliptic curve Discrete Logarithm Problem, which is hard to solve.

ii) Acheives same security with much smaller keys compared to RSA/DH.

iii) Smaller keys → faster computation, less memory, lower power and Ideal for constrained device.

(34) step in the TLS Handshake Process →

i) ClientHello : client sends supported TLS version, cipher suites.

ii) ServerHello : Sends ServerHelloDone to indicate it is ready for client response

iii) clientExchange : client encrypts Pre-master secret with server's public key.

iv) Certificate Verify : Client may sends its certificate and Prove Possession of Private key.

v) Key Derivation : Both sides user Pre-master Secret + both nonces to generate the master Secret.

vi) ExchangeCipherSpec : Client and server Signal they are switching to the newly negotiated symmentic encryption.

Ans to the Q: No: 33

Lattic-based crypto. vs number-theoretic (RSA/DH/ECC) in context of quantum resistance.

i) Assumption: Lattice problems (LWE/RLWE/SIS) vs factoring/dicreate log.

ii) Quantum: No known efficient quantum attacks on lattice problem, believed PQ-secure, RSA/ECC fall to shor.

iii) Math/OPS: Mostly linear algebra mod q. good performance.

(32) TSL Handshake and main steps & how Symmetric keys are established -

i) client → server : clientHellow (version, cipher suite)

ii) server to client : serverHello, certificate, server key exchange, ServerHelloDone.

iii) client varrifies certificate, sends clientKeyExchange.

iv) Both compute pre-master to master secret to derive symmetric keys via PRF.

v) Exchange changeCipherSpec and Finished message to start encrypted traffic.

Assymetric crypto is used for server authentication and for security exchanging the Pre-master secret. Ephemeral DH gives forward security. Symmetric keys are derived from the agreed secret.

(31) Given that -

message = 15, security key = 7.

MAC = (message + secret key) mod 17

Step 1: Compute MAC -

MAC = (15 + 7) mod 17

= 22 mod 17

= 5.

original MAC is 5.

Step 2: Attacket change ~~forge message~~ to 10:

MAC = (10 + 7) mod 17

= 17 mod 17

= 0

∴ Yes, if they Atteker see an exciting (message, MAC) pair, because the schema is linear, Attecker can Adjust MAC by the message difference. If attacker does not see any MAC or key, they cannot compute it.

(40) The Three common IoT-specific attacks -

1. Firmware hijacking: Mitigation: Signed firmware, Secure boot, authenticated update channels, strick code integrity check.

2. physical tempering: Temper evidence hardware, Secure elements / TCM, device attestation, disable debug ports.

3. Botnet: Enforce strong unique credentials, disable default passwords, rate-limit / auth lockout, network segmentation, timely patches, IDS/IPS.